

Overall Evaluations on Benefits of Influence When Disturbed by Rivals

Jianxiong Guo, Yapu Zhang, Weili Wu, *Senior Member, IEEE*

Abstract—Influence maximization (IM) is a representative and classic problem that has been studied extensively before. The most important application derived from the IM problem is viral marketing. Take us as a promoter, we want to get benefits from the influence diffusion in a given social network, where each influenced (activated) user is associated with a benefit. However, there is often competing information initiated by our rivals diffusing in the same social network at the same time. Consider such a scenario, a user is influenced by both my information and my rivals' information. Here, the benefit from this user should be weakened to certain degree. How to quantify the degree of weakening? Based on that, we propose an overall evaluations on benefits of influence (OEBI) problem. We prove the objective function of the OEBI problem is not monotone, not submodular, and not supermodular. Fortunately, we can decompose this objective function into the difference of two submodular functions and adopt a modular-modular procedure to approximate it with a data-dependent approximation guarantee. Because of the difficulty to compute the exact objective value, we design a group of unbiased estimators by exploiting the idea of reverse influence sampling, which can improve time efficiency significantly without losing its approximation ratio. Finally, numerical experiments on real datasets verified the effectiveness of our approaches regardless of performance and efficiency.

Index Terms—Overall evaluations, Influence maximization, Submodularity, Modular-modular procedure, Sampling techniques, Social networks, Approximation algorithm

I. INTRODUCTION

THE online social media, such as Twitter, Facebook, Wechat, and LinkedIn, were booming prosperously in the recent decade and become a dominating method to contact with others and make friends [1]. People are more inclined to share their comments about some hot issues at every moment in these platforms. By the end of December 2019, there are more than 3.725 billion users active in these social media. The relationships among the users on these social platforms can be denoted by social networks. A large number of messages can be shared rapidly over the networks. Subsequently, influence maximization (IM) [2] was formulated to focus on a problem that selects a small subset of users (seed set) for an information cascade to maximize the expected follow-up adoptions (influence spread). It is a natural generalization for viral marketing. The IM problem was based on the two influence diffusion models, independent cascade model (IC-model) and linear threshold model (LT-model), and they can

be summarized into the trigger model. Besides, they [2] proved the expected influence spread is monotone and submodular, thereby a $(1 - 1/e)$ -approximation can be obtained by the greedy algorithm implemented by the Monte-Carlo (MC) simulations.

Since this seminal work, it derives a series of optimization problems, such as profit maximization (PM) [3] [4] [5], competitive IM [6] [7], and rumor blocking [8] [9]. Consider us as a promoter to initiate an information cascade, we aim to get benefits from the influence spread started from our selected seed set in a social network. If a user is activated during the influence diffusion, we can get a benefit associated with her. Suppose it exists cost needed to pay when selecting a seed set, the profit is defined by the total benefits of influence spread minus the cost of this seed set, where the PM problem aims to maximize the expected profit. However, this is only an idealized state, where there is no competitor diffusing its cascade simultaneously. Generally, more than one type of information can flood the same network. In the competitive IM problem, there are multiple information cascades diffusing their respective influence independently, where it assumes a user can only be activated by one cascade successfully. It aims to select a seed set to maximize our own expected influence spread or to minimize the influence spread from other competing cascades (rumor blocking).

Combining the PM and competitive IM problem together, it formulates the competitive PM problem that maximizes our own expected profit when there are multiple information cascades. However, this model has a crucial drawback because each user can only be activated by one cascade. Actually, for a user in a social network, she may be influenced by multiple cascades from different promoters. If a user is activated by our cascade but activated by rivals' cascades contemporarily, the benefit we can get from her will be weakened, even be negative. Let us consider the following example.

Example 1. *Take us as an Apple carrier, we want to popularize a new iPhone across a given network by influence diffusion. If a user is influenced by us, we can get a benefit from her according to her appraisal about our product. When there is a rival, such as Samsung, existing, it will promote its phone by diffusing the influence as well. If a user is influenced by both Samsung and us, its appraisal about our product is very likely to be reduced after comparing it with Samsung. The benefit associated with her will be reduced even to be negative.*

Based on this realistic scenario, we propose an overall evaluations on benefit of influence (OEBI) problem, where we define how to quantify and maximize the benefits of

J. Guo and W. Wu are with the Department of Computer Science, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX, USA; Y. Zhang is with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, CHN.

E-mail: jianxiong.guo@utdallas.edu

Manuscript received April 19, 2005; revised August 26, 2015.

influence because of the rival's disturbance. We show that the OEBI problem is NP-hard and its objective function is not monotone, not submodular, and not supermodular. Because there is no direct approach to approximate it with a theoretical bound, we decompose this objective function into the difference of two monotone and submodular functions. Then, we adopt a modular-modular procedure [10] that replaces the first submodular function with one of its lower bound and the second submodular function with one of its upper bound. Then, a data-dependent approximation ratio can be obtained by this procedure. Moreover, it is #P-hard to compute the exact objective value under the IC-model [11] and LT-model [12]. Even though we can estimate our objective value by use of MC simulations, the terrible time inefficiency is unavoidable, which restricts its scalability to larger networks. Based on the idea of reverse influence sampling (RIS) [13], we design a group of unbiased estimators to estimate the value of our objective function. If the number of samplings is large enough, its estimation error is neglectable. Next, we take this estimator as the input of modular-modular procedure, which reduces the running time greatly while maintaining the approximation guarantee. Finally, we conduct several experiments to evaluate the superiority of our proposed method to other heuristic algorithms, where they support the effectiveness and efficiency of our method strongly.

Organization: Sec. II surveys the-state-of-art works. Sec. III is dedicated to introduce diffusion model, background, and define the OEBI problem formally. The monotonicity, submodularity, and computability are presented in Sec. IV. Sec. V is the main contributions, including algorithm design, sampling techniques, and approximation guarantee. Numerical experiments and performance analysis are presented in Sec. VII and VIII is the conclusion for this paper.

II. RELATED WORKS

Influence Maximization: Kempe *et al.* [2] came up with the IC-model and LT-model, formulated IM problem as a monotone submodular maximization problem, and gave a greedy algorithm that achieves $(1 - 1/e - \epsilon)$ -approximation implemented by MC simulations. Chen *et al.* proved it is #P-hard to compute the expected influence spread given a seed set under the IC-model [11] and LT-model [12]. Besides, they devised two efficient heuristic algorithms to solve the IM problem and evaluate their scalability. Contemporarily, a series of heuristic algorithms emerged, such as cost-effective lazy forward strategy [14] and degree discount heuristics [15]. Brogs *et al.* [13] made a breakthrough. They proposed the concept of RIS to estimate the expected influence spread, which is scalable in practice and has a theoretical bound at the same time. Then, a series of researchers designed more efficient algorithms that achieve $(1 - 1/e - \epsilon)$ -approximation based on the RIS. Tang *et al.* [16] [17] proposed TIM/TIM+ algorithms first and then develop a more efficient IMM based on the martingale analysis. Besides, it was improved further by SSA/DSSA [18] and OPIM [19].

Competitive IM and Profit Maximization: Bharathi *et al.* [6] studied the competitive IM first and generalized it

as a game of influence diffusion with multiple competing cascade. Lu *et al.* [20] created a comparative IC-model that includes all settings of influence propagation from competition to complementarity. Tong *et al.* [21] proposed an independent multi-cascade model and studied a multi-cascade IM problem under this model systematically, where they designed efficient algorithm and obtained a data-dependent approximation guarantee. In the classic PM problem [3] [22], they usually considered the cost of a seed set is modular with respect the seed node in this seed set, which implies the profit function is still submodular but not monotone. It can be generalized as the unconstrained submodular maximization problem, which can be addressed by the double greedy algorithm within $(1/3)$ -approximation and randomized double greedy algorithm within $(1/2)$ -approximation [23]. Tong *et al.* [24] considered the coupon allocation in the PM problem, and designed efficient randomized algorithms to achieve $(1/2 - \epsilon)$ -approximation with high probability. Guo *et al.* [25] proposed a budgeted coupon problem whose domain is constrained and provided a continuous double greedy algorithm with a valid approximation. However, in our model, the formulation of competitiveness and definition of benefit are different from one of the above works.

