Targeted Activation Probability Maximization Problem in Online Social Networks

Yapu Zhang, Jianxiong Guo, Wenguo Yang, and Weili Wu, *Senior Member, IEEE*

***Abstract*—In the past decade, influence maximization becomes one of the fundamental problems in online social networks. It has popular applications such as viral marketing and rumor blocking. This problem asks for some influential users to maximize the expected followers. Unlike traditional influence maximization, we discuss the problem of influence towards a special target user in this paper. We define the targeted activation probability maximization problem, which aims at finding k intermediate users so that a given target user is more likely to be influenced by the start user. Motivated by the need for modeling the diffusion process from one user to another, we propose the Targeted Linear Threshold (TLT) model and Targeted Independent Cascade (TIC) model. We prove that the problem is NP-hard, computation of the objective function is #P-hard, and the objective functions are non-submodular. Moreover, the objective function in the TLT model is an upper bound of that in the TIC model. Based on the sandwich approximation strategy, we obtain their data-dependent approximate solutions. Finally, we use three real datasets to evaluate the effectiveness of our algorithms. The experimental results indicate that our methods can effectively increase the activation probability of the target user.**

***Index Terms*—Online social networks, targeted activation probability, data-dependent approximate solutions.**

I. Introduction

In recent decades, online social networks acted as an information service platform have been developed quickly. There are 4.54 billion users, including 3.725 billion active social media users, by the end of December 2019 [1]. Each user can make friends, promote products, and so on across these networks. Due to its essential applications in economics and epidemiology, a large number of researchers have studied the problem of information diffusion in social networks. A formal study can be traced back to the Influence Maximization (IM) problem [2], which asks for some initial users so as to maximize the followers influenced by these initial users. The authors also proposed two classic models, namely, Independent Cascade (IC) and Linear Threshold (LT) models. Since then, there are considerably related researches based on the two models.

Given an information diffusion model, the traditional IM problem mainly considers maximizing the number of users who adopt the information. However, one may only focus on whether one or more given influential users adopt the information [3]–[5]. To the best of our knowledge, Yang *et al.* [3] first studied the influence towards a target user and proposed the Acceptance Probability Maximization (APM) problem. More specifically, a start user wants to make friends with a target user. The APM problem is to find some intermediate users and send invitations to them step by step so that the target user can become a friend of the start user with the maximum probability. The authors [3] mainly discussed it under an approximate IC model. Different from the IM problem, the initial users are given in the APM problem. That is, both the start user and his/her friends are considered as the initial users. Moreover, only the selected intermediate users can be activated (i.e., the adaptor of the invitation) in the diffusion process. Following this line, Chen *et al.* [6] further considered this problem and extended it to the directed acyclic graphs. Yuan *et al.* [7] considered a constrained active friending problem in the LT model.

In this paper, we study a similar problem, and the following scenario drives our problem. A young scholar wants to send a conference invitation to a mathematician. Due to lacking mutual friends, he can not directly send this invitation or be more likely to be rejected if he sends it directly. Therefore, he hopes that his friends can help him send the invitation, and then his friends send the invitation to their friends and so on. Finally, the mathematician can accept this invitation. Assume that his friends will definitely help him send the invitation and the friends of his friends will help send this invitation with a certain probability. There is a network platform that can provide a recommendation list to assist this young scholar. Given the limited budget, which users should be included in this recommendation list to maximize the acceptance probability of the mathematician? We call this problem the Targeted Activation Probability Maximization (TAPM) problem.

The main difference between our problem and the traditional APM problem are as follows. On the one hand, the diffusion process in the APM problem is approximated by the maximum influence path. And our diffusion process is considered in the general graph. On the other hand, the start user will build a new friendship with the intermediate users at each step for the APM

Manuscript received September 21, 2020; revised October 26, 2020; accepted November 6, 2020. Date of publication November 10, 2020; date of current version March 17, 2021. This work was done during Yapu Zhang's visiting the University of Texas at Dallas as a visiting scholar funded by the China Scholarship Council. This work is supported in part by National Science Foundation under Grants No.1747818 and 1907472. And it was partly supported by the National Natural Science Foundation of China under Grants No.11991022 and 12071459. *(Corresponding author: Wenguo Yang.)*

Yapu Zhang and Wenguo Yang are with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhangyapu16@mails.ucas.ac.cn; yangwg@ucas.edu.cn).

Jianxiong Guo and Weili Wu are with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: jianxiong.guo@utdallas.edu; weiliwu@utdallas.edu).

Digital Object Identifier 10.1109/TNSE.2020.3037106

TABLE I
IMPORTANT NOTATIONS SHOWED IN THIS PAPER

Variable	Description
$G = (V, E)$	an instance of the social network
n, m	the size of nodes set V and edges set E
w_{uv}	influence probability in edge e_{uv}
s, t	the start node and the target node
$N^{in}(v), N^{out}(v)$	the in-neighbors and out-neighbors of v
$N(s)$	a set of nodes in $N^{out}(s) \cup \{s\}$
f	the objective function in the TLT model
h	the objective function in the TIC model
I	the set of intermediate users
B	the backtracking set
g	a realization under a diffusion model

problem, hoping the target can accept the friending invitation when they have sufficient mutual friends. However, the TAPM problem does not need to make friends with intermediate users iteratively. Our goal is to maximize the target user's acceptance probability according to the word-of-mouth effect among the start user, his friends, and these selected intermediate users.

This paper studies the TAPM problem in the Targeted Independent Cascade (TIC) and Targeted Linear Threshold (TLT) models. It is hard to handle since the NP-hardness of the problem and non-submodularity of the objective function in the above models. To address it, we devise the approximation algorithms with the data-dependent ratio. Our main contributions can be concluded as follows.

1) We consider a novel Targeted Activation Probability Maximization (TAPM) problem, which focuses on the influence towards a given user on general graphs. Two diffusion models, namely the Targeted Linear Threshold (TLT) model and Targeted Independent Cascade (TIC) model, are proposed. We show that our problem is NP-hard, and the computation of the objective function is #P-hard under the above two models.
2) We prove that the objective function is supermodular under the TLT model. And it is neither submodular nor supermodular under the TIC model. For the two models, we devise the unbiased estimator for the objective function, respectively. Then, the submodular lower bounds follow. Moreover, we prove that the function in the TLT model is an upper bound for that in the TIC model. Using the Sandwich Approximation method, we obtain their approximate solutions with the data-dependent performance.
3) Finally, in three real-world networks, we use the experimental results to support the correctness and the superiority of our methods.

The following paper is arranged as follows. We review the related works in Section II. We introduce the problem definition and diffusion models in Section III. Sections IV presents our approximation algorithms under the TLT and TIC models, respectively. Furthermore, we perform the experiments in Section V. Finally, our work is concluded in Section 6. For ease of reference, we list the important definitions of variables that are frequently used in Table I.

II. RELATED WORK

With the rapid development of internet technology, online social networks play an important role in our daily life. Henceforth, it attracts many researchers to study related problems in online social networks. Generally, we use a graph structure to describe the social network. Some novel methods are proposed to represent relationships among users in reality [8], [9]. Besides, motivated by its applications such as marketing, there has been extensive research [10], [11]. For instance, Si *et al.* [12], [13] explore the relationships among users' interests and applications. Mao *et al.* [14] proposed some methods to identify influential users for brand communication.

A. Influence Maximization

Among these studies, the influence maximization is a key issue in online social network. The influence maximization problem asks for k initial users such that the followers influenced by these users can be maximized in the social network [2], [15]. This problem is NP-hard and the computation of the objective function is #P-hard [16], [17]. Fortunately, the traditional greedy can give a $(1 - 1/e)$ approximate solution since the objective function is non-negative, monotone non-decreasing and submdoular [18]. Here, a set function $f : 2^V \to R^+$ is submodular if and only if $f(S \cup \{v\}) - f(S) \leq f(T \cup \{v\}) - f(T)$ for any $T \subseteq S \subseteq V$ and $v \in V \setminus S$. f is monotone non-decreasing if and only if $f(T) \leq f(S)$ for any $T \subseteq S$.

