Social Network Monetization via Sponsored Viral Marketing

Parinya Chalermsook*
Max-Planck-Institut für Informatik, Saarbrücken, Germany
parinya@mpi-inf.mpg.de

Atish Das Sarma
eBay Research Labs, eBay Inc., San Jose, USA
atish.dassarma@gmail.com

Ashwin Lall
Denison University, Granville, Ohio, USA
lalla@denison.edu

Danupon Nanongkai†
KTH Royal Institute of Technology, Sweden
danupon@gmail.com

ABSTRACT

Viral marketing is a powerful tool for online advertising and sales because it exploits the influence people have on one another. While this marketing technique has been beneficial for advertisers, it has not been shown how the social network *providers* such as Facebook and Twitter can benefit from it. In this paper, we initiate the study of *sponsored viral marketing* where a social network provider that has complete knowledge of its network is hired by several advertisers to provide viral marketing. Each advertiser has its own advertising budget and a fixed amount they are willing to pay for each user that adopts their product or shares their ads. The goal of the social network provider is to gain the most revenue from the advertisers. Since the products or ads from different advertisers may compete with each other in getting users' attention, and advertisers pay differently per share and have different budgets, it is very important that the social network providers start the "seeds" of the viral marketing of each product at the right places in order to gain the most benefit.

We study both when advertisers have limited and unlimited budgets. In the unlimited budget setting, we give a *tight* approximation algorithm for the above task: we present a polynomial-time $O(\log n)$-approximation algorithm for maximizing the expected revenue, where n is the number of nodes (i.e., users) in the social network, and show that no polynomial-time $O(\log^{1-\epsilon} n)$-approximation algorithm exists, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{poly}\log n})$. In the limited budget setting, we show that it is hopeless to solve the problem (even approximately): unless $\mathsf{P} = \mathsf{NP}$, there is no polynomial-time $O(n^{1-\epsilon})$-approximation algorithm. We perform experiments on several data sets to compare our provable algorithms to several heuristic baselines.

*Work partially done while at IDSIA, Lugano, Switzerland. Supported by the Swiss National Science Foundation project 200020_144491/1 and by Hasler Foundation Grant 11099.

†Part of this work was done while this author was at Nanyang Technological University, Singapore, and University of Vienna, Austria

SIGMETRICS'15, June 15–19, 2015, Portland, OR, USA.

http://dx.doi.org/10.1145/2745844.2745853.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms

Algorithms, Theory

Keywords

social networks, approximation algorithms

1. INTRODUCTION

Online advertising has become the main source of revenue for many companies which provide free services over the Internet, such as Google Search, Facebook, YouTube and Flickr. Since HotWire sold the first banner ads in 1994, online advertising has evolved into many forms and techniques. Some of these techniques, such as Google's *sponsored search*, have benefited their developers tremendously. In this paper, we propose and study a new form of adversing, which could potentially benefit social network providers such as Facebook and Twitter. We call it *sponsored viral marketing*.

We are motivated by the current situation where big social network providers such as Facebook continue to explore various different approaches for monetization. According to the research company eMarketer[1], the current main source of revenue for Facebook is from the *display ads* market where Facebook can use the demographic information from the profiles of hundreds of millions of its users to target the ads directly to them. However, many recent data and reports (e.g. [17, 18, 33]) suggest that it is not clear how much advantage Facebook has over other internet companies which

[1] http://www.emarketer.com/newsroom/index.php/google-display-ad-leader/

can also collect user information in many other ways, like Google, in this market. For a definitive competitive advantage that goes beyond exploiting just user information, one needs to explore new monetization schemes that better exploit the strength and distinguishing features of social networks.

In this paper, rather than leveraging users' demographic information, we aim at exploiting the *word-of-mouth effects* uniquely provided by social networks: people tend to trust their friends' recommendations more than other advertising channels. Even with all the data that social network providers have about each of their users, social influence can be even more reliable in influencing a person's online purchasing behavior. Much research work has been undertaken recently in finding ways to capitalize on this observation (e.g. [3,5,15,28,29,31,36]). There has also been much effort in exploiting this feature by both social network providers and marketers. For example, Facebook used to have *Beacon* as part of its advertisement system to send data from external websites to its website; e.g., if a user buys a movie ticket on Fandango, Facebook would publish the purchase to her friends, presumably increasing their likelihood of watching the movie as well. More recently, Facebook has introduced *Sponsored Stories Ads*: if a user engages with a business's page, app, or event, the business can pay Facebook to highlight this activity to her friends. While there is some evidence that this type of advertising is more effective than display ads (e.g. [16]), it tends to degrade users' comfort with the social network and has led to several lawsuits as the information is usually published without users' consent.

Another way to exploit the word-of-mouth effect is *viral marketing*. This strategy is used by many advertising companies to exploit existing social networks by encouraging users to share product information with their friends, e.g. by offering a discount for each friend adopting the product. The fact that this strategy lets users share their information at will helps avoid degrading users' experiences. It, however, makes this strategy trickier to use: Since information is not forced to spread, there is a chance that, if the strategy is not implemented well, the product information might not spread as widely as it might have otherwise. This difficulty, which exists even when there is one product, has led to an active area of research namely *influence propagation* (e.g. [3, 5, 15, 28, 29, 31, 36]). The situation becomes much more difficult when there are many independent advertising companies competing for users' attention.

In this paper, we study the opportunity of social network providers to generate revenue from advertising companies who want to employ viral marketing in the presence of competition. The viral marketing services provided by the social network providers give many advantages to advertising companies. For example, the social network providers have much more information about their social networks and thus they can employ viral marketing campaigns more efficiently than the independent companies might. More importantly, social network providers can avoid unnecessary competition since they know information being shared in their networks and can organize several viral marketing campaigns from different advertising companies in a *centralized* manner.

Consider, for example, a situation where a phone company wants to advertise its cell phone on a social network. It is willing to give away some free phones (called *seeds*) to some users who express their interest. The hope is that these users will recommend the phone to their friends who, in turn, might buy the phone and recommend it further. There are many existing algorithms that the company can employ (e.g. [28]). However, the company might not be able to do this efficiently by itself. One reason is that it might not have so much information about the social network. Moreover, the company might not be able to measure the effectiveness of its campaign since it cannot track users' behavior in the social network. Further, there might be many other phone companies who want to do the same task. Since the phone companies know nothing about other companies or their approaches, they might end up giving free phones to the same group of people, leading to unnecessarily fierce competition (since most users will buy at most one phone) and sub-optimal revenue for everyone.

Instead, the phone companies might delegate the task of running their campaigns to the social network itself. For each phone sold in the social network, the company pays the social network provider a *commission* (perhaps based on the profit margin of the company). One clear advantage of this approach is that the provider has a better knowledge of the social network structure; so, it might be able to run the viral marketing campaigns more efficiently. Moreover, the social network provider might encourage users to buy phones through its website (e.g. by giving discounts) so that the effectiveness of the campaigns can be measured. More importantly, the social network provider might be able to help companies avoid unnecessary competition; e.g., it might give phones from different companies to users in different locations or social groups. Similarly, this model can be used for many other commodities, from movie tickets to laptops.

In some situations, the spread of the products or ads might not correspond directly to income of the advertising companies. For example, companies might want to promote their new fan page or spread movie trailers. In these situations, companies may impose *budgets* on their campaigns as the maximum amount of money they will pay the social network provider. Since the social network provider has no way to restrict the spread of the product across the network, the advertising company only agrees to pay per user until its advertising budget has been exhausted. This has the effect of making the problem more interesting because otherwise the advertising company which is willing to pay the most per user may dominate over the other advertising companies. We consider both the cases of limited and unlimited budgets.

In this paper, we initiate the study of how a social network provider can *maximize its expected revenue* in the above situation. To model the spread of the products based on friend recommendations, we use a generalization of the *independent cascade model* that was proposed in [22] and popularized in [28]. Unlike most models in the past that consider only a single advertising company, we assume that there are a number of competing advertising companies. Having multiple advertising companies makes the problem considerably more interesting.

The challenge of this problem is to choose the seeds (e.g., free phones) of different campaigns to the right group of users to ensure that many users will adopt these campaigns while campaigns do not interfere with each others' spread too much. Moreover, since different companies might offer different commissions and impose different budgets, it is important to make sure that campaigns from companies that pay more are spread more. This precludes the possibility of optimizing total revenue by simply maximizing the revenue of each advertising company independently.

