Maximizing Multifaceted Network Influence

Yuchen Li♯, Ju Fan†, George V. Ovchinnikov#, Panagiotis Karras*
♯School of Information Systems, Singapore Management University
♯yuchenli@smu.edu.sg
†DEKE Lab & School of Information, Renmin University of China
†fanj@ruc.edu.cn
#Skolkovo Institute of Science and Technology
#ovgeorge@yandex.ru
*Department of Computer Science, Aarhus University
*panos@cs.au.dk

***Abstract*—An information dissemination campaign is often multifaceted, involving several facets or pieces of information disseminating from different sources. The question then arises, how should we assign such pieces to eligible sources so as to achieve the best viral dissemination results? Past research has studied the problem of *Influence Maximization* (IM), which is to select a set of k promoters that maximizes the expected reach of a message over a network. However, in this classical IM problem, each promoter spreads out the same unitary piece of information.**

In this paper, we propose the Optimal Influential Pieces Assignment ($\mathcal{OIPA}$) problem, which is to assign k distinct pieces of an information campaign T to k promoters, so as to achieve the highest viral adoption in a network. We express adoption by users with a logistic model, and show that approximating $\mathcal{OIPA}$ within any constant factor is NP-hard. Even so, we propose a *branch-and-bound* framework for $\mathcal{OIPA}$ with an $(1-1/e)$ approximation ratio. We further optimize this framework with a pruning-intensive *progressive upper-bound estimation* approach, yielding a $(1-1/e-\varepsilon)$ approximation ratio and significantly lower time complexity, as it relies on the power-law properties of real-world social networks to run efficiently. Our extensive experiments on several real-world datasets show that the proposed approaches consistently outperform intuitive baselines, adopted from state-of-the-art IM algorithms. Furthermore, the progressive approach demonstrates superior efficiency with an up to 24-fold speedup over the plain branch-and-bound approach.

I. Introduction

Campaigning in Social Media (SM) is an indispensable tool for promoting business, advocating political standpoints, and spreading information for the social good. The problem of *influence Maximization* (IM) [16] asks to find a set of k initial promoters that maximize the spread of a *unitary* message propagating among users in a network. This problem has been extensively studied over the last decade [16], [33], [7], [32] in the context of viral marketing, network monitoring [19] and recommendation [30]; a survey is given in [21].

However, real-world campaigns often need to be multifaceted: composed of multiple constituent message pieces. For instance, an election campaign aims to inform voters regarding a candidate's policy statements on multiple issues, including taxation, immigration, and healthcare. Studies in consumer behavior confirm that it is *unlikely* to trigger any meaningful actions (e.g., vote for a candidate or adopt a product) when a user only receives a single element of the campaign [34], [13], [11], [18], [29]. In another example, a Youtube channel may try to increase the number of subscribers by spreading viral videos on SM like Facebook or Twitter. Due to the short-lived effect of SM contents [6], the viral video could quickly fade out from a user's memory even if she has watched it and shared it with friends. Only upon watching multiple videos from the same channel would the user turn to a subscriber. Thus, a campaign strategy should spread multiple viral messages aiming to achieve *overlapping* adoption results [1]. An early example of an such approach was a 1919 marketing campaign for a deodorant Odorono by Walter Thompson. Initial research indicated that consumers insisted that they did not need such a product. Thereafter, the campaign set a goal of consumer education independently of product promotion, suggesting that a nonuser "may offend without knowing it"; that caused an outrage at the time, but enormously improved sales. In our terms, a multifaceted promotion campaign was used, as the focus shifted from the product to a wider social concern.

In this paper, we propose the *Optimal Influential Pieces Assignment* ($\mathcal{OIPA}$) problem. Given a multifaceted campaign $\mathbf{T}$ that consists of ℓ pieces, we need to assign k_i promoters to spread the ith piece, where $\sum_i k_i = k$, so as to maximize the number of users who adopt the goal of $\mathbf{T}$ after receiving a subset of the pieces. We assume that each piece is characterized by a unique topic distribution and spreads in the network in a topic-aware manner, governed by topic-dependent influence probabilities between any two users, as in [31], [3]. The influence spread of a promoter with respect to a piece $\mathbf{t} \in \mathbf{T}$ depends on the piece's unique topic distribution; it follows that each promoter may be better positioned to spread one piece than others. We model a user's adoption behavior when receiving multiple pieces by a *logistic* activation model, as in consumer behavior studies [34], [13], [11], [18], [29]: the adoption probability is small when a user receives little information, but increases drastically upon seeing more; after exposure to a sufficient number of pieces, the probability gradient declines as the effect of extra information diminishes.

As an effect of the logistic activation model, the objective function in $\mathcal{OIPA}$ is non-submodular and thus techniques based on a submodularity assumption are inapplicable. Further, we show that it is **NP**-hard to approximate $\mathcal{OIPA}$ within any constant factor, using a novel reduction from the maximum clique problem. Still, we propose a *branch-and-bound* framework and introduce a novel formulation of monotone submodular optimization, which can be approximated with factor of $(1-1/e)$ by means of a greedy heuristic, to obtain a tight upper bound on the unexplored search space. Thereby, we iteratively solve a tractable optimization problem until we compute an upper bound smaller than or equal to the best obtained solution. Thus, this branch-and-bound framework thus guarantees an $(1-1/e)$ approximation ratio. Still, the upper bound estimation procedure is expensive, as it repeatedly scans $O(kn)$ users to select a set of promoters (where n is the number of SM users). We render it more efficient by means of a novel progressive upper bound estimation approach. Instead of scanning a large number candidate promoters, we select a promoter only if its marginal gain in terms of user adoption is larger than a predefined threshold h. By progressively lowering this threshold, we select more promoters as the budget k allows. Due to the power-law principle of social influence, we show that this progressive approach significantly lowers the computational complexity, while ensuring an approximation ratio of $(1-1/e-\varepsilon)$ where ε is a tunable parameter that trades effectiveness with efficiency.

We summarize our contributions as follows:

- We propose a novel $\mathcal{OIPA}$ problem, which is to assign candidate promoters to the propagated facets of a multifaceted campaign in a way that maximizes adoption, while capturing user behavior with a logistic model.
- We show that $\mathcal{OIPA}$ is NP-hard, and also NP-hard to be approximated within any constant factor.
- We propose a branch-and-bound framework for $\mathcal{OIPA}$, and introduce a novel formulation of monotone submodular optimization to tightly bound the unexplored search space for effective pruning. This framework provides an $(1-1/e)$ approximation guarantee.
- We devise an efficient progressive approach for the upper bound evaluation, which achieves an approximation ratio of $(1-1/e-\varepsilon)$ and drastically reduces time complexity.
- We experimentally evaluate the effectiveness and efficiency of our approaches on three real world datasets. The proposed approaches achieve over 215% quality improvement against two intuitive baselines adapted from state-of-the-art IM algorithms, while the progressive approach achieves a 24-fold speedup compared to the plain branch-and-bound approach.

The rest of the paper is organized as follows. We present related work in Section II. Section III introduces preliminaries and the $\mathcal{OIPA}$ problem definition. We present the hardness result in Section IV. We propose the branch-and-bound framework and the progressive upper bound estimation approach in Section V. Section VI reports on our experimental study, and Section VII concludes the paper.

II. Related Work

In this section, we review related works in native, competitive, and comparative Influence Maximization (IM).

Native IM: The objective of the well-studied IM problem [21] is to find a set S of k promoters in a network that maximizes the expected number of influenced users, denoted as $\sigma(S)$. While IM is NP-hard under the popular independent cascade (IC) and linear threshold (LT) influence models, $\sigma(\cdot)$ is a *monotone*[1] and *submodular*[2] function for both IC and LT, hence a constant approximation factor holds for a simple greedy algorithm [16].

Classical influence models treat different viral messages as interchangeable in terms of their influence spread under the model. In contrast, *topic-aware* models differentiate each unique message's influence in a topic-dependent manner [31], [3]. The *topic-aware* IM problem [2] aims to find promoters under such a topic-aware influence model.

All aforementioned works are concerned with the problem of spreading a *unitary* message, as opposed to a message consisting of multiple parts as in $\mathcal{OIPA}$. Still, topic-aware influence models are useful in describing the propagation of message parts. Yet, solutions for native IM problems cannot be extended to address $\mathcal{OIPA}$, in which the adoption probability function is *non-submodular*.

Competitive IM: *Competitive* IM considers a scenario in which each of several competitors spreads a message in the same network, and each user adopts at most one message. Past research has proposed three main objectives for competitive IM problems: (1) maximizing one competitor's influence spread given the opponents' strategies in choosing promoters [4], [8], [14]; (2) finding the equilibrium of a competition using game-theoretical concepts, when the opponents' strategies are unknown beforehand [20], [23]; (3) maximizing the total influence spread of all competitors, which is desirable from the perspective of a network host who is interested to allocate competing campaigns fairly [24], [17].

Competitive IM considers multiple propagating units, like $\mathcal{OIPA}$ does, yet each of these units is a message belonging to a different competitive campaign; they are not pieces constituting a single multifaceted campaign; thus, the competitive IM problem is fundamentally different from $\mathcal{OIPA}$.

Comparative IM: Some works have considered viral marketing with multiple non-competing messages. Datta et al. [10] study the case of campaigns whose influence spreads are independent. Narayanam et al. [27] study a scenario with two sets of *complementary* products, where a product can be adopted only by a user who has already adopted its corresponding product in the other set. Lu et al. [25] introduce a *comparative* influence model which subsumes both competitive and complementary IM: they consider two different kinds of relationships between two campaigns, A and B: in a *competitive* relationship, a user's adoption of A *lowers* the probability to adopt B; in a *complementary* relationship, a user's adoption of A *raises* the probability to adopt B. Two IM problems arise with this comparative influence model: SELF-INFMAX, which is to maximize a campaign's own influence;

[1]A set function g is monotone if for all $A \subseteq B$, $g(A) \leq g(B)$.

[2]A set function g is submodular if for all $A \subseteq B$ and any element $x \notin B$, $g(A \cup \{x\}) - g(A) \geq g(B \cup \{x\}) - g(B)$.

TABLE I: Frequently used notations in the paper

$G(V,E)$	the social graph with vertex and edge sets
n, m, Z	$n = \|V\|, m = \|E\|$ and Z is the set of topics
$\mathbf{T}, l$	the viral campaign with l viral pieces
$V^p \subseteq V$	the set of available promoters
$\boldsymbol{\beta}^v$	the preference vector of v and β_j^v is the preference of v on the viral piece $\mathbf{t}_j$
r_v	the parameter to control the adoption probability of user v
k	the number of possible assignments
$\bar{\mathcal{S}}, S_j \in \bar{\mathcal{S}}$	an assignment plan for $\mathbf{T}$ and the jth seed set for $\mathbf{t}_j \in \mathbf{T}$
$I_{S_j}^v$	an indicator for if a node v is activated by S_j
$\sigma(\cdot)$	the adoption utility (AU) function

and COMPINFMAX, which is to maximize the *incremental* influence of one campaign on another.

