Approximation Algorithms for Coordinating Ad Campaigns on Social Networks

Kartik Lakhotia David Kempe
University of Southern California
klakhoti@usc.edu, david.m.kempe@gmail.com

August 27, 2019

Abstract

We study a natural model of coordinated social ad campaigns over a social network, based on models of Datta et al. and Aslay et al. Multiple advertisers are willing to pay the host — up to a known budget — per user exposure, whether that exposure is sponsored or organic (i.e., shared by a friend). Campaigns are seeded with sponsored ads to some users, but no network user must be exposed to too many sponsored ads. As a result, while ad campaigns proceed independently over the network, they need to be carefully coordinated with respect to their seed sets.

We study the objective of maximizing the network's total ad revenue. Our main result is to show that under a broad class of social influence models, the problem can be reduced to maximizing a submodular function subject to two matroid constraints; it can therefore be approximated within a factor essentially $\frac{1}{2}$ in polynomial time. When there is no bound on the individual seed set sizes of advertisers, the constraints correspond only to a single matroid, and the guarantee can be improved to $1 - 1/e$; in that case, a factor $\frac{1}{2}$ is achieved by a practical greedy algorithm. The $1 - 1/e$ approximation algorithm for the matroid-constrained problem is far from practical; however, we show that specifically under the Independent Cascade model, LP rounding and Reverse Reachability techniques can be combined to obtain a $1-1/e$ approximation algorithm which scales to several tens of thousands of nodes.

Our theoretical results are complemented by experiments evaluating the extent to which the coordination of multiple ad campaigns inhibits the revenue obtained from each individual campaign, as a function of the similarity of the influence networks and the strength of ties in the network. Our experiments suggest that as networks for different advertisers become less similar, the harmful effect of competition decreases. With respect to tie strengths, we show that the most harm is done in an intermediate range.

1 Introduction

Advertising is the most important and successful business model among social network sites. It is widely believed that the adoption of products has a significant social component to it, wherein people recommend products to each other, seek each other's opinion on products, or simply see others use a product. Under advertising campaigns on social networks, users are shown some ads embedded in their newsfeed from friends; the advertisers' hope is that the users will voluntarily share the ads, or adopt the product and be observed by their friends using it.

While models and algorithms for viral marketing have a long history of study (e.g., [12, 17, 16, 19, 21], and [8] for a survey), the focus in all of these works (and a much larger body of literature) has been on campaigns to maximize the reach of a single product. When multiple products have been considered, it has typically been in the context of competition, wherein each member of the network adopts at most one product (e.g., [13, 18, 20]); though see [26] for a model wherein products may exhibit complementarities.

Another very important way in which multiple products' advertising campaigns interact was articulated in two works by Datta, Majumder, and Shrivastava [11] and by Aslay, Lu, Bonchi, Goyal, and Lakshmanan [3]. Both groups of authors observe that marketing campaigns interact in that they involve displaying ads to the same set of users in a social network, and users should not be shown too many ads, lest they feel too inundated with ads and leave the social network [25].[1]

More precisely, both sets of authors assume (explicitly or implicitly) that product exposure comes in two forms: sponsored or user-shared. When a user's friend willingly shares an ad, or the user observes a friend using a product, this is perceived as genuine content. On the other hand, a sponsored ad shown by the network itself is viewed as advertising. Both Datta et al. [11] and Aslay et al. [3] thus impose a constraint on the number of *sponsored* ads that a user can be shown: the number of sponsored ads shown to node v must not exceed a given bound r_v. While only sponsored ads are deemed "annoying," advertisers do profit equally from sponsored or user-shared exposure. Imposing tight constraints on the number of sponsored ads across ad campaigns, and solving the corresponding optimization problem, should thus reduce the exposure of users to annoying sponsored ads. If the coordinated optimization problem is solved well, then this reduction might be achieved without much reduction in network profit. [2]

Both Datta et al. and Aslay et al. study the network's (here also called *host*) goal of maximizing its own payoff (also called *revenue*), by carefully coordinating the ad campaigns of multiple advertisers. In both models, the revenue that can be obtained from an advertiser j is constrained by the advertiser's "budget." In the model of Datta et al., the advertiser pays the host for each user exposure (sponsored or user-shared); however, the number of sponsored ads for advertiser j is bounded by some number $K^{(j)}$. In the standard language of influence maximization (precise definitions are given in Section 2), this translates to constraints on the *seed set sizes* of each advertiser. In contrast, Aslay et al. assume that the advertiser's budget constrains the total payment for exposures; if an advertiser with budget $B^{(j)}$ receives N exposures and is willing to pay $\gamma^{(j)}$ per exposure, the network's revenue is $\min(B^{(j)}, \gamma^{(j)} \cdot N)$.

The advertisers' hard budget constraints in the model of Aslay et al. raise an interesting question: what happens if $\gamma^{(j)} \cdot N > B^{(j)}$, i.e., an advertising cascade reaches *more* network users than the advertiser is willing to pay for? Aslay et al. [3] posit that the network views this negatively, and incurs a *regret* over giving the advertiser free exposure.[3] In their model, the network obtains regret for *undershooting* the budget as well, this time over lost potential revenue. Their goal is then to minimize the total regret. Because the regret is 0 only when the network exactly hits the advertisers'

[1] According to the 2012 Digital Advertising Attitudes Report [32], users may even exhibit negative responses towards products if they receive too many advertisements.

[2] Whether users should be shown ads at all on social networks is an ethical and normative question beyond the scope of our work. We take no position on whether social networking platforms should be considered public utilities, financed by subscription fees, or by ad revenue. Our goal here is to limit users' exposure to sponsored ads while still providing high coverage to individual advertisers in the currently practiced model.

[3] A possible justification is that if this happens repeatedly, advertisers may learn to lower their declared budget and pay less.

budgets, a straightforward reduction from PARTITION style problems shows not only NP-hardness, but also hardness of any multiplicative approximation.

In this article, we depart from the regret objective of Aslay et al. In particular, we do not believe that a network incurs regret from accidentally giving advertisers free exposure. Such regret may well be a psychological reality for humans, but it appears less applicable at the level of a major company. In fact, an equally strong case could be made that giving advertisers free exposure is beneficial, in that it makes the advertisers more likely to run future ad campaigns on the site, which in turn may lead to an increase, not decrease, in future revenue. Once a regret for overshooting the advertiser's budget is not a concern any more, a much more natural objective is to maximize the network's total revenue. (A precise definition of the model is given in Section 2.)

Our main result — in Section 3 — is a general treatment of optimization in this model, subsuming both versions of budgets. We consider very general influence models (potentially different for each advertiser), and only require that the local influence function at every node v for every individual advertiser j is monotone submodular[4]. Under these assumptions, we show that the network's revenue can be approximated to within constant factors. More specifically, when there are no constraints on individual advertisers' seed set sizes (only budget constraints limiting the total number of exposures the advertiser is willing to pay for), then there is a polynomial-time $(1 - 1/\mathrm{e})$-approximation algorithm, and a simple greedy algorithm achieves a $\frac{1}{2}$-approximation. When in addition, there are constraints on the seed set sizes of individual advertisers, we obtain a $\frac{1}{2+\epsilon}$ approximation, for every $\epsilon > 0$.

We show these results by expressing the optimization problem as a monotone submodular maximization problem over suitably chosen matroid domains, much in the style of Vondrák et al.'s result on welfare-maximizing partitions [6, 33]. When there are no constraints on individual seed set sizes, this allows us to bring to bear on the problem Vondrák's polynomial-time $(1 - 1/\mathrm{e})$-approximation algorithm for maximizing a submodular function subject to a matroid constraint [6, 33] (see also subsequent work [7, 28, 34]), and the much simpler $\frac{1}{2}$-approximation algorithm due to Fischer et al. [14] (see also [24]). When there are additional constraints on individual seed set sizes, the feasible ad campaigns can be expressed as the intersection of *two* matroids, and the algorithm of Lee et al. [23] gives a $\frac{1}{2+\epsilon}$ approximation.

While we do not develop new algorithms or heuristics in this part of our work, we consider it an important contribution to explicitly reduce from advertiser competition constraints to the intersections of matroids. Such a characterization was missing from prior work, resulting in heuristics with weaker guarantees and rederiving known arguments. Our work provides a clean optimization framework that allows the incorporation of different constraints in the future, and the application of known powerful optimization techniques and guarantees.

While Vondrák's beautiful $(1 - 1/\mathrm{e})$-approximation algorithm for maximizing a submodular function under a matroid constraint [33] runs in polynomial time, it is not practical.[5] With this concern in mind, in Section 4, we develop a $(1 - 1/\mathrm{e})$-approximation algorithm which scales to moderately sized networks comprising several tens of thousands of nodes, that are often encountered in area-aware advertising [2, 1]. Another natural domain with small networks is targeted health and social intervention programs [36], where network sizes are often only in the hundreds or thousands; it is very plausible that a principal may want to coordinate multiple interventions, such as safe

[4] Recall that a set function f is *submodular* iff $f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$ whenever $S \subseteq T$.

[5] The running time is roughly $O(n^8)$, with large constants. To the best of our knowledge, the full algorithm has never been implemented, but we would be surprised if it scaled to more than 10 nodes.

drug usage, safe sex practices, and others. Unlike our main results, this algorithm is not fully general, providing guarantees only for the Independent Cascade (IC) Model, and satisfying the budget constraint only in expectation, rather than for each solution.

The reason that the algorithm only works for the IC model is that it is based on randomly sampling Reverse Reachable (RR) sets [4, 30, 31]. These RR sets are then used to define and solve a generalized Maximum Coverage LP, which is then rounded using techniques of Gandhi et al. [15]. An additional important benefit of the algorithm is that the fractional LP which it rounds provides an upper bound on the performance of the *optimum* solution. This upper bound is useful in our experimental evaluation, not only of the LP rounding algorithm, but of other algorithms as well.

In Section 5, we describe experiments to evaluate the effects of competition and tie strengths on coordinated advertising campaigns. A comparison of algorithms shows that while the greedy algorithm has a worse worst-case approximation guarantee than LP rounding, it performs (marginally) better in experiments; both algorithms beat several natural heuristics, and get within 85% of the LP-based upper bound on the optimum; this is significantly better than their worst-case guarantees. We also develop a parallel version of the Greedy algorithm that accelerates overall processing by a factor of 12 on a 36-core machine. This allows us to scale to graphs with millions of nodes and dozens of advertisers.

Since approximation algorithms for "standard" Influence Maximization have been extensively studied in the literature, our main interest in the experimental evaluation is the effect of competition between products, and the interaction between competition and tie strengths in the network. Our experiments show that the more similar the advertisers' influence networks, the more the host's payoff decreases compared to the sum of what could be extracted from the advertisers in isolation. This effect is exacerbated when the network has a small number of highly influential nodes, since each advertiser will only be able to target few of these nodes. When nodes' influence is more even across the network, the host loses less revenue, because there are enough different "parts" of the network to extract revenue from all advertisers. When ties are very weak or very strong, the effect of competition is again attenuated, while for intermediate tie strengths, competition can lead to a significant revenue loss compared to treating each advertiser separately.