Our problem is a generalization of the classic problem considered by Kempe et. al. [28] in that ours reduces to it in the case of a single advertising company. As a result, it is straightforward to see that their NP-hardness proof applies to our problem, and their algorithm applies to our problem for the case of a single advertising company. The most significant generalization in our paper is that our goal is to maximize the social network provider's revenue from multiple competing advertising companies.

The main contributions of this paper are as follows:

- We formulate the expected revenue maximization problem via influence propagation in online social networks from the perspective of the social network provider in the presence of multiple competing advertising entities with varying constraints.
- We show a simple polynomial-time $O(\log n)$-approximation algorithm for this problem in the unlimited budget case, where n is the size of the social network.
- We show that the above algorithm is *almost tight* in the sense that, for any $\epsilon > 0$, there is no polynomial-time algorithm that has an approximation ratio of $O(\log^{1-\epsilon} n)$ unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{poly}\log n})$. Since the latter is highly improbable, our abovementioned algorithm is likely the optimal solution for this problem.
- In the limited budget setting, the problem turns out to be very hard to approximate in the sense that there is no $O(n^{1-\epsilon})$ approximation algorithm for any $\epsilon > 0$, unless $\mathsf{P} = \mathsf{NP}$. Hence, it is very unlikely that a reasonable algorithm exists for this problem.
- We perform extensive experiment evaluation on several data sets. We test our algorithms and heuristics along various dimensions and show detailed comparative benefits of our techniques.

Organization. The rest of this paper is laid out as follows. Section 2 defines the terms and notation used in this paper and formalizes the problem. Our proposed approximation algorithms and their guarantees are presented in Section 3, and the corresponding hardness lower bound results are presented in Section 4. Section 5 shows the efficacy of our algorithms using real social network data. In Section 6 we discuss the related work most relevant to us. Finally, we state our conclusions and discuss future work in Section 7.

2. DEFINITIONS

We model the social network by a weighted directed graph $G = (V, E, w)$ with n nodes. Each node corresponds to a user in the social network, and each directed edge (u, v) with weight $w(u, v) \in [0, 1]$ represents the fact that user u can influence user v with probability $w(u, v)$. The spread of product adoption across the social network follows the independent cascade model [22]. That is, the spreading process works in *rounds* as follows. If node u adopts product i at time t, then each neighbor v of u that has not adopted any product in a previous round puts product i into her *consideration set* at time $t + 1$ with probability $w(u, v)$. Note that there might be multiple copies of i in the consideration set; this could happen when more than one neighbor of v adopts product i in the previous round. User v will pick a product from her consideration set to adopt based on her preference ordering $\preceq_v$ over the products, breaking ties randomly. Since the social network provider does not know $\preceq_v$, our algorithm will not assume this knowledge as well—its existence is assumed to make the model well-defined.

We consider the problem of advertising products from m advertisers on the network. Each advertiser i is defined by a quadruple (k_i, p_i, B_i, S_i) where k_i is the number of products the advertiser is willing to give away for free to "seed" the network, p_i is the commission the advertiser pays for each product adoption, B_i is the total budget for the advertiser, and $S_i \subseteq V(G)$ is the set of nodes that we can seed with the advertiser's product (i.e., the set of users willing to use advertiser i's product for free). Our goal is to place seeds for each advertiser i on k_i nodes so as to maximize the expected revenue of the social network owner, as explained below. We note that many seeds could be put on a single node; however, only one seed (i.e. the one that the node prefers the most) will be adopted by such a node. (For example, if we give two phones to a user, she will likely to use only one of them.)

Suppose $\mathcal{T} = \{T_i \subseteq S_i \mid |T_i| \leq k_i\}_{i=1}^m$ is a valid solution to the problem $(G, \mathcal{C})$ where $\mathcal{C} = \{C_i\}_{i=1}^m = \{(k_i, p_i, B_i, S_i)\}_{i=1}^m$. The expected revenue associated with this solution is computed as follows. For each product i, let n_i be a random variable denoting the number of nodes that adopts i after the spreading process is complete. Then, the expected revenue of the social network is

$$\text{rev}(G, \mathcal{C}, \mathcal{T}) = E[\sum_{i=1}^m \min(p_i n_i, B_i)] = \sum_{i=1}^m E[\min(p_i n_i, B_i)].$$

Our objective, given the problem input $(G, \mathcal{C})$, is to identify seeds of nodes in such a way that this expected revenue is maximized. More precisely, we want to compute a set of seeds whose expected revenue is as close as possible to $\mathsf{OPT}(G, \mathcal{C})$, the maximum value of $\text{rev}(G, \mathcal{C}, \mathcal{T}')$ among all possible $\mathcal{T}' = \{T'_i \subseteq S_i \mid |T'_i| \leq k_i\}_{i=1}^m$.

3. APPROXIMATION ALGORITHM FOR UNLIMITED BUDGET SETTING

In this section we present an $O(\log n)$-approximation algorithm for the sponsored viral marketing problem when each company C_i has unlimited budget $B_i = \infty$. Notice that this is still an interesting problem since only the budget is removed and the objective is still to maximize the expected revenue which operates on the *commission* values of p_i. Our algorithm crucially relies on the fact that the *uniform commission* case of the problem, i.e. when all companies offer the same commission per user (p_i), can be approximated within a constant factor. This is done by adapting an algorithm for the *submodular function maximization problem with a matroid constraint* [9, 19] to this setting. We will explain this in Section 3.1. Then, in Section 3.2, we will show how to transform the general case (i.e., non-uniform commission) to the uniform commission case, paying an extra cost of $O(\log n)$ in the approximation factor.

3.1 Uniform Commission Case

First, let us formally define the instances of the *uniform commission* case of the problem. We say that an instance $(G, \mathcal{C})$ is uniform if there is a number p' such that for all company $C_i \in \mathcal{C}$, we have $p_i = p'$. We show later that in this case, the revenue function $\text{rev}(G, \mathcal{C}, \cdot)$ turns out to be a submodular function with respect to the solution set $\mathcal{T}$, thus allowing us to use the theory of submodular optimization to solve the problem.

Maximizing a Monotone Submodular Function subject to a Partition Matroid Constraint. Consider the following problem. We are given a ground set X of n elements and a function $f : 2^X \to \mathbb{R}_+$ that is *monotone non-decreasing submodular* where we say that f is *submodular* if $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ for all $A, B \subseteq X$ and *monotone non-decreasing* if $f(A) \leq f(B)$ for all $A \subseteq B$. Since describing f explicitly (i.e. values of $f(S)$ for all possible sets $S \subseteq X$) can be computationally expensive, we are interested in the *approximate value oracle* model; i.e., there is a polynomial-time algorithm (called an oracle) with parameter $\gamma \geq 1$ that takes a set $S \subseteq X$ as an input and computes the function $\tilde{f}(S)$ such that $f(S)/\gamma \leq \tilde{f}(S) \leq \gamma f(S)$. In the *monotone submodular function maximization problem*, we are given an integer c and want to find a set $S \subseteq X$ such that $|S| \leq c$ and $f(S)$ is maximized.

In this paper, we use the generalization of the above problem where we have an additional restriction on the set S we want to find. This restriction is called the *partition matroid constraint*. In this case, X is partitioned into ℓ disjoint sets $X_1, X_2, \ldots, X_\ell$ where each X_i has associated integer c_i, so the problem instance can be specified by a triple $(X = \bigcup_{i=1}^m X_i, \{c_i\}_{i=1}^m, f)$. Our goal is to find a set $S \subseteq X$ such that $|S \cap X_i| \leq c_i$, for all i, and $f(S)$ is maximized. Observe that this problem generalizes the previous problem since the previous problem corresponds to the setting where $\ell = 1$, $X_1 = X$ and $c_1 = c$.