Non-submodular Maximization: However, many realistic problems derived from the IM do not satisfy the submodularity. For a monotone non-submodular function, we can use the supermodular degree [26] and curvature [27] to analyze the approximation of greedy algorithm to maximize it. Then, Lu *et al.* [20] devised a sandwich approximation framework, which can obtain a data-dependent approximation ratio by maximizing its submodular upper and submodular lower bounds, the return the solution that can maximize the original objective function as the final result. However, our objective function of the OEBI problem is not monotone. For a non-monotone non-submodular function, it can be decomposed into the difference of two submodular functions [28], which can be approximated effectively by the submodular-supermodular procedure [28] and modular-modular procedure [10]. In this paper, we design an efficient randomized algorithm to solve our OEBI problem with a satisfactory approximation guarantee based on the RIS and modular-modular procedure.

III. PROBLEM FORMULATION

In this section, we introduce the diffusion model first and then formulate the OEBI problem.

A. Diffusion Model and Realization

Let $G = (V, E)$ be a directed graph that represents a social network where where $V = \{v_1, v_2, \dots, v_n\}$ is the set of n users, $E = \{e_1, e_2, \dots, e_m\}$ is the set of m directed edges. For each directed edge $(u, v) \in E$, it models their friendship where u (resp. v) is an incoming neighbor (resp. outgoing neighbor) of v (resp. u). Moreover, the set of incoming neighbors (resp. outgoing neighbors) of node $u \in V$ is denoted by $N^-(v)$ (resp. $N^+(v)$).

Given a seed set $S \subseteq V$, the influence diffusion model is a discrete-time stochastic process started from the seed nodes

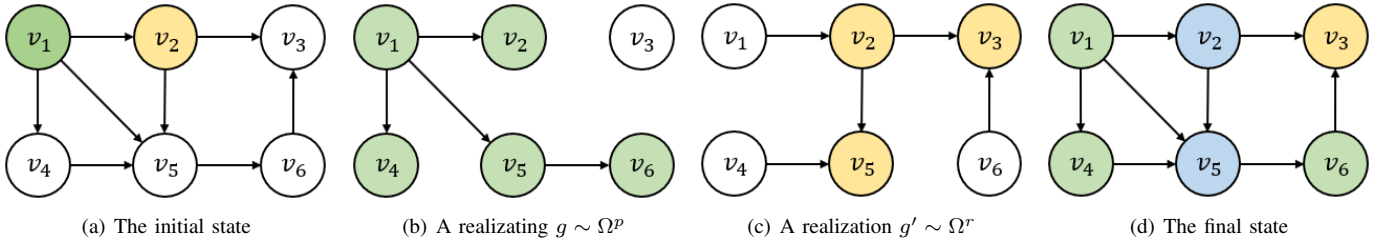


Fig. 1. This is an example to demonstrate the diffusion process caused by a positive cascade and a negative cascade, where the green nodes, yellow nodes, and blue nodes are activated by the positive cascade, rival cascade, and both positive and rival cascades.

in S . In the beginning, all nodes in the seed set S are active, but the other nodes are inactive. At time step t_i , we denote by S_i the current active node set. Thereby we have $S_0 := S$ at t_0 . Under the IC-model [2], there is a diffusion probability $p_{uv} \in (0, 1]$ associated with each edge $(u, v) \in E$. At time step t_i for $i \geq 1$, we have $S_i := S_{i-1}$ first; then, each new activated node $u \in (S_{i-1} \setminus S_{i-2})$ in the last time step has one chance to activate its each inactive outgoing neighbor v with the probability p_{uv} . We add v into S_i if u activates v successfully. The influence diffusion stops when no node can be activated further. The problems we will discuss in the subsequent sections are defaulted on the IC-model, but they can be extended to other influence models easily.

Here, a specific IC-model based on graph G can be defined as $\Omega = (G, P)$ where $P = \{p_{e_1}, p_{e_2}, \dots, p_{e_m}\}$ is the set of m edge probabilities. Given a specific IC-model Ω , we define $g \sim \Omega$ as a realization sampled from Ω , which is an instance of influence diffusion on this probabilistic graph. Under the IC-model, a realization is residual graph built by removing each edge $(u, v) \in E$ with probability $1 - p_{uv}$. Thereby we have $\Pr[g] = \prod_{e \in E(g)} p_e \prod_{e \in E(G) \setminus E(g)} (1 - p_e)$ and there is 2^m potential realizations in total.

Given a seed set $S \subseteq V$ and a realization g , we denote by $I_g(S)$ the set of nodes that can be reachable from at least one node in this seed set. Thus, the expected number of active nodes over all potential realizations (expected influence spread) can be expressed as

$$\sigma_\Omega(S) = \mathbb{E}_{g \sim \Omega} [|I_g(S)|] = \sum_{g \in \mathcal{G}(\Omega)} \Pr[g] \cdot |I_g(S)| \quad (1)$$

where $\mathcal{G}(\Omega)$ is the collection of all possible realizations sampled from Ω . The IM problem is to select a seed set $S \subseteq V$ where $|S| \leq k$ such that the expected influence spread $\sigma(S)$ can be maximized. Given a set function $h : 2^V \rightarrow \mathbb{R}$ and any two sets $S, T \subseteq V$, it is monotone if $h(S) \leq h(T)$ when $S \subseteq T \subseteq V$, submodular if $h(S \cup \{u\}) - h(S) \geq h(T \cup \{u\}) - h(T)$ when $S \subseteq T \subseteq V$ and $u \notin T$, and supermodular if $h(S \cup \{u\}) - h(S) \leq h(T \cup \{u\}) - h(T)$ when $S \subseteq T \subseteq V$ and $u \notin T$. Based on that, we have the expected influence spread $\sigma(\cdot)$ is monotone non-decreasing and submodular under the IC-model [2].

B. Problem Definition

Consider a company, it wants to promote its new product by starting a cascade diffusing over the social network. Obviously,

the expected influence spread is the benefit it can obtain. However, this is only in an ideal world because it does not consider whether there is the other cascade representing a competing product started by a rival company that diffuses over the social network at the same time. Thus, we can no longer evaluate this company's benefit only by the expected influence spread due to the rival's disturbance.

Given a social network $G = (V, E)$, there are multiple cascades diffusing on this network simultaneously. A user is referred as C -active if she is activated by cascade C . Consider such a scenario, we define a positive cascade C_p which represents the influence diffusion for the new product we want to promote over the network. It exists a rival cascade C_r represents the influence diffusion for a competing product started by some rival company. Now, due to the existence of this competing cascade, our benefit from the influence spread of cascade C_p will be disturbed and impaired to some extent. Given a rival seed set S_r , we need to find a positive seed set S_p and start this positive cascade such that it can avoid the negative effects of the rival cascade started from S_r as much as possible.

Next, we discuss how to quantify the disturbance caused by the rival cascade to our benefit. Given a social network $G = (V, E)$, we consider a positive cascade C_p diffuses under the IC-model $\Omega^p = (G, PP)$ and a rival cascade C_r diffuses under the IC-model $\Omega^r = (G, PR)$, where PP (resp. PR) is an edge probability distribution of Ω^p (resp. Ω^r). These two cascades diffuse over the network G respectively and independently. Then, we suppose each node $u \in V$ is associated with a benefit weight $p(u) \in \mathbb{R}_+$, which implies the benefit can be obtained from the fact that u is C_p -active but not C_r -active. In other words, it is the earning from activating user u by our positive cascade but not activating it by the rival cascade. Moreover, we suppose each node $u \in V$ is associated with a disturbed benefit weight $q(u) \in \mathbb{R}$ with $q(u) \leq p(u)$, which implies the earning can be obtained from the fact that u is C_p -active and C_r -active. Here, the disturbed benefit weight describes the degree of disturbance caused by the rival cascade. For a user $u \in V$, her degree of disturbance caused by the rival cascade rests with its disturbed benefit weight $q(u)$. If $q(u) \in [0, p(u)]$, it means that the rival cascade will not cause a negative effect on this node u even though it cuts down the benefit can be obtained from activating this node by positive cascade. If $q(u) \in (-\infty, 0)$, it means that the rival cascade will cause a negative effect on this node. Thus, this q controls the degree of disturbance caused by the rival cascade.