Still, works on comparative IM focus on tuning the *order* in which a user receives the messages of typically *two* campaigns with interdependent propagation processes. In contrast, the $\mathcal{OIPA}$ problem concerns the adoption of a single campaign consisting of several pieces whose propagation processes are mutually independent, yet serve a common goal.

III. PRELIMINARIES

In this section, we introduce a topic-aware influence model that describes how different message pieces spread in a network (Section III-A) and the Optimal Influential Pieces Assignment ($\mathcal{OIPA}$) problem (Section III-B). Before moving on, all frequently used notations are listed in Table I.

A. Topic-aware Influence Model

We model a social network (SN) as a *directed* graph $G(V,E)$, where V is a set of users and each edge $e = (u,v) \in E$ captures the (bidirectional) *friend* or (unidirectional) *follower* relationship from u to v, $|V| = n$, and $|E| = m$. To model how an item propagates in an SN, we adopt the well-studied topic-aware propagation model [3], [2], [9], [22]. This model extracts a set of hidden topics $Z = \{z_1, z_2, \ldots, z_{|Z|}\}$ from social activities (e.g., tweets and replies) propagated on an SN, and explains social influence by means of these topics. Formally, given an edge $e = (u,v)$, a *topic-aware influence probability* $p(e|z)$ models how u *influences* v under topic $z \in Z$; we denote $\mathbf{p}(e)$ as the topic-wise influence vector for e. These probabilities can be learned from logs of past propagation activities [31], [12], [3].

We describe a piece of a viral message by the vector $\mathbf{t} = (t_1, t_2, \ldots, t_{|Z|})$, where t_z is the probability that this piece relates to topic z. Given a seed set of promoters $S \subseteq V$, the influence process of $\mathbf{t}$ starting out from S under the topic-aware influence model unfolds as follows. Initially, all users in $V \setminus S$ are *inactive* and the seed promoters in S are *active*. Each promoter gets one chance to activate each of its neighbors via the corresponding edge. As in [9], [2], we compute the probability that message $\mathbf{t}$ goes through edge e as $p(\mathbf{t}, e) = \mathbf{t} \cdot \mathbf{p}(e)$. Subsequently, any newly activated user u also gets one chance to activate its neighbors. The process terminates when no more users can be activated. Let I_S^u denote a Bernoulli random variable that equals 1 if the item $\mathbf{t}$ propagating from the seed set S successfully activates user u, and 0 otherwise. Then the expected influence spread of $\mathbf{t}$ spreading from S is: $\sigma_{im}(S) = \sum_{v \in V} \mathbb{E}[I_S^v]$.

B. Problem Definition

We consider a viral campaign $\mathbf{T}$ consisting of ℓ viral pieces, $\mathbf{T} = \{\mathbf{t}_1, .., \mathbf{t}_\ell\}$. We assume that each piece $\mathbf{t}_j$ propagates in the network independently[3] of others. We model the user adoption behavior upon receiving some of the viral pieces in a campaign with a *logistic* model [34], [13], [11], [18], [29]. In this model, the probability that user v adopts the campaign $\mathbf{T}$ is a Bernoulli random variable X_v, captured by the logistic function:

$$p[X_v = 1 | I_{S_1}^v .., I_{S_\ell}^v] = \begin{cases} \frac{1}{1+\exp\{\alpha - \beta \cdot \sum_{j=1..\ell} I_{S_j}^v\}} & \text{if } \exists I_{S_j}^v = 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where each $I_{S_j}^v$ is a Bernoulli random variable that indicates whether the piece $\mathbf{t}_j$ propagating from seed set S_j reaches v, and $\alpha, \beta > 0$ are parameters that control the users' turning point for adoption. The larger α is, the harder it is for a user to adopt $\mathbf{T}$, while β weighs the the effect each piece $\mathbf{t}_j$ has on the adoption probability for any user.

As each promoter has a varying potential in spreading message pieces on different topics, our goal is to assign the pieces of $\mathbf{T}$ to a judiciously selected subset of the promoters in S, so that the overall *adoption utility* over all users about $\mathbf{T}$ is maximized. We first formally define the *adoption utility*.

We assign the propagation of each piece $\mathbf{t}_j$ in $\mathbf{T}$ to a subset of promoters $S_j \subseteq S$. Let $\bar{\mathcal{S}}$ denote the collection of all seed subsets that compose our overall assignment plan, $\bar{\mathcal{S}} = \{S_1, ..., S_\ell\}$. We express the overall effectiveness of the assignment plan $\bar{\mathcal{S}}$ via an *adoption utility function* $\sigma(\bar{\mathcal{S}})$:

$$\sigma(\bar{\mathcal{S}}) = \mathbb{E}\left[\sum_{v \in V} X_v\right] = \sum_{v \in V} \mathbb{E}[X_v] = \sum_{v \in V} p[X_v = 1] \quad (2)$$

The Optimal Influential Pieces Assignment ($\mathcal{OIPA}$) problem is to extract the assignment plan that yields the largest adoption utility for a campaign $\mathbf{T}$.

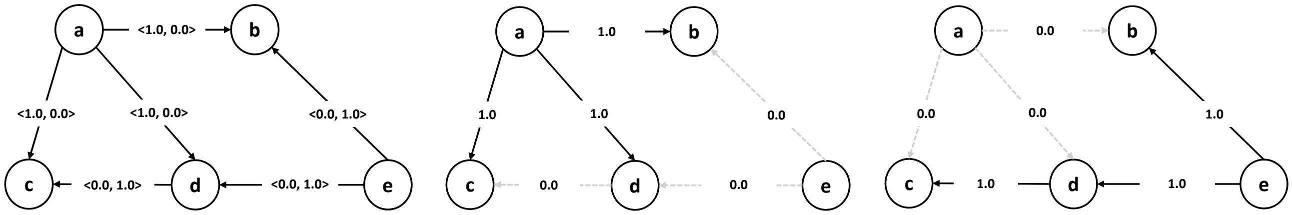
Definition 1 ($\mathcal{OIPA}$)**.** *Given a social graph G, a multifaceted campaign **T** containing ℓ viral pieces, a pool of promoters V^p and a budget k, the* Optimal Influential Pieces Assignment *($\mathcal{OIPA}$) problem is to find an assignment plan $\bar{\mathcal{S}}^*$ that maximizes the overall adoption utility of **T** over G. Formally,*

$$\bar{\mathcal{S}}^* = \underset{|\bar{\mathcal{S}}| \le k}{\operatorname{argmax}} \, \sigma(\bar{\mathcal{S}}) \quad (3)$$

where $|\bar{\mathcal{S}}| = \sum_{j=1,..,\ell} |S_i|$.

Example 1. *Figure 1 presents a running example of an $\mathcal{OIPA}$ instance, using two topics z_1, z_2 with the topic-aware influential probabilities shown on every graph edge. For*

[3]While this *independence assumption* is not general, we show (Section IV) that the problem is intractable even in this restricted case.



(a) Topic-aware influence graph (b) Influence graph viral piece $\mathbf{t}_1$ (c) Influence graph viral piece $\mathbf{t}_2$

Fig. 1: A running example for $\mathcal{OIPA}$. In (a), each edge in contains a topic-aware influence probability vector. In (b), we show the influence graph for $\mathbf{t}_1$ which has a topic distribution of $(1, 0)$. In (c), we show the influence graph for $\mathbf{t}_2$ which has a topic distribution of $(0, 1)$.

instance, the topics could be z_1 =“tax” and z_2 =“healthcare” in a political campaign. We run a campaign on the network with two messages, $\boldsymbol{t}_1, \boldsymbol{t}_2$, having topic distributions $\boldsymbol{t}_1 = (1.0, 0.0)$ and $\boldsymbol{t}_2 = (0.0, 1.0)$ respectively, i.e., one is only about tax and the other only about healthcare. Given a budget of two promoter assignments, $\mathcal{OIPA}$ finds the assignment plan that maximizes the overall adoption utility in the network.

This optimal assignment plan is to assign $\boldsymbol{t}_1$ to user a and $\boldsymbol{t}_2$ to user e. Under this plan, $I^a_{\{a\}} = I^b_{\{a\}} = I^c_{\{a\}} = I^d_{\{a\}} = 1$ and $I^e_{\{a\}} = 0$ for $\boldsymbol{t}_1$ (see Figure 1 (b)), while $I^a_{\{e\}} = 0$ and $I^b_{\{e\}} = I^c_{\{e\}} = I^d_{\{e\}} = I^e_{\{e\}} = 1$ (see Figure 1 (c)). Assume the parameters $\alpha = 3, \beta = 1$, the adoption utility of each user is given by Equation (1). For instance, $p(X_b) = \frac{1}{1+\exp\{3-(1+1)\}} = 0.27$. Eventually, the overall utility of the assignments plan is $\sigma(\{\{a\}, \{e\}\}) = \sum_{i=a,b,c,d,e} p[X_i] = 0.12 + 0.27 \cdot 3 + 0.12 = 1.05$ as user a receives $\boldsymbol{t}_1$, user e receives $\boldsymbol{t}_2$, and others receive both, with probability 1.

IV. Problem Analysis

This section provides a theoretical analysis of $\mathcal{OIPA}$. First, we study the properties of the adoption utility function $\sigma(\bar{\mathcal{S}})$ (Section IV-A). Then, we study the hardness and approximability of $\mathcal{OIPA}$ (Section IV-B).

A. Properties of Adoption Utility Function

Here, we study the properties of the $\mathcal{OIPA}$ objective function, the adoption utility function $\sigma(\bar{\mathcal{S}})$, to gain insights into the optimization problem. While $\sigma(\bar{\mathcal{S}})$ measures the effectiveness of an assignment plan $\bar{\mathcal{S}}$, i.e., a *set of seed sets* assigned to different pieces; thus, $\sigma(\cdot)$ is not a proper set function. To analyze $\sigma(\cdot)$, we define a *containment relationship* between assignment plans.

Definition 2. *An assignment plan $\bar{\mathcal{S}}^a$* contains *another plan $\bar{\mathcal{S}}^b$, i.e., $\bar{\mathcal{S}}^b \subseteq \bar{\mathcal{S}}^a$, if and only if $S^b_j \subseteq S^a_j$ $\forall j = 1, \dots, \ell$.*

We further define the union of one assignment plan with another, and the associated marginal utility gain, as follows.