The problem of maximizing a monotone submodular function subject to a partition matroid constraint is a special case of the problem of maximizing a monotone submodular function subject to an *arbitrary* matroid constraint[2]. The latter problem has been extensively studied, and many constant-approximation algorithms have been previously discovered (e.g. [9,19]). In particular, the tight $\frac{1}{1-1/e}$-approximation algorithm based on pipage rounding and continuous greedy process techniques was presented by Calinescu et al. [10]. Another algorithm which is simpler and computationally more efficient is the greedy 2-approximation algorithm by Fisher, Nemhauser, and Wolsey [19]. We will use the latter algorithm in this paper due to its simplicity and efficiency. This algorithm simply starts with the empty set, and repeatedly adds an element that maximizes marginal gain, while maintaining the partition matroid constraint as an invariant. See Algorithm 1 for details.

Fisher et al. [19] analyzed the greedy algorithm when $\tilde{f}(S) = f(S)$ and showed that this algorithm is a 2-approximation algorithm. Calinescu et al. [9] analyzed the algorithm when $\tilde{f}$ is an approximate oracle (more importantly, they generalized the result of [19] to *p-systems*). We need an approximate-value-oracle version of this result which follows from [9], as stated below.

Lemma 1. *([9, 19]) For a non-negative, monotone submodular function f, let S be a set obtained from Algorithm 1. Let $\tilde{f}$ and $\gamma \geq 1$ be such that $f(S)/\gamma \leq \tilde{f}(S) \leq \gamma f(S)$ for all $S \subseteq X$. Let S^* be a set that maximizes the value of f over all sets satisfying the partition matroid constraint. Then, $f(S) \geq f(S^*)/2\gamma$; in other words, set S gives a 2γ-approximation.*

[2] We do not define the general matroid constraint here, since it is more complicated and not needed in this paper.

Algorithm 1 Greedy 2γ-approximation algorithm for submodular maximization subject to a partition matroid constraint

Input: Ground set $X = \bigcup_i X_i$, constraints $\{c_i\}$ and an oracle computing function $\tilde{f}$ where $\tilde{f}(S) \in (\frac{1}{\gamma} f(S), \gamma f(S))$ for all $S \subseteq X$

Output: Set $S \subseteq X$ such that $f(S) \geq (\max_{S'} f(S'))/(2\gamma)$.

```
S = ∅, A = ∅.
repeat
    A = {e | ∀i : |(S ∪ {e}) ∩ X_i| ≤ c_i}
    if A ≠ ∅ then
        e = arg max_{e'∈A} f̃(S ∪ {e'})
        S = S ∪ {e}
    end if
until A = ∅
return S
```

Algorithm for Uniform Commission Case. We convert an instance $(G, \mathcal{C})$ of our sponsored viral marketing problem into instance $(X, \{c_i\}_{i=1}^m, f)$ of the function maximization problem as follows. For each company $C_i \in \mathcal{C}$, we create $X_i = \{(i, u) \mid u \in S_i\}$ and let $c_i = k_i$. We then let the ground set be $X = \bigcup_i X_i$. Consider a function $f : 2^X \to \mathbb{R}_+$ where $f(S) = \text{rev}(G, \mathcal{C}, \mathcal{T})$ where $\mathcal{T} = \{T_i\}_{i=1}^m$ is defined by $T_i = \{u : (i, u) \in S\}$. In other words, $f(S)$ captures the expected revenue we get when we place seeds of company C_i on all nodes u such that $(i, u) \in S$. Also note that a set S satisfies the partition matroid constraint (i.e., $|S \cap X_i| \leq c_i$ for all i) if and only if the corresponding set $\mathcal{T}$ satisfies the constraint of the sponsored viral marketing problem (i.e., we put at most k_i seeds of company i on nodes in S_i). This leads us to the following simple observation.

Observation 2. $\mathsf{OPT}(G, \mathcal{C}) = \max_{S:(\forall i)|S \cap X_i| \leq c_i} f(S)$. *Moreover, any set $S \subseteq X$ that satisfies the partition matroid constraint can be transformed into a solution $\mathcal{T}$ of the sponsored viral marketing problem such that $f(X) = \text{rev}(G, \mathcal{C}, \mathcal{T})$.*

Thus, if we can compute an α-approximate solution to the problem of maximizing $f(S)$ with the partition matroid constraint above, we will immediately get an α-approximate solution for the sponsored viral marketing problem as well. The key in solving this function maximization problem is the fact that our function f is monotone non-decreasing submodular.

To show this, observe that since every company has the same commission, say p, and an unlimited budget, the revenue we get depends solely on the number of nodes that adopt *any* product; it does not matter so much (in terms of the revenue we get) which product each node adopts in the end. To be precise, let $\sigma(S)$ be the expected number of nodes adopting *any* product when we put the seeds on nodes in $\bigcup_{T_i \in \mathcal{T}} T_i$ where $\mathcal{T}$ is as defined above. Then, the expected revenue that we will get is $f(S) = p\sigma(S)$. Thus, in order to show that function f is monotone non-decreasing submodular, it is enough to show that the function σ is monotone non-decreasing submodular. Note that function σ is nothing but the expected number of nodes that adopt some product when we place seeds on nodes in $\bigcup_{T_i \in \mathcal{T}} T_i$ (it does not matter which companies these seeds belong to). This is the quantity we want to maximize in the problem of maximizing the spread of influence through a social network, which is already shown to be monotone non-decreasing submodular [28]! This allows us to conclude that f is submodular as well.

The above observation also allows us to construct an oracle for computing f: As done in [28], we put seeds of a single product on nodes in $\bigcup_{T_i \in \mathcal{T}} T_i$ and simulate the diffusion process to count the number of nodes that adopt the product. By repeating this polynomial number of times, we are able to obtain $\tilde{\sigma}(S)$ that is an arbitrarily close approximation to $\sigma(S)$, with high probability. Using $\tilde{f}(S) = p\tilde{\sigma}(S)$, we get $\tilde{f}(S)$ that is an arbitrarily close approximation to $f(S)$ as well. Then, we can use Algorithm 1 to obtain a 2γ-approximation of the optimal solution of the sponsored viral marketing problem in the case of uniform commission.

3.2 General Case

Algorithm 2 $O(\log n)$-Approximation Algorithm

Input: Social network G with unknown user preferences (over products) and set $\mathcal{C} = \{C_i\}_{i=1}^m$ of m companies where each company C_i is a quadruple (k_i, p_i, B_i, S_i).

Output: Set of seeds $\mathcal{T} = \{T_i \subset S_i \mid |T_i| \leq k_i\}$ such that $\mathrm{rev}(G, \mathcal{C}, \mathcal{T}) = \Omega(\mathsf{OPT}(G, \mathcal{C})/\log n)$.

1: For any i, define $p'_i = 2^j$ where j is such that $2^j \leq p_i < 2^{j+1}$. Let $C'_i = (k_i, p'_i, B_i, S_i)$ and $\mathcal{C}' = \{C'_i\}_{i=1}^m$.
2: For all j, let $\mathcal{C}'_j = \{C'_i \mid p'_i = 2^j\}$.
3: For all j, compute the solution $\mathcal{T}'_j$ of the instance $(G, \mathcal{C}'_j)$ (using an arbitrary user preference) such that $\mathrm{rev}(G, \mathcal{C}'_j, \mathcal{T}'_j) = \Omega(\mathsf{OPT}(G, \mathcal{C}'_j))$ using the algorithm for the uniform commission case presented in Section 3.1.
4: **return** $\mathcal{T}'_j$ with maximum value of $\mathrm{rev}(G, \mathcal{C}'_j, \mathcal{T}'_j)$ among all possible j.

Our simple $O(\log n)$ approximation algorithm is outlined in Algorithm 2. The algorithm first rounds each company's commission down to the nearest power of two and partitions them into groups based on their approximate commissions: For each company $C_i \in \mathcal{C}$ with commission p_i, the algorithm rounds p_i down to p'_i as shown in Line 1. We use C'_i to denote the same company C_i with the rounded commission p'_i. Then, the algorithm partitions $\mathcal{C}'$ into $\mathcal{C}'_j$ as in Line 2. Observe that all companies in $\mathcal{C}'_j$, for all j, have the same commission. This allows us to compute $\mathrm{rev}(G, \mathcal{C}'_j, \mathcal{T}'_j)$ for all j using the algorithm for the uniform commission case, as shown in Line 3. We then simply return the solution that gives us the most revenue. Note that our algorithm can be made slightly more efficient by considering only j between $j^* - 2\log n$ and j^* where $j^* = \lfloor \log_2(\max_i p'_i) \rfloor$. This fact will be clear in the analysis. It might seem counterintuitive that our algorithm only focuses on collecting the revenue from one commission group, but we show in the next section that this algorithm gives essentially the best possible approximation guarantee (so trying to collect revenues from different commission groups will not help.) We now analyze the algorithm.

