Viral Marketing Meets Social Advertising: Ad Allocation with Minimum Regret

Cigdem Aslay[1,2] Wei Lu[3] Francesco Bonchi[2] Amit Goyal[4] Laks V.S. Lakshmanan[3]

[1]Univ. Pompeu Fabra
Barcelona, Spain
aslayci@acm.org

[2]Yahoo Labs
Barcelona, Spain
bonchi@yahoo-inc.com

[3]Univ. of British Columbia
Vancouver, Canada
{welu,laks}@cs.ubc.ca

[4]Twitter
San Francisco, CA
agoyal@twitter.com

ABSTRACT

Social advertisement is one of the fastest growing sectors in the digital advertisement landscape: ads in the form of promoted posts are shown in the feed of users of a social networking platform, along with normal social posts; if a user clicks on a promoted post, the host (social network owner) is paid a fixed amount from the advertiser. In this context, allocating ads to users is typically performed by maximizing click-through-rate, i.e., the likelihood that the user will click on the ad. However, this simple strategy fails to leverage the fact the ads can propagate virally through the network, from endorsing users to their followers.

In this paper, we study the problem of allocating ads to users through the viral-marketing lenses. We show that allocation that takes into account the propensity of ads for viral propagation can achieve significantly better performance. However, uncontrolled virality could be undesirable for the host as it creates room for exploitation by the advertisers: hoping to tap uncontrolled virality, an advertiser might declare a lower budget for its marketing campaign, aiming at the same large outcome with a smaller cost.

This creates a challenging trade-off: on the one hand, the host aims at leveraging virality and the network effect to improve advertising efficacy, while on the other hand the host wants to avoid giving away free service due to uncontrolled virality. We formalize this as the problem of ad allocation with minimum regret, which we show is NP-hard and inapproximable w.r.t. any factor. However, we devise an algorithm that provides approximation guarantees w.r.t. the total budget of all advertisers. We develop a scalable version of our approximation algorithm, which we extensively test on four real-world data sets, confirming that our algorithm delivers high quality solutions, is scalable, and significantly outperforms several natural baselines.

1. INTRODUCTION

Advertising on social networking and microblogging platforms is one of the fastest growing sectors in digital advertising, further fueled by the explosion of investments in mobile ads. Social ads are typically implemented by platforms such as Twitter, Tumblr, and Facebook through the mechanism of *promoted posts* shown in the "timeline" (or feed) of their users. A promoted post can be a video, an image, or simply a textual post containing an advertising message. Similar to organic (non-promoted) posts, promoted posts can propagate from user to user in the network by means of social actions such as *"likes"*, *"shares"*, or *"reposts"*.[1] Below, we blur the distinction between these different types of action, and generically refer to them all as *clicks*. These actions have two important aspects in common: (1) they can be seen as an explicit form of acceptance or endorsement of the advertising message; (2) they allow the promoted posts to propagate, so that they might be visible to the *"friends"* or *"followers"* of the endorsing (i.e., clicking) users. In particular, the platform may supplement the ads with *social proofs* such as *"X, Y, and 3 other friends clicked on it"*, which may further increase the chance that a user will click [2, 25].

This type of advertisement usually follows a *cost per engagement* (CPE) model. The *advertiser* enters into an agreement with the platform owner, called the *host*: the advertiser agrees to pay the host an amount $cpe(i)$ for each click received by its ad i. The clicks may come not only from the users who saw i as a promoted ad post, but also their (transitive) followers, who saw it because of viral propagation. The agreement also specifies a budget B_i, that is, the advertiser a_i will pay the host the total cost of all the clicks received by i, *up to a maximum of* B_i. Naturally, posts from different advertisers may be promoted by the host concurrently.

Given that promoted posts are inserted in the timeline of the users, they compete with organic social posts and with one another for a user's attention. A large number of promoted posts (ads) pushed to a user by the system would disrupt user experience, leading to disengagement and eventually abandonment of the platform. To mitigate this, the host limits the number of promoted posts that it shows to a user within a fixed time window, e.g., a maximum of 5 ads per day per user: we call this bound the *user-attention bound*, κ_u, which may be user specific [20].

A subtle point here is that ads directly promoted by the host count against user attention bound. On the contrary, an ad i that flows from a user u to her follower v should not count toward v's attention bound. In fact, v is receiving ad i from user u, whom she is voluntarily following: as such, it cannot be considered "promoted".

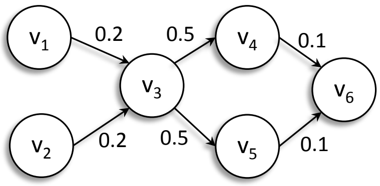
A naïve ad allocation[2] would match each ad with the users most likely to click on the ad. However, the above strategy fails to leverage the possibility of ads propagating virally from endorsing users to their followers. We next illustrate the gains achieved by an allocation that takes viral ad propagation into account.

Proceedings of the VLDB Endowment, Vol. 8, No. 7

[1] `Tumblr`'s CEO David Karp reported (CES 2014) that a normal post is reposted on average 14 times, while promoted posts are on average reposted more than 10 000 times: `http://yhoo.it/1vFfIAc`.

[2] In the rest of the paper we use the form "allocating ads to users" as well as "allocating users to ads" interchangeably.

- Ads = $\{a, b, c, d\}$
- p^i_{uv} (on the edges) are the same $\forall i \in \{a, b, c, d\}$
- $\forall u \in \{v_1, \ldots, v_6\}: \quad \delta(u, a) = 0.9, \quad \delta(u, b) = 0.8,$ $\delta(u, c) = 0.7, \quad \delta(u, d) = 0.6$
- $B_a = 4, B_b = 2, B_c = 2, B_d = 1$
- $\kappa_u = 1 \;\; \forall u \in \{v_1, \ldots, v_6\}$



Allocation $\mathcal{A}$: *maximizing* $\delta(u, i)$
$\langle v_1, a\rangle, \langle v_2, a\rangle, \langle v_3, a\rangle, \langle v_4, a\rangle, \langle v_5, a\rangle, \langle v_6, a\rangle$

$Pr^{\mathcal{A}}(click(v_1, a)) = Pr^{\mathcal{A}}(click(v_2, a)) = 0.9$
$Pr^{\mathcal{A}}(click(v_3, a)) = 1 - (1 - 0.9 \cdot 0.2)^2(1 - 0.9) = 0.93$
$Pr^{\mathcal{A}}(click(v_4, a)) = Pr^{\mathcal{A}}(click(v_5, a)) = 1-(1-0.93\cdot 0.5)(1-0.9) = 0.95$
$Pr^{\mathcal{A}}(click(v_6, a)) = 1 - (1 - 0.95 \cdot 0.1)^2(1 - 0.9) = 0.92$

Expected number of clicks = $2\times 0.9 + 0.93 + 2\times 0.95 + 0.92 = \mathbf{5.55}$

Allocation $\mathcal{B}$: *leveraging virality*
$\langle v_1, a\rangle, \langle v_2, a\rangle, \langle v_3, b\rangle, \langle v_4, c\rangle, \langle v_5, c\rangle, \langle v_6, d\rangle$

$Pr^{\mathcal{B}}(click(v_1, a)) = Pr^{\mathcal{B}}(click(v_2, a)) = 0.9$
$Pr^{\mathcal{B}}(click(v_3, a)) = 1 - (1 - 0.9 \cdot 0.2)^2 = 0.33$
$Pr^{\mathcal{B}}(click(v_4, a)) = Pr^{\mathcal{B}}(click(v_5, a)) = 0.33 \cdot 0.5 = 0.16$
$Pr^{\mathcal{B}}(click(v_6, a)) = 1 - (1 - 0.16 \cdot 0.1)^2 = 0.03$
$Pr^{\mathcal{B}}(click(v_3, b)) = 0.8$
$Pr^{\mathcal{B}}(click(v_4, b)) = Pr^{\mathcal{B}}(click(v_5, b)) = 0.8 \cdot 0.5 = 0.4$
$Pr^{\mathcal{B}}(click(v_6, b)) = 1 - (1 - 0.8 \cdot 0.5 \cdot 0.1)^2 = 0.08$
$Pr^{\mathcal{B}}(click(v_4, c)) = Pr^{\mathcal{B}}(click(v_5, c)) = 0.7$
$Pr^{\mathcal{B}}(click(v_6, c)) = 1 - (1 - 0.7 \cdot 0.1)^2 = 0.14$
$Pr^{\mathcal{B}}(click(v_6, d)) = 0.6$

Expected number of clicks = $2 \cdot 0.9 + 0.33 + 2 \cdot 0.16 + 0.03 + 0.8 + 2 \cdot 0.4 + 0.08 + 2 \cdot 0.7 + 0.14 + 0.6 = \mathbf{6.3}$.

Figure 1: Illustrating viral ad propagation. For simplicity, we round all numbers to the second decimal.

Viral ad propagation: why it matters. For our example we use the toy social network in Fig. 1. We assume that each time a user clicks on a promoted post, the system produces a social proof for such engagement action, thanks to which her followers might be influenced to click as well. In order to model the propagation of (promoted) posts in the network, we can borrow from the rich body of work done in diffusion of information and innovations in social networks. In particular, the *Independent Cascade* (IC) model [19], adapted to our setting, says that once a user u clicks on an ad, she has one independent attempt to try to influence each of her neighbors v. Each attempt succeeds with a probability $p^i_{u,v}$ which depends on the topics of the specific ad i and the influence exerted by u on her neighbor v. The propagation stops when no new users get influenced. Similarly, we model the *intrinsic relevance* of a promoted post i to a user u, as the probability $\delta(u, i)$ that u will click on ad i, based on the content of the ad and her own interest profile, i.e., the prior probability that the user will click on a promoted post in the absence of any social proof. Since the model is probabilistic, we focus on the number of clicks that an ad receives in *expectation*. Formal details of the propagation model, the topic model, and the definition of expected revenue are deferred to § 3.

Consider the example in Fig. 1, where we assume peer influence probabilities (on edges) are equal for all the four ads $\{a, b, c, d\}$. The figure also reports $\delta(u, i)$ and advertiser budgets. For each advertiser, CPE is 1 and the attention bound for every user is 1, i.e., no user wants more than one ad promoted to her by the host. The expected revenue for an allocation is the same as the resulting expected number of clicks, as the CPE is 1. Below, for simplicity, we round all numbers to the second decimal *after* calculating them all.

Let us consider two ways of allocating users to ads by the host. In allocation $\mathcal{A}$, the host matches each user to her top preference(s) based on $\delta(u, i)$, subject to not violating the attention bound. This results in ad a being assigned to all six users, since it has the highest engagement probability for every user. No further ads may be promoted without violating the attention bound. In allocation $\mathcal{B}$, the host recognizes viral propagation of ads and thus assigns a to v_1 and v_2, b to v_3, c to v_4 and v_5, and d to v_6.

Under allocation $\mathcal{A}$, clicks on a may come from all six users: v_1, v_2 click on a with probability 0.9. However, v_3 clicks on a w.p. $(1 - (1 - 0.9 \cdot 0.2)^2(1 - 0.9)) = 0.93$. This is obtained by combining three factors: v_3's engagement probability of 0.9 with ad a, and probability $0.9 \cdot 0.2$ with which each of v_1, v_2 clicks on a and influences v_3 to click on a. In a similar way one can derive the probability of clicking on a for v_4, v_5, and v_6 (reported in the figure). The overall expected revenue for allocation $\mathcal{A}$ is the sum of all clicking probabilities: $2 \times 0.9 + 0.93 + 2 \times 0.95 + 0.92 = \mathbf{5.55}$.