Definition 3. *The* union *of two assignment plans, $\bar{\mathcal{S}}^a$ and $\bar{\mathcal{S}}^b$, is a new assignment plan $\bar{\mathcal{S}} = \bar{\mathcal{S}}^a \cup \bar{\mathcal{S}}^b$ with $S_j = S^a_j \cup S^b_j$, $j = 1, \dots, \ell$. The* marginal gain *of adding $\bar{\mathcal{S}}^b$ to $\bar{\mathcal{S}}^a$ is:*

$$\delta_{\bar{\mathcal{S}}^a}(\bar{\mathcal{S}}^b) = \sigma(\bar{\mathcal{S}}^a \cup \bar{\mathcal{S}}^b) - \sigma(\bar{\mathcal{S}}^a) \tag{4}$$

Definition 4. *The i-*union *of an assignment plan $\bar{\mathcal{S}}^a$ with a seed set S is a new assignment plan $\bar{\mathcal{S}} = \bar{\mathcal{S}}^a \cup_i S$ where $S_j = S^a_j$, $j \neq i$ and $S_i = S^a_i \cup S$. Then the i-*marginal gain *of adding S to $\bar{\mathcal{S}}^a$ is:*

$$\delta^i_{\bar{\mathcal{S}}^a}(S) = \sigma(\bar{\mathcal{S}}^a \cup_i S) - \sigma(\bar{\mathcal{S}}^a) \tag{5}$$

We study the monotonicity and submodularity of $\sigma(\cdot)$ under these containment and union relationships.

Definition 5. *We say that $\sigma(\cdot)$ is* **monotone** iff, *for any two assignment plans $\bar{\mathcal{S}}^a$ and $\bar{\mathcal{S}}^b$ such that $\bar{\mathcal{S}}^a \subseteq \bar{\mathcal{S}}^b$, it holds that $\sigma(\bar{\mathcal{S}}^a) \leq \sigma(\bar{\mathcal{S}}^b)$. We say that $\sigma(\cdot)$ is* **submodular** iff, *for any two such plans and any $\bar{\mathcal{S}}$, $\delta_{\bar{\mathcal{S}}^b}(\bar{\mathcal{S}}) \leq \delta_{\bar{\mathcal{S}}^a}(\bar{\mathcal{S}})$.*

It is trivial to show that $\sigma(\cdot)$ is monotone. However, as the following counterexample shows, $\sigma(\cdot)$ is not submodular.

Example 2. *Building on Example 1, consider assignment plans $\bar{\mathcal{S}}^x = \{\emptyset, \emptyset\}$, $\bar{\mathcal{S}}^y = \{\{a\}, \emptyset\}$, and $\bar{\mathcal{S}} = \{\emptyset, \{e\}\}$. It holds that $\bar{\mathcal{S}}^x \subseteq \bar{\mathcal{S}}^y$. Yet, $\delta_{\bar{\mathcal{S}}^y}(\bar{\mathcal{S}}) = 1.05 - 0.48 = 0.57 > \delta_{\bar{\mathcal{S}}^x}(\bar{\mathcal{S}}) = 0.48 - 0.00 = 0.48$. Thus, $\sigma(\cdot)$ is not submodular.*

The absence of submodularity precludes any greedy approximation algorithm based on that property. Given this negative result, we study the problem's hardness and approximability.

B. Hardness and Approximability

When a campaign consists of one viral piece, $\mathbf{T} = \{\mathbf{t}_0\}$, $\mathcal{OIPA}$ is reduced to a variant of the classical IM problem [16] in which a user v adopts $\mathbf{T}$ with probability $p[X_v = 1] = \frac{1}{1+\exp\{\alpha-\beta \cdot I^v_S\}}$, if v is activated by the seed set S for $\mathbf{t}_0$. Further, if $\alpha \to c \cdot \beta$ for any $c < 1$, then $p[X_v = 1] \to 1$ as $\beta \to \infty$ for a v activated by $\mathbf{t}_0$. Thus, the classical IM problem is fully reduced to this special variant of $\mathcal{OIPA}$. In effect, $\mathcal{OIPA}$ is straightforwardly NP-hard, since IM is NP-hard. Nevertheless, as we have seen, the AU function is not submodular in the general $\mathcal{OIPA}$ case, as the influence spread function is in the IM setting, leading to an efficient greedy approximation algorithm [16]. This fact indicates that $\mathcal{OIPA}$ is harder than IM. That motivates us to study how hard it is to approximate $\mathcal{OIPA}$ within a constant factor.

We study approximability by reduction from the Maximum Clique ($\mathcal{MC}$) problem. Given an instance of $\mathcal{MC}$, Π_a, on a graph $G_{\Pi_a}(V_{\Pi_a}, E_{\Pi_a})$ where $|V_{\Pi_a}| = n$, we construct a corresponding instance of $\mathcal{OIPA}$, Π_b, in polynomial time:

1) We create $3n$ vertices for Π_b which we denote each vertex as x_i, y_i and r_i, $i = 1, \ldots, n$.
2) We set n topics, $Z = \{z_1, \ldots, z_n\}$ and a campaign consisting of $\ell = n$ pieces, $\mathbf{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_n\}$, each piece $\mathbf{t}_i$ having a topic vector where the i^{th} entry is 1 and others 0. In other words, piece $\mathbf{t}_i$ is about topic z_i only.
3) We set an edge $e = (x_i, r_j)$ for all $j \in \{j | j = i \ \vee \ (v_i, v_j) \in E_{\Pi_a}\}$ and set $\mathbf{p}(e) = \mathbf{t}_i$. In other words, we connect each x_i to the r-vertices standing for v_i and its neighbors in G_{Π_a}, with a topic vector consisting of exactly topic z_i.
4) We set an edge $e = (y_i, r_j)$ for all $j \in \{1, \ldots, i-1, i+1, \ldots n\}$ and set $\mathbf{p}(e) = \mathbf{t}_i$. In other words, we connect each y_i to all r-vertices except the one standing for v_i in G_{Π_a}, with a topic vector consisting of exactly topic z_i.
5) We set $\alpha = \ln(2n)^{2n}$ and $\beta = \ln(2n)^2$. With these settings, for a vertex v that receives all n pieces it is $p[X_v = 1] = \frac{1}{2}$, while for a vertex v that receives at most $n-1$ pieces it is $p[X_v = 1] \leq \frac{1}{1+(2n)^2}$.
6) We set the set of available promoters as $V_p = \{x_i | i = 1, \ldots, n\} \cup \{y_i | i = 1, \ldots, n\}$ with budget $k = n$.

In the constructed Π_b $\mathcal{OIPA}$ instance, $p(\mathbf{t}_i, e) = 1$ if and only if e starts from x_i or y_i, hence x_i and y_i are the only eligible promoters for piece $\mathbf{t}_i$. The following lemma establishes the connection between instances Π_a and Π_b.

Lemma 1. *Let clique size $OPT(\Pi_a)$ and adoption utility $OPT(\Pi_b)$ denote the optimal solutions for the $\mathcal{MC}$ instance Π_a and the $\mathcal{OIPA}$ instance Π_b, respectively. Then* $2 \cdot OPT(\Pi_b) - \frac{1}{n} \leq OPT(\Pi_a) \leq 2 \cdot OPT(\Pi_b)$.

Proof. Let C_{Π_a} be the maximum clique vertex set in Π_a. We can then deploy an assignment plan $\bar{\mathcal{S}}$ for Π_b as follows: for each i, if $v_i \in C_{\Pi_a}$, we choose x_i to promote $\mathbf{t}_i$, and y_i otherwise. By this plan, if $v_i \in C_{\Pi_a}$, than r_i receives all n pieces. Thus, $\sigma(\bar{\mathcal{S}}) \geq \sum_{v_i \in C_{\Pi_a}} p[X_{r_i} = 1] = \frac{1}{2} OPT(\Pi_a)$, hence $OPT(\Pi_a) \leq 2 \cdot OPT(\Pi_b)$, the second inequality.

On the other hand, any plan $\bar{\mathcal{S}}$ for Π_b will have the form $\{\{u_{i_1}\}, \ldots, \{u_{i_n}\}\}$ where each u_{i_j} is either some x_{i_j} or some y_{i_j}. Let $C(\bar{\mathcal{S}}) = N(u_{i_1}) \cap N(u_{i_2}) \cap \ldots \cap N(u_{i_n})$ where $N(u_{i_j})$ is the neighbor set of u_j in Π_b. Given the way edges in Π_b are set, since r_i is never a neighbor of y_i, yet r_i is always a neighbor of x_i, only contributions by x_i's survive all intersections in $C(\bar{\mathcal{S}})$. As all vertices r_j corresponding to neighbors of v_i in Π_a are neighbors of x_i in Π_b, any pair of vertices in Π_a that corresponds to a pair $r_i, r_j \in C$ forms an edge in Π_a, i.e., $(v_i, v_j) \in E_{\Pi_a}$. Thus, $C(\bar{\mathcal{S}})$ induces a clique in G_{Π_a}, hence the largest size it can get is $OPT(\Pi_a)$.

If all n pieces are propagated in the network and at least one vertex r_i receives them all, then adoption utility is $\sigma(\bar{\mathcal{S}}) \geq \frac{1}{2}$. On the other hand, if less than n pieces are propagated in the network, then, even if all vertices r_i, $i = 1, \ldots, n$, receive them all, adoption utility is $\sigma(\bar{\mathcal{S}}) < \frac{1}{2n}$. Thus, an optimal plan $\bar{\mathcal{S}}^*$ for Π_b should ensure all n pieces are propagated in the network and at least one vertex receives them all (as long as that is possible). To include all n pieces, since the budget is $k = n$ promoters, $\bar{\mathcal{S}}^*$ should assign each piece to *exactly* one promoter; thus, it should be in the from $\{\{u_1\}, \ldots, \{u_n\}\}$ where u_i is either x_i or y_i, i.e., one of the only eligible promoters of piece $\mathbf{t}_i$. Even so, only nodes r_i in $C(\bar{\mathcal{S}}^*)$ can possibly receive all n pieces, and will do so if and only if $\bar{\mathcal{S}}^*$ selects the x_i corresponding to each of the $OPT(\Pi_a)$ vertices v_i in a maximum clique in Π_a, and the y_i corresponding to any other vertex v_i in Π_a, provided there exists at least one edge (i.e., a clique of size 2) in Π_a. Then it follows that:

$$\begin{aligned} OPT(\Pi_b) &= \sum_{\forall r \in C(\bar{\mathcal{S}}^*)} p[X_r = 1] + \sum_{\forall r \notin C(\bar{\mathcal{S}}^*)} p[X_r = 1] \\ &\leq \frac{1}{2} OPT(\Pi_a) + \frac{2n}{1+(2n)^2} \\ &\leq \frac{1}{2} OPT(\Pi_a) + \frac{1}{2n} \end{aligned}$$

Therefore $2 \cdot OPT(\Pi_b) - \frac{1}{n} \leq OPT(\Pi_a)$. □

Lemma 1 leads to the following hardness result.