However, it takes a lot of time to compute the influence spread at each iteration using the naive greedy algorithm. To be feasible in large-scale social networks, Leskovec *et al.* proposed the cost-effective lazy forward algorithm [19] and Chen *et al.* proposed degree discount heuristics [20]. Recently, Borgs *et al.* achieved a theoretical breakthrough based on the reverse influence sampling [21]. The time complexity of their algorithm is $O(k\ell^2(m+n)\log^2 n/\varepsilon^3)$ under the IC model, where n and m are the sizes of nodes and edges, respectively. Besides, their method can guarantee a $(1 - 1/e - \varepsilon)$-approximate solution with at least $1 - n^{\ell}$ probability. Although it has a strong theoretical guarantee, there still are some rooms to improve its efficiency. Following this line, some researchers devised more efficient algorithms [22]–[25].

B. Non-submodular Influence Maximization

Nowadays, many researchers focus on the information-related problems [26]–[31]. Such an extension of the influence maximization have been studied [28], [29], [32]. In most cases, these problems will lack submodularity. For instance, Chaoji *et al.* studied the content spread in social networks [28] and Wang *et al.* proposed activity maximization problem [29].

Non-submodular influence maximization will no longer have the approximation guarantee using the traditional greedy algorithm and other improved schemes. To the best of our knowledge, there are the following three methods for non-submodular optimization.

1) Global approximation: For any set function, researchers found that it can be written as the difference between

two submodular functions [33]. Based on this idea, some algorithms, such as Submodular-Supermodular [33] and Modular-Modular algorithms [34], are derived. These algorithms always give an approximation of the local optimum.

2) Parameterized method: Some researchers solved the non-submodular problem from the view of supermodular degree [35], [36]. The supermodular degree of an element $u \in V$ by a set function f is define to be $|D_f^+(u)|$ [35], where

$$D_f^+(u) = \{v \in V | \exists S \subseteq V : f(S \cup \{v, u\}) - f(S \cup \{v\}) > f(S \cup \{u\}) - f(S)\}.$$

The supermodular degree of f is

$$D_f^+ = \max_{u \in V} |D_f^+(u)|.$$

Besides, some strategies based on curvature [37], [38] are also proposed. These methods can provide strong theoretical guarantees. However, they are hard to apply to practical problems.

3) Sandwich approximation: Recently, Lu *et al.* [39] proposed the Sandwich Approximation strategy, which can carry out a data-dependent approximate solution based on their submodular lower and upper bounds. More specifically, the method devises the solutions for the original function, lower bound, and upper bound, respectively. It then returns the solution, which can maximize the original objective function as the final result.

C. Targeted Influence Maximization

As an extension of the influence maximization, the problem of influence towards a special target user plays a role in social networks.

Yang *et al.* [3] are first among the researchers who explored this issue, and they defined Acceptance Probability Maximization (APM) problem. For general networks, they showed that the problem is NP-hard, and computing the objective function is #P-hard in the IC model. Then, they modeled an approximate IC model and proposed Selective Invitation with Tree Aggregation (SITA) and In-Node Aggregation (SITINA). Notice that these algorithms are based on the tree, and SITA is not a polynomial-time algorithm. Following them, Chen *et al.* considered the Target Influence Maximization [6], where the goal is to let a boy become a friend of a girl by making new friends influence this girl. In their work, two polynomial-time approximation algorithms are proposed when assuming the network is a directed acyclic graph. Yuan *et al.* [7] then studied a constrained active friending in the LT model. From the view of super-differentials, they devised the approximation algorithms. Furthermore, the problem can be converted into one minimum version to find the minimum size of friending innovations so that the acceptance probability can exceed a given threshold. Tong *et al.* [40] studied the minimum version and presented an approximation algorithm in the LT model for general graphs.

In this paper, we mainly study the targeted activation probability maximization problem, which aims at finding k intermediate users so that a given target user is more likely to be influenced by the start user. Since the problem is non-submodular, we design the methods using the Sandwich strategy. The proposed algorithms can not only produce a data-dependent approximate solution but also apply to large-scale social networks.

III. System Model and Definitions

A. Targeted Activation Probability Maximization Problem

Generally, a social network is described as a directed graph. Given a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, V denotes the users and E represents the relationship between users, respectively. The weight $w_{uv} \in [0, 1]$ on each directed edge e_{uv} means the social influence possibility of u upon v. Let $N^{in}(u) = \{v | e_{vu} \in E\}$, $N^{out}(u) = \{v | e_{uv} \in E\}$ and $N(u) = N^{out}(u) \cup \{u\}$, respectively. Since each weight can be normalized such that its sum is not more than 1, we suppose that $\sum_{v \in N^{in}(u)} w_{vu} \leq 1$ for each node $u \in V$. Given a start node s and a target node t, the Targeted Activation Probability Maximization (TAPM) problem aims to find a set of intermediate nodes $I \subseteq V \setminus N(s)$ with the cardinality of k so as to maximize the activation probability of t. Given a set I, we denote by $f_{<s,t>} : 2^V \rightarrow R^+$ the activation probability of the target node t upon the the nodes in $N(s) \cup I$. Formally, the TAPM problem is

$$\max f_{<s,t>}(I), \quad |I| \leq k.$$

B. Diffusion Models

In this paper, we mainly consider the process in the following two models, which are based on the LT and IC models.

1) Targeted Linear Threshold Model: Motivated by the need for modeling the diffusion process from one user to another, we first propose the Targeted Linear Threshold (TLT) model. The diffusion process under the TLT model unfolds as follows.

1. In the beginning, we activate all nodes in $N(s)$ and determine the threshold $\theta_v \in [0, 1]$ of each node v randomly. We initialize the activation set $S = N(s)$.

2. In each subsequent round, each newly-activated node v tries to activate its each out-neighbor u. If $u \in I \cup \{t\}$, then it can activate u successfully when $\sum_{N^{in}(u) \cap S} w_{vu} \geq \theta_u$. Otherwise, u can never be activated. When u is activated, we add u into the activation set S.

3. The process terminates if there are no newly-activated nodes or the target node t is activated after one round.

Consider an example in Figure 1. Suppose that each weight w_{uv} is 0.2 and the set of intermediate nodes is $I = \{v_2, v_5, v_6\}$. Initially, we activate all nodes in $N(s)$ and each node chooses a threshold in [0,1] randomly. Suppose that the threshold of each node is 0.3. Then, v_2 will be the first newly-activated node by the joint influence of s_1 and s_2. Next, v_5 is activated and it starts to influence t. Unfortunately, t can not be activated since $\theta_t > w_{v_5,t}$ and then the process terminates. Notice that v_1, v_3 and v_4

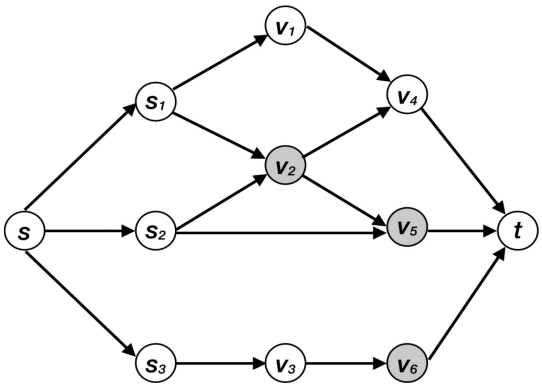


Fig. 1. Example illustrating the diffusion process.

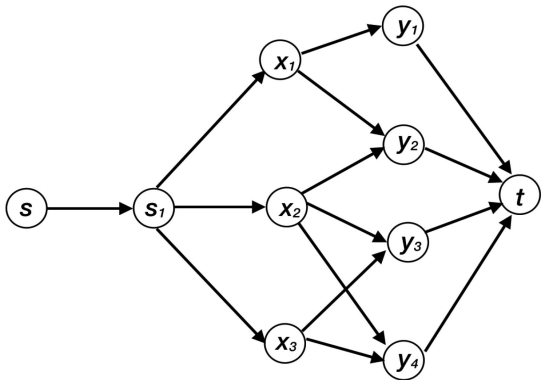


Fig. 2. Example illustrating the NP-hardness: $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_2, y_3, y_4\}$.

can never be activated since they are not the intermediate users. Although v_6 is intermediate node, v_6 will not be active since s and v_6 lack mutual neighbors.

Theorem 1: The TAPM problem is NP-hard under the TLT model.