Under allocation $\mathcal{B}$, the ad a is promoted to only v_1 and v_2 (which click on it w.p. 0.9). Every other user that clicks on a does so solely based on social influence. Thus, v_3 clicks on a w.p. $1-(1-0.9\cdot 0.2)^2 = 0.33$. Similarly one can derive the probability of clicking on a for v_4, v_5, and v_6 (reported in the figure). Contributions to the clicks on b can only come from nodes v_3, v_4, v_5, v_6. They click on b, respectively, w.p. 0.8, $0.8 \cdot 0.5 = 0.4$, $0.8 \cdot 0.5 = 0.4$, and $1 - (1 - 0.8 \cdot 0.5 \cdot 0.1)^2 = 0.08$.

Finally, it can be verified that the expected number of clicks on ad c is $0.7 + 0.7 + (1 - (1 - 0.7 \cdot 0.1)^2)$, while on d is just 0.6. The overall number of expected clicks under allocation $\mathcal{B}$ is **6.3**.

Observations: (1) Careful allocation of users to ads that takes viral ad propagation into account can outperform an allocation that merely focuses on immediate clicking likelihood based on the content relevance of the ad to a user's interest profile. It is easy to construct instances where the gap between the two can be arbitrarily high by just replicating the gadget in Fig. 1.

(2) Even though allocation $\mathcal{A}$ ignores the effect of viral ad propagation, it still benefits from the latter, as shown in the calculations. This naturally motivates finding allocations that expressly exploit such propagation in order to maximize the expected revenue.

In this context, we study the problem of *how to strategically allocate users to the advertisers, leveraging social influence and the propensity of ads to propagate*. The major challenges in solving this problem are as follows. Firstly, the host needs to strike a balance between assigning ads to users who are likely to click and assigning them to "influential" users who are likely to boost further propagation of the ads. Moreover, influence may well depend on the "topic" of the ad. E.g., u may influence its neighbor v to different extents on cameras versus health-related products. Therefore, ads which are close in a topic space *will naturally compete* for users that are influential in the same area of the topic space. Summarizing, a good allocation strategy needs to take into account the different CPEs and budgets for different advertisers, users' attention bound and interests, and ads' topical distributions.

An even more complex challenge is brought in by the fact that uncontrolled virality could be undesirable for the host, as it creates room for exploitation by the advertisers: hoping to tap uncontrolled virality, an advertiser might declare a lower budget for its marketing campaign, aiming at the same large outcome with a smaller cost. Thus, from the host perspective, it is important to make sure the expected revenue from an advertiser is as close to the budget as possible: both undershooting and overshooting the budget results in a *regret* for the host, as illustrated in the following example.

EXAMPLE 1. *Consider again our example in Fig. 1. Rounding to the first decimal, allocation $\mathcal{A}$ leads to an overall regret of* $|4 - 5.6| + |2 - 0| + |2 - 0| + |1 - 0| = \mathbf{6.6}$*: the expected revenue exceeds the budget for advertiser* a *by* 1.6 *and falls short of other advertiser budgets by* $2, 2, 1$ *respectively. Similarly, for allocation* $\mathcal{B}$*, the regret is* $|4-2.5|+|2-1.7|+|2-1.5|+|1-0.6| = \mathbf{2.7}$. □

The host knows it will not be paid beyond the budget of each advertiser, so that any excess above the budget is essentially "free service" given away by the host, which causes regret, and any shortfall w.r.t. the budget is a lost revenue opportunity which causes regret as well. This creates a challenging trade-off: on the one hand, the host aims at leveraging virality and the network effect to improve advertising efficacy, while on the other hand the host wants to avoid giving away free service due to uncontrolled virality.

Contributions and roadmap. In this paper we make the following major contributions:

- We propose a novel problem domain of allocating users to advertisers for promoting advertisement posts, taking advantage of network effect, while paying attention to important practical factors like relevance of ad, effect of social proof, user's attention bound, and limited advertiser budgets (§ 3).
- We formally define the problem of *minimizing regret* in allocating users to ads (§ 3), and show that it is NP-hard and is NP-hard to approximate within any factor (§ 4).
- We develop a simple greedy algorithm and establish an upper bound on the regret it achieves as a function of advertisers' total budget (§ 4.1).
- We then devise a scalable instantiation of the greedy algorithm by leveraging the notion of *random reverse-reachable sets* [5, 24] (§ 5).
- Our extensive experimentation on four real datasets confirms that our algorithm is scalable and it delivers high quality solutions, significantly outperforming natural baselines (§ 6).

To the best of our knowledge, regret minimization in the context of promoting multiple ads in a social network, subject to budget and attention bounds has not been studied before. Related work is discussed in § 2, while § 7 concludes the paper discussing future work. Some of the proofs, omitted due to lack of space, can be found in an extended version of the paper [1].

2. RELATED WORK

Substantial work has been done on viral marketing, which mainly focuses on a key algorithmic problem – *influence maximization* [7, 17, 19]. Kempe et al. [19] formulated influence maximization as a discrete optimization problem: given a social graph and a number k, find a set S of k nodes, such that by activating them one maximizes the expected spread of influence $\sigma(S)$ under a certain propagation model, e.g., the *Independent Cascade* (IC) model. Influence maximization is NP-hard, but the function $\sigma(S)$ is *monotone*[3] and *submodular*[4] [19]. Exploiting these properties, the simple greedy algorithm that at each step extends the seed set with the node providing the largest marginal gain, provides a $(1 - 1/e)$-approximation to the optimum [23]. The greedy algorithm is computationally prohibitive, since selecting the node with the largest marginal gain is #P-hard [7], and is typically approximated by numerous Monte Carlo simulations [19]. However, running many such simulations is extremely costly, and thus considerable effort has been devoted to developing efficient and scalable influence maximization algorithms: in §5 we will review some of the latest advances in this area which help us devise our algorithms.

Datta et al. [9] study influence maximization with multiple items, under a user attention constraint. However, as in classical influence maximization, their objective is to maximize the overall influence spread, and the budget is w.r.t. the size of the seed set, so without any CPE model. Their diffusion model is the (topic-blind) IC model, which also doesn't model the competition among similar items. Du et al. [12] study influence maximization over multiple non-competing products subject to user attention constraints and budget constraints, and develop approximation algorithms in a continuous time setting. Lin et al. [20] study the problem of maximizing influence spread from a website's perspective: how to dynamically push items to users based on user preference and social influence. The push mechanism is also subject to user attention bounds. Their framework is based on Markov Decision Processes (MDPs).

Our work departs from the body of work in this field by looking at the possibility of integrating viral marketing into existing social advertising models and by studying a fundamentally different objective: *minimize host's regret*. A noteworthy feature of our work is that, as will be shown in §6, the budgets we use are such that thousands of seeds are required to minimize regret. Scalability of algorithms for selecting thousands of seeds over large networks has not been demonstrated before.

While social advertising is still in its infancy, it fits in the more general (and mature) area of computational advertising that has attracted a lot of interest during the last decade. The central problem of computational advertising is to find the "best match" between a given user in a given context and a suitable advertisement. The context could be a user entering a query in a search engine ("sponsored search"), reading a web page ("content match" and "display ads"), or watching a movie on a portable device, etc. Considerable work has been done in sponsored search and display ads [10, 13, 14, 16, 22]. Search engines show ads deemed relevant to user-issued queries, so as to maximize click-through rates and in turn, revenue. Revenue maximization in this context is formalized as the well-known *Adwords* problem [**?**]. For a comprehensive treatment, see a recent survey [21]. Our work fundamentally differs from this as we are concerned with the *virality* of ads when making allocations: this concept is still largely unexplored in computational advertising.

Recently, Tucker [25] and Bakshy et al. [2] conducted field experiments on Facebook and demonstrated that adding social proofs to sponsored posts in Facebook's News Feed significantly increased the click-through rate. Their findings empirically confirm the benefits of social influence, paving the way for the application of viral marketing in social advertising, as we do in our work.

3. PROBLEM STATEMENT

The Ingredients. The computational problem studied in this paper is from the host perspective. The host owns: (i) a *directed social graph* $G = (V, E)$, where an arc (u, v) means that v follows u, thus v can see u's posts and can be influenced by u; (ii) a *topic model* for ads and users' interest, defined on a space of K topics; (iii) a *topic-aware influence propagation model* defined on the social graph G and the topic model.

[3] $\sigma(S) \leq \sigma(T)$ whenever $S \subseteq T$.

[4] $\sigma(S \cup \{w\}) - \sigma(S) \geq \sigma(T \cup \{w\}) - \sigma(T)$ whenever $S \subseteq T$.

The key idea behind the topic modeling is to introduce a hidden variable Z that can range among K states. Each topic (i.e., state of the latent variable) represents an abstract interest/pattern and intuitively models the underlying cause for each data observation (a user clicking on an ad). In our setting the host owns a precomputed probabilistic topic model. The actual method used for producing the model is not important at this stage: it could be, e.g., the popular *Latent Dirichlet Allocation* (LDA) [4], or any other method. What is relevant is that the topic model maps each ad i to a topic distribution $\vec{\gamma_i}$ over the latent topic space, formally: $\gamma_i^z = Pr(Z = z|i)$ with $\Sigma_{z=1}^K \gamma_i^z = 1$.

Propagation Model. The propagation model governs the way that ads propagate in the social network driven by social influence. In this work, we extend a simple topic-aware propagation model introduced by Barbieri et al. [3], with Click-Through Probabilities (CTPs) for seeds: we refer to the set of users S_i that receive ad i directly as a promoted post from the host as the *seed set* for ad i. In the *Topic-aware Independent Cascade* model (TIC) of [3], the propagation proceeds as follows: when a node u first clicks an ad i, it has one chance of influencing each inactive neighbor v, independently of the history thus far. This succeeds with a probability that is the weighted average of the arc probability w.r.t. the topic distribution of the ad i:

$$p^i_{u,v} = \sum\nolimits_{z=1}^{K} \gamma_i^z \cdot p^z_{u,v}. \quad (1)$$

For each topic z and for a seed node u, the probability $p^z_{H,u}$ represents the likelihood of u clicking on a promoted post for topic z. Thus the CTP $\delta(u, i)$ that u clicks on the promoted post i in absence of any social proof, is the weighted average (as in Eq. (1)) of the probabilities $p^z_{H,u}$ w.r.t. the topic distribution of i. In our extended TIC-CTP model, each $u \in S_i$ accepts to be a seed, i.e., clicks on ad i, with probability $\delta(u, i)$ when targeted. The rest of the propagation process remains the same as in TIC.

Following the literature on influence maximization we denote with $\sigma_i(S_i)$ the *expected number of clicks* (according to the TIC-CTP model) for ad i when the seed set is S_i. The corresponding expected revenue is $\Pi_i(S_i) = \sigma_i(S_i) \cdot cpe(i)$, where $cpe(i)$ is the cost-per-engagement that a_i and the host have agreed on.

We observe that for a fixed ad i, with topic distribution $\vec{\gamma_i}$, the TIC-CTP model boils down to the standard *Independent Cascade* (IC) model [19] with CTPs, where again, a seed may activate with a probability. We next expose the relationship between the expected spread a $\sigma^{ic}(S)$ for the classical IC model without CTPs, and the expected spread under the TIC-CTP model for a given ad i.