2 Problem Statement

There are m advertisers $j = 1, \ldots, m$ aiming to advertise on a network of n nodes/individuals $V = \{1, \ldots, n\}$. We will use u and v (and their variants) to denote nodes, and j (and variants) exclusively for advertisers.

Product information (or ads or influence) propagates through the social network according to the *general threshold model* [21, 27], defined as follows: For each node v and advertiser j, there is a monotone and submodular *local influence function* $f_v^{(j)} : V \to [0, 1]$ with $f_v^{(j)}(\emptyset) = 0$. Each node v independently chooses thresholds $\theta_v^{(j)} \in [0, 1]$ uniformly at random for each product j. Let $A_t^{(j)}$ be the set of nodes that have shared ad j by round t. (We call such nodes *active for ad* j.) Node v becomes active for ad j in round $t + 1$ iff $f_v^{(j)}(A_t^{(j)}) \geq \theta_v^{(j)}$. The (random) process is seeded with *seed sets* $A_0^{(j)}$ (subject to constraints discussed below). It quiesces when no new activations occur in a single round. At that point, for each ad j, a final (random) set $\hat{A}^{(j)}$ has become active. The general threshold model subsumes most standard models of influence spread, including the Independent Cascade and Linear Threshold models. Notice that (1) the diffusions for different

j proceed independently, except for joint constraints on the seed sets (discussed below), and (2) different ads can in principle follow different diffusion models.

Each advertiser j has a non-negative and monotone value function $w^{(j)}$ giving the payment that j will make to the network site as a function of the active nodes in the end. In full generality, this function could depend on the (random) *set* $\hat{A}^{(j)}$ of nodes that are active for j in the end; in this case, we will assume that $w^{(j)}$ is also submodular. In most cases, however, all nodes in the network will have the same value to advertiser j, in which case $w^{(j)}$ will only depend on the cardinality $|\hat{A}^{(j)}|$. In this case, we will assume that $w^{(j)}$ is (weakly) concave; notice that this is a special case of the previous one, because viewed as a function of $\hat{A}^{(j)}$ (rather than $|\hat{A}^{(j)}|$), the function $w^{(j)}$ is submodular. For notational convenience, and in keeping with much of the prior literature, we write $\sigma^{(j)}(A_0^{(j)}) = |\hat{A}^{(j)}|$ for the (random) number of nodes that are active in the end when the process starts with the node set $A_0^{(j)}$.

A particularly natural case — closest to the definition of Aslay et al. [3] — is when $w^{(j)}(x) = \min(B^{(j)}, \gamma^{(j)} \cdot x)$. Here, $B^{(j)}$ is the advertiser's budget, while $\gamma^{(j)}$ is the amount he is willing to pay per exposure.

The network's goal is to choose seed sets $A_0^{(j)}$ for all advertisers j, subject to additional constraints discussed below, so as to maximize one of the following two global revenue functions:

$$W(A_0^{(1)}, A_0^{(2)}, \ldots, A_0^{(m)}) = \sum_{j=1}^{m} \mathbb{E}\left[w^{(j)}(\sigma^{(j)}(A_0^{(j)}))\right], \tag{1}$$

$$\bar{W}(A_0^{(1)}, A_0^{(2)}, \ldots, A_0^{(m)}) = \sum_{j=1}^{m} w^{(j)}(\mathbb{E}\left[\sigma^{(j)}(A_0^{(j)})\right]). \tag{2}$$

Notice the subtle difference between the two definitions: W captures the expected revenue from advertisers who are charged for each campaign individually, according to their functions $w^{(j)}$. The objective W extends straightforwardly to the case where $w^{(j)}$ depends not only on the cardinality of $\hat{A}^{(j)}$, but on the specific set. In contrast, $\bar{W}$ corresponds to the case in which advertisers are charged according to the expected exposure. An essentially equivalent way of expressing the same objective (up to random noise) is that the revenue is based on the *average* exposure for advertiser j over a large number of campaigns.

Specifically in the context of the linear revenue function with a cap of the budget $B^{(j)}$, this means that under W, advertiser j is *never* charged more than $B^{(j)}$ for any individual campaign, while under $\bar{W}$, advertiser j's budget is not exceeded on average over multiple campaigns.

The main constraint, common to both the models of Aslay et al. and Datta et al., is that each node can be exposed to only a limited number of sponsored ads. We interpret this as saying that for node v, there is an upper bound r_v on the number of seed sets it can be contained in, i.e., $|\{j \mid v \in A_0^{(j)}\}| \leq r_v$.[6]

In addition to the node exposure constraints r_v, following the model of Datta et al., we also allow for constraints on the seed set sizes of each advertiser: for each advertiser j, the number of seed nodes $|A_0^{(j)}|$ cannot exceed $K^{(j)}$. In addition, we can constrain the total number of sponsored ads (more tightly): $\sum_j |A_0^{(j)}| \leq K$. In summary, our target optimization problem is the following:

[6] This model in fact subsumes one in which each sponsored ad exposure only activates v with some probability $p_v^{(j)}$: add a new *ad node* v' with activation function $f_{v'}^{(j)} \equiv 0$, and set $\hat{f}_v^{(j)}(S \cup \{v'\}) = 1 - (1 - p_v^{(j)})(1 - f_v^{(j)}(S))$. Then set $r_v = 0$, so only the new "ad nodes" can be targeted.

Definition 1 (Multi-Product Influence Maximization) *For each advertiser j, choose a seed set $A_0^{(j)}$ with $|A_0^{(j)}| \leq K^{(j)}$, such that $|\{j \mid v \in A_0^{(j)}\}| \leq r_v$.*

Subject to these constraints, maximize $W(A_0^{(1)}, A_0^{(2)}, \ldots, A_0^{(m)})$ or $\bar{W}(A_0^{(1)}, A_0^{(2)}, \ldots, A_0^{(m)})$.

The Multi-Product Influence Maximization problem is clearly NP-hard, as it subsumes the standard influence maximization problem.

3 A General Result

We begin with a very general treatment. Under the assumptions we make (the $w^{(j)}$ are submodular functions of $\hat{A}^{(j)}$, or concave functions of $|\hat{A}^{(j)}|$), the overall objective functions W and $\bar{W}$ are both submodular. In the case of W, this is because a non-negative linear combination of submodular functions (in particular: a convex combination of submodular functions) is submodular. In the case of $\bar{W}$, the reason is that $|\hat{A}^{(j)}|$ is a monotone submodular function of $\hat{A}^{(j)}$, and applying a monotone concave function preserves submodularity. In both cases, we can therefore apply the result of Mossel and Roch [27], which guarantees submodularity of the objective for each advertiser, as a function of the seed set.

To deal with the constraints on node exposures and seed set sizes, we use a technique discussed in [6, 33] in the context of finding welfare-maximizing assignments of items to individuals. We create a disjoint union of separate networks (each with the original n nodes) for each advertiser $j = 1, \ldots, m$, and then consider joint constraints on seed sets that can be selected across all of the separate networks. The overall new network $\hat{\mathcal{G}}$ has nm nodes, one node $u_{v,j}$ for each combination of an original node v and advertiser j. Let $U^{(j)} = \{u_{v,j} \mid v \in V\}$ denote the set of new nodes for advertiser j, $X_v := \{u_{v,j} \mid j = 1, \ldots, m\}$ the set of new nodes corresponding to the node v, an and $U = \bigcup_j U^{(j)} = \bigcup_v X_v$. The influence function for node $u_{v,j}$ is $f_{u_{v,j}}(S) := f_v^{(j)}(\{v' \mid u_{v',j} \in S\})$, and thus only depends on the nodes in the network for advertiser j. Writing $\hat{A} \subseteq U$ for the (random) final set of active nodes in $\hat{\mathcal{G}}$, the objective function is $W(A_0) = \sum_j w^{(j)}(|\{v \mid u_{v,j} \in \hat{A}\}|)$, which is submodular by construction.

Targeting the node $u_{v,j}$ in $\hat{\mathcal{G}}$ corresponds to exposing node v to ad j in the original problem. In this way, the problem is simply to choose a subset $S \subseteq U$. The correspondence between seed sets of $\hat{\mathcal{G}}$ and ad seeding choices in the original problem is that $A_0^{(j)} = \{v \mid u_{v,j} \in A_0\}$. The constraints on the selection of A_0 are then that $|\{j \mid u_{v,j} \in A_0\}| \leq r_v$, $|A_0^{(j)}| \leq K^{(j)}$ (as well as $|A_0| \leq K$).

Because the sets $U^{(j)}$ form a disjoint partition of U, the restriction that at most $K^{(j)}$ nodes from $U^{(j)}$ may be selected defines a partition matroid.[7] Similarly, because the sets X_v form a partition, the constraint that at most r_v nodes from X_v may be selected defines a different partition matroid. The constraint on the total number of selected nodes can be added to either of these partition matroids, and turns it into a truncated partition matroid. Thus, a consequence of this reduction to the optimization problem on $\hat{\mathcal{G}}$ is that the constraints on A_0 form the intersection of two matroids: a partition matroid on the sets $U^{(j)}$, and a truncated partition matroid defined by the sets X_v and the overall constraint of K on the seed set size. The optimization goal is to select a set A_0, subject to these two matroid constraints, so as to maximize a non-negative, monotone, submodular function. The latter is a well-studied problem. The key results for our purposes are the following:

[7] Readers unfamiliar with the standard definitions of matroids and (truncated) partition matroids are directed to Appendix A.1 or [29].

Theorem 2 (Theorem 3.1 of [23]) *There is a simple local search algorithm for maximizing a non-negative non-decreasing submodular function subject to $k \geq 2$ matroid constraints. For any $\epsilon > 0$, the algorithm (with suitable termination condition) provides a polynomial-time $\frac{1}{k+\epsilon}$-approximation.*

Theorem 3 ([14]) *The greedy algorithm, which always adds the next element maximizing the increase in the objective function (subject to not violating the matroid constraint), is a $\frac{1}{k+1}$ approximation algorithm for maximizing a non-negative non-decreasing submodular function subject to k matroid constraints.*

Datta et al. prove an approximation guarantee of 1/3 for a greedy hill climbing algorithm; as discussed above, they do not consider budget constraints on the total number of exposures. Furthermore, they consider only the special case when all advertisers have the same influence functions at all nodes. Their proof shows that the constraints form a p-system for $p = 2$ (a generalization of the intersection of two matroids); they then invoke an analysis of Calinescu et al. [6] for such systems. The same result can be obtained by appealing to Theorem 3 instead. By making the connection to the intersection of matroids explicit[8] and invoking Theorem 2 instead, we improve the approximation guarantee for this problem to essentially $\frac{1}{2}$, under a much more general problem setting.[9]

The model of Aslay et al. does not impose constraints on the seed set sizes of individual advertisers, and the node exposure constraints and the overall seed set size constraint form a *single* truncated partition matroid (as opposed to the intersection of *two* matroids). If we do not consider the penalty of overshooting advertisers' budgets, the resulting payoff function is non-decreasing submodular; therefore, the approximation guarantee can be improved to $(1 - 1/\mathrm{e})$ by appealing to the following theorem of Vondrák et al.