Theorem 3. *Algorithm 2 is returns a solution $\mathcal{T}'_j$ such that $\mathrm{rev}(G, \mathcal{C}', \mathcal{T}'_j) = \Omega(\mathsf{OPT}(G, \mathcal{C})/\log n)$. In other words, it is $O(\log n)$-approximation.*

The theorem states that the seeds assigned by the algorithm result in an expected revenue that is at least a $\Omega(\frac{1}{\log n})$ fraction of the expected revenue of an *optimal* placement of seeds for all advertisers. This essentially suggests that the presented algorithm, while very simple, efficient and easy to implement, is actually very good at choosing from exponentially many possibilities of seeds for every advertiser and make assignments in a manner that helps maximize the social networks' revenue. A good comparison is to observe that if the advertisers operated independently, this could potentially result in a hugely negative competition that not only affected their own revenues but resulted in an arbitrarily large (depending on the number of advertisers) factor of loss to the social network. In the subsequent section, we will also show that our algorihm in fact performs *best possible* from a computational complexity standpoint.

The rest of this section is devoted to proving Theorem 3. First, we note that changing the commission of each company C_i from p_i to p'_i does not change the revenue we get from any solution too much. This is simply because $p'_i \geq p_i/2$ for all i. We thus have the following lemma.

Lemma 4. *For any solution $\mathcal{T}$ of the sponsored viral marketing problem,* $\mathrm{rev}(G, \mathcal{C}, \mathcal{T}) \geq \mathrm{rev}(G, \mathcal{C}', \mathcal{T}) \geq \mathrm{rev}(G, \mathcal{C}, \mathcal{T})/2$. *Consequently,* $\mathsf{OPT}(G, \mathcal{C}, f) \geq \mathsf{OPT}(G, \mathcal{C}', f) \geq \mathsf{OPT}(G, \mathcal{C}, f)/2$.

Let j^* be the minimum integer j such that $p'_i \leq 2^j$ for all i, i.e., $j^* = \log_2(\max_i p'_i)$. Let $\mathcal{C}'' = \cup_{j^* - 2\log n \leq j \leq j^*} \mathcal{C}'_j$; i.e., $\mathcal{C}''$ contains only the companies having "large" commission and ignores the rest of the companies. We show this does not lose us too much revenue.

Lemma 5. $\mathsf{OPT}(G, \mathcal{C}'') \geq \mathsf{OPT}(G, \mathcal{C}')/2$

Proof. Let $p'_{\max} = 2^{j^*}$ (or, equivalently, $p'_{\max} = \max_i p'_i$) and $w_{\max} = \max_{e \in E(G)} w(e)$. Note that

$$\mathsf{OPT}(G, \mathcal{C}') \geq p'_{\max} w_{\max}$$

since we can get a revenue of $p'_{\max} w_{\max}$ by placing a seed of the company with commission $p'_{\max}$ on a node u where u is such that there is an edge uv with $w(uv) = w_{\max}$.

Note further that the revenue we can get from any company in $\mathcal{C}' \setminus \mathcal{C}''$ when we place a seed on a node u is at most $\sum_{v \in V(G)} w(uv) \frac{p'_{\max}}{2n^2} \leq \frac{\mathsf{OPT}(G, \mathcal{C})}{2n}$ since every company in $\mathcal{C}' \setminus \mathcal{C}''$ has commission at most

$$p'_{\max}/2^{2\log n + 1} = p_{\max}/2n^2.$$

Thus, the expected total revenue we get from companies in $\mathcal{C}' \setminus \mathcal{C}''$ is at most $\mathsf{OPT}(G, \mathcal{C}')/2$. In other words,

$$\mathsf{OPT}(G, \mathcal{C}') - \mathsf{OPT}(G, \mathcal{C}'') \leq \mathsf{OPT}(G, \mathcal{C}')/2.$$

The lemma follows. □

Finally, we use the fact that $\mathcal{C}''$ consists of companies having $O(\log n)$ different values of commission to show the following simple lemma.

Lemma 6. $\max_{j^* - 2\log n \leq j \leq j^*} \mathsf{OPT}(G, \mathcal{C}'_j) = \Omega(\frac{\mathsf{OPT}(G, \mathcal{C}'')}{\log n})$.

Proof. Since $\mathcal{C}'' = \bigcup_{j^* - 2\log n \leq j \leq j^*} \mathcal{C}'_j$, we have that

$$\mathsf{OPT}(G, \mathcal{C}'') \leq \sum_{j^* - 2\log n \leq j \leq j^*} \mathsf{OPT}(G, \mathcal{C}'_j).$$

Thus, there exists j such that $\mathsf{OPT}(G, \mathcal{C}'_j) \geq \frac{\mathsf{OPT}(G, \mathcal{C}'')}{2\log n + 1}$. □

Combining all above lemmas, we have that (recall that $\mathcal{T}'_j$ is the solution returned by Algorithm 2)

$$\begin{aligned}\mathrm{rev}(G,\mathcal{C},\mathcal{T}'_j) &\geq \mathrm{rev}(G,\mathcal{C}',\mathcal{T}'_j)\\ &= \Omega\left(\max_{j^*-2\log n\leq j\leq j^*} \mathsf{OPT}(G,\mathcal{C}'_j)\right)\\ &= \Omega\left(\frac{\mathsf{OPT}(G,\mathcal{C}'')}{\log n}\right)\\ &= \Omega\left(\frac{\mathsf{OPT}(G,\mathcal{C}')}{\log n}\right)\\ &= \Omega\left(\frac{\mathsf{OPT}(G,\mathcal{C})}{\log n}\right).\end{aligned}$$

This completes the proof of Theorem 3.

4. HARDNESS OF APPROXIMATION

In this section, we prove our hardness results. The $\Omega(n^{1-\epsilon})$ hardness of approximating the limited-budget case is proven in Section 4.1. We prove $\Omega(\log^{1-\epsilon} n)$ hardness for the unlimited budget setting in Section 4.2.

4.1 Limited Budget Setting

Our reduction starts from the Donation Center Location problem (DCL), introduced by Huang and Svitkina [27]. In DCL, we are given a bipartite graph $G = (A \cup L, E)$ where A is a collection of *agents* and L is a collection of *centers*. $V(G)$ is the number of vertices in G. Each center $\ell \in L$ has limited capacity $c_\ell \in \mathbb{Z}_+$ that represents the maximum number of agents that can be served, and each vertex $a \in A$ has strictly ordered preference ranking $\preceq_a$ of its neighbors in L. Our objective is to compute a triple (A', L', σ) where $L' \subseteq L$ is a set of centers to open, $A' \subseteq A$, and assignment $\sigma : A' \to L'$ such that: (1) The number of agents assigned to any center ℓ is at most c_ℓ, i.e. $|\{a \in A' : \sigma(a) = \ell\}| \leq c_\ell$ and (2) Each $a \in A'$ is assigned to its highest ranked neighbor in L'. We are interested in finding such a triple while maximizing $|A'|$. Our starting point is the following hardness of approximation result due to Chalermsook, Laekhanukit, and Nanongkai [12].

THEOREM 7. *[12] For any positive number $\epsilon > 0$, unless $\mathsf{NP} = \mathsf{ZPP}$, it is hard to approximate* DCL *to within a factor of $|V(G)|^{1-\epsilon}$.*

Now we show an approximation-preserving reduction from DCL to the sponsored viral marketing problem, i.e. given an instance of DCL, we construct an instance of sponsored viral marketing problem such that the value of the optimal solution in the latter problem is the same as that of the former.