Theorem 1. *There is no polynomial-time algorithm to approximate the optimal solution of $\mathcal{OIPA}$ within a factor of $\frac{1}{2} n^{1-\varepsilon}$ for any $\varepsilon > 0$, unless $NP = ZPP$.*

Proof. Let $k \geq 1$. By Lemma 1, if $OPT(\Pi_a) \geq k$ then $OPT(\Pi_b) \geq \frac{k}{2}$, and if $OPT(\Pi_a) \leq \frac{k}{n^{1-\varepsilon}}$ then $OPT(\Pi_b) \leq \frac{k}{2n^{1-\varepsilon}} + \frac{1}{2n} \leq \frac{1}{\frac{1}{2} n^{1-\varepsilon}} \cdot \frac{k}{2}$. Thus, we have a gap-preserving reduction from Π_a to Π_b [35]. If we can approximate $\mathcal{OIPA}$ in polynomial time within a factor of $\frac{1}{2} n^{1-\varepsilon}$, then we can approximate $\mathcal{MC}$ within a factor of $n^{1-\varepsilon}$. This can not be true unless $\mathbf{P} = \mathbf{ZPP}$ [15]. □

V. Approximation Algorithms

Given the results of Section IV, there is no known way to develop a polynomial-time approximation algorithm for the general $\mathcal{OIPA}$ case. Still, social influence follows a power law principle [28]: a few people have significantly larger influence than others. Based on this principle, we propose a branch-and-bound framework that prioritizes promoters with large influence and enables early termination when necessary.

We maintain *partial* candidate plans in a max-heap, sorted by their estimated AU score upper bound. If the upper bound of a partial plan is smaller than the exact AU score of the best currently obtained plan, we safely prune the partial plan. Yet this solution brings nontrivial technical challenges, as we need to: (1) quickly compute the AU score of a candidate plan; (2) derive an effective upper bound for the AU score of a partial plan, and (3) efficiently compute the upper bound function.

This section addresses these challenges. A branch-and-bound technique that efficiently evaluates the AU of any candidate assignment plan is the subject of Section V-A. We present a tight upper bound function for effective pruning that ensures a $(1 - 1/e)$ approximation ratio in Section V-B. Last, we propose an efficient progressive estimation scheme that has a slightly worse $(1 - 1/e - \varepsilon)$ approximation ratio with huge performance boost in Section V-C.

A. AU Estimation

To employ a branch-and-bound framework, we need to evaluate the AU for a large number of candidate assignment plans. Since evaluating the influence spread of any user set in the classical IM problem is $\#P$-hard, it follows that computing the AU for any candidate plan is also $\#P$-hard. Yet we can evaluate AU with good accuracy using an extension of the reverse reachable (RR) sets method [7], [33], [32]. We briefly review the RR set method in the following.

Reverse-Reachable (RR) Sets. Given a homogeneous influence graph $G' = (V', E')$, $|V'| = n'$, where a *single* value $p(e)$ characterizes the activation probability through edge $e = (u, v) \in E'$, the RR set method estimates the expected influence of any seed set $S \in V'$. A *Random* RR set rests on two random choices: (i) an *initial* node x is randomly chosen from the graph, and (ii) the graph is sampled by keeping each edge $e \in E$ with probability $p(e)$. The RR set comprises all vertices from which we can reach x in the sampled graph. Let $I[R_i \cap S \neq \emptyset]$ be a boolean variable that indicates whether RR set R_i intersects with S. Then, after generating θ RR sets, we can estimate the expected influence of S, $\sigma_{IM}(S)$, as:

$$\sigma_{IM}(S) \leftarrow \frac{n'}{\theta} \sum_{i=1,..,\theta} I[R_i \cap S \neq \emptyset]$$

Multi-Reverse-Reachable (MRR) Sets. We extend the RR sets method to the Multi-RR (MRR) sets method to evaluate the AU function in $\mathcal{OIPA}$. As in the RR sets method, we select θ users from V uniformly at random. Yet now, each viral piece $\mathbf{t}_j$ induces a *homogeneous* influence graph where the influence probability of edge e is computed as $p(\mathbf{t}_j, e) = \mathbf{t}_j \cdot \mathbf{p}(e)$. Then, for each selected user v_i, we generate a multi-set of ℓ RR sets, one for each viral piece $\mathbf{t}_j$, $\mathbf{R_i} = \{R_i^1, \ldots, R_i^\ell\}$. Given an assignment plan $\bar{\mathcal{S}} = \{S_1, \ldots, S_\ell\}$, we denote $I^i_{S_j} = I[R_i^j \cap S_j \neq \emptyset]$. Then an unbiased estimator of $\sigma(\bar{\mathcal{S}})$ is:

$$\sigma(\bar{\mathcal{S}}) \leftarrow \frac{n}{\theta} \sum_{i=1,\ldots,\theta} \frac{1}{1 + \exp\{\alpha - \beta \cdot \sum_{j=1,\ldots,\ell} I^i_{S_j}\}]} \quad (6)$$

Lemma 2. *Eqn. 6 gives an unbiased estimator for $\sigma(\bar{\mathcal{S}})$.*

We refer the reader to the appendix for the proof. This estimator is the sum of θ i.i.d random variables $X_i = [1 + \exp\{\alpha - \beta \cdot \sum_{j=1\ldots,\ell} I^i_{S_j}\}]]^{-1}$. We can adopt the Chernoff bound used in the RR sets method [26] to derive the MRR convergence speed. In practice, a large θ ensures the estimated AU score for any $\bar{\mathcal{S}}$ is accurate with a high probability.

Example 3. *Table II shows an example of using MRR samples to estimate AU, following the running example in Figure 1. We sample four vertices, c, a, b, c, and then build the RR sets for $\mathbf{t}_1$ and $\mathbf{t}_2$. We compute the adoption probability for each MRR set shown in the table, and estimate the AU score for the assignment plan $\bar{\mathcal{S}} = \{\{a\}, \{e\}\}$ as $\sigma(\bar{\mathcal{S}}) = \frac{5}{4}(0.27 + 0.12 + 0.27 + 0.27) = 1.16$.*

TABLE II: Example of MRR samples and AU estimation

	vertex	R_i^1 for $\mathbf{t}_1$	R_i^2 for $\mathbf{t}_2$	$p(X_i = 1 \mid \{\{a\},\{e\}\})$
$\mathbf{R_1}$	c	a c	c d e	$\frac{1}{1+\exp\{3-2\}} = 0.27$
$\mathbf{R_2}$	a	a	a	$\frac{1}{1+\exp\{3-1\}} = 0.12$
$\mathbf{R_3}$	b	a b	b e	$\frac{1}{1+\exp\{3-2\}} = 0.27$
$\mathbf{R_4}$	c	a c	c d e	$\frac{1}{1+\exp\{3-2\}} = 0.27$

B. Upper Bound Function by Branch-and-Bound

A naive $\mathcal{OIPA}$ solution would enumerate all possible candidate plans and compute their AU scores using MRR sets. Yet that would incur $\binom{\ell \cdot n}{k}$ AU score estimations. Instead, we estimate an AU upper bound function by branch-and-bound.

Algorithm 1 Branch-and-Bound

1: $\mathbf{R} \leftarrow$ Generate θ MRR sets for $\mathbf{T}$ on G.
2: $\bar{\mathcal{S}} \leftarrow \boldsymbol{\emptyset} = \{S_1 = \emptyset, S_2 = \emptyset, \ldots, S_\ell = \emptyset\}$
3: $\mathbf{V}^p \leftarrow \{V_1 = V^p, V_2 = V^p, ..., V_\ell = V^p\}$
4: $L \leftarrow 0$, $U \leftarrow \infty$, $\bar{\mathcal{S}}^* \leftarrow \bar{\mathcal{S}}$
5: Initialize max heap $H \leftarrow (\bar{\mathcal{S}}, \mathbf{V}^p, U)$
6: **while** $L < U$ **do**
7:     $(\bar{\mathcal{S}}, \mathbf{V}^p, U) \leftarrow$ top of H
8:     **if** $|\bar{\mathcal{S}}| \leq k$ and $|\mathbf{V}^p| \neq 0$ **then**
9:         Select v^* $\forall v^* \in V_j \wedge \forall\, V_j \in \mathbf{V}_p$
10:         $V_j \leftarrow V_j \setminus \{v^*\}$ and update $\mathbf{V}^p$ with V_j
11:         $\bar{\mathcal{S}}^a \leftarrow \bar{\mathcal{S}}$ with v_j included
12:         $\bar{\mathcal{S}}^b \leftarrow \bar{\mathcal{S}}$
13:         $(\bar{\mathcal{S}}^c, L^a, U^a) \leftarrow \mathbf{ComputeBound}(\bar{\mathcal{S}}^a, \mathbf{V}^p, \mathbf{R})$
14:         **if** $L^a > L$ **then**
15:             $L \leftarrow L^a$ and $\bar{\mathcal{S}}^* \leftarrow \bar{\mathcal{S}}^c$
16:         **if** $U^a > L$ **then**
17:             $H \leftarrow H \cup (\bar{\mathcal{S}}^a, \mathbf{V}^p, U^a)$
18:         Repeat Lines 13-17 for $\bar{\mathcal{S}}^b$

Algorithm 1 presents this framework. We first generate θ MRR sets for campaign $\mathbf{T}$ (Line 1). Then we initialize a max heap with each entry denoted as $(\bar{\mathcal{S}}, \mathbf{V}^p, U)$ (Lines 2–5). $\bar{\mathcal{S}}$ is a *partial* plan, $\mathbf{V}^p$ the set of promoters that have not yet been considered and U the upper bound of the corresponding search space. In each iteration of the search loop, we get the top entry of the heap by upper bound value, $(\bar{\mathcal{S}}, \mathbf{V}^p)$, and update the global upper bound U. As long as the partial plan has not been filled up, we select a promoter v^* from $\mathbf{V}^p$, and branch the search space to examine the cases of including v^* and excluding v^* from $\bar{\mathcal{S}}$, to form two new partial plans $\bar{\mathcal{S}}^a$ and $\bar{\mathcal{S}}^b$ (Lines 9–12). For each such partial plan, e.g., $\bar{\mathcal{S}}^a$, we invoke $\mathbf{ComputeBound}(\cdot)$ to get a triple $(\bar{\mathcal{S}}^c, L^a, U^a)$, where U^a is the upper bound for the partial plan and $\bar{\mathcal{S}}^c$ is a *complete* candidate plan with AU score L^a (Line 13). If $\bar{\mathcal{S}}^c$ is better than the current best solution $\bar{\mathcal{S}}^*$, we update the global lower bound L and add the partial solution $\bar{\mathcal{S}}^a$ back to the heap if its upper bound is better than the global lower bound (Lines 14–17). We repeat the process for the other partial solution $\bar{\mathcal{S}}^b$. The algorithms terminates when the global lower bound is larger or equal to the global upper bound.