Theorem 4 ([6, 33]) *The continuous-greedy algorithm of [6, 33] is a polynomial-time $(1-1/\mathrm{e}-\epsilon)$-approximation for maximizing a non-negative monotone submodular function subject to any matroid constraint, for arbitrarily small $\epsilon > 0$.*

While the continuous-greedy algorithm runs in polynomial time, it is not practical. Notice, however, that Theorem 3 with $k = 1$ implies that the greedy algorithm is a $\frac{1}{2}$-approximation for the problem of maximizing the network's expected revenue from coordinated multi-advertiser campaigns, subject to budget constraints or more general concave advertiser value functions.

The main message of this section is that, when focusing on the arguably more natural objective of maximizing the network's revenue (rather than minimizing regret), non-trivial (and in some cases well-established) algorithmic techniques can be leveraged to obtain algorithms with approximation guarantees essentially matching those of standard influence maximization. These guarantees apply in a wide variety of influence models (allowing for completely different influence functions for different advertisers), and under a variety of constraints about the seed sets for different advertisers. In fact, by leveraging subsequent work on submodular maximization under multiple Knapsack and matroid constraints (see, e.g., [7, 28, 23, 34]), one can also obtain (somewhat weaker) approximation guarantees when different seed nodes have different costs (in terms of money or effort) for targeting.

[8] Datta et al. also discuss matroids, though mostly to remark that the constraints do not form a matroid, and thus to motivate an analysis in terms of p-systems.

[9] It should be noted that the analysis of Datta et al. is not really specific to the model they formulate, and could easily be extended to the general problem statement from Section 2. Thus, our contribution here is not so much technical as conceptual, explicitly formulating the general problem statement, and identifying the properties that are really required to obtain the desired guarantees.

4 An LP-rounding Algorithm for the Independent Cascade Model

In this section, we focus on the special case of the Independent Cascade (IC) Model for influence in a network, and a budgeted linear valuation function: the payoff the host receives from advertiser j is $w^{(j)}(\hat{A}^{(j)}) = \min(B^{(j)}, \gamma^{(j)} \cdot |\hat{A}^{(j)}|)$. The IC Model for influence has been widely studied; a reader wishing to briefly review the definition may find a description in Appendix A.2. In talking about the model, we use $p_e^{(j)}$ to denote the probability that an edge $e = (u, v)$ activates node v for product j when node u has become active for product j.

We use the idea of Reverse Reachability sets [4, 30, 31] to reduce the multi-advertiser campaign coordination problem to a generalized Maximum Coverage problem; we then show how to round the LP, using an algorithm of Gandhi et al. [15], to obtain a polynomial-time and reasonably practical $(1 - 1/\mathrm{e})$-approximation algorithm for the problem with both node and advertiser constraints.

This approximation guarantee gives an improvement over the $\frac{1}{2}$-approximation guarantee of the local search algorithm (Theorem 2). When the seed set sizes of individual advertisers are not restricted, the continuous greedy algorithm of Theorem 4 matches the LP rounding algorithm's $(1-1/\mathrm{e})$ guarantee, but the continuous greedy algorithm is completely impractical. The LP rounding algorithm is less efficient than the simple greedy algorithm from Theorem 3, but provides better approximation guarantees.

The downside, compared to the more general treatment in Section 3, is that the results only hold for the IC Model, and the budget constraints on the number of exposures will only be satisfied in expectation: in any one run, the algorithm may charge an advertiser j (significantly) more than $B^{(j)}$.

4.1 The Reverse Reachability Technique

A very useful alternative view of the Independent Cascade Model was first shown in [21], and heavily used in subsequent work: generate graphs $G^{(j)}$ by including each edge e in $G^{(j)}$ independently with probability $p_e^{(j)}$. Then, the distribution of nodes activated by ad j in the end, when starting from the set $A_0^{(j)}$, is the same as the distribution of nodes reachable from $A_0^{(j)}$ in the random graph $G^{(j)}$.

This alternative view forms the basis of the Reverse Reachability Set Technique, first proposed and analyzed by Borgs, Brautbar, Chayes, and Lucier [4], and further refined by Tang, Xiao, and Shi [30, 31]. The primary goal of [4, 30, 31] was to permit a more efficient evaluation of the objective $\sigma(A_0)$. Under both the Independent Cascade and Linear Threshold Models, evaluating $\sigma(A_0)$ is known to be #P-complete [10, 35]. Reverse Reachability sets permit a more efficient (both theoretically and practically) approximate evaluation, as compared to the obvious Monte Carlo simulation.

We explain the Reverse Reachability technique for a single ad, and omit the index j for readability for now. The key insight is the following: let v be an arbitrary node, and consider the (random) set R of all nodes that can reach v in the randomly generated graph G. Then, the probability that v is activated starting from A_0 is equal to the probability that $A_0 \cap R \neq \emptyset$. (See, e.g., [31, Lemma 2].)

More generally, if we draw ρ such sets $R_1, \ldots, R_\rho$ independently, and for independently uniformly random target nodes v, and A_0 intersects $\alpha(A_0)$ of them, then $\frac{n \cdot \alpha(A_0)}{\rho}$ is an unbiased estimator of the expected number of nodes activated when starting from A_0. In order to leverage this insight for computational savings in computing ad campaigns, it is important that the estimate be sufficiently accurate with sufficiently high probability, for a small enough number ρ of reverse reachable sets. A characterization of this accuracy is given by the following lemma from [31] (restated slightly here):

Lemma 5 (Lemma 3 of [31]) *Assume that*

$$\rho \geq (8 + 2\epsilon) \cdot n \cdot \frac{c \log n + \log \binom{n}{k} + \log 2}{OPT \cdot \epsilon^2}. \tag{3}$$

Consider any set S of at most k nodes. With probability at least $1 - n^{-c}/\binom{n}{k}$, the following inequality holds for S:

$$\left| \frac{n \cdot \alpha(S)}{\rho} - \sigma(S) \right| < \frac{\epsilon}{2} \cdot OPT. \tag{4}$$

Here OPT is the maximum influence spread that can be achieved by any set S of at most k nodes.

Lemma 5 implies that when ρ is large enough to allow taking a union bound over all relevant sets S, the fraction of sets R_i that intersect a candidate seed set S is an accurate stand-in for the actual objective function $\sigma(S)$. As already observed by Borgs et al. [4], this reduces the problem of influence maximization to a Maximum Coverage problem: selecting a set of at most k seed nodes that jointly maximize the number of sets R_i containing at least one seed node.

Without loss of generality, assume that $K^{(j)} \leq K$ for all j. Let $\mathrm{OPT}^{(j)}$ be the maximum influence that advertiser j could achieve with $K^{(j)}$ seed nodes, and ignoring the budget constraint $B^{(j)}$. A lower bound $\hat{\mathrm{OPT}}^{(j)}$ on each $\mathrm{OPT}^{(j)}$ can be found by running the TIM algorithm [31] with seed sets of size $K^{(j)}$. For each advertiser j, we draw $\rho^{(j)} \geq 9n^3m^2 \cdot \frac{c \log n + \log m + \log \binom{n}{K^{(j)}} + \log 2}{\hat{\mathrm{OPT}}^{(j)} \cdot \epsilon^2}$ nodes[10] v i.i.d. uniformly (in particular, with replacement) from $U^{(j)}$, and for each such v, we let $R \subseteq U^{(j)}$ be the random reverse reachable set. For later ease of notation, we index the nodes as $v_i \in U$, and the corresponding sets as R_i. Notice that the same node v might appear as v_i for different i, with possibly different sets R_i. For each such index i of a node v_i and set R_i, let $j(i)$ be the unique advertiser j such that $R_i \subseteq U^{(j)}$, and write $\mathcal{R}^{(j)} = \{i \mid j(i) = j\}$ for the set of all sets used to estimate the influences for advertiser j.

Focus on any advertiser j, and the influence of any set $S \subseteq U^{(j)}$ with $|S| \leq K^{(j)}$. Using Lemma 5 with $\epsilon' = \frac{\epsilon}{nm} \leq \frac{1}{2}$ and $c' = c + \log_n(m)$, the influence of S is estimated to within an additive term $\frac{\epsilon}{nm} \cdot \mathrm{OPT}^{(j)}$ with probability at least $1 - n^{-c'}/\binom{n}{K^{(j)}} = 1 - 1/(n^c m) \cdot 1/\binom{n}{K^{(j)}}$. By a union bound over all such sets S (of which there are[11] $\binom{n}{K^{(j)}}$) and all m advertisers $j = 1, \ldots, m$, the influence of *all* such sets S is simultaneously estimated to within an additive term of $\frac{\epsilon}{nm} \cdot \mathrm{OPT}^{(j)}$, with probability at least $1 - 1/n^c$. Assume for the rest of this section that the high-probability event has happened.

Now consider an arbitrary seed set S with $|S| \leq K$, not necessarily contained in just one partition $U^{(j)}$. Let $S^{(j)} = S \cap U^{(j)}$, and let $q^{(j)} = \frac{n \cdot |\{i \in \mathcal{R}^{(j)} | R_i \cap S^{(j)} \neq \emptyset\}|}{\rho}$ be the estimated expected influence of $S^{(j)}$. By the preceding paragraph, the influence of each $S^{(j)}$ is estimated to within an additive $\frac{\epsilon}{nm} \cdot \mathrm{OPT}^{(j)}$, i.e., $|q^{(j)} - \sigma(S^{(j)})| \leq \frac{\epsilon}{nm} \cdot \mathrm{OPT}^{(j)}$. We next want to show that for each j, the actual

[10] This value of $\rho^{(j)}$ is chosen with foresight to apply Lemma 5 and allow the application of various bounds in the following paragraphs.

[11] When running a greedy algorithm, only $O(mnK)$ such sets need to be considered, which leads to significant computational savings. Here, our goal is to leverage the reverse reachability technique not necessarily for significant computational savings, but for improved approximation guarantees.

payoff the host obtains from advertiser j, i.e., $\min(B^{(j)}, \gamma^{(j)} \cdot \sigma(S^{(j)}))$ is estimated to within an additive term $\epsilon/m \cdot \mathrm{OPT}$; then, by summing over all m advertisers j, the total estimation error is at most $\epsilon \cdot \mathrm{OPT}$.