LEMMA 1. *Given an instance of the TIC-CTP model, and a fixed ad i, with topic distribution $\vec{\gamma_i}$, build an instance of IC by setting the probability over each edge (u, v) as in Eq. 1. Now, consider any node u, and any set S of nodes. Let $\delta(u, i)$ be the CTP for u clicking on the promoted post i. Then we have*

$$\delta(u, i)[\sigma^{ic}(S \cup \{u\}) - \sigma^{ic}(S)] = \sigma_i(S \cup \{u\}) - \sigma_i(S). \quad (2)$$

The simple proof, omitted due to space constraints, can be found in [1]. A corollary of the above lemma is that for a fixed $\vec{\gamma_i}$, the expected spread $\sigma_i(\cdot)$ function under the TIC-CTP model, inherits the properties of monotonicity and submodularity from the IC model (see Sec. 2 and [3, 19]). In turn, $\Pi_i(S_i) = cpe(i) \cdot \sigma_i(S_i)$ is also monotone and submodular, being a non-negative linear combination of monotone submodular functions.

Budget and Regret. As in any other advertisement model, we assume that each advertiser a_i has a finite budget B_i for a campaign on ad i, which limits the maximum amount that a_i will pay the host. The host needs to allocate seeds to each of the ads that it has agreed to promote, resulting in an allocation $\mathcal{S} = (S_1, ..., S_h)$. The expected revenue from the campaign may fall short of the budget (i.e., $\Pi_i(S_i) < B_i$) or overshoot it (i.e., $\Pi_i(S_i) > B_i$). An advertiser's natural goal is to make its expected revenue as close to B_i as possible: the former situation is lost opportunity to make money whereas the latter amounts to "free service" by the host to the advertiser. Both are undesirable. Thus, one option to define the host's regret for seed set allocation S_i for advertiser a_i is as $|B_i - \Pi_i(S_i)|$.

Note that this definition of regret has the drawback that it does not discriminate between small and large seed sets: given two seed sets S_1 and S_2 with the same regret as defined above, and with $|S_1| \ll |S_2|$, this definition does not prefer one over the other. In practice, it is desirable to achieve a low regret with a small number of seeds. By drawing on the inspiration from the optimization literature [6], where an additional penalty corresponding to the complexity of the solution is added to the error function to discourage overfitting, we propose to add a similar penalty term to discourage the use of large seed sets. Hence we define the *overall regret* as

$$\mathcal{R}_i(S_i) = |B_i - \Pi_i(S_i)| + \lambda \cdot |S_i|. \quad (3)$$

Here, $\lambda \cdot |S_i|$ can be seen as a penalty for the use of a seed set: the larger its size, the greater the penalty. This discourages the choice of a large number of poor quality seeds to exhaust the budget. When $\lambda = 0$, no penalty is levied and the "raw" regret corresponding to the budget alone is measured. We assume w.l.o.g. that the scalar λ encapsulates CPE such that the term $\lambda|S_i|$ is in the same monetary unit as B_i. How small/large should λ be? We will address this question in the next section.

The overall regret from an allocation $\mathcal{S} = (S_1, ..., S_h)$ to all advertisers is

$$\mathcal{R}(\mathcal{S}) = \sum\nolimits_{i=1}^{h} \mathcal{R}_i(S_i). \quad (4)$$

EXAMPLE 2. *In Example 1, the regrets reported for allocations $\mathcal{A}$ (6.6) and $\mathcal{B}$ (2.7) correspond to $\lambda = 0$. When $\lambda = 0.1$, the regrets change to $6.6 + 0.1 \times 6 = 7.2$ for $\mathcal{A}$ and to $2.7 + 0.1 \times 6 = 3.3$ for $\mathcal{B}$.* □

As noted in the introduction, in practice, the number of ads that can be promoted to a user may be limited. The host can even personalize this number depending on users' activity. We model this using an attention bound κ_u for user u. An allocation $\mathcal{S} = (S_1, ..., S_h)$ is called *valid* provided for every user $u \in V$, $|\{S_i \in \mathcal{S} \mid u \in S_i\}| \leq \kappa_u$, i.e., no more than κ_u ads are promoted to u by the allocation. We are now ready to formally state the problem we study.

PROBLEM 1 (REGRET-MINIMIZATION). *We are given h advertisers $a_1, \ldots, a_h$, where each a_i has an ad i described by topic-distribution $\vec{\gamma_i}$, a budget B_i, and a cost-per-engagement $cpe(i)$. Also given is a social graph $G = (V, E)$ with a probability $p^z_{u,v}$ for each edge $(u, v) \in E$ and each topic $z \in [1, K]$, an attention bound κ_u, $\forall u \in V$, and a penalty parameter $\lambda \geq 0$. The task is to compute a valid allocation $\mathcal{S} = (S_1, \ldots, S_h)$ that minimizes the overall regret:*

$$\mathcal{S} = \underset{\substack{\mathcal{T}=(T_1,\ldots,T_h): T_i \subseteq V \\ \mathcal{T} \text{ is valid}}}{\arg\min} \mathcal{R}(\mathcal{T}).$$

Discussion. Note that $\Pi_i(S_i)$ denotes the expected revenue from advertiser a_i. In reality, the actual revenue depends on the number of engagements the ad *actually* receives. Thus, the uncertainty in $\Pi_i(S_i)$ may result in a loss of revenue. Another concern could be that regret on the positive side ($\Pi_i(S_i) > B_i$) is more acceptable than on the negative side ($\Pi_i(S_i) < B_i$), as one can argue that maximizing revenue is a more critical goal even if it comes at the expense of a small and reasonable amount of free service. Our framework can accommodate such concerns and can easily address them. For instance, instead of defining raw regret as $|B_i - \Pi_i(S_i)|$, we can define it as $|B'_i - \Pi_i(S_i)|$, where $B'_i = (1 + \beta) \cdot B_i$. The idea is to artificially boost the budget B_i with parameter β allowing maximization of revenue while keeping the free service within a modest limit. This small change has no impact on the validity of our results and algorithms. Theorem 2 provides an upper bound on the regret achieved by our allocation algorithm (§ 4.1). The bound remains intact except that in place of the original budget B_i, we should use the boosted budget B'_i. This remark applies to all our results. We henceforth study the problem as defined in Problem 1.

4. THEORETICAL ANALYSIS

We first show that REGRET-MINIMIZATION is not only NP-hard to solve optimally, but is also NP-hard to approximate within any factor (Theorem 1). On the positive side, we propose a greedy algorithm and conduct a careful analysis to establish a bound on the regret it can achieve as a function of the budget (Theorems 2-4).

THEOREM 1. REGRET-MINIMIZATION *is NP-hard and is NP-hard to approximate within any factor.*

PROOF. We prove hardness for the special case where $\lambda = 0$, using a reduction from 3-PARTITION [15].

Given a set $X = \{x_1, ..., x_{3m}\}$ of positive integers whose sum is C, with $x_i \in (C/4m, C/2m)$, $\forall i$, 3-PARTITION asks whether X can be partitioned into m disjoint 3-element subsets, such that the sum of elements in each partition is the same (= C/m). This problem is known to be strongly NP-hard, i.e., it remains NP-hard even if the integers x_i are bounded above by a polynomial in m [15]. Thus, we may assume that C is bounded by a polynomial in m.

Given an instance $\mathcal{I}$ of 3-PARTITION, we construct an instance $\mathcal{J}$ of REGRET-MINIMIZATION as follows. First, we set the number of advertisers $h = m$ and let the cost-per-engagement (CPE) be 1 for all advertisers. Then, we construct a directed bipartite graph $G = (U \cup V, E)$: for each number x_i, G has one node $u_i \in U$ with $x_i - 1$ outneighbors in V, with all influence probabilities set to 1. We refer to members of U (resp., V) as "U" nodes (resp., "V" nodes) below. Set all advertiser budgets to $B_i = C/m$, $1 \le i \le m$ and the attention bound of every user to 1. This will result in a total of C nodes in the instance of REGRET-MINIMIZATION. Since C is bounded by a polynomial in m, the reduction is achieved in polynomial time.

We next show that if REGRET-MINIMIZATION can be solved in polynomial time, so can 3-PARTITION, implying hardness. To that end, assume there exists an algorithm $\mathbf{A}$ that solves REGRET-MINIMIZATION optimally. We can use $\mathbf{A}$ to distinguish between YES- and NO-instances of 3-PARTITION as follows. Run $\mathbf{A}$ on $\mathcal{J}$ to yield a seed set allocation $\mathcal{S} = (S_1, ..., S_m)$. We claim that $\mathcal{I}$ is a YES-instance of 3-PARTITION iff $\mathcal{R}(\mathcal{S}) = 0$, i.e., the total regret of the allocation $\mathcal{S}$ is zero.

($\Longrightarrow$): Suppose $\mathcal{R}(\mathcal{S}) = 0$. This implies the regret of every advertiser must be zero, i.e., $\Pi_i(S_i) = B_i = C/m$. We shall show that in this case, each S_i must consist of 3 "U" nodes whose spread sums to C/m. From this, it follows that the 3-element subsets $X_i := \{x_j \in X \mid u_j \in S_i\}$ witness the fact that $\mathcal{I}$ is a YES-instance. Suppose $|S_i| \neq 3$ for some i. It is trivial to see that each seed set S_i can contain only the "U" nodes, for the spread of any "V" node is just 1. If $|S_i| \neq 3$, then $\Pi_i(S_i) = \sum_{u_j \in S_i} x_j \neq C/m$, since all numbers are in the open interval $(C/4m, C/2m)$. This shows that every seed set S_i in the above allocation must have size 3, which was to be shown.

($\Longleftarrow$): Suppose $X_1, ..., X_m$ are disjoint 3-element subsets of X that each sum to C/m. By choosing the corresponding "U"-nodes we get a seed set allocation whose total regret is zero.

We just proved that REGRET-MINIMIZATION is NP-hard. To see hardness of approximation, suppose $\mathbf{B}$ is an algorithm that approximates REGRET-MINIMIZATION within a factor of α. That is, the regret achieved by algorithm $\mathbf{B}$ on any instance of REGRET-MINIMIZATION is $\le \alpha \cdot OPT$, where OPT is the optimal (least) regret. Using the same reduction as above, we can see that the optimal regret on the reduced instance $\mathcal{J}$ above is 0. On this instance, the regret achieved by algorithm $\mathbf{B}$ is $\le \alpha \cdot 0 = 0$, i.e., algorithm $\mathbf{B}$ can solve REGRET-MINIMIZATION optimally in polynomial time, which is shown above to be impossible unless $P = NP$. □

Algorithm 1: Greedy Algorithm

Input : $G = (V, E)$; λ; attention bounds $\kappa_u, \forall u \in V$; items $\vec{\gamma}_i$ with $cpe(i)$ & budget B_i, $i = 1, \ldots, h$; $\delta(u, i), \forall u \forall i$
Output: $S_1, \ldots, S_h$

1 $S_i \leftarrow \emptyset, \forall i = 1, \ldots, h$
2 **while** *true* **do**
3 $\quad (u, a_i) \leftarrow \arg\max_{v, a_j} \mathcal{R}_j(S_j) - \mathcal{R}_j(S_j \cup \{v\})$,
$\quad\quad$ subject to: $|\{S_\ell | v \in S_\ell\}| < \kappa_v$ **and**
$\quad\quad\quad \mathcal{R}_j(S_j \cup \{v\}) \le \mathcal{R}_j(S_j))$
4 $\quad$ **if** (u, a_i) is **null then** return **else** $S_i \leftarrow S_i \cup \{u\}$

4.1 A Greedy Algorithm

Due to the hardness of approximation of Problem 1, no polynomial algorithm can provide any theoretical guarantees w.r.t. optimal overall regret. Still, instead of jumping to heuristics without any guarantee, we present an intuitive greedy algorithm (pseudocode in Algorithm 1) with theoretical guarantees in terms of the total budget. It is worth noting that analyzing regret w.r.t. the total budget has real-world relevance, as budget is a concrete monetary and known quantity (unlike optimal value of regret) which makes it easy to understand regret from a business perspective.