The **ComputeBound**$(\bar{\mathcal{S}}^a, \mathbf{V}^p, \mathbf{R})$ function estimates the potential of a partial plan $\bar{\mathcal{S}}^a$, i.e., the maximum AU score of all possible complete plans that contain $\bar{\mathcal{S}}^a$. If this bound is lower than the global lower bound, we can safely prune $\bar{\mathcal{S}}^a$ from the search space. Formally, the **ComputeBound** functions solves the following optimization problem:

$$\begin{aligned} &\max_{|\bar{\mathcal{S}}^c| \le k} && \sum_{i=1,..,\theta} \frac{1}{1+\exp\{\alpha - \beta \cdot \sum_{j=1..l} I^i_{S^c_j}\}} \\ &\text{s.t.} && S^c_j \supseteq S^a_j \quad \forall S^c_j \in \bar{\mathcal{S}}^c \end{aligned} \tag{7}$$

While this optimization is still non-submodular, we propose a bound estimation method based on adaptively solving a submodular optimization problem. Figure 2 illustrates the main idea: Although each individual logistic function is not submodular, we choose the submodular function that most tightly upper bounds the function value, by means of a *tangent line* that intersects with the logistic S-curve. Since the objective function in Equation 7 is a sum of logistic functions, the sum of these upper bound functions is submodular. Whenever the branch-and-bound routine searches a subspace which contains a partial assignment plan, we update the upper-bound function as in Figure 2. For example, if we find a user v activated for viral piece $\mathbf{t}_j$ by a partial assignment plan $\bar{\mathcal{S}}^a$, we refine the submodular upper bound function by shifting the tangent line to a larger gradient.

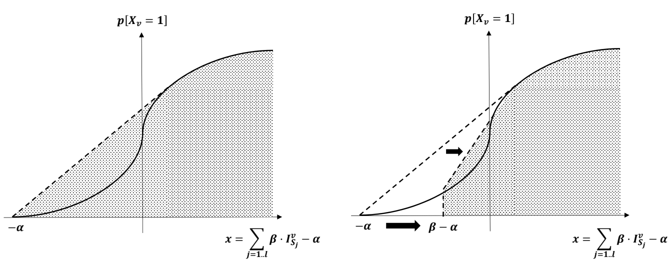


Fig. 2: Example of upper-bound refinement when user v is found to be activated by piece $\mathbf{t}_j$.

Definition 6. *The upper bound function $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$ is defined over a space of all possible assignment plans $\bar{\mathcal{S}} \in V^p \times V^p \times \ldots \times V^p$ s.t.*

$$\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a) = \sum_{i=1,\ldots,\theta} \tau_i(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$$

where $\tau_i(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$ is a minimal monotone submodular function for $\bar{\mathcal{S}}$ s.t. $\bar{\mathcal{S}}^a \subseteq \bar{\mathcal{S}}$ and

$$\tau_i(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a) \ge \frac{1}{1+\exp\{\alpha - \beta \cdot \sum_{j=1..l} I^i_{S_j}\}}$$

We derive $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$ (i.e., discuss how to efficiently obtain the tangent line) in the appendix due to space constraints.

Armed with this upper bound function, Algorithm 2 presents the pseudocode for **ComputeBound**$(\bar{\mathcal{S}}^a, \mathbf{V}^p, \mathbf{R})$, which greedily finds a promoter of maximum marginal gain on $\tau_i(\cdot|\bar{\mathcal{S}}^a)$ in each iteration. As $\tau_i(\cdot|\bar{\mathcal{S}}^a)$ is a submodular function obtained by the tangent-line method, this greedy strategy provides an approximation guarantee. In more detail, Algorithm 2 first refines the upper bound function $\tau(\cdot|\bar{\mathcal{S}}^a)$, in case $\bar{\mathcal{S}}^a$ has a promoter who appears in the i^{th} MRR set. Since at most one promoter, say v, is added to $\bar{\mathcal{S}}^a$ in the branch-and-bound framework (Lines 11–12, Algorithm 1), we only update those $\tau_i(\cdot|\bar{\mathcal{S}}^a)$ functions that correspond to MRR sets in which v appears. Then the algorithm greedily selects the remaining $k-|\bar{\mathcal{S}}^a|$ promoters to form a complete plan $\bar{\mathcal{S}}$ with k assignments (Lines 2–6), and returns $\bar{\mathcal{S}} \cup \bar{\mathcal{S}}^a$ as a candidate solution with its AU score, computed by Equation 6, as a lower bound and $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$ as an upper bound.

Algorithm 2 **ComputeBound**$(\bar{\mathcal{S}}^a, \mathbf{V}^p, \mathbf{R})$

1: Refine $\tau(\cdot|\bar{\mathcal{S}}^a)$ according to the updates in $\bar{\mathcal{S}}^a$
/** Run greedy selection from $\mathbf{V}^p$ **/
2: $\bar{\mathcal{S}} \leftarrow \emptyset$
3: **while** $|\bar{\mathcal{S}}| \le k - |\bar{\mathcal{S}}^a|$ **do**
4: $\quad (v^*, V_j) = \mathrm{argmax}_{v \in V_j \in \mathbf{V}^p}\, \tau(\bar{\mathcal{S}} \cup_j \{v\}|\bar{\mathcal{S}}^a) - \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$
5: $\quad V_j \leftarrow V_j \setminus \{v_j\}$ and update $\mathbf{V}^p$ with V_j
6: $\quad \bar{\mathcal{S}} \leftarrow \bar{\mathcal{S}} \cup_j \{v^*\}$
7: **return** $(\bar{\mathcal{S}} \cup \bar{\mathcal{S}}^a, \sigma(\bar{\mathcal{S}} \cup \bar{\mathcal{S}}^a), \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a))$

Theorem 2. *The branch-and-bound framework using Algorithm 2 for upper bound estimation achieves a $(1-1/e)$ approximation ratio to the MRR-based solution to $\mathcal{OIPA}$.*

Proof. Since $\tau(\cdot)$ is monotone and submodular, and **ComputeBound**$(\cdot)$ selects a candidate solution $\bar{\mathcal{S}}$ by the greedy approach, it is $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a) \ge (1-1/e)\tau(\bar{\mathcal{S}}^*|\bar{\mathcal{S}}^a) \ge (1-1/e)\sigma(\bar{\mathcal{S}}^* \cup \bar{\mathcal{S}}^a)$ for all $|\bar{\mathcal{S}}^*| = |\bar{\mathcal{S}}|$. Let $\bar{\mathcal{S}}^g$ denote the returned assignment plan. For any $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$ that has not been evaluated when the search terminates, we deduce that $\sigma(\bar{\mathcal{S}}^g) \ge \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$. Thus, $\sigma(\bar{\mathcal{S}}^g) \ge (1-1/e)\sigma(\bar{\mathcal{S}}^* \cup \bar{\mathcal{S}}^a)$ for any unexplored partial solution $\bar{\mathcal{S}}^a$. $\bar{\mathcal{S}}^g$ is the best solution among all evaluated ones, hence the guarantee holds. □

The approximation ratio in Theorem 2 holds with respect to the $\mathcal{OIPA}$ solution computed by estimating AU scores $\sigma(\cdot)$ from sampled MRR sets, which incurs a sampling error. Still, when generating millions of MRR sets, this error is negligible.

C. Progressive Upper Bound Estimation

The branch-and-bound framework repeatedly invokes **ComputeBound**$(\cdot)$ to compute an upper bound for each sub-search space that corresponds to a partial assignment plan. Although the greedy selection in Algorithm 2 gives an $(1-1/e)$ approximation ratio, it scans all available promoters in each iteration, incurring $O(kn)$ evaluations of AU marginal gain over $\tau(\cdot|\bar{\mathcal{S}}^a)$ (Line 4), each performed on millions of MRR sets (Definition 6). Nevertheless, due to the power law principle of social influence, most users have small social influence, hence it is unnecessary to scan all promoters when constructing partial plans. Motivated by this observation, we propose a *progressive* upper bound estimation method that maintains a $(1-1/e-\varepsilon)$ approximation ratio, where ε is a tunable parameter that trades efficiency with accuracy. This method is based on an asymptotic bound for the number of $\tau(\cdot|\bar{\mathcal{S}}^a)$ evaluations, based on the power law principle.

The idea of the progressive estimation method is as follows. Instead of scanning all promoters to find the one with maximum marginal gain in $\tau(\cdot|\bar{\mathcal{S}}^a)$ in each iteration, we sort promoters v by their individual $\tau(\cdot|\bar{\mathcal{S}}^a)$. Then, we set a threshold h and include a promoter in the candidate plan if its marginal gain is larger than h. We progressively lower the threshold so as to include more promoters. This method accelerates upper bound estimation thanks to two features: First, the sorting process does not need to rerun in each iteration. We only need to update the position of promoters v whose score is affected by the new promoter v^* in $\bar{\mathcal{S}}^a$, i.e., who coexist in an MRR set with v^*. Second, when the threshold is small enough, the algorithm terminates and returns a plan even if the number of assignments is lower than k. As we will discuss, this second feature tightly bounds the number of $\tau(\cdot)$ evaluations.

Algorithm 3 ComputeBoundPro$(\bar{\mathcal{S}}^a, \mathbf{V}^p, \mathbf{R})$

1: Refine $\tau(\cdot|\bar{\mathcal{S}}^a)$ according to the updates in $\bar{\mathcal{S}}^a$
2: Reorder $v \in V_j \in \mathbf{V}^p$ by $\delta_{\emptyset}(v) = \tau(\emptyset \cup_j \{v\}|\bar{\mathcal{S}}^a) - \tau(\emptyset|\bar{\mathcal{S}}^a)$ for $j = 1, \ldots, \ell$
3: $\texttt{maxinf} \leftarrow \max_{v \in V_j \in \mathbf{V}^p} \delta_{\emptyset}(v)$
4: $h \leftarrow \texttt{maxinf}$
5: $\bar{\mathcal{S}} \leftarrow \emptyset$
6: **while** $|\bar{\mathcal{S}}| \leq k - |\bar{\mathcal{S}}^a|$ **do**
7:     **for** $v \in V_j \in \mathbf{V}^p$ **do**
8:         $\delta_{\bar{\mathcal{S}}}(v) = \tau(\bar{\mathcal{S}} \cup_j \{v\}|\bar{\mathcal{S}}^a) - \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$
9:         **if** $\delta_{\bar{\mathcal{S}}}(v) \geq h$ **then**
10:             $\bar{\mathcal{S}} \leftarrow \bar{\mathcal{S}} \cup_j \{v\}$
11:         **if** $\delta_{\emptyset}(v) < h$ **then**
12:             **break**
13:     $h \leftarrow \frac{h}{(1+\varepsilon)}$
14:     **if** $h \leq \frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}{(k-|\bar{\mathcal{S}}^a|)} \cdot \frac{e^{-1}}{1-e^{-1}}$ **then**
15:         **break**
16: **return** $(\bar{\mathcal{S}} \cup \bar{\mathcal{S}}^a, \sigma(\bar{\mathcal{S}} \cup \bar{\mathcal{S}}^a), \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a))$

Algorithm 3 presents the progressive upper bound estimation. As in **ComputeBound**$(\cdot)$, we refine $\tau(\cdot|\bar{\mathcal{S}}^a)$ if $\bar{\mathcal{S}}^a$ includes a new promoter v^* in the branch-and-bound framework. Line 2 sorts promoters by their individual $\tau(\cdot|\bar{\mathcal{S}}^a)$ scores. Then we set the threshold h to be the largest individual $\tau(\cdot|\bar{\mathcal{S}}^a)$ score among all promoters (Lines 3 – 4). We iteratively lower the threshold by a factor of $(1-\varepsilon)$ to include more promoters in $\bar{\mathcal{S}}$ (Lines 6–15). Lines 11 – 12 break the loop once the individual $\tau(\cdot|\bar{\mathcal{S}}^a)$ score of a promoter is found to be smaller than h. Due to the submodularity of $\tau(\cdot|\bar{\mathcal{S}}^a)$, if $\delta_{\emptyset}(v) < h$ then $\delta_{\bar{S}}(v) < h$. Since promoters are sorted by $\delta_{\emptyset}(v)$, we then have an early termination.