Given a rival seed set $S_r \subseteq V$, the expected overall benefit from our positive seed set S_p can be defined as

$$\begin{aligned} f(S_p) &= \mathbb{E}_{g \sim \Omega^p} \mathbb{E}_{g' \sim \Omega^r} [f_{g,g'}(S_p)] \quad (2) \\ &= \sum_{g \in \mathcal{G}(\Omega^p)} \Pr[g] \sum_{g' \in \mathcal{G}(\Omega^r)} \Pr[g'] \cdot f_{g,g'}(S_p) \quad (3) \end{aligned}$$

where $f(S_p)$ is the expectation over the realizations sampled from the IC-model Ω^p and Ω^r . Given the two realizations $g \sim \Omega^p$ and $g' \sim \Omega^r$, the overall benefit of influence diffusion can be defined as

$$f_{g,g'}(S_p) = \sum_{u \in I_g(S_p) \setminus I_{g'}(S_r)} p(u) + \sum_{u \in I_g(S_p) \cap I_{g'}(S_r)} q(u) \quad (4)$$

where the first term is the benefit from nodes activated only by C_p and the second term is the disturbed benefit from nodes activated by both C_p and C_r .

Let us look at an example shown in Fig. 1. Shown as Fig. 1(a), the positive seed set is $S_p = \{v_1\}$ and the rival seed set $S_r = \{v_2\}$ in the beginning. Then, the influence spread started from S_p is shown as Fig. 1(b), which is a realization sampled from its IC-model Ω^p . Similarly, the influence spread started from S_r is shown as Fig. 1(c), which is a realization sampled from its IC-model Ω^r . From here, we can see that they diffuse respectively and independently. Finally, node v_2 and v_5 are activated by both the positive and rival cascades, thereby we have $I_g(S_p) \cap I_{g'}(S_r) = \{v_2, v_5\}$ shown as Fig. 1(d). Therefore, we have the overall benefit under this realization is $f_{g,g'}(S_p) = p(v_1) + p(v_4) + p(v_6) + q(v_2) + q(v_5)$. The overall evaluations on benefit of influence (OEBI) problem is

Problem 1 (OEBI). *Given a social network $G = (V, E)$, a rival seed set S_r , and a budget k , the OEBI problem is aimed at finding a positive set set $S_p \subseteq V$, where $|S_p| \leq k$, such that its expected overall benefit $f(S_p)$ can be maximized, that is $S_p^* = \arg \max_{|S_p| \leq k} f(S_p)$.*

IV. FURTHER DISCUSSIONS ABOUT OEBI

In this section, we analyze the properties of OEBI first and introduce how to decompose its objective function.

A. The Properties

Given the rival seed set $S_r = \emptyset$, the OEBI problem can be reduced to the classical IM problem if we assume $p(u) = 1$ for each $u \in V$. Thus, the OEBI problem is NP-hard through inheriting the NP-hardness of IM problem [2] under the IC-model. Moreover, it is #P-hard to compute the expected overall benefit because of the #P-hardness to compute the expected influence spread under the IC-model [11]. Next, we will analyze the monotonicity, submodularity, and supermodularity of the expected overall benefit function $f(S_p)$ with respect to S_p step by step.

Theorem 1. *The objective function of the OEBI problem $f(S_p)$ is not monotone with respect to S_p .*

Proof. We consider the simplest case where the graph G has only one node. Here, we have $V = \{v\}$ and $E = \emptyset$. Given a rival seed set $S_r = \{v\}$, the expected overall benefit $f(\{v\}) =$

$q(u)$ and $f(\emptyset) = 0$. Subsequently, we have $f(\{v\}) - f(\emptyset) \geq 0$ if $q(u) \geq 0$; and $f(\{v\}) - f(\emptyset) \leq 0$ if $q(u) \leq 0$. Thus, the monotonicity of $f(S_p)$ depends on the definition of disturbed earning weights. \square

Theorem 2. *The objective function of the OEBI problem $f(S_p)$ is not submodular with respect to S_p and not supermodular with respect to S_p .*

Proof. Take a counterexample to prove it, we assume $p = p(u)$ and $q = q(u)$ for each node $u \in V$ with $q \in (-\infty, -p)$. Shown as Fig. 2, we can see that $f(\{v_2, v_4\}) = 2p - q$ and $f(\{v_1, v_4\}) = 5p - q$. First, we have $f(\{v_2, v_4\}) - f(v_4) = p + q < f(\{v_1, v_2, v_4\}) - f(v_1, v_4) = 0$, thereby $f(S_p)$ is not submodular with respect to S_p . Then, we have $f(\{v_4, v_5\}) - f(v_4) = 2p > f(\{v_1, v_4, v_5\}) - f(v_1, v_4) = 0$, thereby $f(S_p)$ is not supermodular with respect to S_p . \square

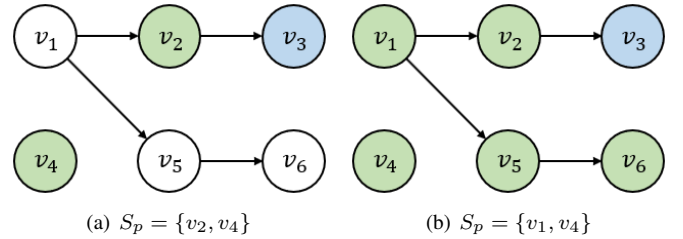


Fig. 2. This is an example to demonstrate the submodularity and supermodularity in Theorem 2.

B. Decomposition of Our Objective Function

From the above subsection, the expected overall benefit is non-monotone, non-submodular, and non-supermodular, therefore, it is hard to get an effective solution with an approximation ratio. Narasimhan *et al.* [28] proposed a DS decomposition, which pointed out any set function can be decomposed into the difference of two submodular set functions. Even that, whether such two submodular set functions can be found in polynomial time is still unknown. Look at the (4), the overall benefit $f_{g,g'}(S_p)$ under the $g \sim \Omega^p$ and $g' \sim \Omega^r$ can be re-arranged as

$$f_{g,g'}(S_p) = \sum_{u \in I_g(S_p)} p(u) - \sum_{u \in I_g(S_p) \cap I_{g'}(S_r)} (p(u) - q(u)) \quad (5)$$

Thus, we can decompose the expected overall benefit as $f(S_p) = w(S_p) - z(S_p)$, where $w(S_p)$ and $z(S_p)$ are defined as follows, that is

$$w(S_p) = \mathbb{E}_{g \sim \Omega^p} \left[\sum_{u \in I_g(S_p)} p(u) \right] \quad (6)$$

$$z(S_p) = \mathbb{E}_{g \sim \Omega^p} \mathbb{E}_{g' \sim \Omega^r} \left[\sum_{u \in I_g(S_p) \cap I_{g'}(S_r)} l(u) \right] \quad (7)$$

where we denote $l(u) = p(u) - q(u)$. Similarly, we denote $w_g(S_p) = \sum_{u \in I_g(S_p)} p(u)$ under the $g \sim \Omega^p$ and $z_{g,g'}(S_p) = \sum_{u \in I_g(S_p) \cap I_{g'}(S_r)} l(u)$ under the $g \sim \Omega^p$ and $g' \sim \Omega^r$.

