The Bang for the Buck: Fair Competitive Viral Marketing from the Host Perspective

Wei Lu[†] Francesco Bonchi[‡] Amit Goyal[†] Laks V.S. Lakshmanan[†]

[†]University of British Columbia
Vancouver, B.C., Canada
{welu,goyal,laks}@cs.ubc.ca

[‡]Yahoo! Research
Barcelona, Spain
bonchi@yahoo-inc.com

ABSTRACT

The key algorithmic problem in viral marketing is to identify a set of influential users (called *seeds*) in a social network, who, when convinced to adopt a product, shall influence other users in the network, leading to a large number of adoptions. When two or more players compete with similar products on the same network we talk about *competitive viral marketing*, which so far has been studied exclusively from the perspective of one of the competing players.

In this paper we propose and study the novel problem of competitive viral marketing from the perspective of the *host*, i.e., the owner of the social network platform. The host sells viral marketing campaigns as a service to its customers, keeping control of the selection of seeds. Each company specifies its budget and the host allocates the seeds accordingly. From the host's perspective, it is important not only to choose the seeds to maximize the collective expected spread, but also to assign seeds to companies so that it guarantees the "bang for the buck" for all companies is nearly identical, which we formalize as the *fair seed allocation* problem.

We propose a new propagation model capturing the competitive nature of viral marketing. Our model is intuitive and retains the desired properties of monotonicity and submodularity. We show that the fair seed allocation problem is NP-hard, and develop an efficient algorithm called *Needy Greedy*. We run experiments on three real-world social networks, showing that our algorithm is effective and scalable.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

Keywords

Social networks, influence propagation, viral marketing

1. INTRODUCTION

Recent years have witnessed tremendous interest in social influence and the phenomenon of influence-driven propagations in social networks, fueled by a variety of applications, among which the most prominent one is *viral marketing*. The key computational problem behind viral marketing is the identification of a set of k influential users, whom should be "targeted" by a viral marketing campaign. Here, targeting means giving free (or price discounted) samples of a product and k represents the company's budget. The targeted users, also called *seeds*, should be those that are well positioned to create word-of-mouth driven cascades, so to transitively convince the largest number of other users to adopt the product.

The bulk of research in this field assumes that there is one company, introducing one product in the market. In other words, there is no competition. However, in the real world, typically multiple players compete with comparable products over the same market. For example, consider consumer technologies such as videogame consoles (X-Box vs. Playstation), digital SLR cameras (Canon vs. Nikon) or smartphones (Android vs. iPhone): since the adoption of these consumer technologies is not free, it is very unlikely that an average consumer will adopt more than one of the competing products. Recognizing this, there has been some recent work on *competitive viral marketing*, where two or more players compete with similar products for the same market. The majority of these studies focus on the best strategy for one of the players [1–4,11,15].

Our motivating observation is that social network platforms are owned by third party such as Facebook and LiveJournal. The owner keeps the proprietary social graph secret[1] for obvious reasons of the company benefits, as well as due to privacy legislation. We call the owner the *host*. Companies that want to run viral campaigns are the host's *clients*. The clients typically do not have direct access to the network and thus cannot choose seeds for their campaign on their own. Any campaign would need the host's permission and privilege to run. Take Facebook as an example, business owners can set up a Facebook Page and create display ads or promoted posts to reach users[2], but they are not able to effectively implement a viral marketing campaign which directly reaches individual users, due to the lack of access to the network graph and privacy concerns.

Motivated by this observation, we propose and study the novel problem of competitive viral marketing *from the host perspective*. We consider a new business model where the host offers viral marketing as a service, for a price. It allows the clients to run campaigns by specifying a seed budget, i.e., number of seeds desired. The host controls the selection of seeds and their allocation to companies. Once seeds are allocated, companies compete for adopters of their products on the common network.

In classical non-competitive influence maximization, the objective is to choose the seeds so as to maximize the expected number of adopters. However, in a competitive setting, from the host's perspective, it is important not only to choose the seeds to maximize

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

[1] http://techcrunch.com/2013/01/24/my-precious-social-graph/

[2] https://www.facebook.com/business

the collective expected number of adoptions across all companies, but also to allocate seeds to companies in a way that guarantees the "*bang for the buck*" for all companies is nearly the same. Intuitively, the bang for the buck for a company is the cost benefit ratio between the expected number of adopters of its product over its number of seeds. We call this the *amplification factor*, as it reflects how investing in a small number of seeds gets amplified by the network effect. If the host allocates the seeds carelessly to its clients, it can result in a wide variance in the amplification factors, leading to resentful clients. Consider the following hypothetical scenario. Suppose Canon and Nikon are two clients with seed budgets 20 and 30, and Facebook, as host, selects 50 seeds. If those 50 seeds are allocated in such a way that Canon ends up getting expected spread of 400 ("bang for the buck" being 20), while Nikon gets 300 ("bang for the buck" being 10), this allocation is unfair and may lead to Nikon to feel resentful.

Motivated by the above, we propose a new propagation model called K-LT by extending the classical Linear Threshold (LT) model [13] to capture the competitive aspect in viral marketing. Intuitively, propagation in our model consists of two phases. A node (user) is in one of three states: *inactive*, *influenced*, or *active*. It adopts a product only in the active state. In the first phase, inactive nodes may become influenced (to adopt a product) as a result of influence coming in from their neighbors. In the second phase, an influenced node makes its choice to adopt one of the products (i.e., becomes active) based on the relative strengths of incoming influence for different products. The model is intuitive and retains the desired properties of monotonicity and submodularity.

We then define the *fair seed allocation* problem whose goal is to allocate seeds to the companies such that their amplification factors are as close to each other as possible, while the total expected number of adoptions over all companies is maximized. The problem is NP-hard and we devise an efficient and effective greedy heuristic to tackle it. To summarize, we make the following contributions:

- We study competitive viral marketing from a campaign host's perspective. We propose the K-LT propagation model and show that in our model, expected influence spread for any individual competing product is monotone and submodular (§4.1).
- We define the problem of Fair Seed Allocation (FSA) and discuss a number of options for formalizing it. As a case study, we focus on minimizing the maximum amplification factor offered to companies (§3.2).
- We show that FSA under K-LT model is NP-hard (§4.1). However, it can be solved exactly using dynamic programming when there are only two companies competing (§4.4).
- We develop an efficient heuristic algorithm, *Needy Greedy*, a natural adaptation of the classic greedy algorithm for the non-competitive setting for any number of companies (§4.3).
- We conduct extensive experiments on three real-world network datasets and our results show that our algorithms are effective and efficient, significantly outperforming two simple baselines including random allocation (§5).

The next section reviews necessary background and related work. In §6, we summarize the paper and discuss future work.

2. BACKGROUND AND RELATED WORK

Kempe et al. [13] modeled viral marketing as a discrete optimization problem, named *influence maximization*, and focusing on two fundamental propagation models: *Independent Cascade* (IC) and *Linear Threshold* (LT). In both models, we are given a directed social graph $G = (V, E)$ with edges $(u, v) \in E$ labeled by influence weights $p_{u,v} \in (0, 1]$. If $(u, v) \notin E$, define $p_{u,v} = 0$. At a given time step, each node is either active (an adopter of product) or inactive. An active node never becomes inactive. Initially all nodes are inactive, and at time 0, a set S of seeds are activated. In the LT model, the sum of incoming weights to v is no more than 1. Each node v chooses a threshold θ_v uniformly at random from $[0, 1]$. If at time t, the total weight from the active neighbors of v is at least θ_v, then v becomes active.