Proof: We show that it is NP-hard with a reduction from the IM problem, which is proved to be NP-hard in [2]. Consider an instance of the IM problem as follows. Let $X = \{x_1, x_2, \ldots, x_q\}$ and $Y = \{y_1, y_2, \ldots, y_p\}$. We define a directed graph $G = (V, E)$: (1) for nodes set V, it includes node t, each node x_i in X and each node y_j in Y; (2) for edges set E, it includes directed edges (x_i, y_j) and (y_j, t) for each $x_i \in X$ and $y_j \in Y$; (3) we let $\sum_{x_i \in N^{in}(y_j)} w_{x_i,y_j} = 1$ and $w_{y_j,t} = 1/|N^{in}(t)| = 1/d$ for $x_i \in X$ and $y_j \in Y$. Notice that w_{x_i,y_j} can be equal to zero. The IM problem aims to find out k nodes to maximize the influence spread. Actually, the maximum value is $k + |Y| + 1$ involving the graph G.

Next, we can define a corresponding TAPM problem as follows. Accordingly, we build a graph $G' = (V', E')$: (1) we copy the graph G as G'; (2) we add nodes s and s_1 into V'; (3) we add edges (s, s_1) and (s_1, x_i) into E', where $w_{s,s_1} = w_{s_1,x_i} = 1$ for each $x_i \in X$. Figure 2 illustrates a construction of G'. Let s and t be the start node and target node in the instance of the TAPM problem. We prove that there is a solution $S \subseteq X$ with k nodes such that the influence spread is equal to $k + |Y| + 1$ in G if and only if there is a solution with $k + |Y|$ nodes such that the activation probability of t is 1 in G'. First, we discuss the sufficient condition. If there is a $k-$size set $S \subseteq X$ such that the influence spread is equal to $k + |Y| + 1$ in G, then selecting $S \cup Y$ will result in t is activated with 1 probability in G'. Then, we discuss the necessary condition. If there is a solution I with $|I| = k + |Y|$ such that the activation probability of t is 1, set I must contain all nodes in Y and the nodes in $I \cap X$ can definitely activate all nodes in Y. We can conclude that $I \cap X$ will be the seed in the IM problem so that the influence spread is $k + |Y| + 1$ with $|I \cap X| = |I| - |Y| = k$. Thus, the theorem is proved. ■

2) Targeted Independent Cascade Model: In the following, we discuss the Targeted Independent Cascade (TIC) model, and the diffusion process unfolds as follows.

1. Initially, we activate all nodes in $N(s)$.

2. In each subsequent round, for each newly-activated nodes v, if its neighbor $u \in I \cup \{t\}$, then it has a single chance to activate u with probability w_{vu}. Otherwise, u can never be active.

3. The process terminates if there are no newly-activated nodes or the target node t is activated after one round.

Again, we use the example in Figure 1 to illustrate the process under an TIC model. We also suppose that each weight w_{uv} is 0.2, and the set of intermediate nodes is $I = \{v_2, v_5, v_6\}$. At beginning, each node in set $N(s) = \{s, s_1, s_2, s_3\}$ is active. Then, each node in $N(s)$ starts to influence their neighbors with a certain probability. Only both v_2 and v_5 can be activated at this time since they belong to I. Notice that the activation probabilities of v_2 and v_5 are 0.36 and 0.2, respectively. Suppose that v_2 is activated and v_5 is not activated. Next, v_2 will try to activate v_5 with probability 0.2. If v_5 becomes active, then it will further influence t. Otherwise, this process terminates.

Theorem 2: The TAPM problem is NP-hard under the TIC model.

Proof: We show that it is NP-hard with a reduction from the IM problem. We construct an instance of the IM problem and define a corresponding instance of the TAPM problem in the TIC model as the case in the TLT model. The difference is that we set $w_{x_i,y_j} = 1$ for each $x_i \in X$ and $y_j \in Y$. Similarly, we can prove that there is a solution $S \subseteq X$ with k nodes such that the influence spread is equal to $k + |Y| + 1 - (1 - d)^{|Y|}$ in G if and only if there is a solution with $k + |Y|$ nodes such that t is activated with $1 - (1 - d)^{|Y|}$ probability in G', where $d = |N^{in}(t)|$. Thus, the theorem is proved. ■

C. Property

Theorem 3: The computation of the TAPM problem is #P-hard under both the TLT model and the TIC model.

Proof: We prove it by a reduction from the IM problem. The computation of influence spread is #P-hard in the LT model [17] and IC model [16]. For any directed graph $G = (V, E)$ and seed set S, we denote the influence spread as $\sigma_G(S)$. We first reduce the computation of influence spread in the LT model to the computation of the activation probability of target node in the TLT model. Let s and t be the start node and target node in

Algorithm 1: Backtracking set B.

Input: G, g, s, t
Output: B

Create a queue Q with the singleton node t
Initialize $B \leftarrow \emptyset$
While Q is not empty
  $v \leftarrow Q.dequeue()$
  If there exists a in-neighbor of v in g
    $u \leftarrow$ the in-neighbor of v in g
    If $u \in B$
      Return $B \leftarrow B \cup \{\pi\}$
    End If
    If $u \in N(s)$
      Return B
    End If
  Else
    Return $B \leftarrow B \cup \{\pi\}$
  End If
  $B \leftarrow B \cup \{u\}$
  $Q.enqueue(u)$
End While
Return B

Algorithm 2: Maximum Probability Paths.

Input: $G = (V, E), t$
Output: l and p

Initialize a set $S \leftarrow V$
Initialize $l[v] \leftarrow \emptyset$ for any node $v \in V$
Initialize $p[t] \leftarrow 1$ and $p[v] \leftarrow 0$ for any node $v \neq t$
While S is not empty
  $u \leftarrow \arg\max_{v \in S}\{p[v]\}$
  remove u from S
  For each in-neighbor v of u
    $temp \leftarrow p[u] \cdot w_{vu}$
    If $temp > p[v]$
      $p[v] \leftarrow temp$
      $l[v] \leftarrow l[u]$ adds node u
    End If
  End For
End While
Return l, p

the TAPM problem, respectively. Denote by $I = \{v_1, \ldots, v_l\}$ all the intermediate nodes between nodes s and t. We have $\sigma_G(N(s)) = f_{<s,v_1>}(I \setminus \{v_1\}) + \quad \ldots + f_{<s,v_l>}(I \setminus \{v_l\}) + f_{<s,t>}(I) + |N(s)|$. Thus, the computation of TAPM problem in the TLT is #P-hard. Similarly, we can prove it in the TIC model. Then, the theorem follows. ■

IV. Our Proposed Approximation Scheme

In this section, for convenience, we denote the objective function $f_{<s,t>}$ as f and h under the TLT model and TIC model, respectively.

A. Targeted Linear Threshold

Definition 1: Given a network $G = (V, E)$, a realization g under the TLT model is generated as follows. Each node v selects at most one node among its in-neighbors $u \in N^{in}(v)$ with probability w_{uv} and no node with probability $1 - \sum_{u \in N^{in}(v)} w_{uv}$.

Similar to the idea in [40], for a realization g in Definition 1, we can judge whether the target node t can be activated by backtracking the nodes that can reach node t. As shown in Algorithm 1, we start to backtrack from t until finding a node in $N(s)$ or no new node. We define by B the backtracking set returned by Algorithm 1. For convenience, if there is no new node, we add an element π into B. Then, if $\pi \in B$, we have $\varphi_g(I) = 0$ for any set $I \subseteq V$. Here, π is a notation introduced for the purpose of analysis. Furthermore, we conclude the following lemma.

Lemma 1: $f(I) = E[\varphi_g(I)]$, where

$$\varphi_g(I) = \begin{cases} 1 & if\ B \subseteq I, \\ 0 & otherwise. \end{cases}$$

Proof: For a realization g in Definition 1, each node selects at most one node. Thus, if there is a path from s to t, then all intermediate nodes in this path are exactly equal to all nodes in B. $E[\varphi_g(I)]$ means the probability that there is a path from s to t such that its intermediate nodes of this path are included in set I. By definition, t is activated if and only if there is a path from s to t, and its intermediate nodes are included in set I. Thus, the lemma is proved. ■

Notice that the intermediate nodes of one path is as follows. As shown in Figure 1, $< s, s_1, v_2, v_5, t >$ is a path from s to t. The intermediate nodes in this path is v_2 and v_5.

Lemma 2: f is monotone non-decreasing supermodular.