Theorem 3. *The function $w(S_p)$ is monotone non-decreasing and submodular with respect to S_p .*

Algorithm 1 Modular-modular

Input: A set function $f : 2^V \rightarrow \mathbb{R}$

- 1: Initialize: $X^t \leftarrow \emptyset, t \leftarrow 0$
- 2: **while** $X^{t+1} \neq X^t$ **do**
- 3: Selects a permutation α^t that contains X^t where the element in X^t are ranked ahead
- 4: $X^{t+1} \leftarrow \arg \max_{|Y| \leq k} \left\{ h_{X^t, \alpha^t}^w(Y) - m_{X^t}^z(Y) \right\}$
- 5: $t \leftarrow t + 1$
- 6: **end while**
- 7: **return** X^t

Proof. The function $w(S_p)$ is the objective function of weighted IM problem. It can be reduced to weighted maximum set cover problem, which is monotone non-decreasing and submodular since $p(u) \geq 0$ for any $u \in V$. \square

Theorem 4. *The function $z(S_p)$ is monotone non-decreasing and submodular with respect to S_p .*

Proof. Given a rival seed set S_r , realization $g \sim \Omega^p$, and $g' \sim \Omega^r$, we consider the monotonicity and submodularity based on $z_{g, g'}(S_p)$. First, it is apparent that $z_{g, g'}(S_p)$ is monotone non-decreasing with respect to S_p . Then, there are two positive seed set S_p^1 and S_p^2 with $S_p^1 \subseteq S_p^2$. For any node in $I_{g'}(S_r)$, if it is reachable from node v but is not reachable from S_p^2 , it must not be reachable from S_p^1 since $S_p^1 \subseteq S_p^2$. Thereby we have $z_{g, g'}(S_p^1 \cup \{v\}) - z_{g, g'}(S_p^1) \geq z_{g, g'}(S_p^2 \cup \{v\}) - z_{g, g'}(S_p^2)$ because of $l(u) \geq 0$ for any $u \in V$, which implies that $z_{g, g'}(S_p)$ is submodular with respect to S_p . Besides, $y(S_p)$ is a linear combination of $z_{g, g'}(S_p)$, thus $z(S_p)$ is monotone non-decreasing and submodular. \square

Therefore, the expected overall benefit $f(S_p)$ has been decomposed into the difference of two monotone submodular functions $w(S_p)$ and $z(S_p)$ definitely.

V. ALGORITHM DEGISN AND SPEEDUP

From the last section, our objective function is not monotone, not submodular, and not supermodular. Fortunately, it can be decomposed into the difference of two monotone submodular functions. Iyer *et al.* [10] proposed a modular-modular procedure to minimize the difference between two submodular functions approximately. First, we need to define the modular upper bound and modular lower bound for a given submodular function.

A. Modular-modular Procedure

Given a submodular function $b(\cdot)$, it has two modular upper bounds based on a given set $X \subseteq V$, that is

$$m_{X,1}^b(Y) = b(X) - \sum_{j \in X \setminus Y} b(j|X \setminus j) + \sum_{j \in Y \setminus X} b(j|\emptyset) \quad (8)$$

$$m_{X,2}^b(Y) = b(X) - \sum_{j \in X \setminus Y} b(j|V \setminus j) + \sum_{j \in Y \setminus X} b(j|X) \quad (9)$$

where $b(S|T) = b(S \cup T) - b(T)$, $m_{X,1}^b(Y) \geq b(Y)$, and $m_{X,2}^b(Y) \geq b(Y)$. They are tight at set X , so we have $m_{X,1}^b(X) = m_{X,2}^b(X) = f(X)$.

Algorithm 2 ModularMax

Input: A permutation α^t and a set X^t

- 1: Initialize: a map $unitValue = \{\}$
- 2: Initialize: a set $X^{t+1} \leftarrow \emptyset$
- 3: $zero \leftarrow h_{X^t, \alpha^t}^w(\emptyset) - m_{X^t}^z(\emptyset)$
- 4: **for each** $u \in V$ **do**
- 5: $unitValue[u] \leftarrow h_{X^t, \alpha^t}^w(\{u\}) - m_{X^t}^z(\{u\}) - zero$
- 6: **end for**
- 7: **for** $i = 1$ to k **do**
- 8: Select $u^* \in \max_{u \in V \setminus X^{t+1}} unitValue[u]$
- 9: **if** $unitValue[u^*] < 0$ **then**
- 10: Break
- 11: **end if**
- 12: $X^{t+1} \leftarrow X^{t+1} \cup \{u^*\}$
- 13: **end for**
- 14: **return** X^{t+1}

Given a set $X \subseteq V$, we define a permutation α of V as $\alpha = \{\alpha(1), \alpha(2), \dots, \alpha(n)\}$ where η 's chain contains X . Denote by $S_i^\alpha = \{\alpha(1), \alpha(2), \dots, \alpha(i)\}$, we have $S_{|X|}^\alpha = X$, in other words, we put all the elements in X prior to the elements in $V \setminus X$. Then, we define

$$h_{X, \alpha}^b(\alpha(i)) = b(S_i^\alpha) - b(S_{i-1}^\alpha) \quad (10)$$

where $h_{X, \alpha}^b(Y) = \sum_{v \in Y} h_{X, \alpha}^b(v)$ and $h_{X, \alpha}^b(Y) \leq b(Y)$ for any $Y \subseteq V$. Here, $h_{X, \alpha}^b(Y)$ is a lower bound of $b(Y)$. It is tight at set X , so we have $h_{X, \alpha}^b(X) = b(X)$.

From the (6) and (7), we adopt the modular-modular procedure to solve it is formulated in Algorithm 1.

Theorem 5. *The objective function $f(X^t)$ is monotone non-decreasing with respect to t . If the $h_{X^t, \alpha^t}^w(Y) - m_{X^t}^z(Y)$ in line 4 of Algorithm 1 reaches a local maximum under the $O(n)$ different permutations α^t and both upper bounds, then the $f(Y)$ is a local maximum.*

Proof. Regardless of what the upper bound we use, at any round t , we have $f(X^{t+1}) = w(X^{t+1}) - z(X^{t+1}) \geq h_{X^t, \alpha^t}^w(X^{t+1}) - m_{X^t}^z(X^{t+1}) \geq h_{X^t, \alpha^t}^w(X^t) - m_{X^t}^z(X^t) = w(X^t) - z(X^t) = f(X^t)$ since the definitions of the upper and lower bounds and the tightness at set X^t .

Suppose the Algorithm 1 converges at $X^{t+1} = X^t$, we consider the $O(n)$ different permutations α^t which are placed with different elements at position $\alpha^t(|X^t|)$ and $\alpha^t(|X^{t+1}|)$. First, we have $h_{X^t, \alpha^t}^w(S_i^\alpha) = w(S_i^\alpha)$, $m_{X^t, 1}^z(X^t \setminus j) = z(X^t) - z(j|X^t \setminus j) = z(X^t \setminus j)$, and $m_{X^t, 2}^z(X^t \cup j) = z(X^t) + z(j|X^t) = z(X^t \cup j)$. At the convergence, we have $h_{X^t, \alpha^t}^w(X^t) - m_{X^t}^z(X^t) \geq h_{X^t, \alpha^t}^w(Y) - m_{X^t}^z(Y)$ for any $Y \subseteq V$ under the $O(n)$ different permutations α^t and both upper bounds. Given a α^t with $\alpha^t(|X^t|) = i$ and $\alpha^t(|X^t| + 1) = j$, we have $f(X^t) = w(X^t) - z(X^t) = h_{X^t, \alpha^t}^w(X^t) - m_{X^t, 1}^z(X^t) \geq h_{X^t, \alpha^t}^w(X^t \setminus i) - m_{X^t, 1}^z(X^t \setminus i) = f(X^t \setminus i)$ and $f(X^t) = w(X^t) - z(X^t) = h_{X^t, \alpha^t}^w(X^t) - m_{X^t, 2}^z(X^t) \geq h_{X^t, \alpha^t}^w(X^t \cup j) - m_{X^t, 1}^z(X^t \cup j) = f(X^t \cup j)$. Therefore, $f(X^t)$ is a local maximum at the convergence. \square

At each iteration in this algorithm, we need to maximize a modular function shown as in line 4 of Algorithm 1, which

can be implemented easily. For example, we can compute the objective value for each node $u \in V$ and then select all those which has a non-negative objective value. At the iteration t , given a permutation α^t and a set X^t , the algorithm that selects a set Y where $|Y| \leq k$ to maximize the modular function $h_{X^t, \alpha^t}^w(Y) - m_{X^t}^z(Y)$ is shown in Algorithm 2. The update rule in Algorithm 2 is according to $h(u|S) = h(u|T) = h(u|\emptyset)$ for any set $S, T \subseteq V$ if $h(\cdot)$ is a modular function.

As for how to select a permutation α^t at each iteration X^t , the optimal solution is to select a permutation α^* such that $\alpha^* \in \arg \max_{\alpha^t} \max_{|Y| \leq k} \{h_{X^t, \alpha^t}^w(Y) - m_{X^t}^z(Y)\}$, however it is very difficult to execute. There are $n!$ permutations in total. Thus, a heuristic choice is to order the permutation α^t according to the magnitude of objective value for each node $u \in V$. We will compare the impact of different permutations on algorithm performance in later experiments.