Given a propagation model (e.g., IC or LT) and a seed set $S \subseteq V$, the expected number of active nodes at the end of the process or the *(expected) spread* is denoted by $\sigma(S)$. The *influence maximization problem* asks for a set $S \subseteq V$, $|S| = k$, such that $\sigma(S)$ is maximum, where k is an input parameter. Under both IC and LT models, the problem is NP-hard [13]. Kempe et al., however, show that the function $\sigma(S)$ is *monotone* (i.e., $\sigma(S) \leq \sigma(T)$ whenever $S \subseteq T$) and *submodular* (i.e., $\sigma(S \cup \{w\}) - \sigma(S) \geq \sigma(T \cup \{w\}) - \sigma(T)$ whenever $S \subseteq T$, and $w \in V \setminus T$). When equipped with such properties, the simple greedy algorithm that at each iteration greedily extends the current set of seeds S with the node w providing the largest marginal gain $\sigma(S \cup \{w\}) - \sigma(S)$, gives a $(1 - 1/e - \epsilon)$-approximation to the optimum [13, 17] (for any $\epsilon > 0$). After [13], considerable work has been done on developing more efficient and scalable influence maximization algorithms [5, 6, 9, 10, 16].

Competitive viral marketing. There have been some recent studies on competitive viral marketing, by extending the IC or the LT model. A common theme among all of them is that they all focus on the client perspective as opposed to the host perspective.

Bharathi et al. [1] and Carnes et al. [4] study the problem from the "follower's perspective". The follower is the player trying to introduce a new product into an environment where a competing product already exists. Both studies show that the problem for the follower maintains the desired properties of monotonicity and submodularity and thus the greedy algorithm can be applied to provide approximation guarantees.

Kostka et al. [15] study competitive influence diffusions under a game-theoretic framework and show that finding the optimal strategy of both the first and second player is NP-Complete. Budak et al. [3] and Chen et al. [11] study the problem of influence blocking maximization, where one entity tries to block the influence propagation of its competitor as much as possible, under extended IC and LT models, respectively. Pathak et al. [18] propose an extension of the voter model to study multiple cascades. Borodin et al. [2] propose extensions to the LT model to deal with competing products. Their work is also from the perspective of one of the competing players. As this work is the most related to our proposal, we present it in greater detail in the next section.

3. MODELS AND PROBLEM DEFINITION

In this section we present the propagation model underlying our work and provide the problem statement. We first introduce our extended LT model (dubbed K-LT) that captures competition, and then provide conceptual justifications of the model. Then we highlight the difference between K-LT and the Weighted-Proportional Competitive (WPCLT) model by Borodin et al. [2].

3.1 The K-LT propagation model

Let K be the number of competing companies (or colors)[3]. Let C_i and S_i with $i \in \{1, 2, \ldots, K\}$, denote the i-th company and its seed set, respectively. Each node $v \in V$ picks an activation threshold θ_v uniformly at random from $[0, 1]$. Initially, all nodes are inactive. At time 0, for each color C_i, a seed set S_i is targeted

[3] We use the two terms interchangeably.

(with disjoint seed sets for different colors). This means that if $u \in S_i$, then u becomes active with color C_i at time 0.

At any time $t \geq 1$, the activation of a node takes place in two phases. First, an *inactive* node v becomes *influenced* when the total incoming influence weight from its in-neighbors (denoted $N^{\text{in}}(v)$) which are active (regardless of colors) reaches v's threshold: $\sum_{\text{active } u \in N^{\text{in}}(v)} p_{u,v} \geq \theta_v$. Then, in a second phase (still at time t), v becomes *active* by picking a color out of those of its in-neighbors that activated at $t-1$.

Let A^i_{t-1} denote the set of nodes that are active with color C_i at the end of time $t-1$ and A_{t-1} denote the set of nodes that are active at the end of time $t-1$, w.r.t. any color. Hence, v becomes active at time t with color C_i with probability $\sum_{u \in A^i_{t-1} \setminus A^i_{t-2}} p_{u,v} \,/\, \sum_{u \in A_{t-1} \setminus A_{t-2}} p_{u,v}$. Once a node becomes active, it remains active and will not switch colors. The diffusion process continues until no more nodes can be activated.

The K-LT model reflects several phenomena of competitive influence propagation that match our daily experience as well as studies in the literature. While the first phase models the threshold behavior in influence propagation, as in the original LT model, the second phase incorporates the *recency effect* in the final decision among competing products. Indeed, it has been recognized in various studies that influence decays very quickly in time, and thus customers are more likely to rely on recent information than on old information, when choosing which product to adopt [12, 19, 20].

Comparisons with the WPCLT model. In the WPCLT model [2], the first phase in which a node is influenced remains exactly the same as in K-LT. The difference lies in the second phase, i.e., the way in which newly influenced nodes decides the color to adopt. In WPCLT, a node v picks a certain C_i with probability equal to the ratio between the total weight from the C_i-active in-neighbors and that from all active in-neighbors. That is, all past exposure are accounted for adoption. Thus, v becomes active with color C_i with probability $\sum_{u \in A^i_{t-1}} p_{u,v} \,/\, \sum_{u \in A_{t-1}} p_{u,v}$.

To fully understand the difference between the WPCLT model and our K-LT, we first need to define the expected spread of influence. Let $\mathbf{S} = \{S_1, ..., S_K\}$ be the set of seeds sets for the various colors, i.e., $\mathbf{S}$ corresponds to a seed set allocation. We use $\mathbf{S}_{-i}$ to denote the set of seed sets for all colors but color C_i, i.e., $\mathbf{S}_{-i} =_{\text{def}} \{S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_K\}$.

DEFINITION 1 (EXPECTED SPREAD). *For a color C_i, we use $\sigma_i(S_i, \mathbf{S}_{-i})$ to denote the expected number of active nodes, or the* expected spread, *w.r.t. C_i, given seed set allocation* $\mathbf{S}$. *We define the* overall expected spread, *denoted $\sigma_{all} =_{\text{def}} \sum_{i=1}^{K} \sigma_i(S_i, \mathbf{S}_{-i})$, to be the expected number of active nodes w.r.t. any color.* □

As we will show in Theorem 1 (§4.1), $\sigma_i(S_i, \mathbf{S}_{-i})$ is *monotone* and *submodular* in S_i under the K-LT model, while this property does not hold in WPCLT, which is somewhat counter-intuitive as noted by the authors that proposed it (cf. [2]). Indeed, $\sigma_i(S_i, \mathbf{S}_{-i})$ being non-monotone means that adding a new seed x to S_i may cause the spread for C_i to go down. This is not desired as a company expects the influence spread to go up when it increases its budget. These counter-intuitive phenomena stem from the possibility that a certain graph structure will allow the seeding of some nodes to trigger multiple "activation attempts" for seeds of a different company, which we show by an example below. For more detailed examples illustrating non-monotonicity and non-submodularity of the WPCLT model, we refer the reader to [2].

EXAMPLE 1 (ACTIVATION IN WPCLT). Consider Figure 1. Suppose that there are two colors with seed sets $S_1 = \{u\}$ and $S_2 = \{w\}$. Also suppose that θ_v and θ_x fall into the interval $(0.5, 1)$. At time step 1, v becomes active w.r.t. color 2 (as $p_{w,v} = 1 > \theta_v$), while x remains inactive (as $p_{u,x} = 0.5 < \theta_x$). Subsequently, at time step 2, x first gets influenced as the total incoming influence weight is now 1. Then, x will activate w.r.t. color 1 with probability 0.5 and color 2 with probability 0.5. □

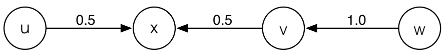


Figure 1: Graph for Example 1.