Proof: It is trivial to know that f is monotone non-decreasing. Next, it suffices to show the supermodularity. Given any two subsets $I_1 \subseteq I_2 \subseteq V$ and a node $v \in V \setminus I_2$, we aim to prove that $f(I_1 \cup \{v\}) - f(I_1) \leq f(I_2 \cup \{v\}) - f(I_2)$. That is, for any realization g, we need to prove that

$$\varphi_g(I_1 \cup \{v\}) - \varphi_g(I_1) \leq \varphi_g(I_2 \cup \{v\}) - \varphi_g(I_2). \quad (1)$$

Notice that $\varphi_g(I)$ is either 0 or 1. We only need to show that $\varphi_g(I_2 \cup \{v\}) - \varphi_g(I_2) = 1$ when $\varphi_g(I_1 \cup \{v\}) - \varphi_g(I_1) = 1$. If $\varphi_g(I_1 \cup \{v\}) - \varphi_g(I_1) = 1$, then $\varphi_g(I_1 \cup \{v\}) = 1$ and $\varphi_g(I_1) = 0$. We can conclude that $v \in B$, where B is the backtracking set involving graph g. Therefore, we have $B \nsubseteq I_2$ and then $\varphi_g(I_2) = 0$. Furthermore, $\varphi_g(I_2 \cup \{v\}) = 1$ since $\varphi_g(I_1 \cup \{v\}) = 1$ and $I_1 \subseteq I_2$. Thus, the lemma is proved. ■

Given sets I and A, we let $\varphi_g^A(I) = \max_{v \in I}\{\varphi_g(A \cup \{v\})\}$. Furthermore, we can construct a submodular lower bound as follows.

Lemma 3: For any set $I \subseteq V$, we first select a subset $A \subseteq I$, and then let $\hat{f}^A(I) = E[\varphi_g^A(I)]$. We have $\hat{f}^A(I) \leq f(I)$. Moreover, $\hat{f}^A(I)$ is submodular.

Proof: For any $v \in I$, $A \cup \{v\}$ is a subset of I. Thus, we have $\varphi_g^A(I) \leq \varphi_g(I)$ and then $\hat{f}^A(I) \leq f(I)$. Given any two subsets $I_1 \subseteq I_2 \subseteq V$ and a node $u \in V \setminus I_2$, it suffices to prove that

$$\varphi_g^A(I_2 \cup \{u\}) - \varphi_g^A(I_2) \leq \varphi_g^A(I_1 \cup \{u\}) - \varphi_g^A(I_1). \quad (2)$$

Algorithm 3: Generate a set $\mathcal{B}$.

Input: $G, s, t, A, \varepsilon, \delta$
Output: $\mathcal{B}$
1: $\Upsilon \leftarrow 1 + (1+\varepsilon) \cdot \frac{4(4-e)\ln(2/\delta)}{\varepsilon^2}$
2: $N \leftarrow 0$
3: **While** $N \leq \Upsilon$
4: Generate a realization g by Definition 1
5: Compute B using Algorithm 1
6: $\mathcal{B} \leftarrow \mathcal{B} \cup \{B\}$
7: **If** $B \subseteq A$
8: $N \leftarrow N + 1$
9: **End If**
10: **End While**
11: **Return** $\mathcal{B}$

Algorithm 4: Semi Sandwich Approximation.

Input: $G, s, t, \varepsilon, \delta$
Output: I
1: Initialize $I \leftarrow \emptyset$
2: $u \leftarrow \arg\max_{v \in N(s)}\{p[v]\}$
3: $A \leftarrow l[u] \setminus \{t\}$
4: Initialize both I_o and I_l are equal to A
5: $\mathcal{B} \leftarrow$ a set returned by Algorithm 3
6: **While** $|I_l| < k$
7: $u \leftarrow \arg\max_{v \in V \setminus I_l}\{\hat{f}^A(I_l \cup \{v\}) - \hat{f}^A(I_l)\}$
8: $I_l \leftarrow I_l \cup \{u\}$
9: **End While**
10: **While** $|I_o| < k$
11: $u \leftarrow \arg\max_{v \in V \setminus I_o}\{f(I_o \cup \{v\}) - f(I_o)\}$
12: $I_l \leftarrow I_l \cup \{u\}$
13: **End While**
14: **Return** $I \leftarrow \arg\max\{f(I_l), f(I_o)\}$

Furthermore, we only need to prove that $\varphi_g^A(I_1 \cup \{u\}) - \varphi_g^A(I_1) = 1$ when $\varphi_g^A(I_2 \cup \{u\}) - \varphi_g^A(I_2) = 1$. If $\varphi_g^A(I_2 \cup \{u\}) - \varphi_g^A(I_2) = 1$, we have $\varphi_g(A \cup \{u\}) = 1$ and $\varphi_g^A(I_2) = 0$. Therefore, $\varphi_g^A(I_1 \cup \{u\}) = \varphi_g(A \cup \{u\}) = 1$ and $\varphi_g^A(I_1) = 0$. Then, the lemma follows. ■

We first plan to determine set A and then obtain an approximate solution I containing A. Next, we discuss the selection of A. Given a set I, we observe that t is activated if and only if there is a path from s to t and the intermediate nodes of this path are included in I. Thus, we compute the maximum probability path from s to t, and select the intermediate nodes in this path as A. Moreover, set A is the optimal solution, if the number of A is exactly equal to the limited budget k and this path is a shortest path from s to t. We utilize the idea of Dijkstra Algorithm and conclude it in Algorithm 2. Let $l[v]$ be the intermediate nodes in the maximum probability path from v to t and $p[v]$ be the corresponding probability. For each $v \in V$, we add $l[v]$ and $p[v]$ into l and p, respectively. Initially, S is equal to V, $l[v]$ is an empty set for any $v \in V$, $p[t] = 1$, and $p[v] = 0$ for $v \neq t$. At each iteration, Algorithm 2 selects a node u with the maximum probability and remove it from S. Then, it updates both path $l[v]$ and probability $p[v]$ for each $v \in N^{in}(u)$. The algorithm ends until $S = \emptyset$. Since we mainly use the idea of Dijkstra Algorithm, the time complexity of Algorithm 2 can be $O(n\log n + m)$.

Since the computation of $f(I)$ is #P-hard, it is expected to estimate $f(I)$ by its unbiased estimator $E[\varphi_g(I)]$. As shown in Lemma 1, given a realization g, $\varphi_g(I)$ can be computed by its corresponding backtracking set B. More specifically, for a realization g and its corresponding set B, $\varphi_g(I) = 1$ if $B \subseteq I$. Thus, we can estimate $f(I)$ well by a number of backtracking sets. Similarly, we can estimate its lower bound $\hat{f}^A(I)$. Using Algorithm 3, we generate a series of backtracking sets $\mathcal{B}$. Let $\widetilde{f}$ and $\widetilde{f}^A$ be the estimation of f and $\hat{f}^A$ computed by the set $\mathcal{B}$, respectively. According to the results in [41], we have the following lemma.

Lemma 4: Given $\varepsilon \in (0,1)$, $\delta > 0$ and set A, let $\mathcal{B}$ be a set returned by Algorithm 3. For any set I containing A, we have $\Pr[|f(I) - \widetilde{f}(I)| \leq \varepsilon \widetilde{f}(I)] \geq 1 - \delta$ and $\Pr[|\hat{f}^A(I) - \widetilde{f}^A(I)| \leq \varepsilon \widetilde{f}^A(I)] \geq 1 - \delta$.

Proof: According to the definition of $\hat{f}^A$, we have $\hat{f}^A(A) = f(A)$. As shown in [41], Algorithm 3 can ensure that $\Pr[|f(A) - \widetilde{f}(A)| \leq \varepsilon \widetilde{f}(A)] \geq 1 - \delta$. For each set $B \in \mathcal{B}$, if $B \subseteq A$, then $B \subseteq I$. Thus, $\widetilde{f}(I)$ and $\widetilde{f}^A(I)$ computed by $\mathcal{B}$ can also satisfy the performance and the lemma follows. ■

Furthermore, we design Algorithm 4 to obtain a solution under the TLT model. First, we pick up the maximum probability path from $v \in N(s)$ to t. Using Algorithm 2, we obtain the the intermediate nodes in the maximum probability path $l[u]$ and we let A contain all nodes in $l[u] \setminus \{t\}$. Then, we generate a set $\mathcal{B}$ according to Algorithm 3. Next, we let both I_l and I_o are equal to A. We iteratively choose the node so as to maximize the marginal gain of the lower bound, i.e., $\arg\max_{v \in V \setminus I_l}\{\hat{f}^A(I_l \cup \{v\}) - \hat{f}^A(I_l)\}$, at each step until $|I_l| = k$. We greedily select the node such that the marginal increment of original function is maximized, i.e., $\arg\max_{v \in V \setminus I_o}\{f(I_o \cup \{v\}) - f(I_o)\}$, at each iteration until $|I_0| = k$. Finally, the algorithm returns the set with the maximum value. Notice that we estimate $\hat{f}^A$ and f according to set $\mathcal{B}$.