According to the (8) and (9), we have two upper bounds for a submodular function. Thereby the upper bound of the optimal value of our expected overall benefit $f(S_p^*)$ can be defined as follows:

$$\pi(X) = \max_{|Y| \leq k} \{ \min \{ m_{X,1}^w(Y), m_{X,2}^w(Y) \} - h_{X,\alpha}^z(Y) \} \quad (11)$$

where $\min \{ m_{X,1}^w(Y), m_{X,2}^w(Y) \}$ is aimed to make this upper bound tighter. It can be solved similar to the process of Algorithm 2. Then, for any set X , we have $\pi(X) \geq \max_{|Y| \leq k} f(Y)$. Denote by S_p^o the seed set returned by Algorithm 1, we have $\pi(S_p^o) \geq f(S_p^*)$, then we are able to estimate the approximation ratio by $f(S_p^o)/\pi(S_p^o)$.

B. Sampling Techniques

Given a seed set S_p , we adopt the technique of reverse influence sampling (RIS) to estimate $f(S_p)$ due to its #P-hardness. Consider the IM problem under the IC-model $\Omega = (G, P)$, we introduce the concept of reverse reachable set (RR-set) first. A random RR-set R can be generated by three steps: (1) selecting a node $u \in V$ uniformly; (2) sampling a realization $g \sim \Omega$; and (3) collecting those nodes in g can reach u and putting them into R . A RR-set rooted at node u is a collection of nodes that are likely to influence u . A larger expected influence spread a seed set S has, the higher the probability that S intersects with a random RR-set is. Given a seed set S and a random RR-set R , we have $\sigma_\Omega(S) = n \cdot \Pr[R \cap S \neq \emptyset]$.

Back to our OEI problem, the expected overall benefit can be denoted by $f(S_p) = w(S_p) - z(S_p)$. Thus, given a seed set S_p , we require to estimate $w(S_p)$ and $z(S_p)$ respectively. Here, we define $p(V) = \sum_{v \in V} p(v)$ and $l(V) = \sum_{v \in V} l(v)$ respectively for convenience. For the $w(S_p)$, a random RR-set R_w can be generated by (1) selecting a node $u \in V$ with probability $p(u)/p(V)$; (2) sampling a realization $g \sim \Omega^p$; and (3) putting those nodes in g can reach u into R_p . Given a seed set S_p and a random RR-set R_w , we have $w(S) = p(V) \cdot \Pr[R_w \cap S_p \neq \emptyset]$. For the $z(S_p)$, a random RR-set R_z can be generated by (1) selecting a node $u \in V$ with probability $l(u)/l(V)$; (2) sampling a realization $g \sim \Omega^p$ and a realization $g' \sim \Omega^r$ independently; and (3) putting those nodes in g can reach u into $R_{z,1}$ and those nodes in g' can reach u into $R_{z,2}$ where $R_z = (R_{z,1}, R_{z,2})$.

Lemma 1. *Given a seed set S_p , a rival seed set S_r , and a random RR-set $R_z = (R_{z,1}, R_{z,2})$, we have*

$$z(S_p) = l(V) \cdot \Pr[S_p \cap R_{z,1} \neq \emptyset \wedge S_r \cap R_{z,2} \neq \emptyset] \quad (12)$$

Proof. We denote by $R_{z,1}(g, u)$ the RR-set rooted at node u under the realization $g \sim \Omega^p$. From the (7), we have $z(S_p) = \mathbb{E}_{g \sim \Omega^p} \mathbb{E}_{g' \sim \Omega^r} [\sum_{u \in I_g(S_p) \cap I_{g'}(S_r)} l(u)] = \sum_{u \in V} \Pr_{g \sim \Omega^p, g' \sim \Omega^r} [S_p \cap R_{z,1}(g, u) \neq \emptyset \wedge S_r \cap R_{z,2}(g', u) \neq \emptyset] \cdot l(u) = l(V) \cdot \sum_{u \in V} \Pr_{g \sim \Omega^p, g' \sim \Omega^r} [S_p \cap R_{z,1}(g, u) \neq \emptyset \wedge S_r \cap R_{z,2}(g', u) \neq \emptyset] \cdot (l(u)/l(V)) = l(V) \cdot \Pr_{g \sim \Omega^p, g' \sim \Omega^r, u} [S_p \cap R_z(g, g', u) \neq \emptyset \wedge S_r \cap R_z(g, g', u) \neq \emptyset]$. The (12) is establish equivalently. \square

As mentioned above, we have to generate two collections of RR sets, $\mathcal{R}_w = \{R_w^1, R_w^2, \dots, R_w^\lambda\}$ to estimate $w(S_p)$ and $\mathcal{R}_z = \{R_z^1, R_z^2, \dots, R_z^\mu\}$ to estimate $z(S_p)$. Then we define the following two estimations

$$F_{\mathcal{R}_w}(S_p) = \frac{1}{\lambda} \cdot \sum_{i=1}^{\lambda} \mathbb{I}[S_p \cap R_w^i \neq \emptyset] \quad (13)$$

$$F_{\mathcal{R}_z}(S_p) = \frac{1}{\mu} \cdot \sum_{i=1}^{\mu} \mathbb{I}[S_p \cap R_{z,1}^i \neq \emptyset \wedge S_r \cap R_{z,2}^i \neq \emptyset] \quad (14)$$

the fraction of RR-sets covered by S_p where $\mathbb{I}[\cdot]$ is an indicator such that $\mathbb{I}[S_p \cap R_w^i \neq \emptyset] = 1$ if $S_p \cap R_w^i \neq \emptyset$, or else $\mathbb{I}[S_p \cap R_w^i \neq \emptyset] = 0$. Then, we have $\hat{w}(S_p) = p(V) \cdot F_{\mathcal{R}_w}(S_p)$, $\hat{z}(S_p) = l(V) \cdot F_{\mathcal{R}_z}(S_p)$, and $\hat{f}(S_p) = \hat{w}(S_p) - \hat{z}(S_p)$. Next, to bound the gap between ground-truth and estimator, we introduce the Chernoff-Hoeffding inequality.

Lemma 2 (Chernoff-Hoeffding). *Let $X_1, X_2, \dots, X_\theta$ be a series of random variables sampled from a distribution X with expectation $\mathbb{E}[X]$ independently and identically in the set $\{0, 1\}$. Given an error $\varepsilon > 0$, we have*

$$\Pr \left[\sum_{i=1}^{\theta} X_i - \theta \cdot \mathbb{E}[X] \geq +\varepsilon \right] \leq \exp \left(-\frac{2\varepsilon^2}{\theta} \right) \quad (15)$$

$$\Pr \left[\sum_{i=1}^{\theta} X_i - \theta \cdot \mathbb{E}[X] \leq -\varepsilon \right] \leq \exp \left(-\frac{2\varepsilon^2}{\theta} \right) \quad (16)$$

According to the Lemma 2, we can get the relationship between $F_{\mathcal{R}_w}(S_p)$ and its real value $w(S_p)$.

Lemma 3. *Given a collection of RR-sets \mathcal{R}_w with $|\mathcal{R}_w| = \lambda$ and any $\delta \in (0, 4)$, we have*

$$\Pr \left[w(S_p) \geq \hat{w}(S_p) - p(V) \sqrt{\frac{1}{2\lambda} \ln \left(\frac{4}{\delta} \right)} \right] \geq 1 - \frac{\delta}{4} \quad (17)$$

$$\Pr \left[w(S_p) \leq \hat{w}(S_p) + p(V) \sqrt{\frac{1}{2\lambda} \ln \left(\frac{4}{\delta} \right)} \right] \geq 1 - \frac{\delta}{4} \quad (18)$$

Proof. To the (17), it is equivalent to prove $\Pr[w(S_p) < \hat{w}(S_p) - p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)}] \leq \delta/4$. Then, we have $\Pr[w(S_p) < p(V) \cdot F_{\mathcal{R}_w}(S_p) - p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)}] = \Pr[\lambda \cdot F_{\mathcal{R}_w}(S_p) - \lambda \cdot w(S_p)/p(V) > \sqrt{(\lambda/2) \ln(4/\delta)}] \leq \exp(-2 \cdot (\lambda/2) \ln(4/\delta)/\lambda) = \delta/4$ based on the (15).