In this example, although u (in color 1) fails to activate x at time step 1, x may still adopt color 1 under WPCLT. The reason is that x gets additional influence from v which has color 2! Thus, seeding w for color 2 ends up "helping" the competitor color 1: u gets a second chance at activating x after failing at first. However, this phenomenon will not occur in K-LT: at time step 2, after getting influenced, x will activate w.r.t. color 2 exclusively, with probability $0.5/0.5 = 1$.

3.2 Problem definition

We are ready to provide the formal problem statement of fair competitive viral marketing from the host perspective. We will focus on the K-LT model hereinafter, unless otherwise specified. Assume that there are K companies, as clients of the host $\mathcal{H}$, competing with similar products (one product each). Before the campaign is run, each company C_i would approach the host, specifying a positive integer b_i as its budget (maximum number of seeds wanted), and it is assumed that $b_1 + b_2 + \ldots + b_K < |V|$. As its business model, $\mathcal{H}$ charges every company a fixed amount of money per requested seed, as well as surcharges proportional to the expected spread achieved. Before defining the problem, we first introduce the important notion of *amplification factor*.

DEFINITION 2 (AMPLIFICATION FACTOR). *The* amplification factor *of C_i, denoted α_i, is the average influence spread that C_i gets per seed, i.e.,*

$$\alpha_i = \frac{\sigma_i(S_i, \mathbf{S}_{-i})}{b_i}. \quad \square \tag{1}$$

Intuitively, after receiving budgets from all companies, $\mathcal{H}$ will allocate each company C_i a seed set S_i, $|S_i| = b_i$, such that (1) the overall influence spread, σ_{all} (Definition 1) is maximized, and (2) the expected influence spread across all companies is as "balanced" as possible, i.e., the amplification factor of each company is as close as possible. Formally, we define the problem of competitive influence maximization from the host's perspective, which consists of two subproblems, as follows.

PROBLEM 1 (OVERALL INFLUENCE MAXIMIZATION). *Given a directed graph $G = (V, E)$ with pair-wise edge weights, numbers $b_1, b_2, \ldots, b_K \in \mathbb{Z}_+$ with $\sum_{i=1}^{K} b_i \leq |V|$, select a seed set $S \subseteq V$ of size $\sum_{i=1}^{K} b_i$, such that σ_{all} is maximized.*

A first observation is that, under both K-LT and WPCLT models, the first phase of activation follows the activation condition of the classic LT model. Therefore, we have the following proposition.

PROPOSITION 1. *Given a directed graph $G = (V, E)$ with edge weights, and K pair-wise disjoint subsets $S_1, S_2, \ldots, S_K$ of V, then under both the K-LT model and the WPCLT model, letting $S = S_1 \cup \ldots \cup S_K$, we have*

$$\sigma_{all} = \sigma_{LT}(S). \tag{2}$$

where σ_{LT} is the spread function for the classical LT model.

This implies that once a seed set S is given, no matter how it gets partitioned into K disjoint subsets, σ_{all} remains the same, i.e, is *invariant* under any K-partition of S. Another consequence of Proposition 1 is that under both K-LT and WPCLT models, Problem 1 is equivalent to the original influence maximization under the LT model, and hence is NP-hard. By the same token, since σ_{LT} is monotone and submodular, selecting the set of seeds S can be done using the classic greedy algorithm outlined in the introduction as for the original LT model, giving a $(1 - 1/e - \epsilon)$-approximate solution to the optimum selection of seeds. In the rest of the paper, we assume that the seeds are selected in this way, and focus on their allocation to the companies.

The goal of our second problem is to allocate seeds among the K clients such that the amplification factor of all companies is as close as possible, so as to maximize fairness. We have various options to formalize this notion of "to be as close as possible". In the following problem statement and hereinafter we adopt as objective function to minimize the maximum amplification factor α_{max}. Intuitively, when the maximum amplification factor is minimized, it balances out all the amplification factors. We believe this min-max objective is a natural choice, as it is widely recognized and adopted in the literature of resource allocation and load balancing [14]. A discussion on several other alternatives is provided in §3.3.

PROBLEM 2 (FAIR SEED ALLOCATION (FSA)). *Given a directed graph $G = (V, E)$ with pair-wise edge weights, numbers $b_1, b_2, \ldots, b_K \in \mathbb{Z}_+$, a set $S \subseteq V$ with $|S| = \sum_{i=1}^{K} b_i$, find a partition of S into K disjoint subsets $S_1, S_2, \ldots, S_K \subseteq S$, such that $|S_i| = b_i$, $i \in [1, K]$, and the maximum amplification factor of any color is minimized.*

Note that although the two problems are formulated separately, the host $\mathcal{H}$ needs to solve both in a sequential order to achieve its goals. In other words, the output of Problem 1, i.e., the union seed set S, is given as input for Problem 2.

3.3 Discussion: choice of objective function

Our goal while partitioning the seed set S is to make the amplification factors as close as possible, so as to maximize the fairness. To achieve this goal, in Problem 2, we defined the objective function as minimizing the maximum amplification factor α_{max}. One can offer similar alternative objective functions, while trying to achieve the same goal. For instance, one can ask for maximizing the minimimun amplification factor α_{min}. Similarly, another objective could be to minimize the difference $\alpha_{max} - \alpha_{min}$, or the ratio $\alpha_{max}/\alpha_{min}$. More sophisticated objective functions can be based on L_1 or L_2 norms. In general, the objective function based on L_p norm could be

$$\left[\sum_{i=1}^{K} \left|\sigma_i(S_i, \mathbf{S}_{-i}) - \sigma_{all} \cdot \frac{b_i}{B}\right|^p\right]^{1/p}, \qquad (3)$$

which we may want to minimize. A comprehensive theoretical analysis of these various objective functions would be an interesting exercise, but it is not the focus of this paper. In the experiments section, we show that our algorithm performs well w.r.t. essentially all of these objectives.

4. MODEL PROPERTIES AND SEED ALLOCATION ALGORITHMS

Before we develop the algorithms for Problem 2, we take a deeper look in the properties of our K-LT model, which will allow us to characterize the complexity of FSA under K-LT and develop efficient and effective seed allocation algorithms.

4.1 Properties of K-LT model

We first show that the expected spread function for individual colors is monotone and submodular (Theorem 1) in the K-LT model. To prove this result, we employ a plot similar to the one in Kempe et al. [13], by establishing the equivalence between the K-LT model and a competitive version of the "live-edge" model (Definition 3). This, importantly, will in turn help us derive a closed-form expression for the spread function (Theorem 2), which will play a pivotal role in the design of our algorithms and characterizing the complexity of FSA: it is NP-hard in general (Theorem 3), but can be solved in polynomial time for $K = 2$.

We start by introducing the competitive live-edge model, by extending the live-edge model defined in [13].

DEFINITION 3 (COMPETITIVE LIVE-EDGE MODEL). Given a directed graph $G = (V, E)$ with edges labeled by influence weights, we can obtain a *possible world* X as follows. Each node v picks at most one of its incoming edges at random, selecting edge (u, v) with probability $p_{u,v}$ and selecting no edge with probability $1 - \sum_{w \in N^{in}(v)} p_{w,v}$. The selected edges are declared "live", while others "blocked". By definition, incoming edges to nodes in the seed set S are blocked. We call a directed path a *live-edge path* if it consists entirely of live edges. □

In a possible world X, we say a node is C_i*-reachable*, if there exists a live-edge path from a node in S_i to v. Note that a node v has at most one incoming live edge, thus there is at most one live-edge path from S to v. Thus, the notion of color rechability is well-defined.