To prove the estimation error bound for an advertiser j, we consider several cases. First, if $S^{(j)} = \emptyset$, then the payoff 0 is estimated completely accurately. Second, if $B^{(j)} \leq \gamma^{(j)}$, i.e., the advertiser's budget is so low that he pays for at most one impression, the payoff is also estimated completely accurately. The reason is that for any $S^{(j)} \neq \emptyset$, all nodes in $S^{(j)}$ will always be shown impressions, so $\gamma^{(j)} \cdot q^{(j)} \geq B^{(j)}$ with probability 1. As a result, the payoff from the advertiser is estimated correctly as $B^{(j)}$. Finally, if $B^{(j)} \geq \gamma^{(j)}$, then $\mathrm{OPT}^{(j)} \leq n \cdot \mathrm{OPT}/\gamma^{(j)}$. The reason is that the optimum does at least as well as using $K^{(j)}$ seed nodes for advertiser j, which would yield payoff $\mathrm{OPT} \geq \min(B^{(j)}, \gamma^{(j)} \cdot \mathrm{OPT}^{(j)}) \geq \gamma^{(j)} \cdot \min(1, \mathrm{OPT}^{(j)}) \geq \gamma^{(j)}\mathrm{OPT}^{(j)}/n$, because $\mathrm{OPT}^{(j)} \leq n$. Thus, the estimation error for the payoff from advertiser j is at most $\gamma^{(j)} \cdot \frac{\epsilon}{nm} \cdot \mathrm{OPT}^{(j)} \leq \frac{\epsilon}{m} \cdot \mathrm{OPT}$.[12]

4.2 The LP and Rounding Algorithm

Notice that except for the budget caps $B^{(j)}$, the objective is a sum of (scaled) coverage functions. This suggests the natural formulation of an LP, which allows us to use a randomized rounding technique due to Gandhi et al. [15].

Recall that each set R_i is generated by a reverse (random) BFS from a randomly sampled node $v \in U$, where the same node v may be sampled for multiple i. Also recall that $j(i)$ is the unique advertiser for the reverse reachable set R_i, and that $\mathcal{R}^{(j)}$ is the set of all reverse reachability sets R_i with $j(i) = j$. We use the decision variable $z_v^{(j)}$ to denote whether node v was targeted for ad j, x_i for the decision variable whether at least one node from the set R_i was selected (for its corresponding ad), and the variable $y^{(j)}$ for the total revenue from advertiser j. The optimization objective can be expressed using the following Integer Linear Program (ILP):

$$\begin{array}{lll}
\text{Maximize} & \sum_j y^{(j)} & \\
\text{subject to} & x_i \leq \sum_{v \in R_i} z_v^{(j(i))} & \text{for all } i \\
 & x_i \leq 1 & \text{for all } i \\
 & \sum_j z_v^{(j)} \leq r_v & \text{for all } v \\
 & \sum_v z_v^{(j)} \leq K^{(j)} & \text{for all } j \\
 & \sum_{j,v} z_v^{(j)} \leq K & \\
 & y^{(j)} \leq \frac{n}{\rho} \cdot \sum_{i \in \mathcal{R}^{(j)}} \gamma^{(j)} x_i & \text{for all } j \\
 & y^{(j)} \leq B^{(j)} & \text{for all } j \\
 & x_i, y^{(j)}, z_v^{(j)} \geq 0 & \text{for all } i, j, v \\
 & z_v^{(j)} \in \{0, 1\} & \text{for all } j, v.
\end{array} \tag{5}$$

The first constraint states that a target node v_i is only covered if at least one node from the set R_i is selected, and the second constraint ensures that the LP derives no benefit from double-covering nodes. The third constraint encodes the bound on the number of targeted ads that v can be exposed to, while the fourth constraint encodes the bound on the seed set sizes of the advertisers

[12] Some of the complications in the proof arose because of the possibility of very different budgets and payoffs per node across advertisers. When the payoff per node for all advertisers is equal, i.e., $\gamma^{(j)} = \hat{\gamma}$ for all j, then $\rho^{(j)} \geq (8 + 2\epsilon)n^2m^2\hat{\gamma}^2 \cdot \frac{c \log n + \log \binom{n}{K} + \log K + \log 2}{\epsilon^2 OPT^2}$ RR set samples are enough to provide accurate payoff estimates with sufficiently high probability.

j. The fifth constraint captures the overall bound on the total number of targeted ads. The sixth and seventh constraints characterize the objective function for advertiser j.

Of course, solving the ILP is NP-hard, so as usual, we consider the fractional LP relaxation obtained by omitting the final integrality constraint $z_v^{(j)} \in \{0,1\}$. Without this constraint, the LP can be solved in polynomial time, obtaining a fractional solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.

To round the fractional solution, we use the dependent rounding algorithm of Gandhi et al. (See Section 2 of [15].) This algorithm can be applied to round fractional solutions whenever the variables to be rounded (here: the $z_v^{(j)}$) can be considered as edges in a bipartite graph, and constraints correspond to bounds on the (fractional) degrees of the nodes of the bipartite graph.[13] It repeatedly rounds the fractional values associated with a subset of edges forming a maximal path or a cycle, until all variables are integral; let $\hat{z}_v^{(j)}$ denote the rounded (integral) values corresponding to $z_v^{(j)}$. The algorithm of [15] provides the following guarantees:

- The probability that the variable $z_v^{(j)}$ is rounded to $\hat{z}_v^{(j)} = 1$ is exactly $z_v^{(j)}$, for all j and v.
- In the bipartite graph, each node's degree is rounded up or down to an adjacent integer. In the context of our LP (5), this means that the number of seed nodes for each advertiser is either rounded up or down to the nearest integer, as is the number of exposures to sponsored ads for each node. In particular, the third and fourth constraints are satisfied with probability 1.
- For any node of the bipartite graph and any subset of incident edges, the rounded values are negatively correlated. In the context of our LP (5), for any advertiser $j = j(i)$ and reverse reachable set R_i, the algorithm guarantees that $\text{Prob}[\hat{z}_v^{(j)} = 0 \text{ for all } v \in R_i] \leq \prod_{v \in R_i} \text{Prob}[\hat{z}_v^{(j)} = 0] = \prod_{v \in R_i} (1 - z_v^{(j)})$.

After rounding all the $z_v^{(j)}$ to $\hat{z}_v^{(j)}$, for each i, we set $\hat{x}_i = 1$ if at least one $v \in R_i$ has $\hat{z}_v^{(j(i))} = 1$, and for each j, we set $\hat{y}^{(j)} = \sum_{i \in \mathcal{R}^{(j)}} \gamma^{(j)} \cdot \hat{x}_i$. Together with the guarantees of Gandhi et al., this ensures that all ILP constraints except possibly the seventh (budget) constraint are satisfied; the budget constraint is discussed more in Section 4.4.

Theorem 6 *Under the IC model, with constraints on node exposures and advertiser seed set sizes, the (modified) Gandhi et al. correlated LP-rounding based algorithm is a polynomial-time* $(1 - 1/\mathrm{e})$*-approximation algorithm, if advertisers pay based on their* expected *exposure.*

Proof. Consider any set $R_i \in \mathcal{R}^{(j)}$. With explanations of some steps given below, the probability

[13] The fifth constraint of our LP (overall budget) does not fit in this framework, since it is a constraint on all variables. We extend the rounding algorithm of Gandhi et al. by one more final loop (see below) which repeatedly picks any two fractional variables and randomly rounds them using the same approach as the rest of the algorithm, until only one fractional variable remains. This last variable can then be rounded randomly by itself.

that $\hat{x}_i = 1$ is

$$\begin{aligned}
\mathrm{Prob}[\hat{x}_i = 1] &= 1 - \mathrm{Prob}[\hat{z}_v^{(j(i))} = 0 \text{ for all } v \in R_i] \\
&\overset{(1)}{\geq} 1 - \prod_{v \in R_i} \mathrm{Prob}[\hat{z}_v^{(j(i))} = 0] \\
&= 1 - \prod_{v \in R_i} (1 - z_v^{(j(i))}) \\
&\overset{(2)}{\geq} 1 - \mathrm{e}^{-\sum_{v \in R_i} z_v^{(j(i))}} \\
&\geq 1 - \mathrm{e}^{-\min(1, \sum_{v \in R_i} z_v^{(j(i))})} \\
&\overset{(3)}{\geq} (1 - 1/\mathrm{e}) \cdot \min(1, \sum_{v \in R_i} z_v^{(j(i))}) \\
&\overset{(4)}{\geq} (1 - 1/\mathrm{e}) \cdot x_i.
\end{aligned}$$

The inequality labeled (1) used the negative dependence discussed above. The inequality labeled (2) used the standard bound that $1 - x \leq \mathrm{e}^{-x}$ for all $x \geq 0$. The inequality labeled (3) uses that the function $x \mapsto 1 - \mathrm{e}^{-x}$ is concave, and hence on the interval $[0, 1]$ is lower-bounded by the straight line which agrees with it at the endpoints $x = 0$ and $x = 1$, which is the function $x \mapsto (1 - 1/\mathrm{e}) \cdot x$. Finally, the inequality labeled (4) uses that $x_i \leq \min(1, \sum_{v \in R_i} z_v^{(j(i))})$ by the first and second LP constraints. Using the definition of $\hat{y}^{(j)}$, we now get that

$$\begin{aligned}
\mathbb{E}\left[\hat{y}^{(j)}\right] = \mathbb{E}\left[\sum_{i \in \mathcal{R}^{(j)}} \gamma^{(j)} \cdot \hat{x}_i\right] &= \sum_{i \in \mathcal{R}^{(j)}} \gamma^{(j)} \cdot \mathrm{Prob}[\hat{x}_i = 1] \\
\geq \sum_{i \in \mathcal{R}^{(j)}} \gamma^{(j)} \cdot (1 - 1/\mathrm{e}) \cdot x_i &\geq (1 - 1/\mathrm{e}) \cdot y^{(j)}.
\end{aligned}$$

■

4.3 No Seed Set Size Restrictions

In the model of Aslay et al., and in our experiments in Section 5, there are no restrictions placed on the seed set sizes of individual advertisers; that is, the fourth set of constraints in the LP (5) is missing. In the absence of these constraints, the correlated LP-rounding algorithm of Gandhi et al. [15] can be further sped up. Note that except for the fifth (sum of seed set sizes) constraint, the $z_v^{(j)}$ constraints are a collection of disjoint stars. As a result, the general correlated rounding algorithm of Gandhi et al. can be replaced with their algorithm for stars. The resulting algorithm runs in linear time and is very simple to implement. It first uses the correlated rounding on pairs of variables $z_v^{(j)}, z_v^{(j')}$ corresponding to the same node (and different advertisers) until no such pairs remain; subsequently, as for the general algorithm, we add one more **while** loop to round pairs of the remaining $z_v^{(j)}$ (which now necessarily correspond to distinct nodes v, v').

Because there is at most one fractional variable remaining incident on any node j or v, the second property of the Gandhi et al. rounding algorithm is preserved. The negative correlation property holds inductively, because at most two values are rounded in each step.