Construction. Given an instance $G = (A \cup L, E)$ of DCL, we create an instance $(G', \mathcal{C}' = \{(k_i, p_i, B_i, S_i)\}_{i=1}^m)$ as follows. The graph G' is a directed bipartite graph $G' = (A \cup L, E')$ on the same sets of vertices and edges. The only difference is that we turn the graph into directed graph: There is a directed edge $\ell a \in E'$ if and only if $\ell a \in E$. The weight of every edge is $w_{\ell a} = 1$. For each $\ell \in L$, we have a company $\mathsf{comp}(\ell)$ with $k_{\mathsf{comp}(\ell)} = 1$, $p_{\mathsf{comp}(\ell)} = 1$, $B_{\mathsf{comp}(\ell)} = c_\ell$, and $S_{\mathsf{comp}(\ell)} = \{\ell\}$. Each node $a \in A$ has the preference $\preceq'_a$ over products $\{\mathsf{comp}(\ell)\}_{\ell\in L}$ in a way that is consistent with its ranking over ℓ, i.e. $\mathsf{comp}(\ell) \preceq'_a \mathsf{comp}(\ell')$ if and only if $\ell \preceq_a \ell'$. This completes the description of our reduction.

Analysis. Let OPT and OPT' denote the optimal revenue of the DCL instance and our instance respectively. We argue that $\mathsf{OPT} = \mathsf{OPT}'$. First, to show that $\mathsf{OPT} \leq \mathsf{OPT}'$, consider any solution (L', A', σ) of DCL. For each node $\ell \in L'$, we put the seed of company $\mathsf{comp}(\ell)$ at node ℓ. We argue that the total profit made from this strategy is $|A'|$: Consider each node $a \in A'$ who sees the seeds at nodes $\{\ell : \ell a \in E'\}$. It must be the case that $\mathsf{comp}(\sigma(a))$ is among the products a considers and must be highest w.r.t. the rank $\preceq'_a$, so we can get the revenue of 1 from a. It is easy to check that each company has enough budget, since for each ℓ, we know that $|\{a : \sigma(a) = \ell\}| \leq c_\ell$.

Now we argue that $\mathsf{OPT}' \leq \mathsf{OPT}$. Let $L' \subseteq L$ be a subset of vertices on which the seeds are placed and $A' \subseteq A$ be the set of nodes that contribute to the total profit. Define $\sigma : A' \to L'$ by setting $\sigma(a)$ to be the node $\ell \in L'$ such that product $\mathsf{comp}(\ell)$ is bought by a. It is easy to check (using the arguments similar to that in the previous paragraph) that (L', A', σ) is a feasible solution for DCL instance, thus implying that $\mathsf{OPT}' \leq \mathsf{OPT}$. We have thus shown the following theorem.

THEOREM 8. *Let $\epsilon > 0$ be sufficiently small constant. Unless $\mathsf{NP} = \mathsf{ZPP}$, there is no $n^{1-\epsilon}$ approximation algorithm for sponsored viral marketing problem in the limited budget setting.*

This result essentially says that the problem of identifying good seed sets to maximize expected revenue for the social network is computationally intractable when advertisers operate under limited budgets. This still does not preclude, from a practical standpoint, an algorithm that conceivably does very well in practice in allocating seeds. Also notice that while we prove the computational intractability of the problem from the social network provider's revenue maximization standpoint, it is important to note that (and perhaps can be deferred to future for a concrete result in this direction) the social network provider presumably can do a significantly better job of maximize profit than individual advertisers may do independently. Our objective compares against the optimal solution from the network's standpoint which is a stringent comparison. More likely if independent advertisers operate in isolation in marketing currently, they are resulting in unnecessarily revenue-detracting competition.

4.2 Unlimited Budget Setting

To prove the hardness of approximation in this setting, we make a connection between our problem and UNIT-DEMAND MIN-BUYING PRICING (UDP-MIN). In the UDP-MIN problem, we are given a collection of items $\mathcal{I}$ and consumers $\mathcal{C}$, where each consumer $C \in \mathcal{C}$ is associated with budget B_C and a subset of items $S_C \subseteq \mathcal{I}$ she is interested in purchasing. Once the price function $p : \mathcal{I} \to \mathbb{R}_+$ is fixed, each consumer C buys the cheapest item in S_C if $\min_{I\in S_C} p(I) \leq B_C$ (in which case, we earn a revenue of $\min_{I\in S_C} p(I)$ from consumer C); otherwise, the consumer buys nothing (in which case, our revenue from this consumer is zero). The objective of the problem is to compute the price function that maximizes the total revenue.

We remark the distinction in the terms we use. If we talk about the items in the pricing problem, we refer to them as *items*, while we refer to those in our problem as *products/ companies*. We need the following theorem which shows the hardness of approximation for UDP-MIN.

THEOREM 9. *[11] For any $\epsilon > 0$, it is hard to approximate* UDP-MIN *to within a factor of* $\log^{1-\epsilon}(|\mathcal{C}| + |\mathcal{I}|)$*, unless* NP *has a quasi-polynomial time algorithm.*

Construction. Consider an instance $(\mathcal{C}, \mathcal{I})$ of UDP-MIN. Denote by $\mathcal{B} = \{B_C\}_{C\in\mathcal{C}}$ the set of all possible consumers' budgets. We create an instance $(G', \mathcal{C}' = \{C'_i\}_{i=1}^m)$ as follows. First the graph $G' = (U \cup V, E)$ is bipartite. Edges are always directed from vertices in U to those in V. For each item $I \in \mathcal{I}$, for each possible budget $B \in \mathcal{B}$, we have a vertex $u(I, B)$ in set U, so the total number of vertices in U is

$$|U| = |\mathcal{B}||\mathcal{I}| \leq |\mathcal{C}||\mathcal{I}|.$$

Now for each consumer $C \in \mathcal{C}$, we have a vertex $v(C)$, so $|V| = |\mathcal{C}|$. There is an edge connecting $u(I, B)v(C) \in E$ if and only if $I \in S_C$ and $B_C \leq B$. Finally, each edge e in the graph has weight $w(e) = 1$. This completes the description of graph G'.

Now we define the companies and their parameters. For each item $I \in \mathcal{I}$, for each possible budget $B \in \mathcal{B}$, we have a company/product $\mathsf{comp}(I, B)$ with the following parameters: the number of seeds $k'_{\mathsf{comp}(I,B)} = 1$, the price $p'_{\mathsf{comp}(I,B)} = B$, and the node set $S_{\mathsf{comp}(I,B)} = \{u(I, B)\}$; recall that the budget of each company is ∞ in this setting. Each node $v(C) \in V$ has the ranking $\preceq_{v(C)}$ of companies $\{\mathsf{comp}(I, B)\}$ based on the values of B, i.e. $\mathsf{comp}(I, B) \preceq_{v(C)} \mathsf{comp}(I', B')$ if and only if $B > B'$. This completes our construction. See Figures 1 and 2 for illustration.

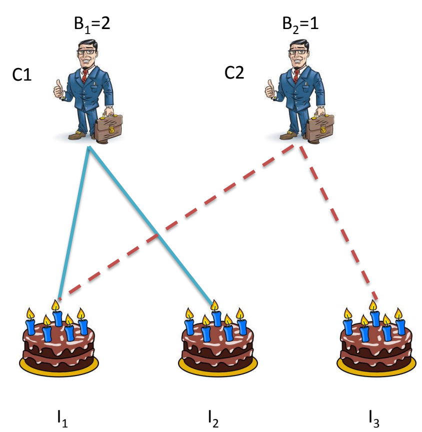


Figure 1: An instance of UDP-MIN. An edge between consumer and item shows that the consumer is interested in that item.

Analysis. Let OPT and OPT$'$ denote the optimal revenue of the pricing instance $(\mathcal{C}, \mathcal{I})$ and our instance $(G', \mathcal{C}')$ respectively. We will argue that OPT $=$ OPT$'$.

To show OPT $\leq$ OPT$'$, consider an optimal price function $p^* : \mathcal{I} \to \mathbb{R}_+$. Observe that any optimal price function must have $p^*(I) \in \mathcal{B}$ for all $I \in \mathcal{I}$: Otherwise, suppose that

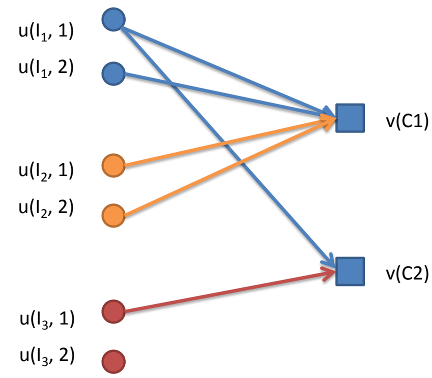


Figure 2: The graph G' obtained from the reduction.