It is easy to see that the spread function under the competitive live-edge model is monotone and submodular. Clearly, each possible world X is a deterministic graph. Let $R_X(\{u\})$ be the set of reachable nodes from a particular node u on live-edge paths, in X. Then the set of nodes reachable from S_i is $R_X(S_i) = \cup_{u \in S_i} R_X(\{u\})$. The function $|R_X(S_i)|$ is clearly monotone and submodular. Finally, the expected number of C_i-reachable nodes according to the live-edge model, $\sum_X \Pr[X] \cdot |R_X(S_i)|$, is a non-negative linear combination of monotone submodular functions, and thus is monotone and submodular (in S_i). Here, $\Pr[X]$ is the probability of the possible world X, which is determined by the choice of live/blocked edges. We now state the submodularity result for K-LT:

THEOREM 1. *Under K-LT model, for any color C_i, the expected spread of influence $\sigma_i(S_i, \mathbf{S}_{-i})$ is monotone and submodular in S_i, with $\mathbf{S}_{-i}$ fixed.*

PROOF. We prove this result by establishing the equivalence between the K-LT model and the competitive live-edge model (Definition 3). We show : Given K colors and their corresponding seed sets $S_1, S_2, \ldots, S_K$ (all disjoint), for any color C_i, *the following two distributions over sets of nodes are equivalent*: (1) The distribution over C_i-active sets obtained by running the K-LT process to completion from $S_1, S_2, \ldots, S_K$, and (2) The distribution over sets of C_i*-reachable nodes* according to the live-edge model. The theorem follows from this claim. We next prove the claim. If a node v has not become active after time step t, then the probability that it becomes C_i-active at $t + 1$ is

$$\frac{\sum_{u \in A_t \setminus A_{t-1}} p_{u,v}}{1 - \sum_{u \in A_{t-1}} p_{u,v}} \cdot \frac{\sum_{u \in A_t^i \setminus A_{t-1}^i} p_{u,v}}{\sum_{u \in A_t \setminus A_{t-1}} p_{u,v}} = \frac{\sum_{u \in A_t^i \setminus A_{t-1}^i} p_{u,v}}{1 - \sum_{u \in A_{t-1}} p_{u,v}},$$

where the former quantity is the probability that v becomes active at $t + 1$, and the latter is the probability that v adopts color C_i, given that v gets activated.

For the competitive live-edge model, we start the "reach-out" process with seed sets $S_1, S_2, \ldots, S_K$. In the first stage, if a node v's selected live-edge is incident on S_i, then v is C_i-reachable from a seed in S_i. We denote the set of such nodes by ${A'}^i_1$. In general, let ${A'}^i_t$ denote the set of nodes which are found to be C_i-reachable from a node in S_i in stage t. In this way, we can obtain sets ${A'}^i_2, {A'}^i_3, \ldots$. Similarly, we can also obtain sets A'_t, $t = 1, 2, 3, \ldots$, which represent the set of nodes reachable from $S_1 \cup S_2 \cup \ldots \cup S_K$ in stage t. Now, if a node v has not yet been determined C_i-reachable by the end of stage t, then the probability that v will be determined C_i-reachable at stage $t + 1$ is the chance that its chosen edge is from $A'_t \setminus A'_{t-1}$, which is $\frac{\sum_{u \in A'_t \setminus A'_{t-1}} p_{u,v}}{1 - \sum_{u \in A'_{t-1}} p_{u,v}}$. Given that, the probability that v proceeds to become C_i-reachable is $\frac{\sum_{u \in {A'}^i_t \setminus {A'}^i_{t-1}} p_{u,v}}{\sum_{u \in A'_t \setminus A'_{t-1}} p_{u,v}}$. By the product rule, the probability that v will be determined to be C_i-reachable at stage $t + 1$, given that it is not already so determined, is $\frac{\sum_{u \in {A'}^i_t \setminus {A'}^i_{t-1}} p_{u,v}}{1 - \sum_{u \in A'_{t-1}} p_{u,v}}$.

Applying induction on time steps (stages), it is easy to see that the distributions over A^i_t and ${A'}^i_t$ are identical, and the same holds for A_t and A'_t, $\forall t$. This was to be shown. □

Closed-form expression for $\sigma_i(S_i, \mathbf{S}_{-i})$. We first introduce the needed notation. By virtue of the equivalence shown in Theorem 1, $\sigma_i(S_i, \mathbf{S}_{-i})$ is equal to the expected number of C_i-reachable nodes under the competitive live-edge model. Let X be a possible world. For simplicity, we write $V - S$ for $V \setminus S$ and $V - S + u$ for $(V \setminus S) \cup \{u\}$ hereinafter. With node-sets as superscripts, we denote the corresponding induced subgraph: e.g., $\sigma^W_{LT}(S)$, where $W \subseteq V$, denotes the expected spread of the seed set S in the subgraph of G induced by the nodes W. When there is no superscript, the entire graph G is meant by default.

We now derive the closed-form expression by establishing connections to the classical LT model. Let $I^{V - \mathbf{S}_{-i}}_X(S_i, v)$ be the indicator function which takes 1 if there exists a node s in S_i and a path from s to v, in a possible world X for the subgraph of G induced on $V - \mathbf{S}_{-i}$ (otherwise the function takes 0). Thus, by definition,

$$\sigma_i(S_i, \mathbf{S}_{-i}) = \sum\nolimits_X \Pr[X] \cdot \sigma_{i,X}(S_i, \mathbf{S}_{-i}),$$

where $\sigma^X_{i,X}(S_i, \mathbf{S}_{-i})$ is the number of C_i-reachable nodes in possible world X. Then, because any live-edge path from any node $u \in S_i$ to v must not go through any node $w \in \mathbf{S}_{-i}$, as all incoming edges to nodes in $\mathbf{S}_{-i}$ are blocked by definition of the live-edge model (in other words, it has the effect of removing nodes in $\mathbf{S}_{-i}$ from G and hence from the possible world X), we have

$$\sigma_i(S_i, \mathbf{S}_{-i}) = \sum\nolimits_X \Pr[X] \cdot \sum\nolimits_{v \in V} I^{V - \mathbf{S}_{-i}}_X(S_i, v).$$

Let $W = V - \mathbf{S}_{-i}$, the set of nodes after removing nodes in $\mathbf{S}_{-i}$. Then, by switching the summations, we have

$$\begin{aligned} \sigma_i(S_i, \mathbf{S}_{-i}) &= \sum\nolimits_{v \in V} \sum\nolimits_X \Pr[X] \cdot I^W_X(S_i, v) \\ &= \sum\nolimits_{v \in V} \Upsilon^W_{S_i,v}\,, \end{aligned} \tag{4}$$

where $\Upsilon_{S_i,v}$ is the probability that there exists a path from S_i to v in the subgraph induced by $V - \mathbf{S}_{-i}$. Since S_i is the seed set for company C_i, it also denotes the probability that v becomes C_i-active on the corresponding subgraph. Note that the indicator function depends only the seed set S_i and the subgraph W, and not on the seeds for other colors. Therefore, $\Upsilon^W_{S_i,v}$ is equal to the probability that v is activated in the subgraph induced by $V - S + u$, *under classical LT model*, with seed set S_i.