4.4 The Budget Constraints

In Theorem 6, we emphasized that advertiser payments have to be based on the expected exposure, rather than the actual exposure. Indeed, using the LP rounding approach, maintaining the budget constraints in each run is not possible without a huge ($\Omega(m)$) loss in the approximation guarantee. The reason is that LP (5) has an integrality gap of $\Omega(m)$ when the budgets are constrained. Consider the following input: the influence graph is a star with n leaves, in which the center node has probability 1 of influencing all its neighbors, and no one else can influence any other node. Thus, any advertiser who gets to advertise to the center node will reach the entire graph.

Suppose that each advertiser j has a budget of $B^{(j)} = n/m$, and the center node v has a constraint of $r_v = 1$. A fractional solution can allocate a fraction of $1/m$ of the center node to each advertiser; each advertiser then reaches n/m nodes, not exceeding his budget, and the host obtains a payoff of n. But any integral solution can only allocate v to one advertiser, who will then pay at most his budget n/m. Thus, any integral solution obtains payoff at most n/m.

This problem disappears if each advertiser's budget must only be met in expectation. In other words, on any given day, the advertiser's budget could be exceeded, even by a lot. But on average, the advertiser does not pay more for impressions than $B^{(j)}$ per day. Because the number of impressions resulting from a particular rounding choice is a random variable in $[0, n]$, standard tail bounds (e.g., Hoeffding Bounds) show that the total payment is with high probability close to the expected payment after about $O(n)$ days.

5 Experiments

In this section, we describe an empirical evaluation of the relationship between the total revenue to the host and the algorithm used, the number of advertisers, and the total number of sponsored ads. More importantly, we also consider the interplay of competition with the influence strengths in the network and the similarity/dissimilarity of the advertisers' influence networks. We conducted our experiments on a 36-core Linux server with an Intel Xeon E5-2695 v4 processor at 2.1GHz and 1TB memory. We develop a parallel version of the Greedy algorithm that scales to networks of millions of nodes and tens of millions of edges, with dozens of advertisers (see Section 5.5).

All of our experiments are based on the Independent Cascade (IC) Model. We employed the Reverse Reachability technique to (approximately) compute the influence of node sets more efficiently. In Section 4, we obtained an upper bound on the number of RR sets $\rho^{(j)}$ to guarantee good approximations with high probability. This bound is quite large, and using the corresponding number of sets would not permit us to scale experiments to large networks. We conducted extensive experiments on the number of RR sets that are sufficient to guarantee high accuracy *in practice*. These experiments are described in detail in Appendix B; the upshot is that $\rho^{(j)} \geq 10n$ RR sets are sufficient to guarantee an estimation error $\epsilon \leq 2\%$. Therefore, for the experiments reported in the remainder of this section, we used a value of $\rho^{(j)} = 10n$.

5.1 Data Sets and Generation of IC Instances

We used the following four real-world networks, whose key statistics are summarized in Table 1. (Two larger networks used for scalability experiments are described in Section 5.5.)

- *Facebook* [22] is an ego network of a user in Facebook, excluding the ego node.
- *Advogato* [22] is a social network whose nodes are users of the Advogato platform; directed edges (u, v) are trust links.
- *DBLP* [22] is a citation network in which nodes are papers, and there is a directed edge from u to v if u cites v.
- *NetHEPT* [9] is a collaboration network generated from co-authorships in high-energy physics publications.

Table 1: Data Set characteristics

Dataset	#nodes	#edges	Type
Facebook	2,889	2,981	Undirected
Advogato	6,542	51,127	Directed
DBLP	12,592	49,743	Directed
NetHEPT	15,229	31,376	Undirected

In creating influence networks (with edge probabilities $p^{(j)}_{u,v}$), for most of our experiments, we wanted to avoid the assumption of uniform probabilities on edges commonly made when evaluating algorithms for influence maximization. The reason is that one key aspect of our work is the notion of coordination/competition between different advertisers. When all edges have uniform probabilities $p^{(j)}_{u,v}$, the value of a node is solely determined by its network position. We therefore define a way of generating edge probabilities non-uniformly that gives some nodes intrinsically more importance. For each node v, we draw a parameter λ_v independently and uniformly from $[0, 0.4]$ for *Facebook*, *DBLP* and *NetHEPT*, and $[0, 0.3]$ for *Advogato*. (The different intervals were chosen to counteract the effects of varying edge densities in the datasets.) We then define $p_{u,v} = \lambda_u \cdot \lambda_v$. We will discuss below how the $p^{(j)}_{u,v}$ are correlated for different j.

For all experiments, we set the constraint on the number of sponsored ads that can be shown to node v to $r_v = 1$, i.e., each node can be shown a sponsored ad from at most one advertiser. This results in maximal inter-advertiser constraints for the seed sets. We also performed experiments varying the value of r_v. The outcome of these experiments is that what matters is mostly the *ratio* r_v/m. As r_v grows (with m fixed), the optimization problem gradually decouples across advertisers. As a result, the most interesting and novel aspects of the model manifest themselves when $r_v = 1$.

For simplicity, we set each advertiser's payoff per exposure to $\gamma^{(j)} = 1$. The overall numbers of sponsored ads K are varied for different experiments, and discussed below. In our experiments, we did not consider individual advertiser seed set constraints $K^{(j)}$; such constraints are heavily studied in traditional Influence Maximization experiments, and we wanted to focus on the novel aspects arising due to advertiser competition.

5.2 Comparison between the Algorithms

Our first set of experiments simply compared the performance of our algorithms and several baseline heuristics. We implemented the following algorithms:

- *Greedy* is the standard greedy algorithm for maximizing a submodular function subject to a matroid constraint (Theorem 3).
- *LP-Rounding* is described in Section 4.2.
- *Max-Degree* considers nodes by non-increasing degrees, and assigns them to advertisers in a round-robin order. Each node v is assigned to r_v consecutive advertisers.
- *Eigen-Centrality* considers nodes by non-increasing eigenvector centrality[14], and assigns them to advertisers in a round-robin order, as with *Max-Degree* .
- $\mathrm{OPT}_{\mathrm{LP}}$ is the value of the optimal fractional solution of the LP (5). It provides an upper bound on the value of the optimal solution, and thus gives us a benchmark to compare the algorithms' performance to, on an absolute scale.

5.2.1 Varying the total number of seeds

In the first set of experiments, we kept the number of advertisers constant at $m = 3$, and varied the total number K of seed nodes. Thus, these experiments are similar to evaluations of standard Influence Maximization algorithms. To avoid strong effects of competition between advertisers (which we are evaluating in later sections), we generated the $\lambda_v^{(j)}$ (and hence the $p_{u,v}^{(j)}$) independently for each j.

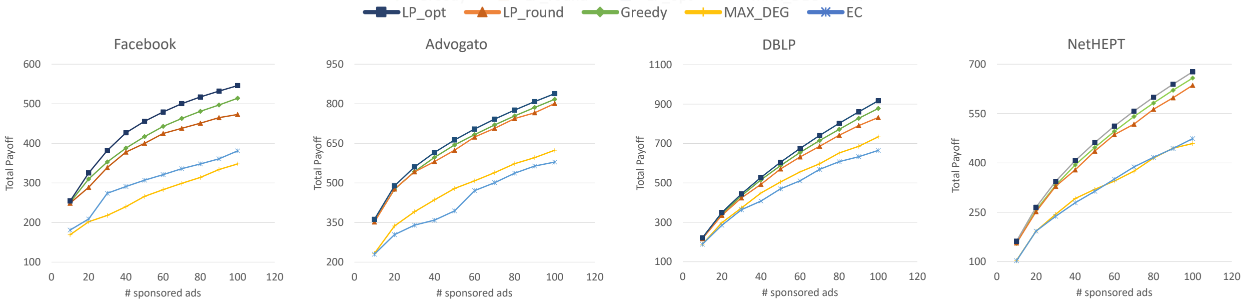

Figure 1: Host payoff vs. number of sponsored ads (K)

Figure 1 shows a comparison of the total host payoff achieved by the algorithms as K is varied from 10 to 100. Both *LP-Rounding* and *Greedy* perform significantly better than *Max-Degree* or *Eigen-Centrality*. This is not surprising, as the heuristics only consider the network *structure*, but not the influence probabilities associated with edges. However, the random generation of edge probabilities still ensures that nodes of high degree or high centrality tend to be more influential; hence, in some scenarios (especially with small K), the payoffs of the *Max-Degree* and *Eigen-Centrality* heuristics are comparable to those of *Greedy* and *LP-Rounding*.

[14] The eigenvector entries of the leading eigenvector of the graph's adjacency matrix.

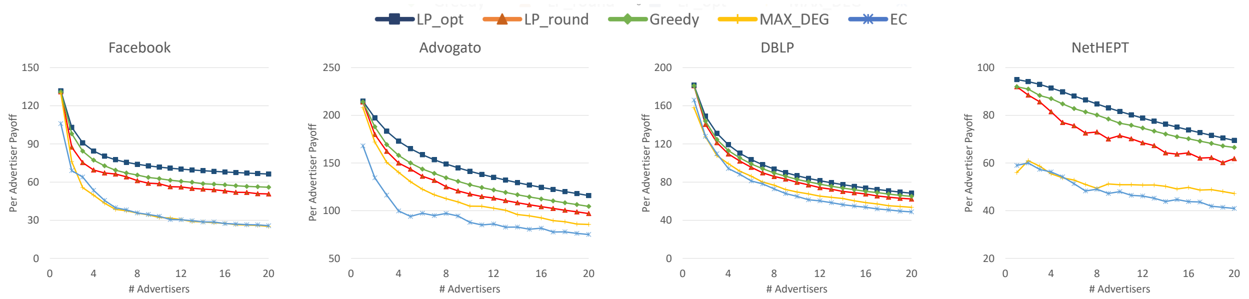

Figure 2: Average per-advertiser payoff vs. number of advertisers (m)

A comparison to the fractional LP solution value shows that both *Greedy* and *LP-Rounding* achieve more than 85% of the optimal payoff, which is significantly more than the respective guarantees of $\frac{1}{2}$ and $1-\frac{1}{e}$. Experimentally, on these instances, *Greedy* performed marginally better than *LP-Rounding*.

5.2.2 Varying the number of advertisers

For the second set of experiments, we varied the number m of advertisers from 1 to 20, scaling the number of seeds as $K = 10m$, to keep an average of 10 seeds per advertiser. Again, we did not add constraints on the budgets or the individual advertisers' seed sets.

Using this set of experiments, we also studied the effect (on payoff) of competition between advertisers, as caused by the constraint that each node can only be chosen as a seed for one advertiser.[15] Therefore, we made the influence probabilities $p^{(j)}_{u,v}$ the same for all j, i.e., the same nodes are most influential for all advertisers.

Figure 2 shows the per-advertiser payoff achieved as m is varied from 1 to 20. A comparison of the algorithms' performance yields results similar to those reported in Section 5.2.1. For a single advertiser, the performance of the *Max-Degree* heuristic is comparable to *LP-Rounding* and *Greedy*. As explained earlier, this observation can be attributed to the random generation of uniform edge probabilities. For larger values of m, there is a significant difference in the performances of the *Greedy* and *LP-Rounding* algorithms vs. the *Max-Degree* and *Eigen-Centrality* heuristics. A likely explanation is the following: when many advertisers compete on the same network, *Greedy* and *LP-Rounding* can alleviate competition for a limited pool of highly influential nodes by indirectly influencing such nodes using other carefully chosen seeds. In contrast, *Max-Degree* and *Eigen-Centrality* do not consider such potential propagation for seed selection.