In the rest of this section, we analyze this progressive upper bound method in terms of approximation ratio and complexity.

The following lemma unveils the approximation properties of the solution obtained by **ComputeBoundPro**$(\cdot)$. To simplify the presentation, we denote $v \in \bar{\mathcal{S}}$ if there exists a $j = 1, .., \ell$ s.t. $v \in S_j \in \bar{\mathcal{S}}$, and denote $\bar{\mathcal{S}}^x \setminus \bar{\mathcal{S}}^y = \{S_1^x \setminus S_1^y, ..., S_\ell^x \setminus S_\ell^y\}$ for any two plans $\bar{\mathcal{S}}^x, \bar{\mathcal{S}}^y$. Last, v^i denotes the ith promoter selected in Algorithm 3.

Lemma 3. *Let $k' = k - |\bar{\mathcal{S}}^a|$. The solution $\bar{\mathcal{S}}$ obtained by Algorithm 3 with $|\bar{\mathcal{S}}| = d \leq k'$ has an approximation ratio of $1 - e^{-d/((1+\varepsilon)k')}$ to the optimal solution w.r.t. $(\bar{\mathcal{S}}^a, \mathbf{V}^p, \mathbf{R})$.*

Proof. Let $\bar{\mathcal{S}}^*$ be the optimal solution for the instance $(\bar{\mathcal{S}}^a, \mathbf{V}^p, \mathbf{R})$ and assume v^i is selected at a given threshold h. Due to the submodularity of $\tau(\cdot|\bar{\mathcal{S}}^a)$, it follows that:

$$\delta_{\bar{\mathcal{S}}}(v) = \begin{cases} \geq h & \textbf{if } v = v^i \\ \leq h \cdot (1+\varepsilon) & \textbf{if } v \in \bar{\mathcal{S}}^* \setminus (\bar{\mathcal{S}} \cup_j \{v^i\}) \end{cases} \tag{8}$$

where $\bar{\mathcal{S}}$ is the current partial plan and $\delta_{\bar{\mathcal{S}}}(v)$ is as defined in Line 8 of Algorithm 3. Equation 8 implies that $\delta_{\bar{\mathcal{S}}}(v^i) \geq \delta_{\bar{\mathcal{S}}}(v)/(1+\varepsilon)$ for $v \in \bar{\mathcal{S}}^* \setminus \bar{\mathcal{S}}$. Thus, we have $\delta_{\bar{\mathcal{S}}}(v^i) \geq \frac{1}{(1+\varepsilon)|\bar{\mathcal{S}}^* \setminus \bar{\mathcal{S}}|} \sum_{v \in \bar{\mathcal{S}}^* \setminus \bar{\mathcal{S}}} \delta_{\bar{\mathcal{S}}}(v) = \frac{1}{(1+\varepsilon)k'} \sum_{v \in \bar{\mathcal{S}}^* \setminus \bar{\mathcal{S}}} \delta_{\bar{\mathcal{S}}}(v)$.

Let $\bar{\mathcal{S}}^i$ be partial plan obtained by the algorithm after including i promoters and v^{i+1} be the promoter included at the $(i+1)$th step. Then we have:

$$\begin{aligned} \tau(\bar{\mathcal{S}}^{i+1}|\bar{\mathcal{S}}^a) - \tau(\bar{\mathcal{S}}^i|\bar{\mathcal{S}}^a) &= \delta_{\bar{\mathcal{S}}^i}(v^{i+1}) \\ &\geq \frac{1}{(1+\varepsilon)k'} \sum_{v \in \bar{\mathcal{S}}^* \setminus \bar{\mathcal{S}}^i} \delta_{\bar{\mathcal{S}}^i}(v) \\ &\geq \frac{1}{(1+\varepsilon)k'} (\tau(\bar{\mathcal{S}}^* \cup \bar{\mathcal{S}}^i|\bar{\mathcal{S}}^a) - \tau(\bar{\mathcal{S}}^i|\bar{\mathcal{S}}^a)) \\ &\geq \frac{1}{(1+\varepsilon)k'} (\tau(\bar{\mathcal{S}}^*|\bar{\mathcal{S}}^a) - \tau(\bar{\mathcal{S}}^i|\bar{\mathcal{S}}^a)) \end{aligned}$$

where the first inequality is obtained from Equation 8, the second inequality is due to a sequence of submodularity relationships that holds for adding $v \in \bar{\mathcal{S}}^* \setminus \bar{\mathcal{S}}^i$ on top of each other vs. adding each one individually to $\bar{\mathcal{S}}^i$, and the last inequality is due to monotonicity. Using the geometric series formula, we derive from this last inequality that:

$$\begin{aligned} \tau(\bar{\mathcal{S}}^d|\bar{\mathcal{S}}^a) &\geq \left(1 - \left(1 - \frac{1}{(1+\varepsilon)k'}\right)^d\right) \tau(\bar{\mathcal{S}}^*|\bar{\mathcal{S}}^a) \\ &\geq \left(1 - e^{\frac{-d}{(1+\varepsilon)k'}}\right) \tau(\bar{\mathcal{S}}^*|\bar{\mathcal{S}}^a) \end{aligned}$$

□

Lemma 3 allows us to analyze the approximation ratio of the progressive estimation method.

Theorem 3. *The branch-and-bound framework using Algorithm 3 for upper bound estimation achieves a $(1 - 1/e - \varepsilon)$ approximation ratio to the MRR-based solution to $\mathcal{OIPA}$.*

Proof. As in the proof of Theorem 2, we only need to prove that the Algorithm 3 achieves a $(1 - 1/e - \varepsilon)$ approximation to the optimization problem of selecting $k' = k - |\bar{\mathcal{S}}^a|$ assignments that maximize $\tau(\cdot|\bar{\mathcal{S}}^a)$. Given a returned plan $\bar{\mathcal{S}}$ with $|\bar{\mathcal{S}}| = d$, we discuss two cases:

First, $d = k'$. The approximation ratio then follows directly from Lemma 3, as we have:

$$\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a) \geq \left(1 - e^{\frac{-1}{1+\varepsilon}}\right)\tau(\bar{\mathcal{S}}^*|\bar{\mathcal{S}}^a) \geq (1 - e^{-1} - \varepsilon)\tau(\bar{\mathcal{S}}^*|\bar{\mathcal{S}}^a)$$

where $\bar{\mathcal{S}}^*$ is the optimal solution.

Second, $d < k'$. Since the algorithm terminates when h falls below $\frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}{k'} \cdot \frac{e^{-1}}{1-e^{-1}}$ (Line 14), we deduce that:

$$\begin{aligned}\tau(\bar{\mathcal{S}}^*|\bar{\mathcal{S}}^a) \leq \tau(\bar{\mathcal{S}}^* \cup \bar{\mathcal{S}}|\bar{\mathcal{S}}^a) &\leq \sum_{v \in \bar{\mathcal{S}}^* \setminus \bar{\mathcal{S}}} \delta_{\bar{\mathcal{S}}}(v) + \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a) \\ &\leq k' \cdot \frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}{k'} \cdot \frac{e^{-1}}{1-e^{-1}} + \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a) \\ &\leq \frac{1}{1-e^{-1}} \cdot \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)\end{aligned}$$

Thus, the theorem is proved. □

Eventually, we discuss the complexity of the proposed progressive approach. We focus on the number of $\tau(\cdot|\bar{\mathcal{S}}^a)$ evaluations as it is the bottleneck of the upper bound estimation process. Algorithm 3 enables an early termination when we find $\delta_{\emptyset}(v) < h$ for some v. Moreover, since h progressively decreases by a factor of $(1+\varepsilon)$ and h is bounded as $h \in [\frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)\cdot e^{-1}}{(k-|\bar{\mathcal{S}}^a|)(1-e^{-1})}, \mathtt{maxinf}]$, the number of scans s over all promoters is bounded as follows:

$$s \leq \log_{1+\varepsilon}\left(\mathtt{maxinf} \cdot \frac{(k - |\bar{\mathcal{S}}^a|)(1-e^{-1})}{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)\cdot e^{-1}}\right) \leq \log_{1+\varepsilon} 2k \tag{9}$$

Next, we examine how many evaluations are invoked for each scan. The following lemma is based on the power law principle of social influence.

Lemma 4. *The number of $\tau(\cdot|\bar{\mathcal{S}}^a)$ evaluations for each iteration in Algorithm 3 is $O\left(k \cdot \frac{n}{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}\right)$.*

Proof. We only evaluate a promoter v if $\delta_{\emptyset}(v) \in \left[\frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)\cdot e^{-1}}{(k-|\bar{\mathcal{S}}^a|)(1-e^{-1})}, \mathtt{maxinf}\right] \subset \left[\frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}{2k}, \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)\right]$. By the power law principle, the fraction of nodes that have an influence of x, i.e., $P(x)$ is $P(x) \sim x^{-\alpha}$, $2 < \alpha < 3$.