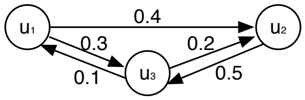


Figure 2: Example of Adjusted Marginal Gain

Adjusted marginal gain. Next, we introduce the notion of *adjusted marginal gain*, which is *key to solving Problem 2.*

DEFINITION 4 (ADJUSTED MARGINAL GAIN). Given a set S of seeds, for any $u \in S$, the *adjusted marginal gain* of u, denoted δ_u, is the expected spread of influence of $\{u\}$ on the graph induced by $V - S + u$ under the classical LT model. That is, $\delta_u = \sigma^{V-S+u}_{LT}(\{u\})$. □

Consider the example in Fig. 2. Suppose $S = \{u_1, u_2\}$ is the seed set. Then, one can verify that δ_{u_1} is the expected spread of u_1 on graph consisting of u_1 and u_3 only, which is $1 + 0.3 = 1.3$.

Next, we show the following useful result for the K-LT model, which says that given a set of seeds S selected by the host, the expected spread for company C_i only depends on the seeds S_i allocated it, and not on how the remaining seeds $S - S_i$ are distributed among the other companies.

THEOREM 2. *Consider an allocation of seed sets, where the seed set $S_i \subseteq S$ is assigned to company C_i and the remaining seeds $S - S_i$ are allocated arbitrarily to other companies (denoted by $\mathbf{S}_{-i}$). Then under the K-LT model,*

$$\sigma_i(S_i, \mathbf{S}_{-i}) = \sum\nolimits_{u \in S_i} \delta_u \tag{5}$$

PROOF (THEOREM 2). Consider the right hand side of the equation. Since S is the set of all seeds, that is, $S = S_i + \mathbf{S}_{-i}$. we have by Definition 4,

$$\begin{aligned} \sum\nolimits_{u \in S_i} \delta_u &= \sum\nolimits_{u \in S_i} \sigma^{V - \mathbf{S}_{-i} - S_i + u}_{LT}(\{u\}) \\ &= \sum\nolimits_{u \in S_i} \sum\nolimits_{v \in V} \Upsilon^{V - \mathbf{S}_{-i} - S_i + u}_{u,v}\,, \end{aligned}$$

where $\Upsilon^{V - \mathbf{S}_{-i} - S_i + u}_{u,v}$ is the probability with which v is activated given seed set $\{u\}$, on the subgraph induced by the nodes $V - \mathbf{S}_{-i} - S + u$, under LT model. We next make use of the proof of Theorem 1 of [10]. There, it is shown that, under LT model, $\Upsilon_{S_i,v} = \sum_{u \in S_i} \Upsilon^{W - S_i + u}_{u,v}$, for any set $S_i \subseteq W \subseteq V$, where $\Upsilon_{S_i,v}$ is the probability that v becomes active, given seed set S_i, on the subgraph induced by the nodes $W - S_i + u$. Let $W = V - \mathbf{S}_{-i}$, then by switching the summations and applying this result, we get

$$\sum\nolimits_{u \in S_i} \delta_u = \sum\nolimits_{v \in V} \Upsilon^W_{S_i,v}\,.$$

From the equivalence with the live-edge model, and Eq. 4, the theorem follows. □

Consider again the example shown in Fig. 2. Suppose there are two companies with $S_1 = \{u_1\}$ and $S_2 = \{u_2\}$. Then, $\sigma_1(S_1, \mathbf{S}_{-1}) = \delta_{u_1} = 1.3$. Similarly, $\sigma_2(S_1, \mathbf{S}_{-2}) = \delta_{u_2} = 1.5$. Also, note that $\sigma_{all} = \sigma_{LT}(\{u_1, u_2\}) = 2.8$.

4.2 NP-Hardness of Fair Seed Allocation

Having established the notion of adjusted marginal gain, we are ready to prove the complexity of Problem 2 (Fair Seed Allocation).

THEOREM 3. *The Fair Seed Allocation problem under the K-LT model is NP-hard.*

PROOF. We prove the theorem by reduction from 3-PARTITION [7]. In 3-PARTITION, we are given a set A of $3 \cdot m$ elements, and a size $s(a) \in Z^+$ for each element. Let Y be the sum of sizes of all elements, i.e., $Y = \sum_a s(a)$, then the question is whether there exists a partition of A into m disjoint subsets $A_1, ..., A_m$, each with exactly 3 elements, such that the sum of sizes of elements in each subset is the same, i.e., $\sum_{a \in A_i} s(a) = Y/m$. This problem is known to be strongly NP-hard [7]. Recall that a problem is strongly NP-hard if it remains NP-hard even when the numerical parameters of the problem are bounded by a polynomial of the input size. In the context of 3-PARTITION problem, it implies that the problem remains NP-hard even when Y is bounded by a polynomial in m.

Let $\mathcal{I}$ be an instance of 3-PARTITION. We reduce it to an instance $\mathcal{J}$ of FSA as follows. Create m companies, and for each element $a \in A$ with size $s(a)$, create a seed u_a in instance $\mathcal{J}$, with its adjusted marginal gain set to $\delta_{u_a} := s(a)$. Set the seed budget of each company to 3. Suppose there exists a polynomial time algorithm $\mathcal{A}$ that provides an optimal solution to FSA. Then by running this algorithm on instance $\mathcal{J}$ and checking whether the maximum amplification factor is exactly $Y/3m$ or not, we can separate the YES-instances from the NO-instances of 3-PARTITION, which is not possible unless P = NP.

Above, we performed the reduction entirely in terms of adjusted marginal gains, instead of creating a graph, which is a required input to FSA. It is easy to create an input graph whose seed nodes u_a satisfy the adjusted marginal gains above. E.g., create $3 \cdot m$ disjoint trees, each rooted at a node u_a. The root u_a has exactly $s(a) - 1$ children, with influence weights on all edges set to 1. Since the trees are disjoint, $\delta_{u_a} = s(a)$. Notice this reduction is polynomial time in m since Y is a polynomial in m. □

When $K = 2$, the FSA problem resembles the PARTITION problem, which is weakly NP-hard and admits an exact dynamic programming algorithm in pseudo-polynomial time. We can adapt it to solve FSA. In our case, the dynamic programming algorithm (See §4.3) is truly polynomial in the size of the input, since the number of nodes is a natural bound on all adjusted marginal gains.

4.3 The Needy Greedy algorithm

Suppose there are K companies approaching a host for running a competitive viral campaign and each company C_i specified a seed budget b_i. Let $B = \sum_{i=1}^{K} b_i$. The host owns a social graph G. How can the host effectively find seeds and allocate them to the K companies? As pointed out by Proposition 1, the host can select B seeds using the classic greedy algorithm. Let S be this set. The real challenge is in finding a good partition of S into disjoint subsets $S_1, ..., S_K$ such that the allocation of S_i to company C_i minimizes the maximum amplification factor. Given the hardness result above, a natural question is whether we can devise efficient heuristic algorithms that work well in practice. In this section, we develop an algorithm called *Needy Greedy* (NG for short), which works for any K. Later, in §4.4, we will give an algorithm based on dynamic programming, for $K = 2$.

The Needy Greedy algorithm takes advantage of Theorem 2, which says that given the set S of seeds, the expected spread of C_i is solely determined by the seeds $S_i \subseteq S$ that are allocated to company C_i, and it can be calculated by summing the adjusted marginal gains of the seeds in S_i in appropriate subgraphs. Thus, we first find all seeds S using the classic greedy algorithm. Then we determine the adjusted marginal gains δ_u of the seeds (Definition 4) and keep seeds sorted in non-increasing order of the gains.