Another observation is that as a result of the competition, per-advertiser payoff decreases and hence, the total host payoff does not scale linearly in m. For a single advertiser, *DBLP* and *Advogato* have significantly higher payoff than *Facebook* and *NetHEPT*. With 20 advertisers, on the *DBLP* and *Facebook* networks, the average payoff per advertiser decreases by factors of 2.8 and 2.4, respectively, while the decrease for *Advogato* is a factor of 2. Comparatively, *NetHEPT* exhibits only modest competition, and its per-advertiser payoff decreases only by a factor of 1.4. A possible cause for the

[15] If there were no constraint on the number of sponsored ads per node, there would be no competition, and the total payoff would increase linearly in m. Note also that the total influence of the seed sets of all advertisers is different from the influence of their seed sets' union if assigned to one advertiser: while no two advertisers can choose the same seed node, if the different seed nodes *influence* the same nodes later, they will both derive utility (and the host revenue) from those exposures.

higher decreases may be the very skewed degree distributions of the *DBLP* and *Facebook* networks, which results in a smaller number of extremely valuable nodes, and the fact that *Advogato* and *DBLP* are directed, resulting in less symmetry between nodes. The strong competition for *DBLP* and *Facebook* can also be observed in the fact that the gradient of the per-advertiser payoff is very negative for small values of m. For larger m, the decrease becomes less pronounced, because there are many marginally useful seed nodes to choose from.

5.3 Effects of Competition

Next, we focused in detail on the effects of competition on total host payoff. More specifically, the question we were interested in is the following: how does the similarity or dissimilarity of influence networks for different advertisers affect revenue? If the influence networks are very similar, then high-value seed nodes for one advertiser will typically also be high-value for others, and the constraint that no node must be chosen by more than a given number of advertisers (in our experiments: 1) will constrain the reach of seed sets. On the other hand, if the influence networks are very different, then different advertisers might focus on different parts of the network, and the host could potentially derive significantly higher payoff.

Since the purpose of these experiments is not to compare the performance of algorithms, but to draw qualitative insights, we ran these experiments only using the *Greedy* algorithm, which had performed best in our earlier experiments.

For these experiments, we fixed the number of advertisers to $m = 20$ and total seeds to $K = 200$. To cover a spectrum of different similarities between networks, we used the following generative model. All advertisers initially have the same edge probabilities $p_{u,v}$. We can assume for simplicity that the graph is complete, by setting $p_{u,v} = 0$ whenever (u, v) is not an edge. A parameter $s \in \{0, 1, \ldots, 200\}$ will capture the similarity between the networks for different advertisers. Each advertiser j's influence strengths $p^{(j)}_{u,v}$ are generated independently as follows. Starting from $p_{u,v}$, perform $s \cdot \frac{n}{100}$ *node swaps* of the following form: select two vertices u, v independently and uniformly at random, and switch all their associated influence probabilities, i.e., set $\hat{p}^{(j)}_{u,w} = p^{(j)}_{v,w}$ and $\hat{p}^{(j)}_{v,w} = p^{(j)}_{u,w}$ for all $w \neq u, v$, and $\hat{p}^{(j)}_{u,v} = p^{(j)}_{v,u}$ and $\hat{p}^{(j)}_{v,u} = p^{(j)}_{u,v}$.

The effect is that the influence networks for all advertisers j are exactly isomorphic to each other, i.e., no advertiser has an a priori better network. However, the larger the value of s, the more independent the networks are, which we expected to lead to more potential to derive payoff from all advertisers simultaneously.

Figure 3 shows the payoff as a function of s. First, notice that the host's payoff does indeed increase steeply in s, nearly linearly for a non-trivial segment. This shows that competition between advertisers for high-impact nodes indeed restricts the host's payoff; as the advertisers' influence networks become more dissimilar, the host can extract more payoff from the joint ad campaigns.

Next, we observe that the relative increase in payoff (comparing no swaps vs. $2n$ swaps) is noticeably higher in the *Facebook* (a factor of 2.4) and *DBLP* (a factor of 2.7) networks, compared to the *Advogato* (a factor of 1.8) and *NetHEPT* (a factor of 1.3) networks. This aligns with our earlier observations: the *Facebook* and *DBLP* data sets seem to have fewer high-influence nodes, as compared to the more even influence of nodes in *Advogato* and *NetHEPT*. Thus, ensuring more independence among the isomorphic copies of the *Facebook* and *DBLP* graphs creates more potential for additional payoff.

Further, the payoff saturates around just $n/3$ swaps for *NetHEPT*, as opposed to around n

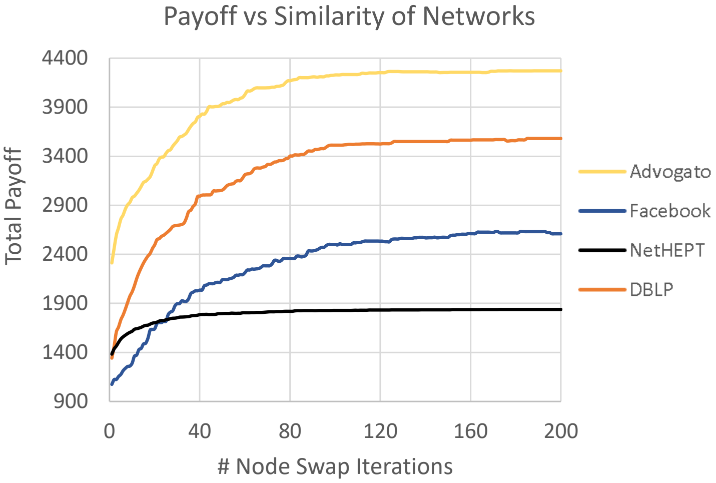

Figure 3: Host payoff vs. influence network similarity.

swaps for *Facebook*, *Advogato* and *DBLP*. This is likely because networks with more competition require more swaps to completely realize the potential of essentially independent campaigns.

5.4 Effect of Influence Probabilities

For our final set of experiments, we were interested in the interplay between competition and edge strengths. We expected two counter-acting effects: as the probabilities on edges increase, more different seed nodes may become capable of reaching the same large part of the network, thus reducing the negative effects of competition. On the other hand, as the edge probabilities decrease, most cascades will not spread beyond a few nodes; as a result, all parts of the network may provide small influence, so again, the additional detrimental effects of competition could be reduced.

For m advertisers, we chose a combined seed set size of $K = 10m$, and gave each advertiser a budget of $B^{(j)} = n/5$. Different from the earlier experiments, we assigned *uniform* probabilities of p to the edges, and varied the value of p. Again, because our goal was to study the interplay between competition and parameters of the model (rather than comparing algorithms), we only used the *Greedy* algorithm. We were interested in the host's payoff increase as the number of advertisers is increased from 1 to 20. We call the ratio between the two quantities the *relative payoff at* $m = 20$, and denote it by α.

Figure 4 shows the relative payoff α as a function of p. The two counter-acting effects produce — for all four networks — a local minimum in α. As p grows large, the ratio saturates at 20; this is not surprising, as with high enough probabilities, essentially any node will reach the entire graph, and there is in effect no competition, resulting in a 20-fold increase in host payoff.

For *Advogato* and *NetHEPT* (the two graphs which earlier showed less susceptibility to competition), the ratio α is minimized at the same $p = 0.02$. However, the underlying reasons appear to be different. *Advogato* has a high edge density; as a result, advertisements spread to a large portion of the graph even for small p; in particular, already for $p \geq 0.1$, the reach of every ad campaign j matches the budget $B^{(j)}$. The curve for *NetHEPT* looks similar, but for different reasons. Here, the reason appears to be that *NetHEPT* is sparse and exhibits little competition from the start, which is confirmed by the fact that α never drops below 15. Here, the advertisers' budgets $B^{(j)}$

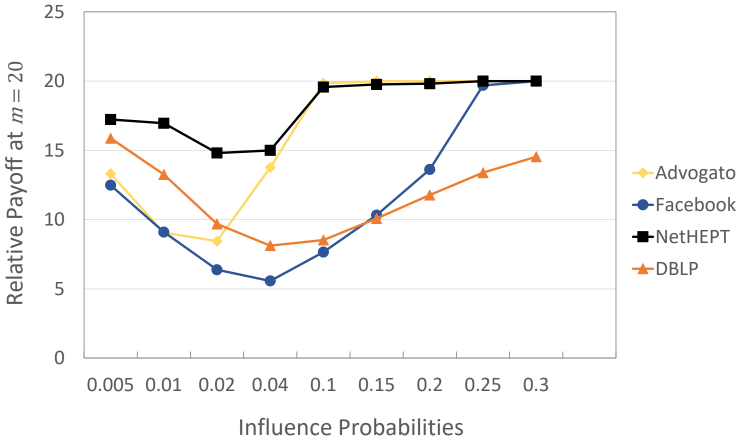

Figure 4: Relative payoff with $m = 20$ advertisers vs. influence probability p (shown in non-linear scale to display interesting regimes)

are only fully extracted around $p = 0.25$, but the decentralized nature of the graph ensures lack of harmful competition much earlier, which is why α saturates around $p = 0.1$.

For *Facebook* and *DBLP*, the two graphs exhibiting more competition, the minimum is attained at $p = 0.04$. Notice that the minimum value of α for these data sets is significantly smaller and the saturation of $\alpha = 20$ happens later than for the other networks. This behavior again shows the stronger competition. Specifically for *DBLP*, even at $p = 0.3$, the network only extracted about half of the sum of all advertiser budgets.

5.5 Scalability and Parallelization

We evaluated the scalability of the Greedy algorithm on the following two large datasets:

- *Pokec* [22] is a friendship network from the Slovakian social network Pokec, containing 1,632,803 individuals (nodes), and 30,622,564 directed edges (directed friendships).
- *Flixster* [22] is a social network of a movie rating site where people with similar cinematic taste can connect with each other. It contains 2,523,386 individuals and 7,918,801 (undirected) edges.

We evaluated scalability by measuring the runtime as a function of the number of advertisers m. The LP Rounding algorithm does not scale to such large datasets; its limit is a few tens of thousands of nodes and edges.

In order to utilize all 36 cores on the server, we parallelized the RR set construction in the Greedy algorithm, which contributes most to the overall execution time. Figure 5 shows that the parallel algorithm can compute seed sets on the large graphs in scenarios with large numbers of competing advertisers, even when the sequential algorithm fails to terminate in a reasonable amount of time. The parallel Greedy algorithm accelerates RR-set construction by a factor about 22 on average, and is overall about 12 times faster than the sequential version.