Similarly, to the (18), it is equivalent to prove $\Pr[w(S_p) > \hat{w}(S_p) + p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)}] \leq \delta/4$. Then, we have $\Pr[w(S_p) > p(V) \cdot F_{\mathcal{R}_w}(S_p) + p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)}] =$

$\Pr[\lambda \cdot F_{\mathcal{R}_w}(S_p) - \lambda \cdot w(S_p)/p(V) < -\sqrt{(\lambda/2) \ln(4/\delta)}] \leq \exp(-2 \cdot (\lambda/2) \ln(4/\delta)/\lambda) = \delta/4$ based on the (16). \square

Given an unbiased estimator $\hat{w}(S_p)$, an upper bound and a lower bound of $w(S_p)$ can be defined with at least $1 - \delta/4$ probability. Given an unbiased estimator $\hat{z}(S_p)$, an upper bound and a lower bound of $z(S_p)$ can be defined with at least $1 - \delta/4$ probability. That is

$$w_u(S_p) = \hat{w}(S_p) + p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)} \quad (19)$$

$$w_l(S_p) = \hat{w}(S_p) - p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)} \quad (20)$$

Given a collection of RR-sets \mathcal{R}_z with $|\mathcal{R}_z| = \mu$, any $\delta \in (0, 4)$, and an unbiased estimator $\hat{z}(S_p)$, an upper bound and a lower bound of $z(S_p)$ can be defined with at least $1 - \delta/4$ probability in the same way. That is

$$z_u(S_p) = \hat{z}(S_p) + l(V) \cdot \sqrt{(1/(2\mu)) \ln(4/\delta)} \quad (21)$$

$$z_l(S_p) = \hat{z}(S_p) - l(V) \cdot \sqrt{(1/(2\mu)) \ln(4/\delta)} \quad (22)$$

Based on the (19)–(21), we can derive a lower bound for our objective value $f(S_p)$ naturally.

Lemma 4. *Given any seed set $S_p \subseteq V$, we can take $w_u(S_p) - z_l(S_p)$ as an upper bound of $f(S_p)$ with at least $1 - \delta/2$ probability and $w_l(S_p) - z_u(S_p)$ as a lower bound of $f(S_p)$ with at least $1 - \delta/2$ probability.*

Proof. To estimate the $f(S_p)$, we have $\Pr[f(S_p) \leq w_u(S_p) - z_l(S_p)] \geq \Pr[(w(S_p) \leq w_u(S_p)) \wedge (z(S_p) \geq z_l(S_p))] = (1 - \delta/4) \cdot (1 - \delta/4) \geq 1 - \delta/2$. Similarly, we have $\Pr[f(S_p) \geq w_l(S_p) - z_u(S_p)] \geq \Pr[(w(S_p) \geq w_l(S_p)) \wedge (z(S_p) \leq z_u(S_p))] = (1 - \delta/4) \cdot (1 - \delta/4) \geq 1 - \delta/2$. \square

Next, we are going to discuss how to compute the upper bound of our objective value $\pi(S_p^\circ)$ according to the solution S_p° returned by Algorithm 1. The value of $\hat{\pi}(S_p)$ can be obtained by $\hat{f}(S_p)$, which has been decomposed as $\hat{f}(S_p) = \hat{w}(S_p) - \hat{z}(S_p)$. Here, $\hat{w}(S_p)$ and $\hat{z}(S_p)$ are monotone and submodular with respect to S_p as well since they can be reduced to the set coverage problem. Therefore, for any set X , we have $\hat{\pi}(X) \geq \max_{|Y| \leq k} \hat{f}(Y)$. From the Lemma 4, the objective value $f(S_p)$ is upper bounded by $w_u(S_p) - z_l(S_p)$ with a high probability. Thereby we have the following conclusion.

Lemma 5. *Given the solution S_p° returned by Algorithm 1, for any seed set $S_p \subseteq V$ and any $\delta \in (0, 4)$, we have*

$$f(S_p) \leq \hat{\pi}(S_p^\circ) + p(V) \sqrt{\frac{1}{2\lambda} \ln\left(\frac{4}{\delta}\right)} + l(V) \sqrt{\frac{1}{2\mu} \ln\left(\frac{4}{\delta}\right)} \quad (23)$$

holds with at least $1 - 2/\delta$ probability.

Proof. According to the Lemma 4, we have $\Pr[f(S_p) \leq w_u(S_p) - z_l(S_p)] \geq 1 - \delta/2$. Then, $f(S_p) \leq w_u(S_p) - z_l(S_p) = \hat{w}(S_p) - \hat{z}(S_p) + p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)} + l(V) \cdot \sqrt{(1/(2\mu)) \ln(4/\delta)} = \hat{f}(S_p) + p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)} + l(V) \cdot \sqrt{(1/(2\mu)) \ln(4/\delta)} \leq \hat{\pi}(S_p^\circ) + p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)} + l(V) \cdot \sqrt{(1/(2\mu)) \ln(4/\delta)}$, which holds with at least $1 - \delta/2$ probability. \square

Theorem 6. *The approximation guarantee achieved by the solution S_p° returned by Algorithm 1 satisfies as follows: $f(S_p^\circ)/\max_{|S_p| \leq k} f(S_p) \geq$*

$$\frac{w_l(S_p^\circ) - z_u(S_p^\circ)}{\hat{\pi}(S_p^\circ) + p(V) \sqrt{\frac{1}{2\lambda} \ln\left(\frac{4}{\delta}\right)} + l(V) \sqrt{\frac{1}{2\mu} \ln\left(\frac{4}{\delta}\right)}} \quad (24)$$

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

Proof. Based on the Lemma 4, we have $f(S_p^\circ) \geq w_l(S_p^\circ) - z_u(S_p^\circ)$ holds with at least $1 - \delta/2$ probability. Then based on the Lemma 5, we have $\max_{|S_p| \leq k} f(S_p) \leq \hat{\pi}(S_p^\circ) + p(V) \cdot \sqrt{(1/(2\lambda)) \ln(4/\delta)} + l(V) \cdot \sqrt{(1/(2\mu)) \ln(4/\delta)}$ holds with at least $1 - \delta/2$ probability. Thereby the approximation (24) is established with at least $1 - \delta$ probability. \square

VI. NUMERICAL EXPERIMENTS

In this section, we carry out several experiments on different datasets to validate the performance of our proposed algorithms. It aims to test the efficiency of modular-modular procedure, shown as Algorithm 1, and its effectiveness compared to other heuristic algorithms. All of our experiments are programmed by python, and run on Windows machine with a 3.40GHz, 4 core Intel CPU and 16GB RAM. There are four datasets used in our experiments: (1) NetScience [29]: a co-authorship network, co-authorship among scientists to publish papers about network science; (2) Wiki [29]: a who-votes-on-whom network, which comes from the collection Wikipedia voting; (3) Bitcoin [30]: a who-trusts-whom network of people who trade using Bitcoin on a platform called Bitcoin Alpha. The statistics information about these four datasets is represented in Table I. For an undirected graph, each undirected edge is replaced with two reversed directed edges.

TABLE I
THE DATASETS STATISTICS ($K = 10^3$)

Dataset	n	m	Type	Avg.Degree
Netscie	0.40 K	1.01 K	undirect	5.00
Wikivot	1.00 K	3.15 K	directed	6.20
Bitcoin	4.00 K	25.1 K	directed	12.5

A. Experimental Settings

The diffusion process is based on the IC-model by default. Under the IC-model, we set the diffusion probability $p_{uv} = 1/|N^-(v)|$ for each $(u, v) \in E$ as the inverse of v 's in-degree, which has been given by many existing researches about the IM problem. For each node $u \in V$, there is a benefit weight and a disturbed wight associated with it. We sample its benefit weight $p(u)$ from $[0, 1]$ uniformly and sample its disturbed benefit weight $q(u)$ from $[-1, p(u)]$ uniformly.