Algorithm 1: NEEDY-GREEDY (NG)

```
Input  : $S$ (with $\delta_u, \forall u \in S$) and $b_i, \forall i \in \{1, \ldots, K\}$.
Output: A $K$-partition of $S$, with $|S_i| = b_i, \forall i$.
1 Initialize $S_i = \emptyset, \forall i$;
2 for each $u \in S$ do
3     $T \leftarrow \{i \mid i \in \{1, 2, \ldots, K\}, |S_i| < b_i\}$;
4     $j \leftarrow \arg\min_{i \in T}\{\sigma_i(S_i, \mathbf{S}_{-i})/b_i\}$;
5     $S_j \leftarrow S_j \cup \{u\}$;
```

Needy Greedy (Algorithm 1) takes as input the seeds with adjusted marginal gain sorted In addition, it is given the budgets of various companies. It starts by initializing all seed sets S_i to be empty (line 1). Then, we process each seed $u \in S$ (line 2). Let T be the set of of companies for which the budget has not yet been exhausted, i.e., $|S_i| < b_i$ (line 3). Note that we can allocate a new seed only to such companies. In line 4, we find the company j which has the least amplification factor. Finally, we add the seed u to S_j (line 5). In Sec. 5, we will show how NG can be adapted to deal with other objective functions (L_p-norms) for FSA.

Time complexity. The time complexity of NG is $O(B \log K)$, as there are $|S| = B$ iterations, in each of which the algorithm examines each company to determine i^*. Using a min-heap, we can perform the search and update in $O(\log K)$ time.

4.4 The case of two companies: a dynamic programming algorithm

In the special case of $K = 2$ (only two competing companies), the FSA problem can be solved exactly by a dynamic programming (DP) algorithm in polynomial time[4]. Also note that when $K = 2$, it can also be shown that the *minimizing the objective function based on the L_p-norm, $\forall p \geq 1$ (Eq. 3), is all equivalent to minimizing the maximum amplification factor.*

FSA with $K = 2$ resembles the *partition problem*, in which we are given a collection of positive integers, and the question is whether there exists a partition of two subcollections such that the sum of elements in the two subcollections is the same. While there is some similarity among the two problems, FSA comes with cardinality constraints, in the form of seed set budgets. In addition, while the partition problem involves integers, FSA involves adjusted marginal gains of seeds which are real numbers, and thus we should pay attention to precision.

The dynamic programming algorithm is set up as follows. First, let the seed set be $S = \{u_1, u_2, \ldots, u_B\}$ (recall $B = \sum_i b_i$) and let S^j denote the "partial" seed set $\{u_1, ..., u_j\}$ for $j \in \{1, 2, \ldots, B\}$. Then, we define

$$P(j, \mu, \ell) = \begin{cases} 1, & \text{if } \exists Q \subseteq S^j \text{: } |Q| = \ell \text{ and } \sigma_1(Q, S^j - Q) = \mu \\ 0, & \text{otherwise} \end{cases}$$

Here j keeps track of the horizon, i.e., which seeds from S have been explored; ℓ is the size of a seed set Q, such that with Q allocated to C_1 and $S^j - Q$ allocated to C_2, $\sigma_1(Q, S^j - Q)$, is exactly μ. The size of Q is bounded by b_1, the budget of C_1.

Since μ represents the spread, it virtually can take any real value in $[1, \sigma_{all}]$. To keep the DP table size finite, we can round the spread μ at any level of precision desired. E.g., if we want precision up to two decimal places, all we need to do is amplify all real numbers involved in the calculation, namely the adjusted marginal gains δ_u and the spread μ by 100 and round all results to the nearest

[4] For any fixed precision of the real numbers involved.

integer. In the rest of this section, we assume some fixed precision and that the appropriate amplification and rounding are done.

Dynamic programming formulation. Notice that there is a subset of S^j of size ℓ, which when allocated to C_1, yields the spread μ, if and only if one of the following is true: (1) There is a subset of $Q \subseteq S^{j-1}$ of size ℓ, which when allocated to C_1, yields spread μ; or (2) There is a subset of S^{j-1} of size $\ell - 1$, which does not give spread μ for C_1 itself, but will if we add u_j to C_1's allocation. More formally, $P(j, \mu, \ell) = 1$ if $P(j-1, \mu, \ell) = 1$, or $P(j-1, \mu - \delta_{u_j}, \ell - 1) = 1$. This gives rise to the following dynamic programming equation:

$$P(j, \mu, \ell) = \max\{P(j-1, \mu, \ell), P(j-1, \mu - \delta_{u_j}, \ell - 1)\}$$

with the base case $P(1, 0, 0) = 1$.

Notice that, in the ideal partition, the spread of C_1 would be exactly $Z = \frac{b_1}{B} \cdot \sigma_{all}$. This is the best possible allocation w.r.t. minimizing maximum amplification factor. Thus, after the entire DP table is populated, to obtain the partition, we can set our target be the number t obtained by amplifying and rounding the number Z as outlined earlier. Modulo our precision, if $P(B, t, b_1) = 1$, then we have found this ideal partition; if not, find the number t' such that $P(B, t', b_1) = 1$ and $|t - t'|$ is minimized. This represents the optimal solution at the chosen level of precision.

Time complexity. From the ranges for j, μ, and ℓ, the size of the DP table is $O(b_1(b_1 + b_2)|V|)$ which determines its running time. Note that typically, b_1 and b_2 are much smaller than $|V|$. In our implementation, we apply a couple of optimizations. First, there is no need to populate cells with $\ell > j$, Second, if $\mu < \delta_{u_j}$, there is no need to examine the second argument in the RHS of the dynamic programming equation.

5. EMPIRICAL EVALUATIONS

5.1 Experiments settings

To evaluate the effectiveness of our proposed algorithms (NG and DP) developed for FSA and compare them with several baselines, we conduct simulations on three real-world networks – *Epinions*, *Flixster*, and *NetHEPT*. Table 1 presents the statistics of the datasets. We use the classic greedy algorithm to select the union seed set S, and following Kempe et al. [13], $10,000$ iterations of Monte Carlo (MC) simulations are run to estimate influence spread. This is an expensive step and limits the size of the graph we can work on. The scale can be extended by using scalable heuristic algorithms for the LT model [6, 10], but this is not the focus of our experiments, which is on testing seed allocation algorithms. Implementations are in C++ and all experiments were run on a Windows 7 machine with 2.66GHz Intel i5 CPU and 6GB RAM.

Preparation of datasets. We use models in [8] to compute edge weights. For Epinions, we apply both the Bernoulli and Jaccard models and then normalize the weights. In Bernoulli, the influence weight on edge (u, v) is calculated as $p_{u,v} = A_{u2v}/A_u$ where A_{u2v} is the number of actions that v performed after u, and A_u is the total number of actions u performed. In Jaccard, $p_{u,v} = A_{u2v}/A_{u|v}$ where $A_{u|v}$ is the number of actions either u or v has performed. After computing these weights, we normalize them to ensure that the sum of incoming weights to any node is 1.

NetHEPT is a collaboration network from the High Energy Physics Theory section on `arXiv.org` with nodes representing authors and edges representing co-author relationships. We calculate the weights as $p_{u,v} = A_{u,v}/N_v$ where $A_{u,v}$ is the number of papers u and v co-authored. Flixster is a friendship network from social movie site `Flixster.com`, for which $A_{u,v}$ is the number of movies rated by both u and v. In both datasets, N_v is the normalizing factor to ensure the sum of weights incoming to v is 1.

Table 1: Statistics of network datasets.

	Epinions	Flixster	NetHEPT
Number of nodes	76K	7.6K	15K
Number of edges	509K	50K	62K
Average out-degree	13.4	6.5	4.12
Maximum out-degree	3079	197	64
#Connected components	11	761	1781
Largest component size	76K	2861	6794

Table 2: Test cases with varying budget distribution. (N,F): NetHEPT & Flixster. (E): Epinions.