$B' < p^*(I) < B$ for some $B', B \in \mathcal{B}$ that are consecutive in values. We could have increased $p^*(I)$ to B because any consumer C who can afford I will still be able to afford it with the new price. Let $\mathcal{C}^* \subseteq \mathcal{C}$ be the set of consumers who contribute to the value of OPT, i.e. consumers who made purchases. For each consumer $C \in \mathcal{C}^*$, let $\sigma(C) \in S_C$ denote the item with minimum price $p^*(I)$ in S_C. This is the item that consumer C purchases.

Our strategy to collect revenue is as follows: For each $I \in \mathcal{I}$, each company $\mathsf{comp}(I, p^*(I))$ puts its seed at $u(I, p^*(I))$. The seeds of other companies are not placed anywhere. Now consider each node $v(C)$ for $C \in \mathcal{C}^*$ and let $I^* = \sigma(C)$. It suffices to argue that the revenue we get from this node is $p^*(I^*)$. Notice that there is an edge $u(I^*, p^*(I^*))v(C) \in E$, so node $v(C)$ adopts product $\mathsf{comp}(I^*, p^*(I^*))$. Since $I \in S_C$ if and only if $v(C)$ adopts $\mathsf{comp}(I, p(I))$, it must be the case that $\mathsf{comp}(I^*, p^*(I^*))$ is ranked highest w.r.t. $\preceq_{v(C)}$ (because the price $p^*(I^*)$ is lowest). This means that node $v(C)$ picks $\mathsf{comp}(I^*, p^*(I^*))$ and therefore pays the price of $p^*(I^*)$.

Next, we prove that OPT$' \leq$ OPT. Let $\mathcal{C}^* \subseteq \mathcal{C}'$ be the set of companies who placed the seeds in the optimal strategy. We partition $\mathcal{C}^*$ into $\mathcal{C}^* = \bigcup_{I\in\mathcal{I}} \mathcal{C}^*_I$ where $\mathcal{C}^*_I$ contains the companies of the form $\mathsf{comp}(I, B)$ for some B.

CLAIM 10. *For any $I \in \mathcal{I}$, we can assume without loss of generality that $|\mathcal{C}^*_I| \in \{0, 1\}$.*

PROOF. Assume that there are $B, B' \in \mathcal{B}$ such that both $\mathsf{comp}(I, B)$ and $\mathsf{comp}(I, B')$ belong to $\mathcal{C}^*_I$ such that $B < B'$. We argue that no node $v(C)$ buys $u(I, B')$ which allows us to remove the seed at $u(I, B')$ while preserving the revenue. Assume otherwise that some $v(C)$ bought $u(I, B')$. From the construction, there must also be an edge $u(I, B)v(C)$ and $u(I, B') \prec_{v(C)} u(I, B)$. This is impossible because $v(C)$ would have preferred to buy $u(I, B)$ instead. $\square$

The above claim allows us to "recover" the price function p^* for $(\mathcal{C}, \mathcal{I})$ that collects the same revenue. We set the price $p^*(I) = \infty$ if $|\mathcal{C}^*_I| = 0$; otherwise, we set $p^*(I) = B$ where $\mathsf{comp}(I, B) \in \mathcal{C}^*_I$. Now, let $V^* \subseteq V$ be the subset of nodes that contribute to the value of OPT'. For each consumer C such that $v(C) \in V^*$, let $\mathsf{comp}(I^*, B^*)$ be the product chosen by $v(C)$, so node $v(C)$ contributes B^* to the value of OPT. We argue that consumer C also pays the value of B^* with respect to the price p^*. This will conclude the proof.

Assume to the contrary that, in the pricing instance $(\mathcal{C}, \mathcal{I})$, consumer C buys some item $\tilde{I}$ with $p^*(\tilde{I}) < B^*$. This implies that $u(\tilde{I}, p^*(\tilde{I}))v(C) \in E$, and that $\mathsf{comp}(I^*, B^*) \prec_{v(C)} \mathsf{comp}(\tilde{I}, p^*(\tilde{I}))$. Hence, $\mathsf{comp}(\tilde{I}, p^*(\tilde{I}))$ would have been bought instead of $\mathsf{comp}(I^*, B^*)$.

Gap Analysis. Notice that the size of the instance is $|V(G')| \le |\mathcal{C}||\mathcal{I}|$, so the hardness gap $\log^{1-\epsilon}(|\mathcal{C}| + |\mathcal{I}|)$ translates to $\Omega(\log^{1-\epsilon}|V(G')|)$ as desired. We conclude by stating our result formally.

THEOREM 11. *Let $\epsilon > 0$ be any sufficiently small constant. Then, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\mathrm{poly}\log n})$, there is no $\log^{1-\epsilon} n$ approximation algorithm for sponsored viral marketing problem even when each advitiser has unlimited budget.*

Notice that this result proves the *near*-optimality of our algorithm presented in the previous section.

5. EXPERIMENTAL EVALUATION

We experimentally tested our algorithms against several heuristic baselines via simulations. In addition to Algorithm 2 from Section 3, we introduce a heuristic extension of this algorithm. The point of this section is to empirically show that a coordinated strategy by the social network provider will always outperform uncoordinated advertisers. All the simulations in this section were run on real social network data. All the code was written in C and run on an Intel Core 2 Duo running Ubuntu 12.04.2.

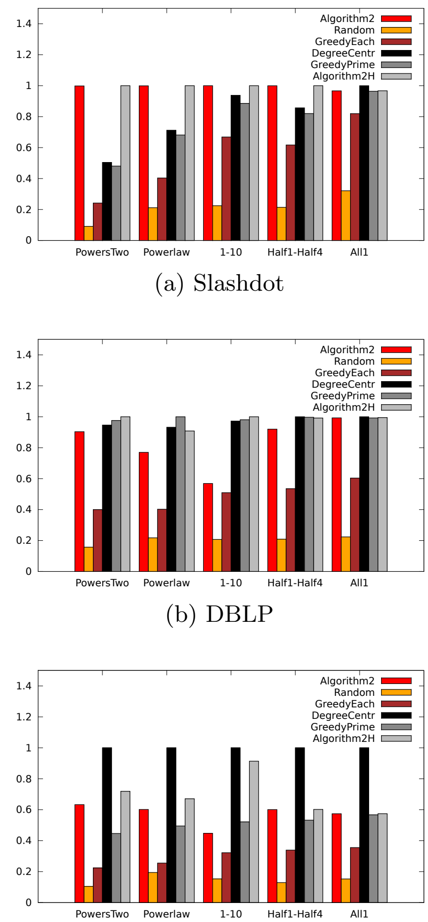
We ran our simulations on social networks made available by the Stanford Network Analysis Project (SNAP)[3]. More specifically, we used a Slasdot user directed network collected in April 2009 (82168 nodes, 948464 edges), a DBLP coauthorship dataset (317080 nodes, 1049866 edges), and a directed Epinions who-trusts-whom social network (75879 nodes, 508837 edges).

Recall that Algorithm 2 identifies the highest tier of commissions and assigns seeds only to them. It is indeed sometimes better to omit the low commission advertisers as they may decrease overall revenue by taking away nodes from higher commission advertisers. In the case that large parts of the network are not influenced (e.g., if the propagation probability is low), however, it is better to include these lower commission advertisers as they do not interfere. Additionally, there are business reasons why they should not be neglected. Motivated by this, we introduce a heuristic version of our algorithm (Algorithm 2H, described below) to avoid this issue.

We compared the following algorithms:

- **Algorithm 2** is the algorithm from Section 3.
- **Random** assigns seeds uniformly at random (included as a baseline).
- **GreedyEach** is the result when each advertiser picks seeds independently of each of the others.