Consider the modular-modular procedure, we have to define a modular lower bound for the function $w(\cdot)$ and a modular upper bound for the function $z(\cdot)$. Here, we denote ‘‘modmod-1’’ to imply that we use the first upper bound $m_{X,1}^{\hat{z}}(Y)$ defined in (8) and ‘‘modmod-2’’ to imply that we use the second upper

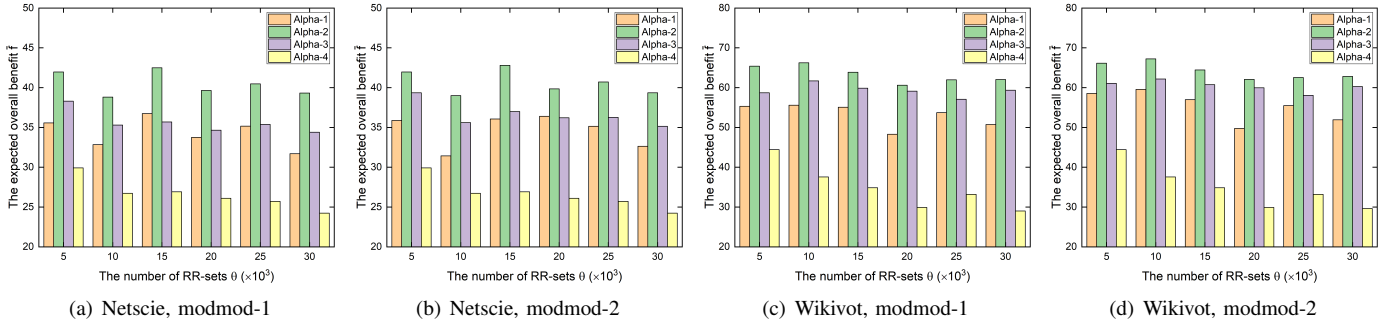


Fig. 3. The performance comparison of four permutation selections under the different datasets and upperbounds.

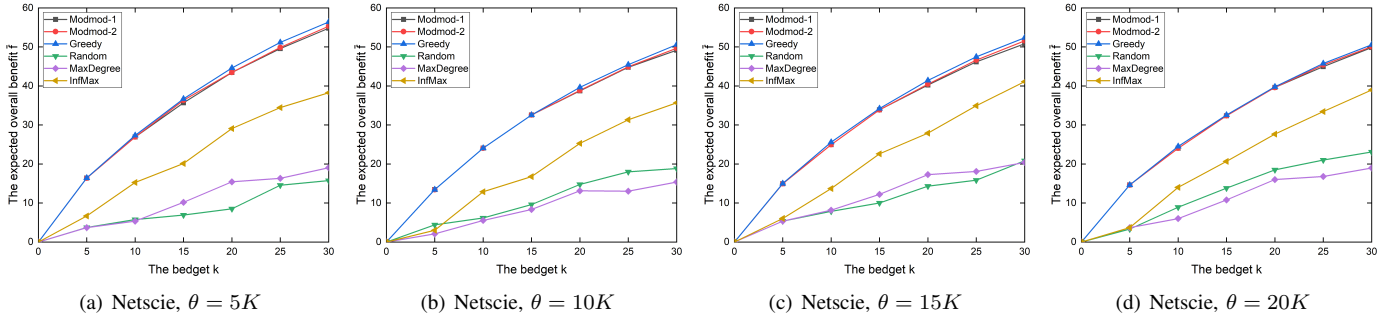


Fig. 4. The performance comparison with other heuristic algorithms under the Netscie dataset.

bound $m_{X,2}^z(Y)$ defined in (9). Then, we need to compare our modular-modular procedure with other heuristic algorithms, especially for the greedy algorithm. The greedy algorithm is shown in Algorithm 3, which selects the node with the maximum marginal expected overall benefit at each iteration until there is no positive marginal gain can be obtained. Other heuristic algorithms are shown as follows: (1) Random: it selects k nodes uniformly from the node set; (2) MaxDegree: it selects k nodes with the largest out-degree; and (3) InfMax: it is similar to the greedy algorithm, but substitutes the overall benefit $f(\cdot)$ with benefit $w(\cdot)$. They are all estimated on the same group of RR-sets, where the number of random RR-set R_w and R_z is denoted by $\theta = \lambda = \mu$.

Algorithm 3 Greedy

Input: A set function $f : 2^V \rightarrow \mathbb{R}$

- 1: Initialize: $S_p \leftarrow \emptyset$
- 2: **for** $i = 1$ to k **do**
- 3: Select u^* such that $u^* \in \arg \max_{u \in V \setminus S_p} f(u|S_p)$
- 4: **if** $f(u^*|S_p) < 0$ **then**
- 5: Break
- 6: **end if**
- 7: $S_p \leftarrow S_p \cup \{u^*\}$
- 8: **end for**
- 9: **return** S_p

To get a lower bound, the optimal permutation selections is very hard, thus we give several heuristic strategies to get that efficiently. For the permutation α^t that contains X^t at each iteration, there are four heuristic selection strategies. They are (1) Alpha-1: rearrange X^t and $V \setminus X^t$ randomly

and respectively, and then concatenate them together as a α^t ; (2) Alpha-2: sort X^t and $V \setminus X^t$ respectively from largest to smallest according to the expected overall benefit $f(u)$ for each $u \in V$, and then concatenate them together as a α^t ; (3) Alpha-3: sort X^t and $V \setminus X^t$ respectively from largest to smallest according to the expected benefit $w(u)$ for each $u \in V$, and then concatenate them together as a α^t ; and (4) Alpha-4: sort X^t and $V \setminus X^t$ respectively from smallest to largest according to the $z(u)$ for each $u \in V$, and then concatenate them together as a α^t .

B. Experimental Results

1) *Permutation selections:* Fig. 3 shows the performance comparison of modular-modular procedure under the aforementioned four permutation selections. Shown as Fig. 3, the solution achieved under the Alpha-2 that permutes according to the expected overall benefit has the best performance. Thus, in the follow-up experiments, we default that modular-modular procedure is implemented under the Alpha-2. The performance under the Alpha-3 is slightly worse that under the Alpha-2. The performance under the Alpha-4 is extremely worse, which implies this heuristic selection is invalid. Moreover, the random permutation selection Alpha-1 is unstable, which is sometimes good sometimes bad.

2) *Performance of different algorithms:* Fig. 4, Fig. 5, and Fig. 6 show the performance comparison with other heuristic algorithms under the different datasets. In these figures, we test the algorithms under the different number of RR-sets. Obviously, the estimations will be more and more accurate as the number of RR-sets increases, but the gap looks inconspicuous from these figures. Then, we have several observations

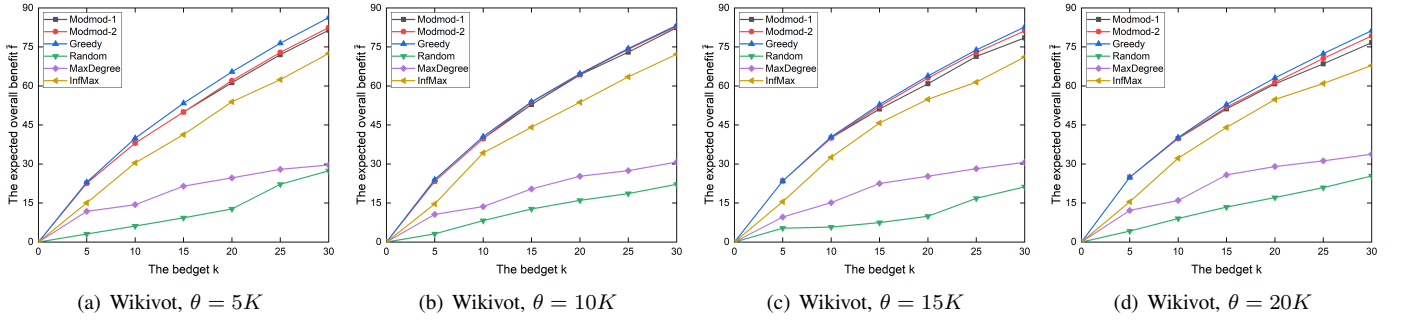


Fig. 5. The performance comparison with other heuristic algorithms under the Wikivot dataset.