Due to the large graph sizes and extremely skewed degree distributions in *Pokec* and *Flixster*, the highly influential nodes had significantly more impact, and the overall payoff was up to two orders of magnitude larger than for the *Facebook*, *Advogato*, *NetHEPT* and *DBLP* networks. This

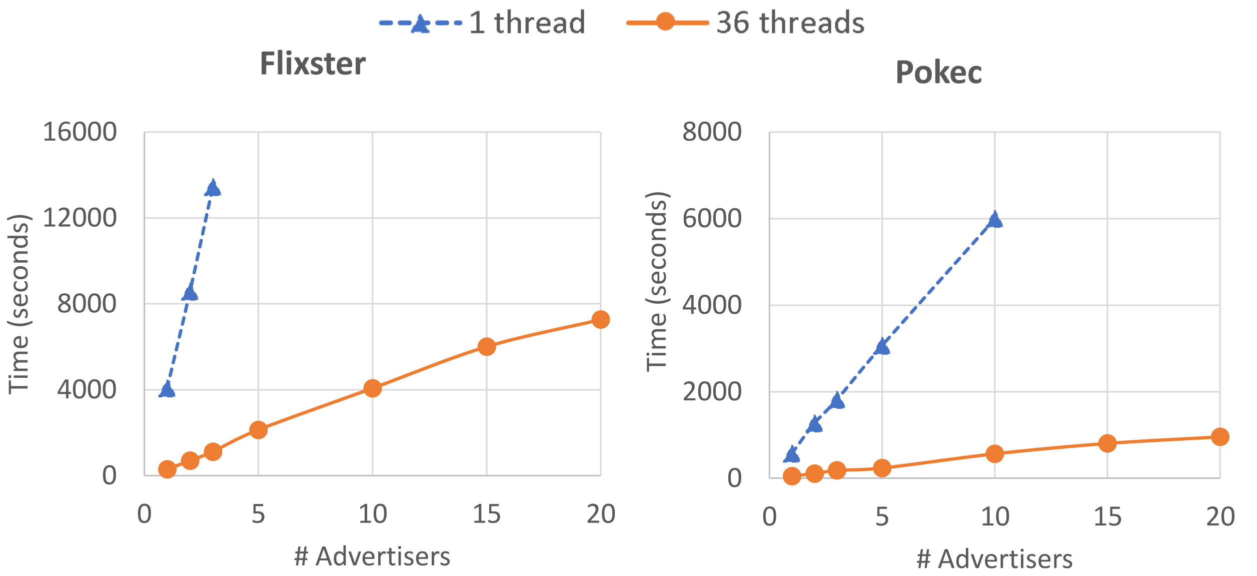

Figure 5: Running times of the sequential and parallel Greedy algorithm on the *Flixster* and *Pokec* datasets.

resulted in slower seed set selection, which became a significant factor when the RR-set construction was parallelized. As a result, we see that the runtime of our parallel implementation (especially in *Pokec*) scales sub-linearly with respect to m when there are more than a dozen advertisers.

6 Conclusions and Future Work

We explored the problem of maximizing the host's payoff for a budgeted multi-advertiser setting with constraints on the ad exposure of each node and each advertiser's seed set size. We showed that under a very general class of influence models, this problem can be cast as maximizing a submodular function subject to two matroid constraints. It can therefore be approximated to within essentially a factor $\frac{1}{2}$ using an algorithm due to Lee et al. [23]; this improves on the $\frac{1}{3}$-approximation guarantee of [11]. When there are no constraints on individual advertisers' seed set sizes, only on the total number of seeds, the constraints form just a single truncated partition matroid. Therefore, the algorithm of Vondrák et al. [6, 33] gives $(1 - \frac{1}{e})$-approximation algorithm, while a simple greedy algorithm [14] achieves a $\frac{1}{2}$-approximation. For the special case of the Independent Cascade model, the Reverse Reachability Technique [4, 30, 31] can be combined with a linear program and rounding due to [5, 15] to yield a more efficient $(1 - 1/e)$-approximation algorithm, even under both types of constraints.

Our experiments show that in practice, the greedy algorithm slightly outperforms the LP-rounding based algorithm, despite its worse approximation guarantee. They also reveal that competition between advertisers leads to a loss in payoff for the host when the advertisers' influence networks are more similar, and particularly so when the networks have few highly influential nodes — in that case, these nodes cannot simultaneously be used as seeds for all advertisers, and the average exposure per advertiser becomes significantly lower than the individual exposures of the advertisers on isolated networks.

Our work was in part motivated by considering a more “natural” objective function than the notion of regret from [3]. If one believes that giving advertisers free exposure is detrimental to the host company, excess exposure could alternatively be modeled as a negative term in the objective function. In additional work not included here, we show that if the penalty per unit excess exposure is constant, the greedy algorithm and LP rounding algorithm still yield constant-factor approximation guarantees.

There are several natural directions for future work. Perhaps most directly, the LP-rounding based algorithm only satisfies the budget constraint in expectation; indeed, we have shown a large integrality gap for the LP. It is a natural question whether a $(1 - 1/e)$-approximation guarantee can be obtained while always satisfying the budget constraints, and without invoking the heavy Continuous Greedy machinery discussed in Section 3.

An interesting empirical study would be to evaluate to what extent influential nodes in one advertiser’s network are also influential in other advertisers’ networks. Within typical social networks, one would expect significant differences based on individuals’ expertise; on the other hand, the use of celebrities to endorse products entirely outside their realm of expertise shows that humans appear willing to project expertise in one area on other areas as well.

Acknowledgments

We would like to thank Ajitesh Srivastava for helpful technical discussions. David Kempe was supported in part by NSF Grant 1619458 and ARO MURI grant 72924-NS-MUR.

References

[1] Facebook reach ads. https://www.facebook.com/business/help/218841515201583.

[2] Hyperlocal advertising. https://www.wordstream.com/blog/ws/2018/01/25/hyperlocal-marketing, 2018.

[3] Cigdem Aslay, Wei Lu, Francesco Bonchi, Amit Goyal, and Laks VS Lakshmanan. Viral marketing meets social advertising: Ad allocation with minimum regret. *Proc. VLDB Endowment*, 8(7):814–825, 2015.

[4] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proc. 25th ACM-SIAM Symp. on Discrete Algorithms*, pages 946–957, 2014.

[5] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *Proc. 12th Intel. Conf. on Integer Programming and Combinatorial Optimization*, pages 182–196, 2007.

[6] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[7] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proc. 43rd ACM Symp. on Theory of Computing*, pages 783–792, 2011.

[8] Wei Chen, Laks V.S. Lakshmanan, and Carlos Castillo. *Information and Influence Propagation in Social Networks.* Synthesis Lectures on Data Management. Morgan & Claypool, 2013.

[9] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[10] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Proc. 10th Intl. Conf. on Data Mining*, pages 88–97, 2010.

[11] Samik Datta, Anirban Majumder, and Nisheeth Shrivastava. Viral marketing for multiple products. In *Proc. 10th Intl. Conf. on Data Mining*, pages 118–127, 2010.

[12] Pedro Domingos and Matthew Richardson. Mining the network value of customers. In *Proc. 7th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[13] Pradeep Dubey, Rahul Garg, and Bernard de Meyer. Competing for customers in a social network: The quasi-linear case. In *Proc. 2nd Workshop on Internet and Network Economics (WINE)*, pages 162–173, 2006.

[14] Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. An analysis of approximations for maximizing submodular set functions — ii. *Mathematical Programming Study*, 8:73–87, 1978.

[15] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006.

[16] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12:211–223, 2001.

[17] Jacob Goldenberg, Barak Libai, and Eitan Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 9:1, 2001.

[18] Sanjeev Goyal and Michael Kearns. Competitive contagion in networks. In *Proc. 44th ACM Symp. on Theory of Computing*, pages 759–774, 2012.

[19] Jason D. Hartline, Vahab S. Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *17th Intl. World Wide Web Conference*, pages 189–198, 2008.

[20] Xinran He and David Kempe. Price of anarchy for the n-player competitive cascade game with submodular activation functions. In *Proc. 9th Conference on Web and Internet Economics (WINE)*, pages 232–248, 2013.

[21] David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of influence in a social network. *Theory of Computing*, 11(4):105–147, 2015.

[22] Jérôme Kunegis. Konect: the koblenz network collection. In *22nd Intl. World Wide Web Conference*, pages 1343–1350, 2013.

[23] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. In *Proc. 12th Intl. Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 244–257, 2009.

[24] Benny Lehmann, Daniel J. Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[25] Shuyang Lin, Qingbo Hu, Fengjiao Wang, and Philip S. Yu. Steering information diffusion dynamically against user attention limitation. In *Proc. 14th Intl. Conf. on Data Mining*, pages 330–339. IEEE Computer Society, 2014.

[26] Wei Lu, Wei Chen, and Laks V. S. Lakshmanan. From competition to complementarity: Comparative influence diffusion and maximization. *Proc. VLDB Endowment*, 9(2):60–71, 2015.

[27] Elchanan Mossel and Sebastien Roch. Submodularity of influence in social networks: From local to global. *SIAM Journal on Computing*, 39(6):2176–2188, 2010.

[28] Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *Proc. 22nd ACM-SIAM Symp. on Discrete Algorithms*, pages 1098–1116, 2011.

[29] James G. Oxley. *Matroid Theory.* Oxford University Press, 1992.

[30] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proc. 34th ACM SIGMOD Intl. Conference on Management of Data*, pages 1539–1554, 2015.

[31] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: near-optimal time complexity meets practical efficiency. In *Proc. 33rd ACM SIGMOD Intl. Conference on Management of Data*, pages 75–86, 2014.

[32] Upstream and YouGov. Digital advertising attitudes report. `http://cache-www.upstreamsystems.com/wp-content/uploads/2014/02/yougov12_report.pdf`, 2012.

[33] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. 40th ACM Symp. on Theory of Computing*, pages 67–74, 2008.

[34] Jan Vondrák. Symmetry and approximability of submodular maximization problems. In *Proc. 50th IEEE Symp. on Foundations of Computer Science*, pages 651–670, 2009.

[35] Chi Wang, Wei Chen, and Yajun Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery Journal*, 25(3):545–576, 2012.

[36] Bryan Wilder, Laura Onasch-Vera, Juliana Hudson, Jose Luna, Nicole Wilson, Robin Petering, Darlene Woo, Milind Tambe, and Eric Rice. End-to-end influence maximization in the field. In *Proc. 17th Intl. Conf. on Autonomous Agents and Multiagent Systems*, 2018.

A Review of Basic Concepts

In this section, we provide definitions and descriptions of several concepts we believe to be standard, but which some readers may want to review.