[3]https://snap.stanford.edu



(a) Slashdot

(b) DBLP

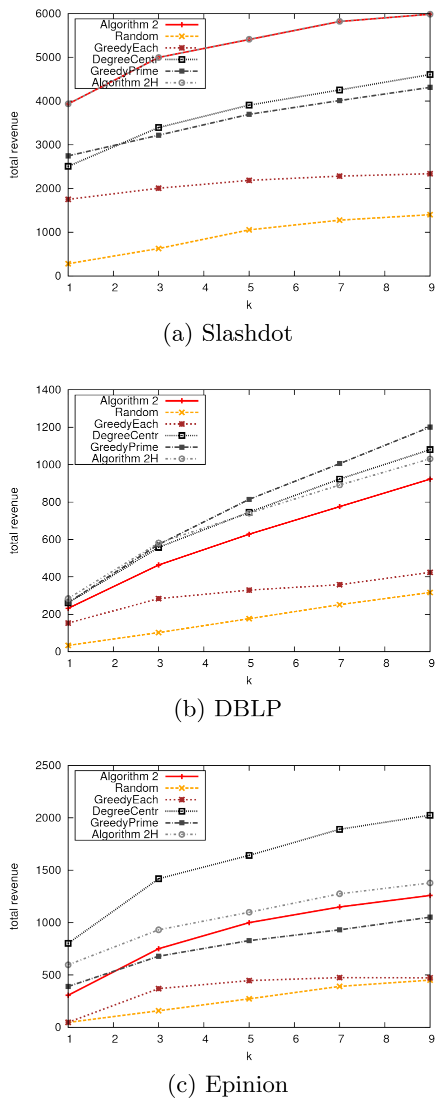
(c) Epinion

Figure 3: Relative performance of algorithms with different commission (p) distributions. (Note: the revenues for each distribution have been normalized by the best algorithm to make all of them fit the same scale.)

- **DegreeCentrality** is a heuristic in which the highest degree nodes are used as seeds. The higher commission advertisers are assigned the higher degree nodes.
- **GreedyPrime** computes the greedy solution for all the advertiser's seeds assuming equal commissions and then assigns these nodes to the advertisers in non-increasing order of commission.
- **Algorithm 2H** is a heuristic version of Algorithm 2. It adds unassigned seeds from Algorithm 2 to the highest degree unassigned nodes and returns the better of this solution and that of Algorithm 2 (thus the performance of Algorithm 2H will never be worse than that of Algorithm 2).

Note that the **GreedyEach** algorithm is the most likely algorithm used by uncoordinated advertisers. We will show that the other algorithms (except Random) outperform this one empirically with many different parameter settings.

In the runs of each of the greedy algorithms, the revenue was estimated using 100 runs of simulation. Using more runs

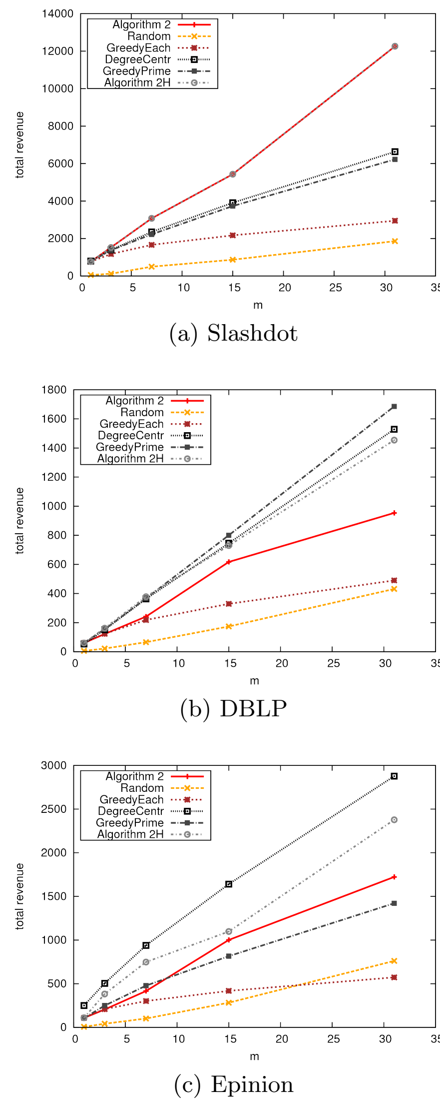


(a) Slashdot

(b) DBLP

(c) Epinion

Figure 4: Varying the number of seeds per advertiser (k) on all data sets



(a) Slashdot

(b) DBLP

(c) Epinion

Figure 5: Varying the number advertiser (m) on all data sets

did not significantly change the results. The final evaluation of each algorithm was performed with 10,000 runs of simulation. All the values in the following graphs are the means of ten independent runs of each algorithm.

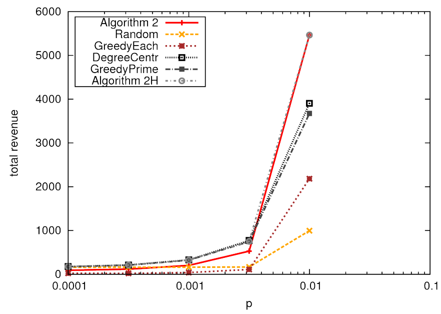
Since our algorithm divides advertisers into tiers by commission, we first studied the effect of the advertiser commission distribution on the performance of each of the above algorithms. In particular, we tried the following distributions:

- PowersTwo: $1, 2^1, 2^2, \ldots, 2^9$
- PowerLaw: $1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 4, 4, 8$
- 1-10: $1, 2, 3, \ldots, 10$
- Half1Half4: 5 1's, 5 4's
- All1: 10 1's

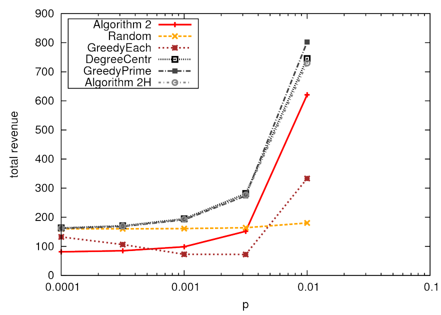
From Figure 3, we can see that Algorithm 2H was best or close to best for all distributions in Slashdot and DBLP. For the Epinions dataset, the degree centrality heuristic was consistently the best. The connectivity of this network is such that using high degree nodes works better for it than the greedy method.

For the remaining experiments, we set default values and varied parameters one-by-one to see the effect this would have on the performance of the algorithms. Unless otherwise stated, all the following results used a uniform propagation probability of $P = 0.01$ (1%), fifteen advertisers with commissions distributed by a power law as described above ($m = 15$), and $k = 5$ seeds per advertiser.

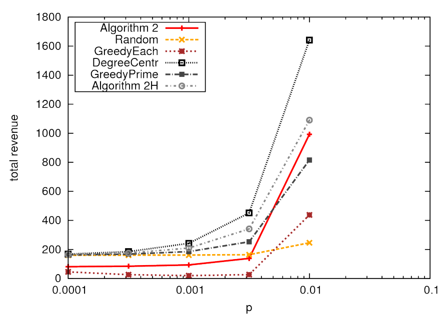
Figure 4 shows the effect of varying the number of seeds per advertiser (k). In the case of the Slashdot data set, Algorithm 2 performs the best. The DBLP dataset shows a case in which the heuristic performs better since it spreads unseeded advertiser on unreached parts of the network. As in the previous figure, the degree centrality heuristic is best on the Epinions data. Similar results can be observed when varying the number of advertisers (m) in Figure 5.



(a) Slashdot



(b) DBLP



(c) Epinion

Figure 6: Varying the propagation probability (P) on all data sets

When we vary the propagation probability in Figure 6, the advantage of Algorithm 2H over Algorithm 2 becomes apparent. For very low probabilities, there are parts of the social network that advertisers are not reaching and it makes sense to seed these with the low commission advertisers. Hence, we see Algorithm 2H outperform Algorithm 2 in most cases. However, when the probability rises to the extent that most of the network is being influenced, Algorithm 2 rapidly catches up.

Note that in all cases our algorithms greatly outperform GreedyEach, the result when all the advertisers greedily assign their seeds in an uncoordinated manner. This demonstrates the advantage of a coordinated strategy by the social network owner over uncoordinated efforts by the advertisers.

6. RELATED WORK

There is a large body of work studying the correlation of activity among friends in online communities; see examples in [20, 32, 34]. Most are forms of diffusion research, built on the premise that user engagement is contagious. As such, a user is more likely to adopt new products or behaviors if her friends do so [4, 29]; and large cascades of behavior can be triggered by the actions of a few individuals [20, 32].