K	Equal budgets case	Unequal budgets case
2	$b_1 = b_2 = 30$ (N,F) $b_1 = b_2 = 15$ (E)	$b_1 = 20, b_2 = 40$ (N,F) $b_1 = 10, b_2 = 20$ (E)
3	$b_1 = b_2 = b_3 = 20$ (N,F) $b_1 = b_2 = b_3 = 10$ (E)	$b_1 = 10, b_2 = 20, b_3 = 30$ (N,F) $b_1 = 5, b_2 = 10, b_3 = 15$ (E)
6	$b_1 = \ldots = b_6 = 10$ (N,F)	$b_1 = b_2 = b_3 = 5$ (N,F) $b_4 = b_5 = b_6 = 10$
	$b_1 = \ldots = b_6 = 5$ (E)	$b_1 = b_2 = b_3 = 4$ (N,F) $b_4 = b_5 = b_6 = 6$ (E)

Baselines. We compare our algorithms with two simple baselines: Random and Alternating. In Random, seeds are assigned uniformly at random to different companies (with budget constraints obeyed). The Alternating heuristic first fixes a random permutation of the K companies, and then allocates seeds to the companies in a round-robin fashion according to that order.

Competition settings. We vary K, the number of competing entities, to be 2, 3 and 6. For NetHEPT and Flixster, we take 60 seeds, while for the much larger Epinions dataset, we take 30 seeds so that MC simulations can finish within a reasonable amount of time (48 hours). The budget distributions we choose to test are summarized in Table 2. For instance, for $K = 3$, we consider "equal budgets" cases, where each company has a budget of 20 (on NetHEPT and Flixster) or 10 (on Epinions). Likewise, we also consider "unequal budgets" cases where the three companies have budgets of 10, 20 and 30 (on NetHEPT and Flixster) or 5, 10 and 15 (on Epinions). We believe these various cases are representative.

After the seeds are chosen, we compute the adjusted marginal gain of all seeds (Definition 4), whose statistics are presented in Table 3. As can be seen, the variance is much larger in both Epinions graphs than in NetHEPT and Flixster. Especially, NetHEPT has the most concentrated values. We run NG, Random, and Alternating for all settings, and also DP for the cases of $K = 2$.

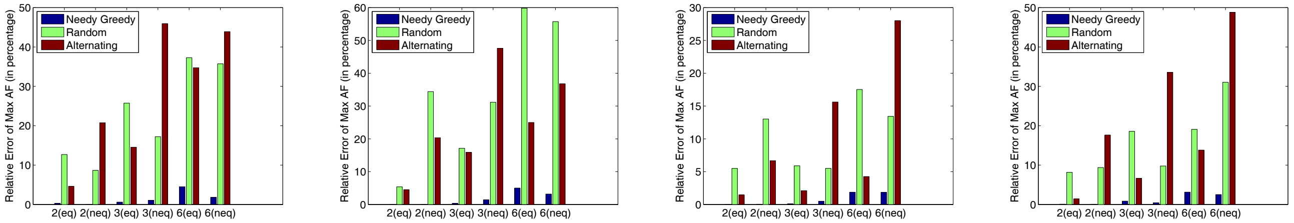
5.2 Experimental results and analysis

Evaluation metrics. To compare accuracy (i.e., quality of partition), we obtain the maximum amplification factor (denoted by α_{max}) output by the algorithm and calculate its *relative error* w.r.t. the *theoretical lower bound* σ_{all}/B (in which case the allocation is perfectly fair, with every company having the same amplification factor). The relative error is defined as follows.

$$\texttt{RelativeError}(\alpha_{max}) = \frac{\alpha_{max} - \sigma_{all}/B}{\sigma_{all}/B} \times 100\%. \qquad (6)$$

By definition, it is nonnegative as $\alpha_{max} \geq \sigma_{all}/B$ always holds: σ_{all}/B is the average spread-to-budget ratio, while α_{max} is the maximum ratio in a solution.

Comparing relative errors. The relative errors of the maximum amplification factor achieved by NG, Random and Alternating



(a) Epinions (Bernoulli)

(b) Epinions (Jaccard)

(c) NetHEPT

(d) Flixster

Figure 3: Relative error of the maximum amplification factors in three algorithms: Needy Greedy, Random and Alternating. On the X-axis, 2(eq) refers to the setting of $K = 2$, budgets equal, while (neq) refers to unequal budgets.

Table 3: Statistics of adjusted marginal gains.

	Mean	Median	Max	Min	Std. Dev.
Epinions (Bernoulli)	328.4	265.4	829.1	122.1	158.9
Epinions (Jaccard)	437.6	375.0	1304	143.6	252.5
NetHEPT	26.30	25.87	51.68	16.01	6.38
Flixster	74.77	65.05	216.4	30.65	31.91

solutions are illustrated in Fig. 3. As can be seen, NG consistently outperform the two baselines, achieving significantly smaller errors in all cases. In fact, *the errors of NG never exceed* 5.1% in all cases. By contrast, the errors by Random and Alternating can be as bad as 59.8% and 48.8%, respectively.

In most of the cases when $K = 2$ or 3, the error by NG is under 2.0%. The largest NG error is 5.02%, observed on Epinions (Jaccard) with 6(eq), while the smallest is 0.01%, achieved on NetHEPT with 2(eq), in which case Random (5.5%) and Alternating (1.4%) both have errors two orders of magnitude larger. As another example, on Epinions (Bernoulli) with 6(neq), NG, Random, and Alternating have an error of 1.83%, 35.7%, and 43.9%, respectively. These results straightaway establish the effectiveness of our Needy Greedy algorithm.

A deeper look into NG solutions. We graphically summarize the amplification factors using the *box-and-whisker diagram* in Fig. 4. In the plot, a box (shown in color blue) represents a subset of the data points from the lower quartile to the the upper quartile. The red line inside the box is the median. Two black bars outside the box correspond to the minimum and maximum, and a red plus sign means the extremum could be considered as an "outlier". Intuitively, the more "compressed" the box is, the more balanced the partition is. The green line is the theoretical lower bound, σ_{all}/B.

As can be seen from Fig. 4, in all cases, the difference between maximum amplification factor α_{max} and the minimum amplification factor α_{min} is small. For instance, for Epinions (Bernoulli) with 6(eq), $\alpha_{max} - \alpha_{min} = 24.8$ (or 7.8% of α_{min}, which is 318.4). Similarly, on NetHEPT and Flixster, with 6(neq), $\alpha_{max} - \alpha_{min}$ is 0.72 (2.7% of α_{min}) and 3.56 (4.9% of α_{min}), respectively. We also observe that the error tends to enlarge when K increases. Besides, the difference is relatively higher for Epinions, and we believe this is because the size of the whole seed set is small (which is 30, compared to 60 on other datasets).

Dynamic programming. We next consider the special case when only two companies compete ($K = 2$). The DP algorithm, being theoretically optimal, must produce the best partitions, and as expected, its accuracy is better than NG: the relative errors are very close to zero on all datasets, as shown in Table 4. For instance, on NetHEPT with 2(eq), DP (with precision up to two decimal places) achieves an error of 0.0004%, while NG achieves 0.013%.

Table 4: Relative error of the maximum amplification factor for Dynamic Programming (with precision up to 2 decimal places)

	Epinions Bernoulli	Epinions Jaccard	NetHEPT	Flixster
$b_1 = b_2$	0.0013%	0.0040%	0.0004%	0.0049%
$\frac{b_1}{b_2} = \frac{1}{2}$	0.0016%	0.0026%	0.0004%	0.0028%

For the same case, if the precision drops to one and zero decimal place, the error increases to 0.007% and 0.23%, respectively. The trend is similar for other cases, and hence we omit them here.