Since $\delta_{\emptyset}(v)$ measures the individual τ score of v and is positively correlated to the influence of v, we can model the number of promoters that have influence score in $[\frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}{2k}, \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)]$, denoted as $w[\frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}{2k}, \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)]$, by the power law principle:

$$\begin{aligned}&w\left[\frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}{2k}, \tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)\right] = \\ &n \cdot \frac{\int_{\frac{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}{2k}}^{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)} x^{-\alpha}\mathrm{d}x}{\int_1^{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)} x^{-\alpha}\mathrm{d}x} \sim n \cdot \left(\frac{2k}{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}\right)^{\alpha-1} < k \cdot \frac{n}{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)}\end{aligned}$$

where the last inequality holds for $\alpha > 2$. The lemma thus follows. □

We summarize the above analysis in the following theorem:

Theorem 4. *The number of $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$ evaluations in Algorithm 3 is $O(\frac{n}{\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)} \cdot k\log_{1+\varepsilon}(2k))$.*

The proof follows from Equation (9) and Lemma 4. In practice, $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$ is large especially for large k, i.e., $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a) \sim n$, hence the number of $\tau(\cdot|\bar{\mathcal{S}}^a)$ evaluations is small. Compared to the greedy selection that performs $O(nk)$ evaluations, the progressive approach effectively avoids unnecessary evaluations while maintaining a $(1 - 1/e - \varepsilon)$ approximation ratio.

VI. Experimental Evaluation

A. Experimental Setup

Datasets. We conduct experiments on the following three real datasets. 1) `lastfm` is a social music sharing dataset from an online site[4]. `lastfm` contains a social network and an action log which records users' activities of voting items (i.e., "a log of past propagation" in [3]). 2) `dblp` is a DBLP coauthor graph which is downloaded from an online academic search service[5]. 3) `tweet` is a social network built from the retweet and reply actions of users in Twitter[6]. We adopt the TIC model [3] to learn the topic-aware influence probabilities $p(e|z)$ (see Section III) for `lastfm` based on its action logs. Following the previous settings in [3], [2], [9], we set the number of topics of `lastfm` as 20. Since `dblp` has no action log, we follow the settings in [9] to use research fields as topics and compute $p(e|z)$ of two authors by categorizing their related conferences using the topics. For `tweet` dataset, we consider all hashtags of an individual users as a document and apply LDA [5] on all the documents to obtain the topic distribution of each user. Given an edge $e = (u, v)$, we compute $p(e|z)$ based on the topic distribution of u and v. The statistics of the datasets are listed in Table III.

TABLE III: Statistics of Datasets

Datasets	`lastfm`	`dblp`	`tweet`
Number of Vertices	1.3K	0.5M	10M
Number of Edges	15K	6M	12M
Average Degrees	8.7	11.9	1.2
Number of Topics	20	9	50
Sample Time	1.2s	5.7s	23.9s

Compared Methods. To the best of our knowledge, there is no existing work for $\mathcal{OIPA}$. Thus, we compare the following baselines together with our proposed approaches.

- IM: We run the state-of-the-art IM algorithm on graph G to obtain k seed nodes (denoted as S) under the IC model [32]. Subsequently, we compare the adoption utility among using S to spread each viral piece $\mathbf{t}_i \in \mathbf{T}$ and the viral piece with the maximum utility is chosen to be spread by S.
- TIM: We construct an influence graph $G_{\mathbf{t}_i}$ for each $\mathbf{t}_i \in \mathbf{T}$. For each $G_{\mathbf{t}_i}$ constructed, we run the IM algorithm on $G_{\mathbf{t}_i}$ to obtain k seed nodes (denoted as S_i) [9] and we select S_i to spread $\mathbf{t}_i$ which achieves the largest adoption utility.
- BAB: The branch and bound algorithm proposed in Section V-B. We terminate the search when the utility difference between the upper bound and the best obtained solution is within 1% error ratio.

[4]http://www.last.fm/
[5]http://dblp.uni-trier.de/xml/
[6]http:/snap.stanford.edu/

- BAB-P: The progress upper bound estimation techniques proposed in Section V-C to speedup BAB.

Note that for all compared approaches, we need to generate θ RR sets for each viral pieces. For a fair comparison, we fix $\theta = 10^6$ across all experiments. Note that when comparing the efficiency of different approaches, we exclude the sampling time for generating RR sets since the time is the same for all compared approaches. The sampling overheads can be found in Table III.

Parameter and Query. The parameters in this experiments are: (1) k is the number of promoters selected for the campaign **T**. (2) ℓ is the number of viral pieces in **T**. For each viral piece, we generate the topic vector by uniformly sampling a non-zero topic dimension. (3) α, β are parameters in the logistic model. We fix $\beta = 1$ and vary β/α to test the performance against increasing difficulty levels for the user to adopt the idea promoted by **T**. (4) For $\mathbf{V}^p$, we select 10% users from V since in reality not all users are eligible for promoting ads. The parameter setup can be found in Table IV.

TABLE IV: Parameters in the experiments

Parameter	Values
k	10, 20,…, **50**, …, 100
ℓ	1, 2, **3**, 4, 5
β/α	0.3, **0.5**, 0.7
ε	0.1, 0.2, …, **0.5**, …, 0.9

Experiment Settings. All the methods are implemented with C++ and ran on a Ubuntu 14.04 server (Intel(R) Xeon(R) CPU E5-2650 v2 with 256G memory).

B. Tunning Parameter ε

Recall that BAB-P progressively lowers the threshold h for a promoter to be included in the solution (Sec. V-C) by a ratio of $(1 + \varepsilon)$. We examine how the parameter ε affects the solution quality of BAB-P. The results are illustrated in Figure 3. The adoption utility shows a descending trend when ε rises. Thus, the larger ε is, the easier an promoter is included, which could potentially degrade utility. When lowering ε from 0.1 to 0.9, the adoption utility drops by 0.08%, 6.6% and 1.4% for `lastfm`, `dblp` and `tweet` respectively, a result which aligns with the theoretical result presented in Theorem 3. For the remaining part of the experiments, we fix the parameter ε to be 0.5.

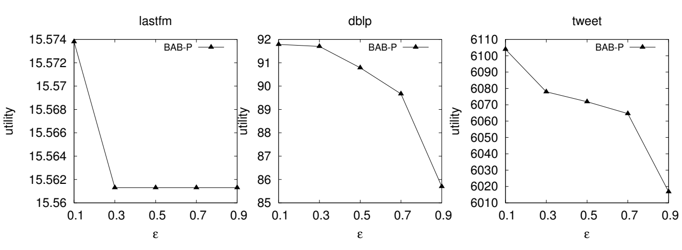


Fig. 3: Tunning parameter ε for BAB-P.

C. Varying the Number k of Promoters

The experimental results for the compared approaches with varying k are shown in Fig. 4. The utility of all proposed methods increases with a larger k, which is expected since more promoters would enable a wider spread of viral messages and lead to higher adoption utility of the campaign. IM and TIM show inferior utilities than those of BAB and BAB-P. IM has the worst utility because it completely ignores influence behavior of viral pieces with different topic distribution. TIM has better results compared with IM since it chooses the promoters who collectively maximize the spread of a single viral piece. However, users have a low probability to adopt the campaign if they only receive one viral piece thus the overall adoption probability could be low. Our proposed BAB and BAB-P achieve superior adoption utility as the branch and bound framework ensures the theoretical guarantee. On top of that, BAB-P demonstrates competitive solution quality with BAB with near-equivalent adoption utilities. Thus, the progressive upper bound estimation technique introduced in BAB-P does not show significant quality degradation even for a larger k empirically.

The efficiency results of all proposed approaches are also shown in Fig. 4 (y-axis is plotted in log-scale). The run time increases for all compared approaches as more promoters are selected into a plan. IM and TIM are efficient since they simply run the greedy maximum cover algorithm on the generated samples. BAB takes much longer run time as it needs to repeatedly invoke the **ComputeBound**$(\cdot)$ functions, each of which is a greedy procedure to select the promoters, until the procedure stops (Algorithm 1). BAB-P optimizes the computation of **ComputeBound**$(\cdot)$ via the progressive upper bound estimation technique with early termination and demonstrates superior efficiency (Algorithm 2), achieving up to 24x, 22x, 8.1x speedups compared with BAB in `lastfm`, `dblp` and `tweet` respectively. BAB-P shows great scalability for a larger k and has converging performance with TIM. This is because BAB-P does not need to scan all candidate promoters to obtain an upper bound estimation and could early terminate even when there are less than k promoters selected. On the contrary, TIM, IM and BAB need to iteratively scan all candidates and thus their performance degrades more severely than BAB-P for larger k.

D. Varying the Number of Viral Pieces ℓ

We examine the effect of conducting a campaign with varying the number of viral pieces (Fig. 5). The utility of all the compared approaches increases as more viral pieces are being promoted. This is because, the adoption probability of a user u increases with the number of viral pieces influencing u when $\beta = 1$, as the model defined by Eqn. 1. The qualities of IM and TIM degrade with larger ℓ compared with BAB and BAB-P since they do not optimize towards multiple viral pieces. Take `tweet` for an example, BAB achieves quality gain of 71x, 2.9x against IM and TIM respectively when $l = 5$, meanwhile BAB-P has competitive qualities against BAB. It is worth-noting that the quality of IM and TIM is severely poor in `tweet`. This is because the average number of non-zero topic influence probability (i.e., $p(e|z)$)) in `tweet` is only 1.5 across all edges in the dataset and `tweet` more topics than `lastfm` and `dblp`. Under this scenario,

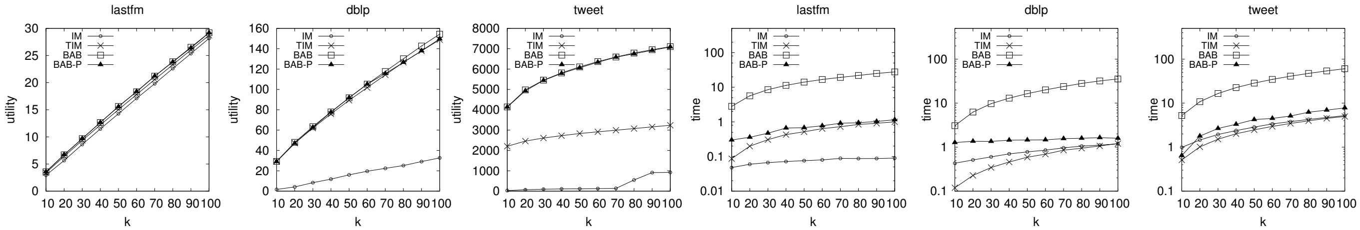


Fig. 4: Varying number k of promoters.

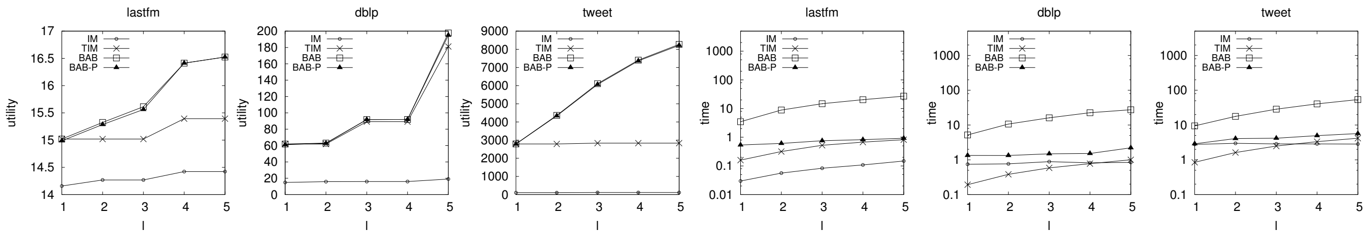


Fig. 5: Varying number of viral pieces ℓ.