A number of theoretical models have been developed to model influence cascades [14,30]. In the context of social networks, the threshold model [24] and the independent cascade model [22] are widely applied. The seminal work of Kempe et al. [28] includes the general cascade model which forms the basis of our work. There has also been a long line of work on viral marketing starting from [15, 31].

The concept of competition in influence maximization has also been studied extensively in recent years. A paper that adopts a follower's perspective, by Carnes et al. [10], considers an advertiser trying to maximize influence when a competing technology is already present. An extension of the threshold models under the competitive setting was studied by Borodin et al. [7] and another model was proposed by Bharathi et al. [5]. In more recent work, He et al. [26] consider the setting where an advertiser wants to *block* the influence of a competitor. Another recent paper that considers competition and how to defeat it in an adversarial setting is due to Shirazipourazad et al. [35].

Goyal-Kearns [23] study competitive contagion in networks and explore equilibria and price of anarchy. Tzoumas et al. [37] study questions around the existence of pure strategy Nash equilibria by focusing on 2-player competitive diffusion games. More recently Goldberg-Liu [21] provide approximation algorithms for finding the smallest set of nodes that can trigger a cascade that results in every node in the graph adopting the technology.

Datta et al. [13] consider viral marketing with multiple products with a profit maximization setting that is similar to ours. They study the problem of selecting seed nodes for multiple products to maximize the overall influence. Like us, they have a maximum budget for each product and have a certain set of allowed seeds corresponding to each product. Their work differs from ours in two fundamental ways: (1) they consider a non-competitive setting in the sense that each node can get influenced by multiple products and end up adopting more than one; and (2) they enforce a hard constraint on each node on how many different products may choose it as a seed. Notice that our problem is far more general given that we consider a competitive model with individual users having their preferences across products. A very recent work on similar lines is due to Borodin et al. [6]. Their setting is similar to us in that there are competitive agents such as advertisers who want to spread their ads or products on the network, and a centralized social network owner who wishes to optimize total influence (or social welfare). However, they consider the game theoretic problem where each advertiser has a true demand on number of seeds they would like to instantiate and the social network owner want to design mechanisms where each agent is incentivized to reveal their true demand. There have been several other papers as well in general influence maximization type questions and we list some other recent ones here [1, 2, 8, 25].

7. CONCLUSIONS

As several companies fight for the attention of users in the social context to leverage influence spread, competition for the same users on similar products becomes inevitable. A natural question that arises in this framework is how do the social network advertisement managers reconcile all these conflicting objectives from various companies, while maximizing their profits, by taking advantage of users' influence on one another. In this paper, we explore a very generic framework that takes the social network's viewpoint with the goal of revenue maximization. The holistic theoretical study here captures the motivation, algorithmic techniques, as well as the boundaries or limits of this framework. Specifically, in the case of unlimited budgets, we show a polynomial time $O(\log n)$ approximation algorithm for maximizing the expected revenue and further prove that no polynomial time approximation algorithm exists, assuming $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{\text{poly}\log n})$, with a guarantee of $O(\log^{1-\epsilon} n)$. In the limited budget setting, the problem appears intractable as we show that no $O(n^{1-\epsilon})$-approximation algorithm exists under the assumption that $\mathsf{P} \neq \mathsf{NP}$. In terms of experimental evaluation, we tested our techniques against several baseline heuristics and demonstrated that a coordinated strategy by the social network operator will vastly outperform an uncoordinated strategy by the advertisers.

From an overall motivational standpoint, several directions remain to be explored. It would be interesting to extend our work to a setting where advertisers' demands can be admitted online, as often in practice new advertisers, budgets, and products arrive in a continuous manner, without a clean notion of timesteps or rounds. Further, the associated budgets could be changing with time and/or may need to be handled across various data centers. Another orthogonal yet very important consideration is that in a real social network, often one can only learn approximate edge influence probabilities with varying degrees of accuracy, and these may even change with time.

There also remain specific theoretical open questions to be addressed and we mention two here. First, recall that in our case the preference function $\preceq_v$ of each node is random. That is, if there are many products in the consideration set at the same time, each user picks the product in a random fashion. The hardness result however does not apply to this case. Secondly, if the seeds of each company could be placed at any node, the hardness result would not apply. Conceivably there may be a good approximation algorithm even in the limited budget setting.

8. REFERENCES

[1] D. Agrawal, C. Budak, and A. El Abbadi. Information diffusion in social networks: observing and affecting what society cares about. In *CIKM*, pages 2609–2610, 2011.

[2] D. Agrawal, C. Budak, and A. El Abbadi. Information diffusion in social networks: Observing and influencing societal interests. *PVLDB*, 4(12):1512–1513, 2011.

[3] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD'08*.

[4] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06*.

[5] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *WINE*, 2007.

[6] A. Borodin, M. Braverman, B. Lucier, and J. Oren. Strategyproof mechanisms for competitive influence in networks. In *WWW*, 2013.

[7] A. Borodin, Y. Filmus, and J. Oren. Threshold models for competitive influence in social networks. In *WINE*, 2010.

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

[9] G. Călinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011. Also in IPCO'07 and STOC'08.

[10] T. Carnes, C. Nagarajan, S. M. Wild, and A. van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *ICEC*, pages 351–360, 2007.

[11] P. Chalermsook, J. Chuzhoy, S. Kannan, and S. Khanna. Improved hardness results for profit maximization pricing problems with unlimited supply. In *APPROX-RANDOM*, pages 73–84, 2012.

[12] P. Chalermsook, B. Laekhanukit, and D. Nanongkai. Graph products revisited: Tight approximation hardness of induced matching, poset dimension, and more. In *SODA*, 2013.

[13] S. Datta, A. Majumder, and N. Shrivastava. Viral marketing for multiple products. In *ICDM*, 2010.

[14] P. S. Dodds and D. J. Watts. A generalized model of social and biological contagion. *Journal of Theoretical Biology*, 2005.

[15] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, pages 57–66, 2001.

[16] J. Edwards. Chart: Facebook's sponsored stories are way more effective than display ads. *Business Insider*, 2012.

[17] A. Efrati. Google expected to surpass facebook in display-ad sales. *The Wall Street Journal*, 2012.

[18] S. Fiegerman. Google will overtake facebook in u.s. display ad revenue this year [report]. *Mashable Business*, 2012.

[19] M. Fisher, G. Nemhauser, and L. Wolsey. An analysis of approximations for maximizing submodular set functions-II. In M. Balinski and A. Hoffman, editors, *Polyhedral Combinatorics*, volume 8 of *Mathematical Programming Studies*, pages 73–87. Springer Berlin Heidelberg, 1978.

[20] M. Gladwell. *The Tipping Point: How Little Things Can Make a Big Difference*. 2002.

[21] S. Goldberg and Z. Liu. The diffusion of networking technology. In *SODA*, 2013.

[22] J. Goldenberg, B. Libai, and E. Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, 2001.

[23] S. Goyal and M. Kearns. Competitive contagion in networks. In *STOC*, pages 759–774, 2012.

[24] M. Granovetter. Threshold Models of Collective Behavior. *The American Journal of Sociology*, 1978.

[25] J. D. Hartline, V. S. Mirrokni, and M. Sundararajan. Optimal marketing strategies over social networks. In *WWW*, pages 189–198, 2008.

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

[27] C.-C. Huang and Z. Svitkina. Donation center location problem. In *FSTTCS*, pages 227–238, 2009.

[28] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*.

[29] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 2007.

[30] M. E. J. Newman. Spread of epidemic disease on networks. *Physical Review E*, 2002.

[31] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[32] E. M. Rogers. *Diffusion of Innovations*. Simon and Schuster, 2003.

[33] M. Rogowsky. Will facebook generate more revenue than google? *Quora*, 2012.

[34] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion. In *WWW '11*.

[35] S. Shirazipourazad, B. Bogard, H. Vachhani, A. Sen, and P. Horn. Influence propagation in adversarial setting: how to defeat competition with least amount of investment. In *CIKM*, 2012.

[36] S. Simon and K. R. Apt. Choosing products in social networks. In *WINE*, 2012.

[37] V. Tzoumas, C. Amanatidis, and E. Markakis. A game-theoretic analysis of a competitive diffusion process over social networks. In *WINE*, 2012.