The algorithm starts by initializing all seed sets to be empty (line 1). It keeps selecting and allocating seeds until regret can no longer be minimized. In each iteration, it finds a user-advertiser pair (u, a_i) such that u's attention bound is not reached (that is, $|\{S_i | u \in S_i\}| < \kappa_u$) and adding u to S_i (the seed set of a_i) yields the largest decrease in regret among all valid pairs. Clearly, we want to ensure that regret does not increase in an iteration (that is, $\mathcal{R}_i(S_i \cup \{u\}) < \mathcal{R}_i(S_i)$) (line 3). The user u is then added to S_i. If no such pair can be found, that is, regret cannot be reduced further, the algorithm terminates (line 4).

Before stating our results on bounding the overall regret achieved by the greedy algorithm, we identify extreme (and unrealistic) situations where no such guarantees may be possible.

Practical considerations. Consider a network with n users, one advertiser with a CPE of 1 and a budget $B \gg n$. Assume CTPs are all 1. Clearly, even if all n users are allocated to the advertiser, the regret approaches 100% of B, as most of the budget cannot be tapped.

At another extreme, consider a dense network with n users (e.g., clique), one advertiser with a cpe of 1 and a budget $B \ll n$. Suppose the network has high influence probabilities, with the result that assigning *any* one seed u to the advertiser will result in an expected revenue $\Pi(\{u\}) \gg B$. In this case, the allocation with the least regret is the empty allocation (!) and the regret is exactly B!

In many practical settings, the budgets are large enough that the marginal gain of any one node is a small fraction of the budget and small enough compared to the network size, in that there are enough nodes in the network to allocate to each advertiser in order to exhaust or exceed the budget.

4.2 The General Case

In this subsection, we establish an upper bound on the regret achieved by Algorithm 1, when every candidate seed has essentially an unlimited attention bound. For convenience, we refer to the first term in the definition of regret (cf. Eq. 3) as *budget-regret* and the second term as *seed-regret*. The first one reflects the regret arising from undershooting or overshooting the budget and the second arises from utilizing seeds which are the host's resources. For a seed set S_i for ad i, the *marginal gain* of a node $x \in V \setminus S_i$ is defined as $MG_i(x|S_i) := \Pi_i(S_i \cup \{x\}) - \Pi_i(S_i)$. By submodularity, the marginal gain of any node is the greatest w.r.t. the empty seed set, i.e., $MG_i(x|\emptyset) = \Pi_i(\{x\})$. Let p_i be the maximum marginal gain of any node w.r.t. ad i, as a fraction of its budget B_i, i.e., $p_i := \max_{x \in V} \Pi_i(\{x\})/B_i$. As discussed at the end of the previous subsection, we assume that the network and the budgets are such that $p_i \in (0, 1)$, for all ads i. In practice, p_i tends to be a small fraction of the budget B_i. Finally, we define $p_{max} := \max_{i=1}^{h} p_i$ to be the maximum p_i among all advertisers.

THEOREM 2. *Suppose that for every node u, the attention bound $\kappa_u \geq h$, the number of advertisers, and that $\lambda \leq \delta(u, i) \cdot cpe(i)$, $\forall$ user u and ad i. Then the regret incurred by Algorithm 1 upon termination is at most*

$$\sum_{i=1}^{h} \frac{p_i B_i + \lambda}{2} + \lambda \cdot \sum_{i=1}^{h} \left(1 + s_{opt}^{i} \lceil \ln \frac{1}{p_i/2 - \lambda/2B_i} \rceil \right),$$

where s_{opt}^{i} is the smallest number of seeds required for reaching or exceeding the budget B_i for ad i.

Discussion: The term $\delta(u, i) \cdot cpe(i)$ corresponds to the expected revenue from user u clicking on i (without considering the network effect). Thus, the assumption on λ, that it is no more than the expected revenue from any one user clicking on an ad, keeps the penalty term small, since in practice click-through probabilities tend to be small. Secondly, the regret bound given by the theorem can be understood as follows. Upon termination, the budget-regret from Greedy's allocation is at most $(1/2)p_{max}B$ (plus a small constant $\lambda/2$). The theorem says that Greedy achieves such a budget-regret while being frugal w.r.t. the number of seeds it uses. Indeed, its seed-regret is bounded by the minimum number of seeds that an optimal algorithm would use to reach the budget, multiplied by a logarithmic factor.

PROOF OF THEOREM 2. We establish a series of claims.

CLAIM 1. *Suppose S_i is the seed set allocated to advertiser a_i and $\Pi_i(S_i) < B_i$. Then the greedy algorithm will add a node x to S_i iff $|\Pi_i(S_i \cup \{x\}) - B_i| < |\Pi_i(S_i) - B_i|$ and $x = \arg\max_{w \in V \setminus S_i}(|\Pi_i(S_i) - B_i| - |\Pi_i(S_i \cup \{w\}) - B_i|)$, with ties broken arbitrarily.* (Proof in [1].)

CLAIM 2. *The budget-regret of Greedy for advertiser a_i, upon termination, is at most $(p_i B_i + \lambda)/2$.*

PROOF OF CLAIM: Consider any iteration j. Let x be the seed allocated to advertiser a_i in this iteration. The following cases arise.
• Case 1: $\Pi_i(S_i \cup \{x\}) < p_i B_i$. By submodularity, for any node $y \in V \setminus (S_i \cup \{x\}) : MG_i(y|S_i \cup \{x\}) \leq MG_i(y|\emptyset) \leq p_i B_i$. Thus, from Claim 1, we know the algorithm will continue adding seeds to S_i until Case 2 (below) is reached.
• Case 2: $\Pi(S_i \cup \{x\}) \geq p_i B_i$.
• Case 2a: $\Pi(S_i \cup \{x\}) < B_i$. If x is the last seed added to S_i, then $\forall y \in V \setminus (S_i \cup \{x\}) : B_i - \Pi(S_i \cup \{x\}) + \lambda(|S_i| + 1) < \Pi_i(S_i \cup \{x\} \cup \{y\}) - B_i + \lambda(|S_i| + 2)$. Notice that upon adding any such y, a cross-over must occur w.r.t. B_i: suppose otherwise, then adding y would cause net drop in regret and the algorithm would just add y to $S_i \cup \{x\}$, a contradiction. Simplifying, we get $B_i - \Pi_i(S_i \cup \{x\}) < \Pi_i(S_i \cup \{x\} \cup \{y\}) - B_i + \lambda$. Also by submodularity, we have $\Pi_i(S_i \cup \{x\} \cup \{y\}) - \Pi_i(S_i \cup \{x\}) \leq p_i B_i$. Thus,
$\implies \Pi_i(S_i \cup \{x\} \cup \{y\}) - B_i + B_i - \Pi_i(S_i \cup \{x\}) \leq p_i B_i$.
$\implies 2(B_i - \Pi_i(S_i \cup \{x\})) - \lambda \leq p_i B_i$.
$\implies B_i - \Pi_i(S_i \cup \{x\}) \leq (p_i B_i + \lambda)/2$.
• Case 2b: $\Pi_i(S_i \cup \{x\}) > B_i$. Since Greedy just added x to S_i, we infer that $\Pi_i(S_i) < B_i$ and $[B_i - \Pi_i(S_i)] + \lambda|S_i| \geq \Pi_i(S_i \cup \{x\}) - B_i + \lambda(|S_i| + 1)$.
$\implies B_i - \Pi_i(S_i) \geq \Pi_i(S_i \cup \{x\}) - B_i + \lambda$. Clearly, x must be the last seed added to S_i, as any future additions will strictly raise the regret. By submodularity, we have
$\Pi_i(S_i \cup \{x\}) - \Pi_i(S_i) \leq p_i B_i$.
$\implies \Pi_i(S_i \cup \{x\}) - B_i + B_i - \Pi_i(S_i) \leq p_i B_i$.
$\implies 2(\Pi_i(S_i \cup \{x\}) - B_i) + \lambda \leq p_i B_i$.
$\implies \Pi_i(S_i \cup \{x\}) - B_i \leq (p_i B_i - \lambda)/2$.
By combining both cases, we conclude that the budget-regret of Greedy for a_i upon termination is $\leq (p_i B_i + \lambda)/2$. □

Next, define $\eta_0 = B_i$. Let S_i^j be the seed set assigned to advertiser a_i by Greedy after iteration j. Let $\eta_j := \eta_0 - \Pi_i(S_i^j)$, i.e., the shortfall of the achieved revenue w.r.t. the budget B_i, after iteration j, for advertiser a_i.

CLAIM 3. *After iteration j, $\exists x \in V \setminus S_i^j : \Pi_i(S_i \cup \{x\}) - \Pi_i(S_i) \geq 1/s_{opt}^{i} \cdot \eta_j$, where s_{opt}^{i} is the minimum number of seeds needed to achieve a revenue no less than B_i.*

PROOF OF CLAIM: Suppose otherwise. Let S_i^* be the seeds allocated to advertiser a_i by the optimal algorithm for achieving a revenue no less than B_i. Add seeds in $S_i^* \setminus S_i^j$ one by one to S_i^j. Since none of them has a marginal gain w.r.t. S_i that is $\geq 1/s_{opt}^{i} \cdot \eta_j$, it follows by submodularity that $\Pi_i(S_i^j \cup S_i^*) \leq \Pi(S_i^j) + s_{opt}^{i} \cdot 1/s_{opt}^{i} \cdot \eta_j < B_i$, a contradiction. □

It follows from the above proof that $\eta_j \leq \eta_{j-1} \cdot (1 - 1/s_{opt}^{i})$, which implies that $\eta_j \leq 1/\eta_{j-1} \cdot e^{-1/s_{opt}^{i}}$. Unwinding, we get $\eta_j \leq \eta_0 \cdot e^{-j/s_{opt}^{i}}$. Suppose Greedy stops in ℓ iterations. We showed above that the budget-regret of Greedy, for advertiser a_i, at the end of this iteration, is either at most $(p_i \cdot B_i + \lambda)/2$ or is at most $(p_i B_i - \lambda)/2$ depending on the case that applies. Of these, the latter is more stringent w.r.t. the #iterations Greedy will take, and hence w.r.t. the #seeds it will allocate to a_i. So, in iteration $\ell - 1$, we have $\eta_{\ell-1} \geq (p_i B_i - \lambda)/2$. That is,
$\eta_{\ell-1} = B_i \cdot e^{-(\ell-1)/s_{opt}^{i}} \geq (p_i B_i - \lambda)/2$, or
$\implies e^{-(\ell-1)/s_{opt}^{i}} \geq (p_i - \lambda/B_i])/2$.
$\implies \ell \leq 1 + s_{opt}^{i} \cdot \lceil \ln\{1/(p_i/2 - \lambda/2B_i)\} \rceil$. Notice that this is an upper bound on $|S_i^{\ell}|$. We just proved

CLAIM 4. *When Greedy terminates, the seed-regret for advertiser a_i, upon termination, is at most $\lambda \cdot (1 + s_{opt}^{i} \cdot \lceil \ln\{1/(p_i/2 - \lambda/2B_i)\} \rceil)$.* □

Combining all the claims above, we can infer that the overall regret of Greedy upon termination is at most $\sum_{i=1}^{h}(p_i B_i + \lambda)/2 + \lambda \sum_{i=1}^{h}[1 + s_{opt}^{i}(1 + \lceil \ln\{1/(p_i/2 - \lambda/2B_i)\} \rceil$. □

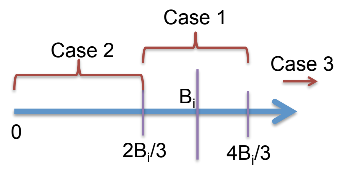


Figure 2: Interpretation of Theorem 3.