First, the time complexity of generating A is $O(n\log n + m)$. Denote by p the maximum probability from $v \in N(s)$ to t. We have the expected number of experiments in Algorithm 3 is Υ/p. Meanwhile, the time complexity of generating each backtracking set B is $O(m)$. Thus, the time complexity is $O(m\Upsilon/p)$ for Algorithm 3. Notice that the time complexity is linear to the size of $\mathcal{B}$ when computing both I_l (line 6-9) and I_o (line 10-13). Thus, the expected time of Algorithm 4 is $O((k+m)\Upsilon/p + n\log n)$.

Theorem 4: Let I_{LT}^* and I_{LT}^Abe the optimal solution for the original function f and its submodular lower bound $\hat{f}^A$, respectively. Algorithm 4 derives a $\frac{\hat{f}^A(I_{LT}^A)}{f(I_{LT}^*)}(1 - 1/e - \varepsilon)$-approximate solution with at least $1 - \delta$ probability.

Proof: Due to $\hat{f}^A$ is submodular and Lemma 4, Algorithm 4 can return a solution I_l sunch that $\hat{f}^A(I_l) \geq (1 - 1/e - \varepsilon)\hat{f}^A(I_{LT}^A)$ with at least $1 - \delta$ probability. Thus, we have

$$f(I_l) \geq \hat{f}^A(I_l) \geq (1 - 1/e - \varepsilon)\hat{f}^A(I_{LT}^A) \geq \frac{\hat{f}^A(I_{LT}^A)}{f(I_{LT}^*)}(1 - 1/e - \varepsilon)f(I_{LT}^*),$$

holds at least $1 - \delta$ probability. ■

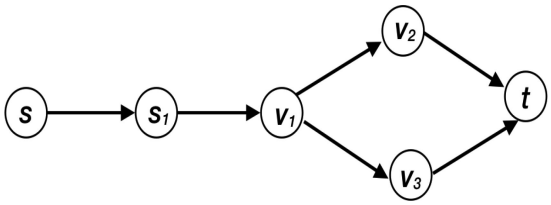


Fig. 3. Counterexample illustrating the submodularity.

Notice that $\hat{f}^A(I^A_{LT}) \geq \hat{f}^A(I^*_{LT})$ if $A \subseteq I^*_{LT}$. In that case, Algorithm 4 returns a $\frac{\hat{f}^A(I^*_{LT})}{f(I^*_{LT})}(1-1/e-\varepsilon)$ with at least $1-\delta$ probability. Given a set I, t can be active if only if there is a path from s to t and all intermediate nodes belong to I. Thus, the optimal solution must contain at least all intermediate nodes in one path. To obtain the $\frac{\hat{f}^A(I^*_{LT})}{f(I^*_{LT})}(1-1/e-\varepsilon)$ approximate solution, we can iteratively choose A as the intermediate nodes in each path from s to t. And then select the optimal value among them.

B. Targeted Independent Cascade

We first denote the realization under the *TIC* model. For a set $I \subseteq V$, we then introduce its reduced graph for this realization.

Definition 2: Given a network $G=(V,E)$, a realization g under the TIC model is generated as follows. Each edge e_{uv} is removed with probability $1-w_{uv}$.

Definition 3: Let s and t be the start node and target node, respectively. Given a realization g in Definition 2 and a set I, we generate a reduced graph $g(I)$ from g as follows. We delete each edge e_{uv} from g if u or v does not belong to set $I \cup N(s) \cup \{t\}$.

Similar to the case under the TLT model, we have the following results.

Lemma 5: Given any set $I \subseteq V$, we have $h(I) = E[\phi_g(I)]$, where $\phi_g(I)=1$ if there is a path from s to t in the reduced graph $g(I)$, and $\phi_g(I)=0$ otherwise.

Proof: By definitions, the target node t is activated if and only if there is path from s to t in a reduced graph $g(I)$. Thus, the lemma follows. ■

Next, we discuss the submodularity of h and the relationship between the objective function h under the TIC model and the objective function f under the TLT model.

Lemma 6: h is neither submodular nor supermodular.

Proof: Consider a counterexample shown in Figure 3. Let each weight $w_{uv}=0.5$ for any edge e_{uv}. We have $h(\{v_1\}) - h(\emptyset) \leq h(\{v_1,v_2\}) - h(v_2)$. Thus, h is not submodular. Moreover, h is not supermdular since $h(\{v_1,v_2\}) - h(v_1) \geq h(\{v_1, v_2, v_3\}) - h(\{v_1,v_3\})$ holds. ■

Similarly, we can construct a submodular lower bound as follows.

Lemma 7: For any set $I \subseteq V$, we first select a subset $A \subseteq I$, and then let $\hat{h}^A(I) = E[\phi^A_g(I)]$, where $\phi^A_g(I) = \max_{v\in I}\{\phi_g(A\cup\{v\})\}$. We have $\hat{h}^A(I) \leq h(I)$. Moreover, $\hat{h}^A(I)$ is submodular.

Proof: The proof is similar to Lemma 3. ■

Lemma 8: For any $I \subseteq V$, $h(I) \leq f(I)$.

Algorithm 5: Sandwich Approximation.

Input: $G, s, t, A, \varepsilon, \delta$
Output: I

Initialize $I \leftarrow \emptyset$ and $I_u \leftarrow \emptyset$
Initialize $I_o \leftarrow A$ and $I_l \leftarrow A$
$\Upsilon \leftarrow 1 + (1+\varepsilon)\cdot\frac{4(4-e)\ln(2/\delta)}{\varepsilon^2}$
$N \leftarrow 0$
While $N \leq \Upsilon$
  $g \leftarrow$ a realization generated by Definition 2
  $\mathcal{G} \leftarrow \mathcal{G} \cup \{g\}$
  For each e_{uv} in g
    If $u \notin A \cup N(s) \cup \{t\}$ or $v \notin A \cup N(s) \cup \{t\}$
      delete e_{uv} from g
    End If
  End For
  If there is a path from s to t in g
    $N \leftarrow N+1$
  EndIf
End While
While $|I_l| < k$
  $u \leftarrow \arg\max_{v\in V\setminus I_l}\{\hat{h}^A(I_l\cup\{v\}) - \hat{h}^A(I_l)\}$
  $I_l \leftarrow I_l \cup \{u\}$
End While
While $|I_o| < k$
  $u \leftarrow \arg\max_{v\in V\setminus I_o}\{h(I_o\cup\{v\}) - h(I_o)\}$
  $I_o \leftarrow I_o \cup \{u\}$
End While
$I_u \leftarrow$ a set returned by Algorithm 4
Return $I \leftarrow \arg\max\{h(I_l), h(I_o), h(I_u)\}$

Proof: We first claim that the activation probability of each node under the TLT model is no less than that under the TIC model. Suppose that there are ℓ active in-neighbors of node v at time t and this set of in-neighbors is $\{u_1, u_2, \ldots, u_\ell\}$. Then, node v becomes active under the TIC and TLT models with probability $1-\prod_{i=1}^{\ell}(1-w_{u_iv})$ and $\sum_{i=1}^{\ell} w_{u_iv}$, respectively. We prove $1-\prod_{i=1}^{\ell}(1-w_{u_iv}) \leq \sum_{i=1}^{\ell} w_{u_iv}$ by induction. When $\ell=1$, it is obvious that the above inequality holds. Assume that it still holds when $\ell-1$. We have $1-\prod_{i=1}^{\ell}(1-w_{u_iv}) = 1-\prod_{i=1}^{\ell-1}(1-w_{u_iv})\cdot(1-w_{u_\ell v}) = 1-\prod_{i=1}^{\ell-1}(1-w_{u_iv}) + \prod_{i=1}^{\ell-1}(1-w_{u_iv})\cdot w_{u_lv} \leq \sum_{i=1}^{\ell-1} w_{u_iv} + w_{u_\ell v} = \sum_{i=1}^{\ell} w_{u_iv}$. Then, we can conclude that the probability that the target node t is activated in the TLT model is no less than it in the TIC model. Thus, the lemma is proved. ■

We design Algorithm 5 to solve the TAPM problem under the TIC model. And this algorithm is based on the above results and the sandwich approximation strategy. First, we generate a series of realizations $\mathcal{G}$ according to Definition 2. Similarly, we initialize I_l and I_o as a given set A, which is generated as Algorithm 4. Then, we choose iteratively node satisfying $\max_{v\in V\setminus I_l}\{\hat{h}^A(I_l\cup\{v\}) - \hat{h}^A(I_l)\}$ and $\max_{v\in V\setminus I_o}\{h(I_o\cup\{v\}) - h(I_o)\}$ into I_l and I_o until $|I_l| = |I_o| = k$, respectively. Moreover, according to Lemma 8, the set I_u returned by Algorithm 4 is considered as a solution of its upper bound. Finally, our result is the set with the maximum value of h among I_l, I_o and I_u.