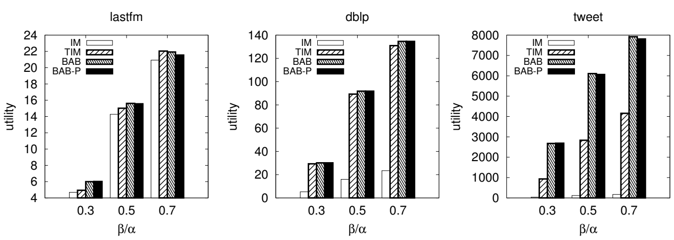


Fig. 6: Varying the ratio β/α.

optimizing a single viral piece results in low influence spread thus produces poor overall adoption utility.

The runtime of all the compared approaches increases with larger ℓ and the performance trend follows previous observations: given competitive solution qualities, BAB-P is able to achieve substantial efficiency improve over BAB. Although IM and TIM have better overall performance, this is at the expense of significant lower utility.

E. Varying the Ratio β/α

The experimental results for the compared approaches with varying β/α are shown in Fig. 6. The utility shows an increasing trend when the ratio is set to larger values. Since we fix $\beta = 1$, β/α rises when α drops. When that happens, the probability of a user v adopting a campaign increases (Eqn. 1), which leads to higher overall utility. We note that BAB and BAB-P has significant better utilities than IM and TIM, with greater improvement ratio for a smaller β/α (larger α). For example, in `tweet`, BAB has 190% utility improvement over TIM when $\beta/\alpha = 0.7$, whereas the improvements pump to 280% when $\beta/\alpha = 0.3$. This is because for a smaller β/α, it is harder for a user to adopt the campaign for receiving one viral piece only. Thus, it requires a more sophisticated optimization technique to find promoters for different viral pieces. We omit the efficiency results since they demonstrate similar patterns with previous results.

VII. Conclusion & Future Work

We introduced a novel formulation of a real-world setting where a network influence campaign $\mathbf{T}$ is multifaceted, consisting of various pieces pertaining to its different facets. We described user adoption behavior in this setting by a logistic model and introduced the *Optimal Influential Piece Assignment* ($\mathcal{OIPA}$) that assigns campaign elements to k promoters in a way to maximizes adoption utility. We showed that it is NP-hard to approximate $\mathcal{OIPA}$ within any constant factor. Nevertheless, we developed a branch-and-bound framework that solves $\mathcal{OIPA}$ with an $(1 - 1/e)$ approximation ratio, a novel formulation of monotone and submodular optimization to compute an upper bound of the unexplored search space, and a progressive pruning-intensive approach to efficiently compute this bound. We show that this progressive approach has an approximation factor of $(1 - 1/e - \varepsilon)$ and much lower time complexity than the plain branch-and-bound approach, by virtue of the power law principle. Our experiments show that our solutions achieve adoption utility superior to that of two intuitive baselines adapted from state-of-the-art IM approaches, while the progressive approach achieves up to 24-fold speedups over the plain approach.

In this work, the viral pieces are spread in the network independently. It would be interesting to study the interdependence of different viral pieces while still optimizing the adoption utility. However, as we have shown the problem to be intractable in general, a promising future direction would be to relax the adoption behavior model in a way that would render the problem tractable, i.e., monotone and submodular, yet add interdependence considerations as well.

Acknowledgement. Yuchen Li was supported by the Singapore MOE Tier 1 grant MSS18C001. Ju Fan was supported by National Natural Science Foundation of China (No. 61602488, No. 61632016), the Research Funds of Renmin University of China (No. 18XNLG18) and the Tencent Social Ads Rhino-Bird Focused Research Grant.

VIII. Appendix

Proof of Lemma 2. We show the lemma as follows:

$$Eqn.\ 6 = \mathbb{E}\left[\frac{n}{\theta}\sum_{i=1,..,\theta}\frac{1}{1+\exp\{\alpha-\beta\cdot\sum_{j=1..l}I^i_{S_j}\}]}\right]$$
$$= \frac{n}{\theta}\sum_{i=1,..,\theta}\mathbb{E}\left[\frac{1}{1+\exp\{\alpha-\beta\cdot\sum_{j=1..l}I^i_{S_j}\}]}\right]$$
$$= \frac{n}{\theta}\sum_{i=1,..,\theta}\mathbb{E}\left[\sum_{v\in V}\frac{1}{n}\cdot\frac{1}{1+\exp\{\alpha-\beta\cdot\sum_{j=1..l}I^v_{S_j}\}]}\right]$$
$$= \sum_{v\in V}\sum_{i=1,..,\theta}\frac{1}{\theta}\cdot\mathbb{E}[p(X_v=1)] = \sigma(\bar{\mathcal{S}})$$

Derivation of $\tau(\bar{\mathcal{S}}|\bar{\mathcal{S}}^a)$. To simplify the derivation, let $x(\bar{\mathcal{S}}) = (\sum_{j=1..l}\beta\cdot I^i_{S_j}-\alpha)$ and $\tau(x(\bar{\mathcal{S}})|\bar{\mathcal{S}}^a) = \frac{1}{1+e^{-x}}$. We further simplify and write $\tau(x)$ instead of $\tau(x(\bar{\mathcal{S}})|\bar{\mathcal{S}}^a)$ when the context is clear. To derive the tangent line $L: y = w\cdot x + b$ where w is the gradient of the line and b is the intercept, we note that this line is unique due to the S shape curve of the logistic function. Suppose we have $\tau(x(\bar{\mathcal{S}}^a) = \beta|\bar{\mathcal{S}}^a)) = \frac{1}{1+e^{-\beta}}$ and we want to refine $\tau(\cdot)$ by adjusting L, there will be two points on the line: $p_1 = (\beta, \frac{1}{1+e^{-\beta}})$, $p_2 = (t, \frac{1}{1+e^{-t}})$ and $w = \tau'(t) = (1-\tau(t))\tau(t) = \frac{e^t}{(1+e^t)^2}$. We wish to express t in terms of β. Unfortunately, we can show that: $\frac{e^{-t}(\beta-t+1)+1}{(1+e^{-t})^2} = \frac{1}{1+e^{-\beta}}$. Neither t nor e^{-t} can be expressed as a close form function of β. Thus, in order to quickly obtain $\tau(\cdot)$, we devise a binary search routine since we know that L is unique and it always lies above $\frac{1}{1+e^{-x}}$. Algorithm 4 shows how to obtain the gradient w and other parameters of the line can be easily obtained with w. We know that w must lie in $(0, 1/4)$ and thus we recursively divide the interval to get the gradient that makes the line just tangent to $\frac{1}{1+e^x}$.

Algorithm 4 Refine(β)

```
1: L ← 0, U ← 1/4
2: while U − L > ε do
3:     w = (U + L)/2
4:     t = log((1+√(1−4w))/(1−√(1−4w)))
5:     v = w · t + 1/(1+e^β) − w · β
6:     if v = 1/(1+e^{−t}) then return w
7:     if v > 1/(1+e^{−t}) then U ← w
8:     else L ← w
9: return U
```

References

[1] A. Anderson, K. Meethan, R. S. Miles, S. Miles, et al. *The changing consumer: Markets and meanings*. Routledge, 2005.

[2] C. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates. Online topic-aware influence maximization queries. In *EDBT*, 2014.

[3] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *ICDM*, 2012.

[4] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *WINE*, pages 306–311, 2007.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[6] J. Bollen, H. Mao, and A. Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*, 2011.

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

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

[9] S. Chen, J. Fan, G. Li, J. Feng, K. Tan, and J. Tang. Online topic-aware influence maximization. *PVLDB*, 8(6), 2015.

[10] S. Datta, A. Majumder, and N. Shrivastava. Viral marketing for multiple products. In *ICDM*, pages 118–127, 2010.

[11] G. Feder, R. Just, and D. Zilberman. Adoption of agricultural innovations in developing countries: A survey. *Economic Development and Cultural Change*, 33(2):255–98, 1985.

[12] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, pages 241–250, 2010.

[13] W. H. Greene. *Econometric Analysis*. Prentice Hall, 5. edition, 200.

[14] X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM*, pages 463–474, 2012.

[15] J. Hstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Acta Mathematica*, pages 627–636, 1996.

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

[17] A. Khan, B. Zehnder, and D. Kossmann. Revenue maximization by viral marketing: A social network host's perspective. In *ICDE*, pages 37–48, 2016.

[18] S. Y. Lee. Examining the factors that influence early adopters' smartphone adoption: The case of college students. *Telemat. Inf.*, 31(2):308–318, 2014.

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

[20] H. Li, S. S. Bhowmick, J. Cui, Y. Gao, and J. Ma. Getreal: Towards realistic selection of influence maximization strategies in competitive networks. In *SIGMOD*, pages 1525–1537, 2015.

[21] Y. Li, J. Fan, Y. Wang, and K. Tan. Influence maximization on social graphs: A survey. *IEEE TKDE*, 30(10):1852–1872, 2018.

[22] Y. Li, J. Fan, D. Zhang, and K.-L. Tan. Discovering your selling points: Personalized social influential tags exploration. In *SIGMOD*, pages 619–634, 2017.

[23] S.-C. Lin, S.-D. Lin, and M.-S. Chen. A learning-based framework to handle multi-round multi-party influence maximization on social networks. In *KDD*, pages 695–704, 2015.

[24] W. Lu, F. Bonchi, A. Goyal, and L. V. Lakshmanan. The bang for the buck: Fair competitive viral marketing from the host perspective. In *KDD*, pages 928–936, 2013.

[25] W. Lu, W. Chen, and L. V. S. Lakshmanan. From competition to complementarity: Comparative influence diffusion and maximization. *PVLDB*, 9(2):60–71, 2015.

[26] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995.

[27] R. Narayanam, e. P. A. Nanavati, Amit A.", T. De Bie, and N. Cristianini. Viral marketing for product cross-sell through social networks. In *PKDD*, pages 581–596, 2012.

[28] A. Rajaraman and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2011.

[29] W. Sierzchula, S. Bakker, K. Maat, and B. van Wee. The influence of financial incentives and other socio-economic factors on electric vehicle adoption. *Energy Policy*, 68(C):183–194, 2014.

[30] X. Song, B. L. Tseng, C.-Y. Lin, and M.-T. Sun. Personalized recommendation driven by information flow. In *SIGIR*, 2006.

[31] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *SIGKDD*, pages 807–816, 2009.

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

[33] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD*, 2014.

[34] K. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2009.

[35] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., 2001.