4.3 The Case of $\lambda = 0$

In this subsection, we focus on the regret bound achieved by Greedy in the special case that $\lambda = 0$, i.e., the overall regret is just the budget-regret. While the results here can be more or less seen as special cases of Theorem 2, it is illuminating to restrict attention to this special case. Our first result follows.

THEOREM 3. *Consider an instance of* REGRET-MINIMIZATION *that admits a seed allocation whose total regret is bounded by a third of the total budget. Then Algorithm 1 outputs an allocation $\mathcal{S}$ with a total regret $\mathcal{R}(\mathcal{S}) \leq \frac{1}{3} \cdot B$, where $B = \sum_{i=1}^{h} B_i$ is the total budget.*

In the proof of this theorem [1] three cases arise, as illustrated in Fig. 2. The crux of the proof consists in showing that If the revenue achieved falls in Case 2, then Greedy will add more seeds until Case 1 is reached and that Case 3 will never happen.

The regret bound established above is conservative, and unlike Theorem 2, does not make any assumptions about the marginal gains of seed nodes. In practice, as previously noted, most real networks tend to have low influence probabilities and consequently, the marginal gain of any single node tends to be a small fraction of the budget. Using this, we can establish a tighter bound on the regret achieved by Greedy.

THEOREM 4. *On any input instance that admits an allocation with total regret bounded by $\min\{\frac{p_{max}}{2}, 1 - p_{max}\} \cdot B$, Algorithm 1 delivers an allocation $\mathcal{S}$ so that $\mathcal{R}(\mathcal{S}) \leq \min\{\frac{p_{max}}{2}, 1 - p_{max}\} \cdot B$.*

We note that this claim generalizes Theorem 3. In fact, the two bounds: $\frac{p_{max}}{2}$ and $1 - p_{max}$ meet at the value of $1/3$ when $p_{max} = 2/3$. In practice, p_{max} may be much smaller, making the bound better. Full proofs of both theorems are in [1].

5. SCALABLE ALGORITHMS

Algorithm 1 (Greedy) involves a large number of calls to influence spread computations, to find the node for each advertiser a_i that yields the maximum decrease in regret $\mathcal{R}_i(S_i)$. Given any seed set S, computing its *exact* influence spread $\sigma(S)$ under the IC model is #P-hard [7], and this hardness trivially carries over to the topic-aware IC model [3] with CTPs. A common practice is to use Monte Carlo (MC) simulations to estimate influence spread [19]. However, accurate estimation requires a large number of MC simulations, denoted r, which is prohibitively expensive and not scalable: Algorithm 1 runs in $O(\sum_{i=1}^{h} \frac{B_i}{cpe(i)} \cdot (1 + \frac{p_{max}}{2}) \cdot |V| \cdot |E| \cdot r)$ time, where $\sum_{i=1}^{h} \frac{B_i}{cpe(i)} \cdot (1 + \frac{p_{max}}{2})$ is the maximum possible number of seed selection iterations. Thus, to make Algorithm 1 scalable, we need an alternative approach.

In the influence maximization literature, considerable effort has been devoted to developing more efficient and scalable algorithms [5, 7, 8, 18, 24]. Of these, the IRIE algorithm proposed by Jung et al. [18] is a state-of-the-art heuristic for influence maximization under the IC model and is orders of magnitude faster than MC simulations. We thus use a variant of Greedy, GREEDY-IRIE, where IRIE replaces MC simulations for spread estimation. It is one of the strong baselines we will compare our main algorithm with in §6. In this section, we instead propose a scalable algorithm with guaranteed approximation for influence spread.

Recently, Borgs et al. [5] proposed a quasi-linear time randomized algorithm based on the idea of sampling *"reverse-reachable"* (RR) sets in the graph. It was improved to a near-linear time randomized algorithm – *Two-phase Influence Maximization (TIM)* – by Tang et al. [24]. Cohen et al. [8] proposed a sketch-based design for fast computation of influence spread, achieving efficiency and effectiveness comparable to TIM. We choose to extend TIM as it is the current state-of-the-art influence maximization algorithm and is more adapted to our needs.

In this section, we adapt the essential ideas from Greedy, RR-sets sampling, and the TIM algorithm to devise an algorithm for REGRET-MINIMIZATION, called Two-phase Iterative Regret Minimization (TIRM for short), that is much more efficient and scalable than Algorithm 1 with MC simulations. Our adaptation to TIM is non-trivial, since TIM relies on knowing the exact number of seeds required. In our framework, the number of seeds needed is driven by the budget and the current regret and so is dynamic. We first give the background on RR-sets sampling, review the TIM algorithm [24], and then describe our TIRM algorithm.

5.1 Reverse-Reachable Sets and TIM

RR-sets Sampling: Brief Review. We first review the definition of RR-sets, which is the backbone of both TIM and our proposed TIRM algorithm. Conceptually speaking, a random RR-set R from G is generated as follows. First, for every edge $(u, v) \in E$, remove it from G w.p. $1 - p_{u,v}$: this generates a possible world X. Second, pick a *target* node w uniformly at random from V. Then, R consists of the nodes that can reach w in X. This can be implemented efficiently by first choosing a target node $w \in V$ uniformly at random and performing a breadth-first search (BFS) starting from it. Initially, create an empty BFS-queue Q, and insert all of w's in-neighbors into Q. The following loop is executed until Q is empty: Dequeue a node u from Q and examine its *incoming* edges: for each edge (v, u) where $v \in N^{in}(u)$, we insert v into Q w.p. $p_{v,u}$. All nodes dequeued from Q thus form a RR-set.

The intuition behind RR-sets sampling is that, if we have sampled sufficiently many RR-sets, and a node u appears in a large number of RR sets, then u is likely to have high influence spread in the original graph and is a good candidate seed.

TIM: Brief Review. Given an input graph $G = (V, E)$ with influence probabilities and desired seed set size s, TIM, in its first phase, computes a lower bound on the optimal influence spread of any seed set of size s, i.e., $OPT_s := \max_{S \subseteq V, |S|=s} \sigma^{ic}(S)$. Here $\sigma^{ic}(S)$ refers to the spread w.r.t. classic IC model. TIM then uses this lower bound to estimate the number of random RR-sets that need to be generated, denoted θ. In its second phase, TIM simply samples θ RR-sets, denoted $\mathbf{R}$, and uses them to select s seeds, by solving the Max s-Cover problem: find s nodes, that between them, appear in the maximum number of sets in $\mathbf{R}$. This is solved using a well-known greedy procedure: start with an empty set and repeatedly add a node that appears in the maximum number of sets in $\mathbf{R}$ that are not yet "covered".

TIM provides a $(1 - 1/e - \epsilon)$-approximation to the optimal solution OPT_s with high probability. Also, its time complexity is $O((s + \ell)(|V| + |E|) \log |V| / \epsilon^2)$, while that of the greedy algorithm (for influence maximization) is $\Omega(k|V||E| \cdot \mathrm{poly}(\epsilon^{-1}))$.

Theoretical Guarantees of TIM. Consider any collection of random RR-sets, denoted $\mathbf{R}$. Given any seed set S, we define $F_{\mathbf{R}}(S)$

as the fraction of $\mathbf{R}$ covered by S, where S covers an RR-set iff it overlaps it. The following proposition says that for any S, $|V| \cdot F_{\mathbf{R}}(S)$ is an unbiased estimator of $\sigma^{ic}(S)$.

PROPOSITION 1 (COROLLARY 1, [24]). *Let $S \subseteq V$ be any set of nodes, and $\mathbf{R}$ be a collection of random RR sets. Then, $\sigma^{ic}(S) = \mathbb{E}[|V| \cdot F_{\mathbf{R}}(S)]$.*

The next proposition shows the accuracy of influence spread estimation and the approximation gurantee of TIM. Given any seed set size s and $\varepsilon > 0$, define $L(s, \varepsilon)$ to be:

$$L(s,\varepsilon) = (8+2\varepsilon)n \cdot \frac{\ell \log n + \log \binom{n}{s} + \log 2}{OPT_s \cdot \varepsilon^2}, \tag{5}$$

where $\ell > 0, \epsilon > 0$.

PROPOSITION 2 (LEMMA 3 & THEOREM 1, [24]). *Let θ be a number no less than $L(s, \varepsilon)$. Then for any seed set S with $|S| \leq s$, the following inequality holds w.p. at least $1 - n^{-\ell}/\binom{n}{s}$:*

$$\big||V| \cdot F_{\mathbf{R}}(S) - \sigma^{ic}(S)\big| < \frac{\varepsilon}{2} \cdot OPT_s. \tag{6}$$

Moreover, with this θ, TIM returns a $(1 - 1/e - \epsilon)$-approximation to OPT_s w.p. $1 - n^{-\ell}$.

This result intuitively says that as long as we sample enough RR-sets, i.e., $|\mathbf{R}| \geq \theta$, the absolute error of using $|V| \cdot F_{\mathbf{R}}(S)$ to estimate $\sigma^{ic}(S)$ is bounded by a fraction of OPT_s with high probability. Furthermore, this gives approximation guarantees for influence maximization. Next, we describe how to extend the ideas of RR-sets sampling and TIM for regret minimization.

5.2 Two-phase Iterative Regret Minimization

A straightforward application of TIM for solving REGRET-MINIMIZATION will not work. There are two critical challenges. First, TIM requires the number of seeds s as input, while the input of REGRET-MINIMIZATION is in the form of monetary budgets, and thus we do not know the precise number of seeds that should be allocated to each advertiser beforehand. Second, our influence propagation model has click-through probabilities (CTPs) of seeds, namely $\delta(u,i)$'s. This is not accounted for in the RR-sets sampling method: it implicitly assumes that each seed becomes active w.p. 1.

We first discuss how to adapt RR-sets sampling to incorporate CTPs. Then we deal with unknown seed set sizes.

RR-sets Sampling with Click-Through Probabilities. Recall that in our model, when a node u is chosen as a seed for advertiser a_i, it has a probability $\delta(u,i)$ to accept being seeded, i.e., to actually click on the ad.

For ease of exposition, in the rest of this subsubsection only, we assume that there is only one advertiser, and the CTP of each user u for this advertiser is simply $\delta(u) \in [0,1]$. The technique we discuss and our results readily extend to any number of advertisers. A naive way to incorporate CTPs is as follows. For all $u \in V$, w.p. $\delta(u)$, mark it "live", and w.p. $1 - \delta(u)$, mark it "blocked". After that, generate RR-sets as described in the previous subsection, with an additional check, whether a node is live. Precisely, unless both node v is live and the associated edge $e = (u,v)$ has a positive outcome (w.p. $p(e)$), u will not be added to the BFS queue in the RR-set generation process.