Notice that we can not compute both h and $\hat{h}^A$ using $\mathcal{G}$ directly. According to the current set I_o and I_l, we first need

generate its reduced graph for each realization $g \in \mathcal{G}$ and then obtain h and $\hat{h}^A$. For a reduced graph $g(I)$, if there is a path from s to t, then $\phi_g(I) = 1$. Thus, we can compute $\phi_g(I)$ and $\phi_g^A(I)$ according to Depth First Search. We have the time complexity of generation I_l and I_o are $O(km\Upsilon/p)$. In summary, the expected time of Algorithm 5 is $O(km\Upsilon/p + n\log n)$.

Let h and $\hat{h}^A$ computed by the realizations $\mathcal{G}$. We can have the following performance when estimating h and $\hat{h}^A$.

Lemma 9: Given $\varepsilon \in (0,1)$ and $\delta > 0$, for a set I containing A, we have $\Pr[|h(I) - \widetilde{h}(I)| \leq \varepsilon\widetilde{h}(I)] \geq 1 - \delta$ and $\Pr[|\hat{h}^A(I) - \widetilde{h}^A(I)| \leq \varepsilon\widetilde{h}^A(I)] \geq 1 - \delta$.

Proof: The proof is similar to Lemma 4. ■

Theorem 5: Let I_{IC}^*, I_{IC}^A and I_{LT}^A be the optimal solutions for h, $\hat{h}^A$ and $\hat{f}^A$, respectively. Algorithm 5 derives a $\max\{\frac{\hat{h}^A(I_{IC}^A)}{h(I_{IC}^*)}, \frac{h(I_u)}{h(I_{IC}^*)} \cdot \frac{\hat{f}^A(I_{LT}^A)}{f(I_u)}\}(1 - 1/e - \varepsilon)$-approximate solution with at least $1 - \delta$ probability.

Proof: Due to the submodularity of $\hat{h}^A$ and Lemma 9, the solution I_l returned by Algorithm 5 can ensure that $\hat{h}^A(I_l) \geq (1 - 1/e - \varepsilon)\hat{h}^A(I_{IC}^A)$ with at least $1 - \delta$ probability. Thus, we have

$$\begin{aligned} h(I_l) &\geq \hat{h}^A(I_l) \geq (1 - 1/e - \varepsilon)\hat{h}^A(I_{IC}^A) \\ &\geq \frac{\hat{h}^A(I_{IC}^A)}{h(I_{IC}^*)}(1 - 1/e - \varepsilon)h(I_{IC}^*). \end{aligned}$$

Furthermore, the solution I_u returned by Algorithm 4 can ensure that $\hat{f}^A(I_u) \geq (1 - 1/e - \varepsilon)\hat{f}^A(I_{LT}^A)$ with at least $1 - \delta$ probability. Thus, we have

$$\begin{aligned} h(I_u) &= \frac{h(I_u)}{f(I_u)} f(I_u) \\ &\geq \frac{h(I_u)}{f(I_u)} \hat{f}^A(I_u) \\ &\geq \frac{h(I_u)}{f(I_u)}(1 - 1/e - \varepsilon)\hat{f}^A(I_{LT}^A) \\ &\geq \frac{h(I_u)}{f(I_u)} \cdot \frac{\hat{f}^A(I_{LT}^A)}{h(I_{IC}^*)}(1 - 1/e - \varepsilon)h(I_{IC}^*), \end{aligned}$$

hold with at least $1 - \delta$ probability, respectively. The theorem is proved. ■

V. Performance Analysis

A. Experimental Setup

1) Datasets: We complete our experiments in *Wikipedia*, *HepTh*, and *Facebook*. All these three datasets can be obtained from J. Leskovec [42]. We double the number of edges for *Facebook* since it is undirected. The important information on these networks is shown in Table II.

Wikipedia captures a voting activity. If user i votes on user j, then there is an edge from i to j. *HepTh* is a citation graph from the e-print arXiv. An edge from i to j means paper i cites paper j. *Facebook* is collected from Facebook pages, where the nodes denote the pages, and edges are mutual hobbies among them.

2) Settings: We conduct the tests under both the TLT model and TIC model. For convenience, we consider the influence probability w_{uv} is equal to $1/|N^{in}(v)|$ on each edge e_{uv}. For each network, we randomly sample 50 pairs of start node s and target node t. Meanwhile, we make each pair (s,t) satisfy that there is at least one simple path with the activation probability of t is no less than 0.005. We use the average of the results yielding the above 50 pairs as our final experimental results. To estimate the objective function, we generate a series of realizations. As for this process, we set that $\varepsilon = 0.5$ and $\delta = 0.01$.

TABLE II
Important Information on Networks

Name	Wikipedia	HepTh	Facebook
#Nodes	7.1K	28K	135K
#Edges	104K	353K	1.4M
Type	Directed	Directed	Undirected

When comparing with other algorithms, we evaluate the objective function using 10,000 Monte-Carlo simulations. All experiments are completed on a machine with a 3.6 GHz quad-core processor.

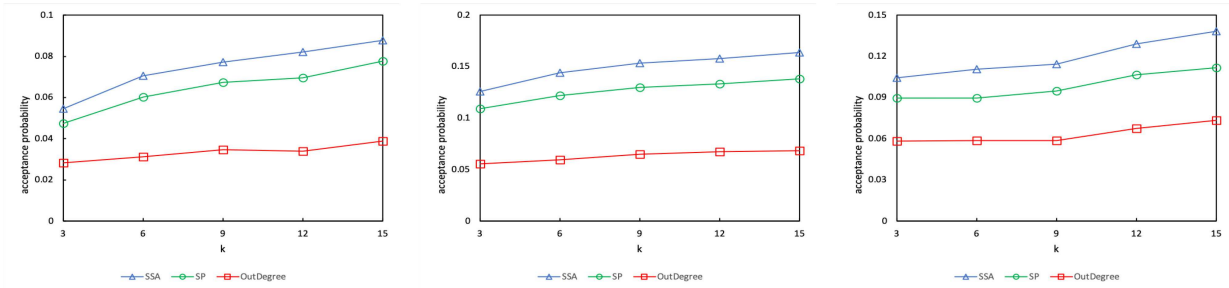
3) Algorithms: Although our proposed problem is similar to the active friending problem, there are differences mentioned in Section 1. Moreover, we notice that Lin *et al.* [3] and Chen *et al.* [6] discuss the problem by assuming the network is a tree or a directed acyclic graph. And the algorithms in [40] aims to solve the minimum version. Thus, we mainly consider the following algorithms.

1) Semi Sandwich Approximation (SSA): As shown in Algorithm 4, it is proposed to solve the problem under the TLT model.
2) Sandwich Approximation (SA): As shown in Algorithm 5, it is proposed to solve the problem under the TIC model.
3) Shortest Path (SP): The algorithm iteratively chooses the intermediate nodes on their shortest paths from s to t until the budget constraint is satisfied. It is also considered in [3], [40].
4) OutDegree: It selects top k nodes with the maximum out-degree, which is considered a baseline.