A.1 Matroids and (truncated) Partition Matroids

Definition 7 (Matroid) *A* matroid *is a non-empty, downward-closed*[16] *set system $(X, \mathcal{I})$, with the following* exchange property*: if $S, S' \in \mathcal{I}$, and $|S| < |S'|$, then there exists some element $x \in S' \setminus S$ such that $S \cup \{x\} \in \mathcal{I}$. The sets in $\mathcal{I}$ are called* independent sets*.*

Definition 8 (Partition Matroid, Truncation) *A partition matroid is defined as follows: Let $X_1, X_2, \ldots, X_p$ be a disjoint partition of X, and $b_1, b_2, \ldots, b_p$ be non-negative integers. A set S is independent (i.e., in $\mathcal{I}$) iff $|S \cap X_i| \leq b_i$ for all i.*

A truncation *of a matroid $(X, \mathcal{I})$ is obtained by replacing $\mathcal{I}$ with $\{S \in \mathcal{I} \mid |S| \leq k\}$, i.e., by discarding all sets of size exceeding k. It is easy to see that any truncation of a matroid is again a matroid.*

A.2 The Independent Cascade Model

The Independent Cascade (IC) Model [17, 16, 21] is defined as follows. (We give the generalization to multiple ads directly here.) For each ad j and (directed) edge $e = (u, v)$, there is a known probability $p_e^{(j)}$ with which u will succeed in activating v. Starting from a seed set $A_0^{(j)}$, in each round t, each newly activated (for ad j) node u can make one attempt to activate each currently inactive (for j) neighbor v. The attempt succeeds with probability $p_{u,v}^{(j)}$, independently of other activation attempts. If at least one activation attempt (for j) on v is successful, v will become active for j at time $t+1$, i.e., become part of $A_{t+1}^{(j)}$. As proved, e.g., in [21], the Independent Cascade Model is a special case of the General Threshold Model from Section 2, by setting $f_v^{(j)}(S) = 1 - \prod_{u \in S}(1 - p_{u,v}^{(j)})$.

A very useful alternative view of the Independent Cascade Model was first shown in [21], and heavily used in subsequent work: generate graphs $G^{(j)}$ by including each edge e in $G^{(j)}$ independently with probability $p_e^{(j)}$. Then, the distribution of nodes activated by ad j in the end, when starting from the set $A_0^{(j)}$, is the same as the distribution of nodes reachable from $A_0^{(j)}$ in the random graph $G^{(j)}$.

B Number of Reverse Reachable Sets vs. Estimation Error

An important part of all algorithms for the optimization problem is being able to evaluate the objective function. Doing so exactly is #P-complete [10, 35]. Good approximations can be obtained by using Reverse Reachable (RR) sets, as described in Section 4. However, the theoretical bounds from Section 4 that *guarantee* a good approximation of the objective, while polynomial, are sufficiently large that algorithms using this many RR sets would not scale well. The goal of this section is to experimentally evaluate how many RR sets yield good estimates of the objective function *in practice*.

[16] That is, $\mathcal{I} \neq \emptyset$, and if $S \in \mathcal{I}$, then $S' \in \mathcal{I}$ whenever $S' \subseteq S$.

To do so, we ran experiments on the four networks from Section 5: *Facebook*, *Advogato*, *NetHEPT* and *DBLP*. For every advertiser, we created independent influence networks as described in Section 5. In our experiments, we varied the number of advertisers m from 1 to 20 and the number of RR sets from mn to $100mn$. For simplicity, we assumed that each advertiser j's value function is simply the (unscaled, and unbudgeted) number of nodes exposed to j's ad.

The main idea is to obtain several independent estimates of the objective function by drawing independent samples of RR sets. When the number of RR sets in each sample is large enough, the estimates should be similar to each other according to different metrics, such as absolute error, variance, etc.

B.1 Absolute Error

In the first set of experiments, we computed reference seed sets $A_0^{(j)}$ by running the Greedy algorithm using $200mn$ RR sets for estimates.[17]

Having computed the seed sets $A_0^{(j)}$, we estimated their payoff (for all advertisers) using different numbers of RR sets. The absolute error of an estimate is the sum, over all advertisers, of the absolute difference of the estimate using the $200mn$ RR sets and the estimate of the newly drawn RR sets.

In Figure 6, we plot the absolute error for all four networks, as we vary the number of RR sets drawn. Each plot was obtained by averaging the absolute error over 100 independent draws of the evaluation RR sets. We observe that for almost all network sizes n and numbers of advertisers m, the error is less than 2% once $\rho^{(j)} \geq 10n$ for all j; at that point, the error curve also nearly flattens out.

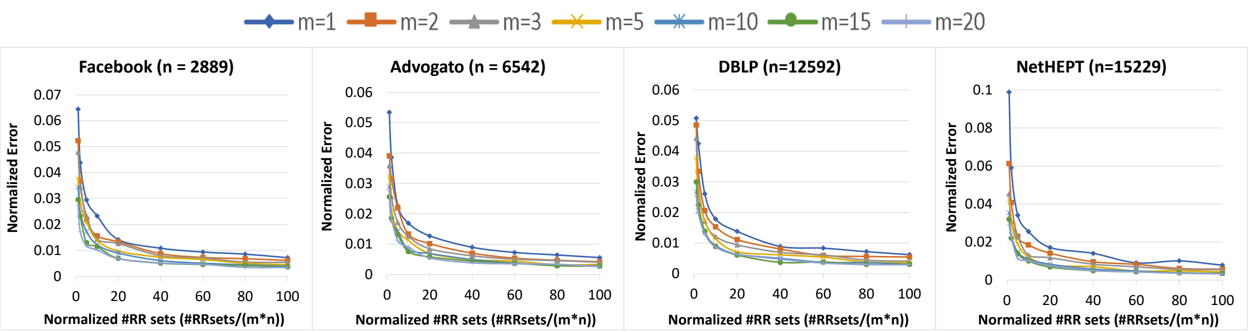

Figure 6: Absolute error in payoff estimation for different numbers of RR sets and advertisers m. The x-axis plots the ratio between the total number of RR sets and mn.

B.2 Standard Deviation

As a second error measure, we considered the standard deviation. The experimental setup was identical to the one in Section B.1. The only difference is that instead of the total absolute error, in Figure 7, we plot the standard deviation of the estimations, normalized by the estimated mean.

The plots are nearly identical to those in Figure 6, confirming that our conclusions about estimation errors are robust to the specific error measure used.

[17] Whether these were actually good seed sets is secondary, as we used them to measure the quality of the *estimation* only. Running the Greedy algorithm ensured that we avoided trivial cases, such as accidentally choosing seed sets with extremely small influence.

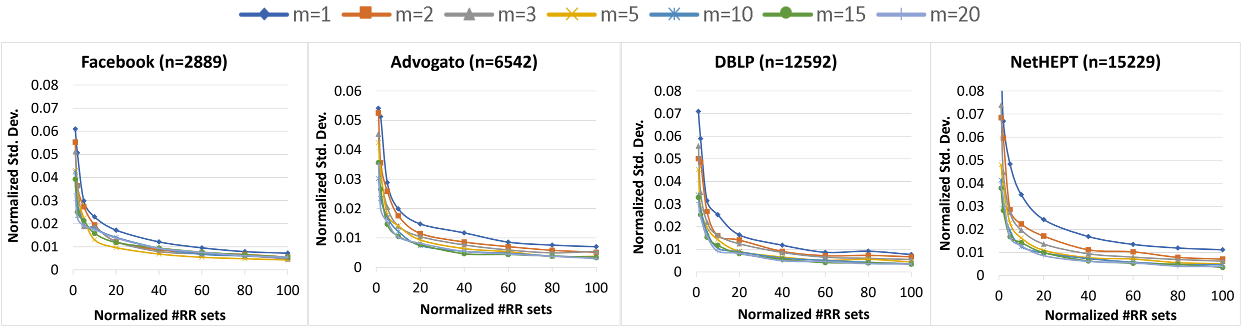

Figure 7: Standard deviation of the estimated payoff, normalized by the mean, for different numbers of RR sets and advertisers m. The x-axis plots the ratio between the total number of RR sets and mn.

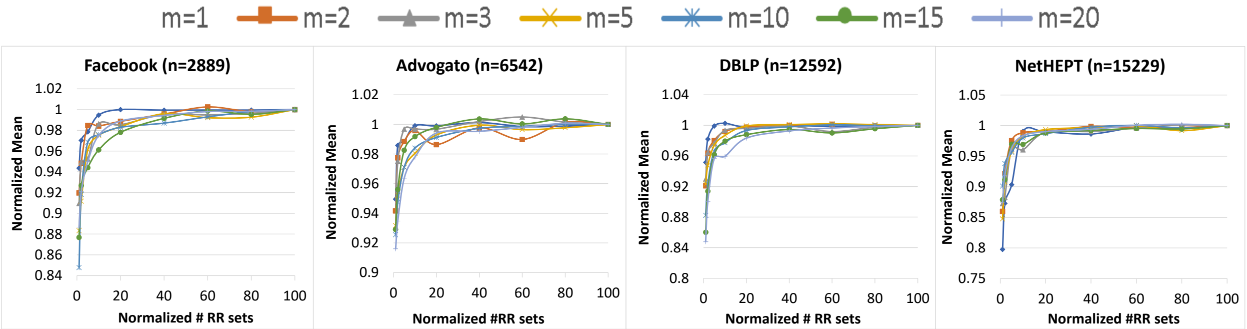

Figure 8: Average payoff of selected seed set normalized by the payoff of seed set obtained with very large number of RR sets ($100mn$). The x-axis plots the ratio of total number of RR sets and $m * n$.

B.3 Impact on Optimization

In the third set of experiments, we analyzed the impact of the number of RR sets on the quality of the seed sets selected by the Greedy algorithm. Unlike in Sections B.2 and B.1, the Greedy algorithm was now run with a limited number of RR samples for estimation. (The number is specified as the number of RR sets per advertiser.) In other words, the Greedy algorithm used estimates of higher variance, and thus was expected to produce suboptimal seed sets. To evaluate the performance, we normalized the objective value against the objective value that a Greedy algorithm using $100mn$ RR sets would obtain.

Figure 8 plots the quality of the Greedy algorithm, normalized by the average payoff. Plots are shown for all four networks and different values of m, and are averaged over 100 runs. We observe that with a small number of RR sets, such as $\rho^{(j)} = n$, the quality of the selected seed sets can be quite poor, and the average payoff obtained can be 10–20% less than what can be achieved using a larger number of RR sets, such as $\rho^{(j)} = 100n$. This is because with small $\rho^{(j)}$, the estimation errors are high; thus, the Greedy algorithm may overestimate the objective value of some seed sets and underestimate the objective value of other seed sets.[18]

For almost all cases, when $\rho^{(j)} = 10n$, the payoff achieved was at least 98% of the payoff with $100n$ RR sets per advertiser. As in Sections B.1 and B.2, the curves nearly flatten out for $\rho^{(j)} \geq 10n$. Hence, we infer that empirically, about $10n$ RR sets per advertiser are sufficient to obtain very accurate estimates of the objective function, and to guarantee that the greedy algorithm performs almost as well as with much larger numbers of RR sets.

[18] Recall that payoffs were estimated with Reverse Reachability sets by computing the fraction of RR sets that contain at least one of the nodes from the seed set.