For clarity, call the random RR-sets generated with CTPs incorporated as above, *RR-Sets with CTPs* (*RRC-sets* for short). Let $\mathbf{Q}$ be a collection of RRC-sets. Similar to $F_{\mathbf{R}}(S)$, for any set S, we define $F_{\mathbf{Q}}(S)$ to be the fraction of $\mathbf{Q}$ that overlap with S. Let $\sigma^{icctp}(S)$ be the influence spread of a seed set S under the IC model with CTPs. We first establish a similar result to Proposition 1 which says that $|V|F_{\mathbf{Q}}(S)$ is an unbiased estimator of $\sigma^{icctp}(S)$.

LEMMA 2. *Given a graph $G = (V,E)$ with influence probabilities on edges, for any $S \subseteq V$, $\sigma^{icctp}(S) = \mathbb{E}[|V| \cdot F_{\mathbf{Q}}(S)]$.*

PROOF. We show the following equality holds:

$$\sigma^{icctp}(S)/|V| = \mathbb{E}[F_{\mathbf{Q}}(S)]. \tag{7}$$

The LHS of (7) equals to the probability that a node chosen uniformly at random can be activated by seed set S where a seed $u \in S$ may become live with CTP $\delta(u)$, while the RHS of (7) equals to the probability that S intersects with a random RRC-set. They both equal to the probability that a randomly chosen node is reachable by S in a possible world corresponding to the IC-CTP model. ☐

In principle, RRC-sets are those we should work with for the purpose of seed selection for REGRET-MINIMIZATION. However, note that by Equation (5) and Proposition 2, the number of samples required is inversely proportional to the value of the optimal solution OPT_s. However, in reality, click-through rates on ads are quite low, and thus OPT_s, taking CTPs into account, will decrease by at least two orders of magnitude (e.g., OPT_s with CTP 0.01 would become 100 times smaller than OPT_s with CTP 1). This in turn translates into at least two orders of magnitude more RRC-sets to be sampled, which ruins scalability.

An alternative way of incorporating CTPs is to pretend as though all CTPs were 1. We still generate RR-sets, and use the estimations given by RR-sets to compute revenue. More specifically, for any $S \subseteq V$ and any $u \in V \setminus S$, we compute the marginal gain of u w.r.t. S, namely $\sigma_C(S \cup \{u\}) - \sigma_C(S)$, by $\delta(u) \cdot |V| \cdot [F_{\mathbf{R}}(S \cup \{u\}) - F_{\mathbf{R}}(S)]$. This avoids sampling of numerous RRC-sets.

We can show that in expectation, computing marginal gain in IC-CTP model using RRC-sets is essentially equivalent to computing it under the IC model using RR-sets in the manner above.

THEOREM 5. *Consider any $u \in S$ and any $S \subseteq V$. Let $\delta(u)$ be the probability that u accepts to become a seed. Let $\mathbf{R}$ and $\mathbf{Q}$ be a collection of RR-sets and of RRC-sets, respectively. Then,*

$$\delta(u)(\mathbb{E}[F_{\mathbf{R}}(S \cup \{u\})] - \mathbb{E}[F_{\mathbf{R}}(S)]) = \mathbb{E}[F_{\mathbf{Q}}(S \cup \{u\})] - \mathbb{E}[F_{\mathbf{Q}}(S)].$$

This theorem (proof in [1]) shows even with CTPs, we can still use the usual RR-sets sampling process for estimating spread efficiently and accurately as long as we multiply marginal gains by CTPs. *This result carries over to the setting of multiple advertisers.*

Iterative Seed Set Size Estimation. As mentioned earlier, TIM needs the required number of seeds s as input, which is not available for the REGRET-MINIMIZATION problem. From the advertiser budgets, there is no obvious way to determine the number of seeds. This poses a challenge since the required number of RR-sets (θ) depends on s. To circumvent this difficulty, we propose a framework which first makes an initial guess at s, and then iteratively revises the estimated value, until no more seeds are needed, while concurrently selecting seeds and allocating them to advertisers.

For ease of exposition, let us first consider a single advertiser a_i. Let B_i be the budget of a_i and let s_i be the true number of seeds required to minimize the regret for a_i. We do not know s_i and estimate it in successive iterations as $\tilde{s}_i^t$. We start with an estimated value for s_i, denoted $\tilde{s}_i^1$, and use it to obtain a corresponding θ_i^1 (*cf.* Proposition 2). If $\theta_i^t > \theta_i^{t-1}$ (assuming $\theta_i^0 = 0$ for all i), we will need to sample an additional $(\theta_i^t - \theta_i^{t-1})$ RR-sets, and use all RR-sets sampled up to this iteration to select $(\tilde{s}_i^t - \tilde{s}_i^{t-1})$ additional seeds. After adding those seeds, if a_i's budget B_i is not yet reached, this means more seeds can be assigned to a_i. Thus, we will need

Algorithm 2: TIRM

Input : $G = (V, E)$; attention bounds $\kappa_u, \forall u \in V$; items $\vec{\gamma}_i$ with $cpe(i)$ & budget B_i, $i = 1, \ldots, h$; CTPs $\delta(u, i), \forall u \forall i$

Output: $S_1, \cdots, S_h$

1 **foreach** $j = 1, 2, \ldots, h$ **do**
2 $\quad S_j \leftarrow \emptyset; Q_j \leftarrow \emptyset;$ `// a priority queue`
3 $\quad s_j \leftarrow 1; \theta_j \leftarrow L(s_j, \varepsilon); \mathbf{R}_j \leftarrow \mathsf{Sample}(G, \gamma_j, \theta_j);$
4 **while** *true* **do**
5 $\quad$ **foreach** $j = 1, 2, \ldots, h$ **do**
6 $\quad\quad (v_j, cov_j(v_j)) \leftarrow \mathsf{SelectBestNode}(\mathbf{R}_j)$; `// Alg 3`
7 $\quad\quad F_{\mathbf{R}_j}(v_j) \leftarrow cov_j(v_j)/\theta_j;$
8 $\quad i \leftarrow \arg\max_{j=1}^{h} \mathcal{R}_j(S_j) - \mathcal{R}_j(S_j \cup \{v_j\})$
$\quad\quad\quad$ subject to: $\mathcal{R}_j(S_j \cup \{v_j\}) < \mathcal{R}_j(S_j);$
$\quad$ `//(user, ad) pair with max drop in regret.`
9 $\quad$ **if** $i \neq$ **NULL then**
10 $\quad\quad S_i \leftarrow S_i \cup \{v_i\};$
11 $\quad\quad Q_i.\mathsf{insert}(v_i, cov_i(v_i));$
12 $\quad\quad \mathbf{R}_i \leftarrow \mathbf{R}_i \setminus \{R \mid v_i \in R \ \wedge \ R \in \mathbf{R}_i\};$
13 $\quad$ `//remove RR-sets that are covered;`
14 $\quad$ **else return** ;
15 $\quad$ **if** $|S_i| = s_i$ **then**
16 $\quad\quad s_i \leftarrow s_i + \lfloor \mathcal{R}_i(S_i)/(cpe(i) \cdot n \cdot \delta(v_i, i) \cdot F_{\mathbf{R}_i}(v_i)) \rfloor;$
17 $\quad\quad \theta_i \leftarrow \max\{L(s_i, \varepsilon), \theta_i\};$
18 $\quad\quad \mathbf{R}_i \leftarrow \mathbf{R}_i \cup \mathsf{Sample}(G, \gamma_i, \max\{0, L(s_i, \varepsilon) - \theta_i)\};$
19 $\quad\quad \Pi_i(S_i) \leftarrow \mathsf{UpdateEstimates}(\mathbf{R}_i, \theta_i, S_i, Q_i);$
$\quad\quad$ `//revise estimates to reflect newly added RR-sets;`
20 $\quad\quad \mathcal{R}_i(S_i) \leftarrow |B_i - \Pi_i(S_i)|;$

Algorithm 3: SelectBestNode($\mathbf{R}_j$)

Output: $(u, cov_j(u))$

1 $u \leftarrow \arg\max_{v \in V} |\{R \mid v \in R \ \wedge \ R \in \mathbf{R}_j\}|$
$\quad\quad$ subject to: $|\{S_l | v \in S_l\}| < \kappa_v;$
2 $cov_j(u) \leftarrow |\{R \mid u \in R \ \wedge \ R \in \mathbf{R}_j\}|;$ `//find best seed for ad` a_j `as well as its coverage.`

Algorithm 4: UpdateEstimates($\mathbf{R}_i, \theta_i, S_i, Q_i$)

Output: $\Pi_i(S_i)$

1 $\Pi_i(S_i) \leftarrow 0$;
2 **for** $j = 0, \ldots, |S_i| - 1$ **do**
3 $\quad (v, cov(v)) \leftarrow Q_i[j]$;
4 $\quad cov'(v) \leftarrow |\{R \mid v \in R, R \in \mathbf{R}_i\}|;$
5 $\quad Q_i.\mathsf{insert}(v, cov(v) + cov'(v));$
6 $\quad \Pi_i(S_i) \leftarrow$
$\quad \Pi_i(S_i) + cpe(i) \cdot n \cdot \delta(v, i) \cdot ((cov(v) + cov'(v))/\theta_i);$
$\quad$ `//update coverage of existing seeds w.r.t. new RR-sets added to collection.`

another iteration and we further revise our estimation of s_i. The new value, $\tilde{s}_i^{t+1}$, is obtained by adding to $\tilde{s}_i^t$ the floor function of the ratio between the current regret $\mathcal{R}_i(S_i)$ and the marginal revenue contributed by the $\tilde{s}_i^t$-th seed (i.e., the latest seed). This ensures we do not overestimate, thanks to submodularity, as future seeds have diminishing marginal gains.

Algorithm 2 outlines TIRM, which integrates the iterated seed set size estimation technique above, suitably adapted to multi-advertiser setting, along with the RR-set based coverage estimation idea of TIM, and uses Theorem 5 to deal with CTPs. Notice that the core logic of the algorithm is still based on greedy seed selection as outlined in Algorithm 1. Algorithm TIRM works as follows. For every advertiser a_i, we initially set its seed budget s_i to be 1 (a conservative, but safe estimate), and find the first seed using random RR-sets generated accordingly (line 3). In the main loop, we follow the greedy selection logic of Algorithm 1. That is, every time, we identify the valid user-advertiser pair (u, a_i) that gives the *largest decrease in total regret* and allocate u to S_i (lines 6 to 12), paying attention to the attention bound of u (line 1 of Algorithm 3). If $|S_i|$ reaches the current estimate of s_i after we add u, then we increase s_i by $\lfloor \mathcal{R}_i(S_i)/(cpe(i) \cdot n \cdot F_{\mathbf{R}_i}(u)) \rfloor$ (lines 15 to 20), as described above, as long as the regret continues to decrease. Note that after adding additional RR-sets, we should update the spread estimation of current seeds w.r.t. the new collection of RR-sets (line 19). This ensures that future marginal gain computations and selections are accurate. This is effectively a *lower bound* on the number of additional seeds needed, as subsequent seeds will not have marginal gain higher than that of u due to submodularity. As in Algorithm 1, TIRM terminates when all advertisers have saturated, i.e., no additional seed can bring down the regret. Note that in Algorithm 4, we update the estimated revenue of existing seeds w.r.t. the additional RR-sets sampled, to keep them accurate.

Estimation Accuracy of TIRM. At its core, TIRM, like TIM, estimates the spread of chosen seed sets, even though its objective is to minimize regret w.r.t. a monetary budget. Next, we show that the influence spread of seeds estimated by TIRM enjoys bounded error guarantees similar to those chosen by TIM (see Proposition 2).