B. Experimental Results

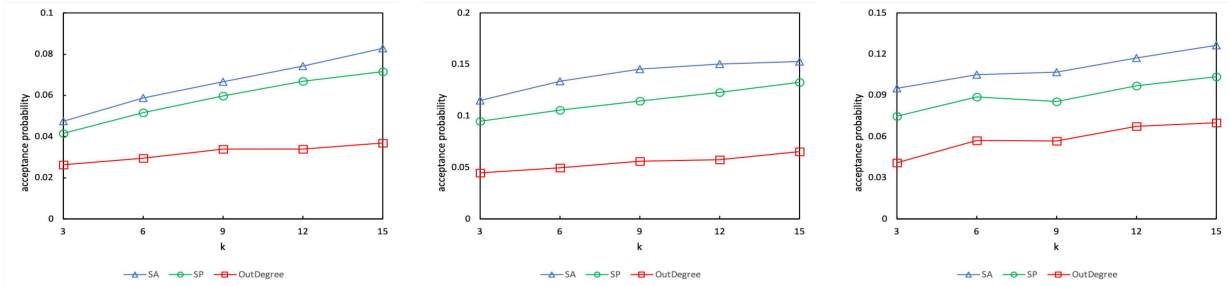
1) The Impact of k: First, we compare the activation probability of target node using the above algorithms by varying the budget k. As shown in Figure 4 and Figure 5, our proposed algorithms outperform other algorithms under both the TLT model and TIC model. Meanwhile, as the budget k increases, the activation probability will increase. We observe that the OutDegree algorithm is not as bad as we expected. That is because that the distance of the shortest path from s to t is 3 in some cases. It means that the intermediate nodes in a path can be an empty set. Generally, if the distance of the shortest path is 3, their activation probability will be large. In fact, the activation probability shown in OutDegree algorithm is very close to the value when budget $k = 0$.

Notice that in the TLT model, we have $\sum_{v \in N^{in}(u)} w_{vu} \leq 1$ for each node $u \in V$. However, $\sum_{v \in N^{in}(u)} w_{vu}$ can be larger than 1 in the TIC model. Thus, we conduct other experiments in which w_{uv} is chosen uniformly at random from [0,1]. We use



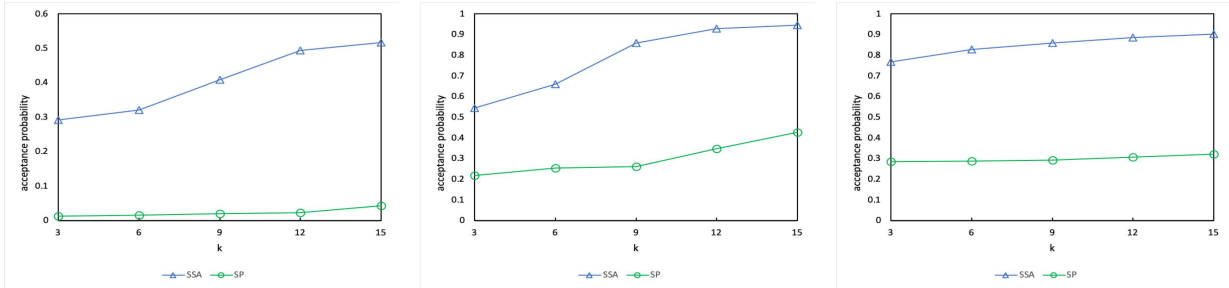
(a) Wikipedia (b) HepTh (c) Facebook

Fig. 4. Activation probability by varying budget k under the TLT model.



(a) Wikipedia (b) HepTh (c) Facebook

Fig. 5. Activation probability by varying budget k under the TIC model.



(a) Wikipedia (b) HepTh (c) Facebook

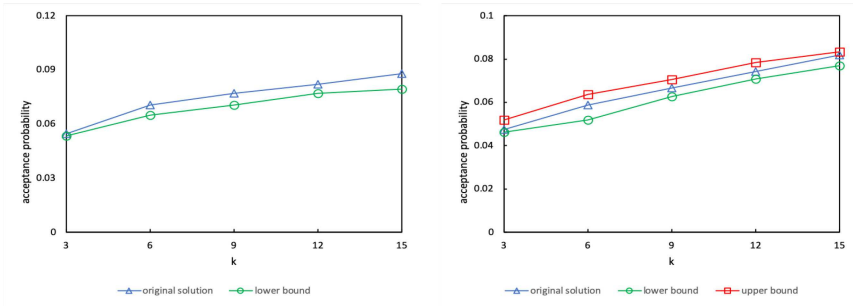
Fig. 6. Activation probability by varying budget k under the TIC model when w_{uv} is uniformly at random from [0,1].

a SSA algorithm to obtain the solution since the upper bound under the TIC model comes from the solution to the TLT model. As shown in Figure 6, the activation probability of the target increases when the budget k increases. At the same time, the activation probability is close to 1 in HepTh and Facebook when $k = 15$. Moreover, the results from the SSA is ten times larger than the solutions from the SP in Wikipedia and three times larger than that in both *HepTh* and *Facebook*.

2) Approximation Ratio: Although both the SSA and SA algorithms carry a data-dependent approximation ratio, they are hard to compute directly. To show this approximation ratio, for *Wikipedia*, we compute the activation probability of the target node for both the lower bound and the original function using the solution returned by the SSA algorithm, respectively. As shown in Figure 7(a), we see that the difference between f and $\hat{f}^A$ is small no matter what the size of the budget k. Furthermore, we consider the objective function f under the TLT model as an upper bound of the objective function h under the TIC model. We present the value among h, h^A and f in Figure 7(b) using the solution returned by SA algorithm. Similarly, their results are very close, which illustrates that our algorithms can provide a good performance guarantee.

3) The Impact of A: Our algorithms are based on a given set A. Next, we focus on the impact of A. As shown in the SSA algorithm, we select the intermediate nodes on the maximum probability path as A directly. To reflect the influence of A, we consider the top 3 maximum probability paths for *Wikipedia*. Let ℓ be the number of selected paths. That is, we use the intermediate nodes on the top ℓ maximum probability paths as the set A. Figure 8 shows the final results when the size of the budget k is equal to 15. We observe that the results do not vary much, no matter what the size of A.

4) Running Time: Since other methods are heuristics, we do not compare our algorithm to them. Figure 8 shows the running



(a) TLT model (b) TIC model

Fig. 7. The performance of the bounds.

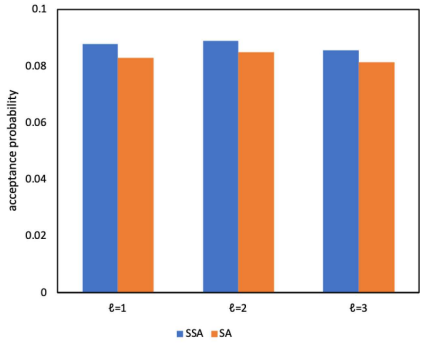


Fig. 8. The Performance when changing the size of A in Wikipedia.

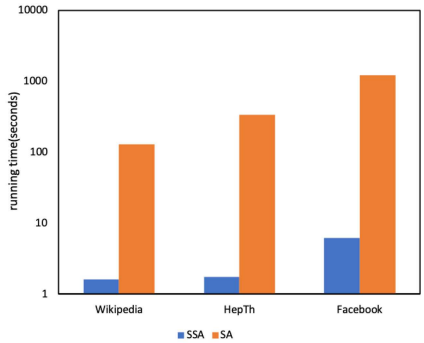


Fig. 9. Running time.

time of the SSA and SA algorithms when the budget k is equal to 15 on three networks. Here, we use a log scale on y-axis to show their difference. According to Figure 9, we find that the small dataset takes less time to run. In the meantime, the running time of the SA algorithm is larger than that of the SSA algorithm. That is mainly because that the SSA algorithm takes much more time to estimate the objective function. In section V, we show that the expected time of SSA and SA are $O((k+m)\Upsilon/p + n\log n)$ and $O(km\Upsilon/p + n\log n)$, respectively.

VI. Conclusion and Future Work

In this paper, we discuss the targeted activation probability maximization problem, which asks for a certain number of intermediate users so that one user can successfully influence a given target with a maximum probability. Two different diffusion models, namely the TIC and TLT model, are proposed. We show that the problem is NP-hard, and computing the objective function is #P-hard under the above models. To estimate them, we devise their unbiased estimators. Moreover, we show that the objective function is an upper bound of that in the TIC model. Based on the Sandwich Approximation strategy, we obtain their data-dependent approximate solutions, respectively. Our experimental results on three networks show the significance of our methods.

Although we obtained a submodular lower bound of the objective function in the TLT model, a submodular upper is still hard to give. In the future, we will try to devise an available upper bound. Also, we will try to design another approximate algorithm with a better performance guarantee.

References

[1] K. Smith, "126 amazing social media statistics and facts," https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/, Dec. 2019.