Running time. The running time of DP is reasonable: it finishes within 1.2 *seconds* in all cases. NG is three orders of magnitude faster, completing within 5 *milliseconds* in all cases, while producing allocations with quality comparably close to those of DP. Random and Alternating have similar running time as NG's. We also note that all algorithms require adjusted marginal gains as input, whose computational overhead, however, is much smaller compared to the greedy algorithm for selecting seeds. For example, on Epinions (Jaccard), computing adjusted marginal gains for all seeds takes only 3 *minutes*, compared to 48 *hours* taken by the greedy algorithm to select 30 seeds.

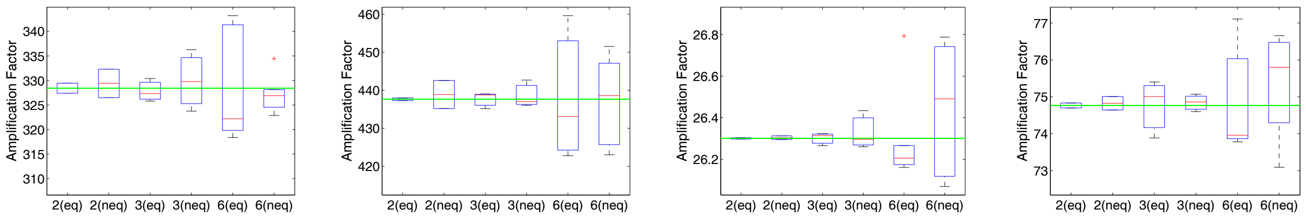
Extended results: NG with L_p-norms. As mentioned in §3.3, the fair allocation objective can also be defined using the L_p-norms (Eq. 3). Our algorithm NG is easily extensible to L_p-norms. The adapted NG iterates through every seed, and in any iteration $j + 1$, it assigns the seed to the company whose deviation $|\sigma_i(S_i, \mathbf{S}_{-i}) - \frac{b_i}{B} \cdot \sigma_{all}|^p$ is the largest among those with budget not yet exhausted. Here $\sigma_i(S_i, \mathbf{S}_{-i})$ is the value by the end of iteration j. Since the power function is monotone,[5] the allocation choice made by NG is essentially the same for all p, and thus it suffices to test only L_1.

In Fig. 5, we compare the objectives of using L_1-norms and of using α_{max}[6]. When the budgets are equal, NG with both objectives has exactly the same performance, but when the budgets are unequal, NG does better on minimizing the maximum amplification factor than it does on minimizing the L_1-norm. For instance, on Epinions (Bernoulli) with 6(neq), the relative error (Eq. 6) is 1.8% and 4.4% for the α_{max} and L_1-norm objectives, respectively. It implies that our choice of objective function, which is minimizing the maximum amplification factor, is a better one.

In sum, we have demonstrated the effectiveness and efficiency of our proposed algorithms (NG and DP) for the FSA problem. We show that NG, while not optimal, produces partitions comparable to those by DP, with very small errors, and it is significantly better than the Random and Alternating baselines. We also show that NG performs reasonably well for various other objective functions.

[5] Note that it's the absolute value that is raised to power p.

[6] We only show results on two Epinions graphs with $K = 3$ and 6; the results for other cases are similar, and hence omitted.



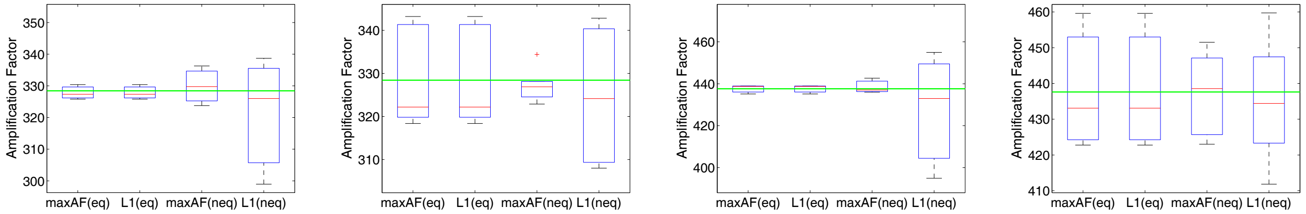
(a) Epinions (Bernoulli)

(b) Epinions (Jaccard)

(c) NetHEPT

(d) Flixster

Figure 4: Box-and-whisker diagrams on amplification factors by Needy Greedy.



(a) Epinions (Bernoulli), $K = 3$

(b) Epinions (Bernoulli), $K = 6$

(c) Epinions (Jaccard), $K = 3$

(d) Epinions (Jaccard), $K = 6$

Figure 5: Box-and-whisker diagrams on amplification factors for comparing objective functions.

6. CONCLUSIONS

Influence maximization has received significant attention recently. However, there is a gap between this problem and real-world viral marketing, and we believe this paper takes a step towards closing this gap by studying influence maximization under a more realistic setting, in which: (i) there is competition, and (ii) the network is owned by a host as in real life, and the competing companies cannot just autonomously set up their campaigns, but have to buy viral marketing as a service from the host.

We posed the novel problem of Fair Seed Allocation in which the host must allocate influential users to competing companies to guarantee "the bang for the buck" for the competitors is as balanced as possible. We proved that the problem is in general NP-hard, and developed two algorithms for it: Needy-Greedy and dynamic programming (for the case of two companies). We performed simulations on real world networks and showed that our algorithms are both effective and efficient.

Investigation of competitive influence propagation and fair seed allocation under other propagation models (e.g., in which the virality of different products may vary), other business models for the host and for the companies, and game-theoretic aspects are three examples of many fruitful directions for further research.

Acknowledgments. This research is supported by a grant from the Business Intelligence Network (BIN) of the Natural Sciences and Engineering Research Council (NSERC) of Canada, and also in part by the Institute for Computing, Information and Cognitive Systems (ICICS) at the University of British Columbia.

7. REFERENCES

[1] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *WINE'07*.

[2] A. Borodin, Y. Filmus, and J. Oren. Threshold models for competitive influence in social networks. In *WINE'10*.

[3] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *WWW'11*.

[4] T. Carnes, C. Nagarajan, S. M. Wild, and A. van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *ICEC'07*.

[5] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD'10*.

[6] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM'10*.

[7] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, 1979.

[8] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Learning influence probabilities in social networks. In *WSDM'10*.

[9] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. A data-based approach to social influence maximization. *PVLDB*, 5(1):73–84, 2011.

[10] A. Goyal, W. Lu, and L. V. S. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM'11*.

[11] X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM'12*.

[12] T. Hogg and G. Szabó. Diversity of User Activity and Content Quality in Online Communities. In *ICWSM'09*.

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

[14] J. M. Kleinberg and É. Tardos. *Algorithm design*. Addison-Wesley, Massachusetts, 2006.

[15] J. Kostka, Y. A. Oswald, and R. Wattenhofer. Word of mouth: Rumor dissemination in social networks. In *SIROCCO'08*.

[16] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *KDD'07*.

[17] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - i. *Mathematical Programming*, 14(1):265–294, 1978.

[18] N. Pathak, A. Banerjee, and J. Srivastava. A generalized linear threshold model for multiple cascades. In *ICDM'10*.

[19] G. Peng and J. Mu. Technology Adoption in Online Social Networks. *Journal of Product Innovation Management*, 28(s1):133-145, 2011.

[20] S. Zhao, R. Meyer, and J. Han. The Enhancement Bias in Consumer Decisions to Adopt and Utilize Product Innovations *Research Collection Lee Kong Chian School of Business (Open Access)*, 2003.