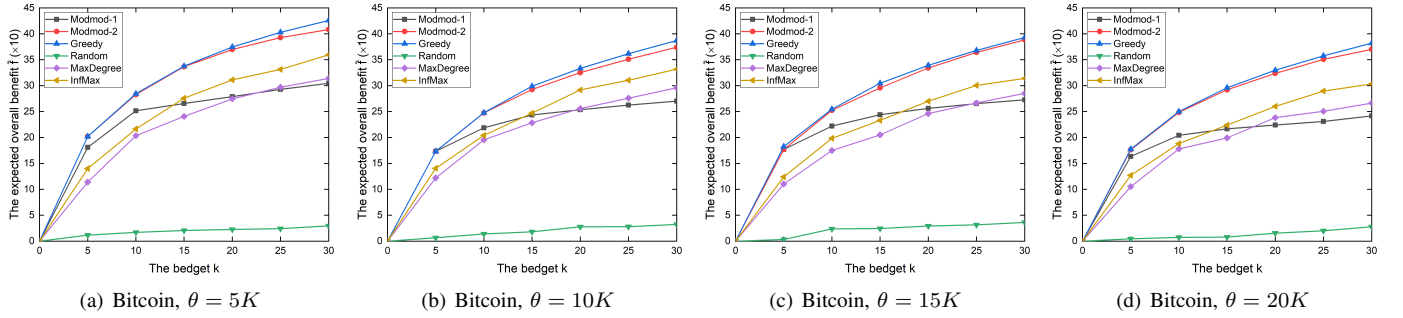


Fig. 6. The performance comparison with other heuristic algorithms under the Bitcoin dataset.

TABLE II

 APPROXIMATION OF MODULAR-MODULAR PROCEDURE WHEN $k = 20$

θ	Netscie		Wikivot		Bitcoin	
	md-1	md-2	md-1	md-2	md-1	md-2
5 K	0.51	0.51	0.44	0.44	0.31	0.41
10K	0.50	0.50	0.47	0.47	0.31	0.42
15K	0.50	0.51	0.50	0.50	0.32	0.42
20K	0.52	0.53	0.51	0.51	0.32	0.45

TABLE III

 RUNNING TIME OF MODULAR-MODULAR PROCEDURE WHEN $k = 20$

θ	Netscie		Wikivot		Bitcoin	
	md-1	md-2	md-1	md-2	md-1	md-2
5 K	09	28	24	083	255	0935
10K	17	53	44	154	232	1190
15K	23	64	65	410	445	2587
20K	27	57	82	285	535	2481

as follows. First, the expected overall benefit increases as the budget increases at least on a budget less than 30. Then, the performances achieved by greedy and modmod-2 algorithms are very close under all datasets. The performances achieved by modmod-1 are unstable under the different datasets, which has good results under the Netscie and Wikivot datasets but a bad result under the Bitcoin dataset. It implies that the

selection of upper bound is a critical factor that affects the results of the modular-modular procedure.

C. Approximation and Running Time:

The approximation and running time of modular-modular procedure when $k = 20$ are shown in Table II and Table III. Here, we set the parameter $\delta = 0.1$, which means that the approximation ratio shown as II can be satisfied with at least 0.9 probability. From the Table II, we can see that the approximation ratio improves as the number of RR-sets increases since the estimation errors in (23) can be reduced. From the table III, the running time increases as the number of RR-sets increases generally because the modular maximization process shown as Algorithm 2 is more time-consuming. However, it is still uncertain since the number of iterations varies under different circumstances, where modmod-2 needs to update X^t more times than modmod-1.

VII. CONCLUSIONS

In this paper, we consider the disturbance of rival's influence on our benefits we can get from the social networks and propose an OEBI problem formally, which is a generalization for a number of realistic scenarios. Then, we quantify this disturbance, define its objective function, and show its properties. To solve it, we decompose it into the difference of two submodular functions and apply modular-modular procedure to get a solution according to their lower bound and upper bound. Then, we design an efficient unbiased estimate to approximate it with a data-dependent approximation guarantee but reduce running time significantly. These results are verified by numerical simulations based on real-world datasets.

ACKNOWLEDGMENT

This work is partly supported by National Science Foundation under grant 1747818 and 1907472.

REFERENCES

- [1] Z. Zhang, R. Sun, X. Wang, and C. Zhao, "A situational analytic method for user behavior pattern in multimedia social networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 520–528, 2017.
- [2] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137–146.
- [3] W. Lu and L. V. Lakshmanan, "Profit maximization over social networks," in *2012 IEEE 12th International Conference on Data Mining*, IEEE, 2012, pp. 479–488.
- [4] Y. Dong, Z. Ding, F. Chiclana, and E. Herrera-Viedma, "Dynamics of public opinions in an online and offline social network," *IEEE Transactions on Big Data*, pp. 1–1, 2017.
- [5] J. Guo, T. Chen, and W. Wu, "Continuous activity maximization in online social networks," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.
- [6] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *International workshop on web and internet economics*. Springer, 2007, pp. 306–311.
- [7] J. Guo and W. Wu, "A novel scene of viral marketing for complementary products," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 4, pp. 797–808, 2019.
- [8] G. A. Tong, W. Wu, L. Guo, D. Li, C. Liu, B. Liu, and D.-Z. Du, "An efficient randomized algorithm for rumor blocking in online social networks," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.
- [9] J. Guo, T. Chen, and W. Wu, "A multi-feature diffusion model: Rumor blocking in social networks," *arXiv preprint arXiv:1912.03481*, 2019.
- [10] R. Iyer and J. Bilmes, "Algorithms for approximate minimization of the difference between submodular functions, with applications," in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012, pp. 407–417.
- [11] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 1029–1038.
- [12] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *2010 IEEE international conference on data mining*. IEEE, 2010, pp. 88–97.
- [13] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2014, pp. 946–957.
- [14] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 420–429.
- [15] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 199–208.
- [16] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 2014, pp. 75–86.
- [17] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1539–1554.
- [18] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 695–710.
- [19] J. Tang, X. Tang, X. Xiao, and J. Yuan, "Online processing algorithms for influence maximization," in *Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 991–1005.
- [20] W. Lu, W. Chen, and L. V. Lakshmanan, "From competition to complementarity: comparative influence diffusion and maximization," *Proceedings of the VLDB Endowment*, vol. 9, no. 2, pp. 60–71, 2015.
- [21] G. Tong, R. Wang, and Z. Dong, "On multi-cascade influence maximization: Model, hardness and algorithmic framework," *arXiv preprint arXiv:1912.00272*, 2019.
- [22] J. Tang, X. Tang, and J. Yuan, "Profit maximization for viral marketing in online social networks," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE, 2016, pp. 1–10.
- [23] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," *SIAM Journal on Computing*, vol. 44, no. 5, pp. 1384–1402, 2015.
- [24] G. Tong, W. Wu, and D.-Z. Du, "Coupon advertising in online social systems: Algorithms and sampling techniques," *arXiv preprint arXiv:1802.06946*, 2018.
- [25] J. Guo, T. Chen, and W. Wu, "Budgeted coupon advertisement problem: Algorithm and robust analysis," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.
- [26] U. Feige and R. Izsak, "Welfare maximization and the supermodular degree," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 2013, pp. 247–256.
- [27] Z. Wang, B. Moran, X. Wang, and Q. Pan, "Approximation for maximizing monotone non-decreasing set functions with a greedy method," *Journal of Combinatorial Optimization*, vol. 31, no. 1, pp. 29–43, 2016.
- [28] M. Narasimhan and J. Bilmes, "A submodular-supermodular procedure with applications to discriminative structure learning," in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, 2005, pp. 404–412.
- [29] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015. [Online]. Available: <http://networkrepository.com>
- [30] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," <http://snap.stanford.edu/data>, jun 2014.



Jianxiong Guo is a Ph.D. candidate in the Department of Computer Science at the University of Texas at Dallas. He received his B.S. degree in Energy Engineering and Automation from South China University of Technology in 2015 and M.S. degree in Chemical Engineering from University of Pittsburgh in 2016. His research interests include social networks, data mining, IoT application, blockchain, and combinatorial optimization.



Yapu Zhang received the B.S. degree in Mathematics and Applied Mathematics from Northwest University, Xi'an, China, in 2016. She is a Ph.D. candidate in the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China. Her research interests include social networks and approximation algorithms.



Weili Wu received the Ph.D. and M.S. degrees from the Department of Computer Science, University of Minnesota, Minneapolis, MN, USA, in 2002 and 1998, respectively. She is currently a Full Professor with the Department of Computer Science, The University of Texas at Dallas, Richardson, TX, USA. Her research mainly deals in the general research area of data communication and data management. Her research focuses on the design and analysis of algorithms for optimization problems that occur in wireless networking environments and various

database systems.