[2] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining.*, 2003, pp. 137–146.

[3] D.-N. Yang, W.-C. Lee, W. Chen, and H.-J. Hung, "Maximizing acceptance probability for active friending in on-line social networks," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD '13)*, Aug. 2013, pp. 713–721.

[4] Y. Li, D. Zhang, and K.-L. Tan, "Real-time targeted influence maximization for online advertisements," in *Proc. VLDB Endowment: 41st Int. Conf. VLDB Endowment*, Res. Collection School Inf. Syst., Kohala Coast, HI, USA, vol. 8, pp. 1070–1081. Aug. 31–Sep. 4, 2015. [Online]. Available: https://ink.library.smu.edu.sg/sis_research/4022

[5] J. Guo, Y. Li, and W. Wu, "Targeted protection maximization in social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1645–1655, Jul.–Sep. 2019.

[6] H. Chen, W. Xu, X. Zhai, Y. Bi, A. Wang, and D.-Z. Du, "How could a boy influence a girl?," in *Proc. 10th Int. Conf. Mobile Ad-hoc Sensor Netw.*, 2014, pp. 279–287.

[7] J. Yuan, W. Wu, Y. Li, and D. Du, "Active friending in online social networks," in *Proc. 4th IEEE/ACM Int. Conf. Big Data Comput., Appl. Technol.*, 2017, pp. 139–148.

[8] Y. Hou, H. Chen, C. Li, J. Cheng, and M. C. Yang, "A representation learning framework for property graphs," in *Proc. 25th ACM SIGKDD Int. Conf.*, 2019, pp. 65-73.

[9] H. Si, Z. Chen, W. Zhang, J. Wan, J. Zhang, and N. N. Xiong, "A member recognition approach for specific organizations based on relationships among users in social networking twitter," *Future Gener. Comput. Syst.*, vol. 92, pp. 1009–1020, 2019. [Online]. Available: https://doi.org/10.1016/j.future.2018.07.060

[10] Y. Liu, A. Liu, N. N. Xiong, T. Wang, and W. Gui, "Content propagation for content-centric networking systems from location-based social networks," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, no. 10, pp. 1946–1960, Oct. 2019. [Online]. Available: https://doi.org/10.1109/TSMC.2019.2898982

[11] G. Liao, X. Huang, N. N. Xiong, and C. Wan, "An intelligent group event recommendation system in social networks," *arXiv:2006.08893*, 2020. [Online]. Available: https://arxiv.org/abs/2006.08893
[12] H. Si, H. Wu, L. Zhou, J. Wan, N. Xiong, and J. Zhang, "An industrial analysis technology about occupational adaptability and association rules in social networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1698–1707, Mar. 2020. [Online]. Available: https://doi.org/10.1109/TII.2019.2926574
[13] H. Si *et al.*, "Association rules mining among interests and applications for users on social networks," *IEEE Access*, vol. 7, pp. 116 014–116 026, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2925819
[14] Y. Mao, L. Zhou, and N. Xiong, "Identify influential nodes in online social network for brand communication," *arXiv:2006.14104*, 2020. [Online]. Available: https://arxiv.org/abs/2006.14104
[15] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2001, pp. 57–66.
[16] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2010, pp. 1029–1038.
[17] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Proc. IEEE Int. Conf. Data Mining.* , 2010, pp. 88–97.
[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-i," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
[19] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 420–429.
[20] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 199–208.
[21] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2014, pp. 946–957.
[22] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2014, pp. 75–86.
[23] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*. ACM, 2015, pp. 1539–1554.
[24] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," in *Proc. ACM Int. Conf. Manag. Data*, 2016, pp. 695–710.
[25] J. Tang *et al.*, "Efficient approximation algorithms for adaptive seed minimization," in *Proc. Int. Conf. Manag. Data*, 2019, pp. 1096–1113.
[26] Shahzad *et al.*, "Real time modbus transmissions and cryptography security designs and enhancements of protocol sensitive information." *Symmetry*, vol. 7, no. 3, pp. 1176–1210, 2015.
[27] K. Huang, Q. Zhang, C. Zhou, N. Xiong, and Y. Qin, "An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 47, no. 10, pp. 2704–2713, Oct. 2017.
[28] V. Chaoji, S. Ranu, R. Rastogi, and R. Bhatt, "Recommendations to boost content spread in social networks," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 529–538.
[29] Z. Wang, Y. Yang, J. Pei, L. Chu, and E. Chen, "Activity maximization by effective information diffusion in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 11, pp. 2374–2387, Nov. 2017.
[30] W. Wu, N. Xiong, and C. Wu, "Improved clustering algorithm based on energy consumption in wireless sensor networks," *IET Netw.*, vol. 6, no. 3, pp. 47–53, 2017. [Online]. Available: https://doi.org/10.1049/iet-net.2016.0115
[31] Q. Zhang, C. Zhou, N. Xiong, Y. Qin, X. Li, and S. Huang, "Multimodel-based incident prediction and risk assessment in dynamic cybersecurity protection for industrial control systems," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 10, pp. 1429–1444, Oct. 2017.
[32] G. Tong *et al.*, "An efficient randomized algorithm for rumor blocking in online social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 845–854, Apr.-Jun. 2020.
[33] M. Narasimhan and J. A. Bilmes, "A submodular-supermodular procedure with applications to discriminative structure learning," *ArXiv*, 2008. [Online]. Available: http://arxiv.org/abs/1207.0560
[34] R. Iyer and J. Bilmes, "Algorithms for approximate minimization of the difference between submodular functions, with applications," *ArXiv*, 2012. [Online]. Available: http://arxiv.org/abs/1207.1404
[35] U. Feige and R. Izsak, "Welfare maximization and the supermodular degree," in *Proc. 4th Conf. Innov. Theor. Comput. Sci.*, 2013, pp. 247–256.
[36] M. Feldman and R. Izsak, "Building a good team: Secretary problems and the supermodular degree," in *Proc. 28th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2017, pp. 1651–1670.
[37] Z. Wang, B. Moran, X. Wang, and Q. Pan, "Approximation for maximizing monotone non-decreasing set functions with a greedy method," *J. Combinatorial Optimi.*, vol. 31, no. 1, pp. 29–43, 2016.
[38] W. Bai and J. A. Bilmes, "Greed is still good: Maximizing monotone submodular+ supermodular functions," in *Proc. Int. Conf. Mach. Learn.*, 2018. [Online]. Available: http://arxiv.org/abs/1801.07413
[39] W. Lu, W. Chen, and L. V. Lakshmanan, "From competition to complementarity: Comparative influence diffusion and maximization," *Proc. VLDB Endowment*, vol. 9, no. 2, pp. 60–71, 2015.
[40] G. Tong, R. Wang, X. Li, W. Wu, and D.-Z. Du, "An approximation algorithm for active friending in online social networks," in *Proc.IEEE 39th Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 1264–1274.
[41] P. Dagum, R. Karp, M. Luby, and S. Ross, "An optimal algorithm for monte carlo estimation," *SIAM J. Comput.*, vol. 29, no. 5, pp. 1484–1496, 2000.
[42] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

Yapu Zhang received the B.S. degree in mathematics and applied mathematics from Northwest University, Xi'an, China, in 2016. She is currently working toward the Ph.D. degree with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China. Her research interests include social networks and approximation algorithms.

Jianxiong Guo received the B.S. degree in energy engineering and automation from the South China University of Technology in 2015 and the M.S. degree in chemical engineering from the University of Pittsburgh in 2016. He is currently working toward the Ph.D. degree with the Department of Computer Science, the University of Texas, Dallas. His research interests include social networks, data mining, IoT application, blockchain, and combinatorial optimization.

Wenguo Yang received the M.A. and Ph.D. degrees in operation research and control theory from Beijing Jiaotong University, China, in 2003, and from the University of Chinese Academy of Sciences, China, in 2006, respectively. He is currently a Professor with the School of Mathematics, University of Chinese Academy of Sciences, Beijing, China. His research interest includes social networks, robust optimization, nonlinear combinatorial optimization, emergency management, and telecommunication network optimization.

Weili Wu (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the Department of Computer Science, University of Minnesota, Minneapolis, MN, USA, in 1998 and 2002, respectively. She is currently a Full Professor with the Department of Computer Science, University of Texas, Dallas, TX, USA. She is involved in the design and analysis of algorithms for optimization problems that occur in wireless networking environments and various database systems. Her research interests include data communication and data management.