THEOREM 6. *At any iteration t of iterative seed set size estimation in Algorithm* TIRM, *for any set S_i of at most $s = \sum_{j=1}^{t} s^j$ nodes, $|n \cdot F_{\mathbf{R}^t}(S_i) - \sigma_i(S_i)| < \frac{\varepsilon}{2} \cdot OPT_s$ holds with probability at least $1 - n^{-\ell}/\binom{n}{s}$, where $\sigma_i(S)$ is the expected spread of seed set S_i for ad i.*

PROOF. When $t = 1$, our claim follows from Proposition 2. When $t > 1$, by definition of our iterative sampling, the number of RR-sets, $|\mathbf{R}^t| = \max_{j=1,\ldots,t} L_j$, where $L_j = L(\sum_{a=1}^{j} s^a, \varepsilon)$. This means at any iteration t, the number of RR-sets is always sufficient for Eq. (6) to hold. Hence, for the set S_i containing seeds accumulated up to iteration t, our claim on the absolute error in the estimated spread of S_i holds, by Proposition 2. □

Time Complexity of TIRM. For each ad i, let τ_i be the number of iterations. The total number of seeds TIRM generates for i is thus $\sum_{t=1}^{\tau_i} \lfloor \mathcal{R}_i^t / MG_i^t \rfloor + 1$, where $\mathcal{R}_i^t$ and MG_i^t are used to compute the incremental update for s_i in line 16 in Algorithm 2. Clearly, this sum is upper-bounded by $\hat{s}_i := B_i(1 + p_{max}/2)/\min_{t=1}^{\tau_i} MG_i^t$, where B_i is the budget of ad i. Therefore, adapting the time complexity of TIM [24], the time complexity if TIRM is $\sum_{i=1}^{h}((\hat{s}_i + \ell) \cdot (|V| + |E|) \cdot \log |V| \cdot \varepsilon^{-2})$.

6. EXPERIMENTS

We conduct an empirical evaluation of the proposed algorithms. The goal is manifold. First, we would like to evaluate the quality of the algorithms as measured by the regret achieved, the number of seeds they used to achieve a certain level of budget-regret, and the extent to which the attention bound (κ) and the penalty factor (λ) affect their performance. Second, we evaluate the efficiency and scalability of the algorithms w.r.t. advertiser budgets, which indirectly control the number of seeds required, and w.r.t. the number of advertisers. We measure both running time and memory usage.

Datasets. Our experiments are based on four real-world social networks, whose basic statistics are summarized in Table 1. Of the four datasets, we use FLIXSTER and EPINIONS for our quality experiments and DBLP and LIVEJOURNAL for scalability experiments. FLIXSTER is from a social movie-rating site (`http://www.flixster.com/`). The dataset records movie ratings

	FLIXSTER	EPINIONS	DBLP	LIVEJOURNAL
#nodes	30K	76K	317K	4.8M
#edges	425K	509K	1.05M	69M
type	directed	directed	undirected	directed

Table 1: Statistics of network datasets.

	Budgets			CPEs		
Dataset	mean	max	min	mean	max	min
FLIXSTER	375	200	600	5.5	5	6
EPINIONS	215	100	350	4.35	2.5	6

Table 2: Advertiser budgets and cost-per-engagement values

from users along with their timestamps. We use the topic-aware influence probabilities and the item-specific topic distributions provided by the authors of [3], who learned the probabilities using maximum likelihood estimation for the TIC model with $K = 10$ latent topics. In our quality experiments, we set the number of advertisers h to be 10, and used 10 of the learnt topic distributions from Flixster dataset, where for each ad i , its topic distribution $\vec{\gamma_i}$ has mass 0.91 in the i-th topic, and 0.01 in all others. CTPs are sampled uniformly at random from the interval $[0.01, 0.03]$ for all user-ad pairs, in keeping with real-life CTPs (see §1).

EPINIONS is a who-trusts-whom network taken from a consumer review website (`http://www.epinions.com/`). For Epinions, we similarly set $h = 10$ and use $K = 10$ latent topics. For each ad i, we use synthetic topic distributions $\vec{\gamma_i}$, by borrowing the ones used in FLIXSTER. For all edges and topics, the topic-aware influence probabilities are sampled from an exponential distribution with mean 30, via the inverse transform technique [11] on the values sampled randomly from uniform distribution $\mathcal{U}(0, 1)$.

For scalability experiments, we adopt two large networks DBLP and LIVEJOURNAL (both are available at `http://snap.stanford.edu/`). DBLP is a co-authorship graph (undirected) where nodes represent authors and there is an edge between two nodes if they have co-authored a paper indexed by DBLP. We direct all edges in both directions. LIVEJOURNAL is an online blogging site where users can declare which other users are their friends.

In all datasets, advertiser budgets and CPEs are chosen in such a way that the total number of seeds required for all ads to meet their budgets is less than n. This ensures no ads are assigned empty seed sets. For lack of space, we do not enumerate all the numbers, but rather give a statistical summary in Table 2. Notice that since the CTPs are in the 1-3% range, the effective number of targeted nodes is correspondingly larger. We defer the numbers for DBLP and LIVEJOURNAL to §6.2.

All experiments were run on a 64-bit RedHat Linux server with Intel Xeon 2.40GHz CPU and 65GB memory. Our largest configuration is LIVEJOURNAL with 20 ads, which effectively has $69M \cdot 20 = 1.4B$ edges; this is comparable with [24], whose largest dataset has 1.5B edges (Twitter).

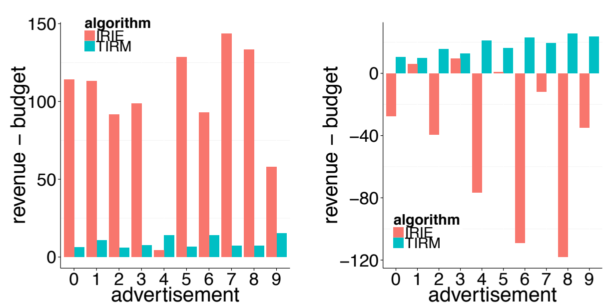
Algorithms. We test and compare the following four algorithms.

- MYOPIC: A baseline that assigns every user $u \in V$ in total κ_u most relevant ads i, i.e., those for which u has the highest expected revenue, not considering any network effect, i.e., $\delta(u, i) \cdot cpe(i)$. It is called "myopic" as it solely focuses on CTPs and CPEs and effectively ignores virality and budgets. Allocation $\mathcal{A}$ in Fig. 1 follows this baseline.
- MYOPIC+: This is an enhanced version of MYOPIC which takes budgets, but not virality, into account. For each ad, it first ranks users w.r.t. CTPs and then selects seeds using this order until budget is exhausted. User attention bounds are taken into account by going through the ads round-robin and advancing to the next seed if the current node u is already assigned to κ_u ads.
- GREEDY-IRIE: An instantiation of Algorithm 1, with the IRIE heuristic [18] used for influence spread estimation and seed selection. IRIE has a damping factor α for accurately estimating influence spread in its framework. Jung et al. [18] report that $\alpha = 0.7$ performs best on the datasets they tested. We did extensive testing on our datasets and found that $\alpha = 0.8$ gave the best spread estimation, and thus used 0.8 in all quality experiments.
- TIRM: Algorithm 2. We set ε to be 0.1 for quality experiments on FLIXSTER and EPINIONS, and 0.2 for scalability experiments on DBLP and LIVEJOURNAL (following [24]).

For all algorithms, we evaluate the final regret of their output seed sets using Monte Carlo simulations (10K runs) for neutral, fair, and accurate comparisons.

6.1 Results of Quality Experiments

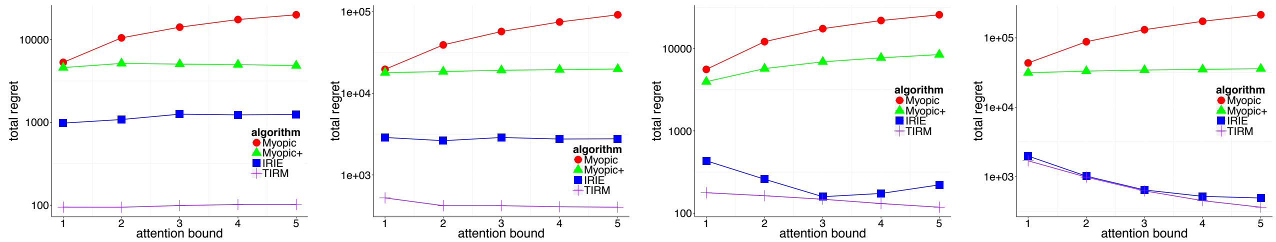
Overall regret. First, we compare overall regret (as defined in Eq. (4)) against attention bound κ_u, varied from 1 to 5, with two choices 0 and 0.5 for λ. Fig. 3 shows that the overall regret (in log-scale) achieved by TIRM and GREEDY-IRIE are significantly lower than that of MYOPIC and MYOPIC+. For example, on FLIXSTER with $\lambda = 0$ and $\kappa_u = 1$, overall regrets of TIRM, GREEDY-IRIE, MYOPIC, and MYOPIC+, expressed relative to the total budget, are 2.5%, 26.1%, 122%, 141%, respectively. On EPINIONS with the same setting, the corresponding regrets are 6.5%, 15.9%, 145%, and 205%. MYOPIC, and MYOPIC+ typically always overshoot the budgets as they are not vitality-aware when choosing seeds. Notice that even though MYOPIC+ is budget conscious, it still ends up overshooting the budget as a result of not factoring in virality in seed allocation. In almost all cases, overall regret by TIRM goes down as κ_u increases. The trend for MYOPIC and MYOPIC+ is the opposite, caused by their larger overshooting with larger κ_u. This is because they will select more seeds as κ_u goes up, which causes higher revenue (hence regret) due to more virality.



(a) FLIXSTER (b) EPINIONS

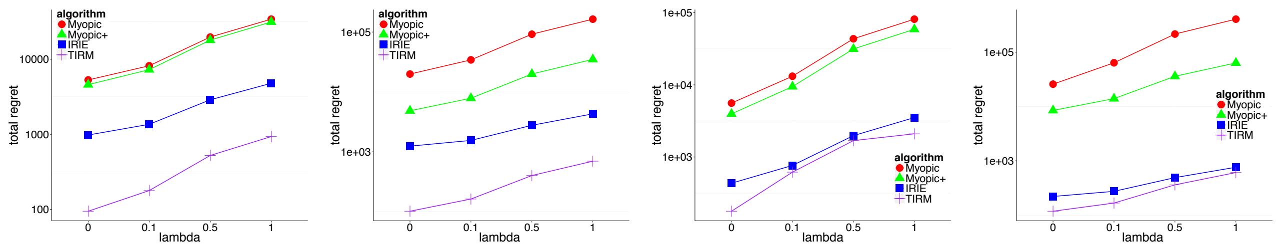
Figure 5: Distribution of individual regrets ($\lambda = 0$, $\kappa_u = 5$).

We also vary λ to be 0, 0.1, 0.5, and 1 and show the overall regrets under those values in Fig. 4 (in log-scale), with two choices 1 and 5 for κ_u. As expected, in all test cases as λ increases, the overall regret also goes up. The hierarchy of algorithms (in terms of performance) remains the same as in Fig. 3, with TIRM being the consistent winner. Note that even when λ is as high as 1, TIRM still wins and performs well. This suggests that the λ-assumption ($\lambda \leq \delta(u, i) \cdot cpe(i), \forall$ user u and ad i) in Theorem 2 is conservative as TIRM can still achieve relatively low regret even with large λ values.



(a) FLIXSTER ($\lambda = 0$) (b) FLIXSTER ($\lambda = 0.5$) (c) EPINIONS ($\lambda = 0$) (d) EPINIONS ($\lambda = 0.5$)

Figure 3: Total regret (log-scale) vs. attention bound κ_u



(a) FLIXSTER ($\kappa_u = 1$) (b) FLIXSTER ($\kappa_u = 5$) (c) EPINIONS ($\kappa_u = 1$) (d) EPINIONS ($\kappa_u = 5$)

Figure 4: Total regret (log-scale) vs. λ

Drilling down to individual regrets. Having compared overall regrets, we drill down into the budget-regrets (see §4) achieved for different individual ads by TIRM and GREEDY-IRIE. Fig. 5 shows the distribution of budget-regrets across advertisers for both algorithms. On FLIXSTER, both algorithms overshoot for all ads, but the distribution of TIRM-regrets is much more uniform than that of GREEDY-IRIE-regrets. E.g., for the fourth ad, GREEDY-IRIE even achieves a smaller regret than TIRM, but for all other ads, their GREEDY-IRIE-regret is at least 3.8 times as large as the TIRM-regret, showing a heavy skew. On EPINIONS, TIRM slightly overshoots for all advertisers as in the case of FLIXSTER, while GREEDY-IRIE falls short on 7 out of 10 ads and its budget-regrets are larger than TIRM for most advertisers. Note that MYOPIC and MYOPIC+ are not included here as Figs. 3 and 4 have clearly demonstrated that they have significantly higher overshooting[5].

Number of targeted users. We now look into the distinct number of nodes targeted at least once by each algorithm, as κ_u increases from 1 to 5. Intuitively, as κ_u decreases, each node becomes "less available", and thus we may need more distinct nodes to cover all budgets, causing this measure to go up. The stats in Table 3 confirm this intuition, in the case of TIRM, GREEDY-IRIE, and MYOPIC+. MYOPIC is an exception since it allocates an ad to every user (i.e., all $|V|$ nodes are targeted). Note that on EPINIONS, TIRMtargeted more nodes than GREEDY-IRIE. The reason is that GREEDY-IRIE tends to overestimate influence spread on EPINIONS, resulting in pre-mature termination of Greedy. When MC is used to estimate ground-truth spread, the revenue would fall short of budgets (see Fig. 5). The behavior of GREEDY-IRIE is completely the opposite on FLIXSTER, showing its lack of consistency as a pure heuristic.

6.2 Results of Scalability Experiments

We test the scalability of TIRM and GREEDY-IRIE on DBLP and LIVEJOURNAL. For simplicity, we set all CPEs and CTPs to 1 and λ to 0, and the values of these parameters do not affect running time

[5]Their regrets are all from overshooting the budget on account of ignoring virality effects.

FLIXSTER	$\kappa_u = 1$	2	3	4	5
TIRM	868	352	319	263	257
GREEDY-IRIE	3.7K	1.7K	1.5K	1237	1222
MYOPIC	29K	29K	29K	29K	29K
MYOPIC+	27K	13K	9.6K	7.5K	6.6K
EPINIONS	$\kappa_u = 1$	2	3	4	5
TIRM	4.4K	901	396	233	175
GREEDY-IRIE	3.1K	826	393	251	183
MYOPIC	76K	76K	76K	76K	76K
MYOPIC+	55K	28K	19K	15K	13K

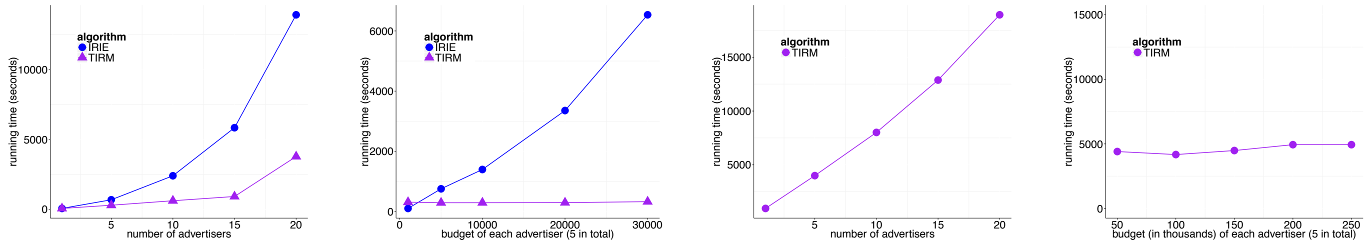
Table 3: Number of nodes targeted vs. attention bounds ($\lambda = 0$)

or memory usage. Influence probabilities on each edge $(u, v) \in E$ are computed using the Weighted-Cascade model [7]: $p^i_{u,v} = \frac{1}{|N^{in}(v)|}$ for all ads i. We set $\alpha = 0.7$ for GREEDY-IRIE and $\varepsilon = 0.2$ for TIRM, in accordance with the settings in [18, 24]. Attention bound $\kappa_u = 1$ for all users. We emphasize that our setting is fair and ideal for testing scalability as it simulates a fully competitive case: all advertisers compete for the same set of influential users (due to all ads having the same distribution over the topics) and the attention bound is at its lowest, which in turn will "stress-test" the algorithms by prolonging the seed selection process.

We test the running time of the algorithms in two dimensions: Fig. 6(a) & 6(c) vary h (number of ads) with per-advertiser budgets B_i fixed (5K for DBLP, 80K for LIVEJOURNAL), while Fig. 6(b) & 6(d) vary B_i when fixing $h = 5$. Note that GREEDY-IRIE results on LIVEJOURNAL (Fig. 6(c) & 6(d)) are excluded due to its huge running time, details to follow.

At the outset, notice that TIRM significantly outperforms GREEDY-IRIE in terms of running time. Furthermore, as shown in Fig. 6(a), the gap between TIRM and GREEDY-IRIE on DBLP becomes larger as h increases. For example, when $h = 1$, both algorithms finish in 60 secs, but when $h = 15$, TIRM is 6 times faster than GREEDY-IRIE.

On LIVEJOURNAL, TIRM scales almost linearly w.r.t. the number of advertisers, It took about 16 minutes with $h = 1$ (47 seeds chosen) and 5 hours with $h = 20$ (4649 seeds). GREEDY-IRIE



(a) DBLP (h) (b) DBLP (budgets) (c) LIVEJOURNAL (h) (d) LIVEJOURNAL (budgets)

Figure 6: Running time of TIRM and GREEDY-IRIE on DBLP and LIVEJOURNAL

DBLP	$h = 1$	5	10	15	20
TIRM	2.59	12.6	27.1	40.6	60.8
GREEDY-IRIE	0.16	0.30	0.48	0.54	0.84
LIVEJOURNAL	$h = 1$	5	10	15	20
TIRM	3.72	15.6	32.5	47.7	60.9

Table 4: Memory usage (GB)

took about 6 hours to complete for $h = 1$, and did not finish after 48 hours for $h \geq 5$. When budgets increase (Fig. 6(b)), GREEDY-IRIE's time will go up (super-linearly) due to more iterations of seed selections, but TIRM remains relatively stable (barring some minor fluctuations). On LIVEJOURNAL, TIRM took less than 75 minutes with $B_i = 50K$ (254 seeds). Note that once h is fixed, TIRM's running time depends heavily on the required number of random RR-sets (θ) for each advertiser rather than budgets, as seed selection is a linear-time operation for a given sample of RR-sets. Thus, the relatively stable trend on Fig. 6(b) & 6(d) is due to the subtle interplay among the variables to compute $L(s, \varepsilon)$ (Eq. 5); similar observations were made for TIM in [24].

Table 4 shows the memory usage of TIRM and GREEDY-IRIE. As TIRM relies on generating a large number of random RR-sets for accurate estimation of influence spread, we observe high memory consumption by this algorithm, similar to the TIM algorithm [24]. The usage steadily increases with h. The memory usage of GREEDY-IRIE is modest, as its computation requires merely the input graph and probabilities. However, GREEDY-IRIE is a heuristic with no guarantees, which is reflected in its relatively poor regret performance compared to TIRM. Furthermore, as seen earlier, TIRM scales significantly better than GREEDY-IRIE on all datasets.

7. CONCLUSIONS AND FUTURE WORK

In this work, we build a bridge between viral marketing and social advertising, by drawing on the viral marketing literature to study influence-aware ad allocation for social advertising, under real-world business model, paying attention to important practical factors like relevance, social proof, user attention bound, and advertiser budget. In particular, we study the problem of regret minimization from the host perspective, characterize its hardness and devise a simple scalable algorithm with quality guarantees w.r.t. the total budget. Through extensive experiments we demonstrate its superior performance over natural baselines.

Our work takes a first step toward enriching the framework of social advertising by integrating it with powerful ideas from viral marketing and making the latter more applicable to real online marketing problems. It opens up several interesting avenues for further research. Studying continuous-time propagation models, possibly with the network and/or influence probabilities not known beforehand (and to be learned), and possibly in presence of hard competition constraints, is a direction that offers a wealth of possibilities for future work.

8. REFERENCES

[1] C. Aslay et al. Viral Marketing Meets Social Advertising: Ad Allocation with Minimum Regret. *arXiv:1412.1462*, 2014.

[2] E. Bakshy et al. Social influence in social advertising: evidence from field experiments. In *EC*, pp. 146–161, 2012.

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

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] C. Borgs et al. Maximizing social influence in nearly optimal time. In *SODA*, pp. 946–957. SIAM, 2014.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[7] W. Chen et al. Scalable influence maximization for prevalent viral marketing in large-scale social networks. *KDD* 2010.

[8] E. Cohen et al. Sketch-based influence maximization and computation: Scaling up with guarantees. In *CIKM*, 2014.

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

[10] N. R. Devanur, B. Sivan, and Y. Azar. Asymptotically optimal algorithm for stochastic adwords. In *EC*, pp. 388–404, 2012.

[11] L. Devroye. Sample-based non-uniform random variate generation. In *18th conf. on winter simulation*, pp. 260–265. ACM, 1986.

[12] N. Du et al. Budgeted influence maximization for multiple products. *arXiv:1312.2164*, 2014.

[13] J. Feldman et al. Online stochastic packing applied to display ad allocation. In *ESA*, pp. 182–194, 2010.

[14] J. Feldman et al. Online ad assignment with free disposal. In *WINE*, pp. 374–385, 2009.

[15] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, 1979.

[16] G. Goel et al. Online budgeted matching in random input models with applications to adwords. In *SODA*, pp. 982–991, 2008.

[17] A. Goyal et al. A data-based approach to social influence maximization. *PVLDB*, 5(1):73–84, 2011.

[18] K. Jung, W. Heo, and W. Chen. IRIE: scalable and robust influence maximization in social networks. In *ICDM*, pp. 918–923, 2012.

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

[20] S. Lin, Q. Hu, F. Wang, and P. S. Yu. Steering information diffusion dynamically against user attention limitation. In *ICDM*, 2014.

[21] A. Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[22] V. S. Mirrokni et al. Simultaneous approximations for adversarial and stochastic online budgeted allocation. *SODA* 2012.

[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - i. *Mathematical Programming*, 1978.

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

[25] C. Tucker. Social advertising. Working paper available at SSRN: http://ssrn.com/abstract=1975897, 2012.