# Adaptive Seeding in Social Networks

Lior Seeman
Department of Computer Science
Cornell Univeristy
Ithaca, NY
lseeman@cornell.cs.edu

Yaron Singer
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA
yaron@seas.harvard.edu

*Index Terms*—social networks; influence maximization; approximation algorithms; stochastic optimization; submodularity;

*Abstract*—The algorithmic challenge of maximizing information diffusion through word-of-mouth processes in social networks has been heavily studied in the past decade. While there has been immense progress and an impressive arsenal of techniques has been developed, the algorithmic frameworks make idealized assumptions regarding access to the network that can often result in poor performance of state-of-the-art techniques.

In this paper we introduce a new framework which we call Adaptive Seeding. The framework is a two-stage stochastic optimization model designed to leverage the potential that typically lies in neighboring nodes of arbitrary samples of social networks. Our main result is an algorithm which provides a constant factor approximation to the optimal adaptive policy for any influence function in the Triggering model.

## I. Introduction

For several decades there have been numerous studies on the way in which information spreads from one individual to another in a society, and an interest in understanding how to instigate such processes of information diffusion. With the exploding adoption of social networking services in the past decade, information diffusion through social networks is becoming easier to analyze and predict, as well as to engineer. A problem which received considerable attention in this context is that of *influence maximization*. As elegantly formulated by Kempe, Kleinberg, and Tardos [1], influence maximization is the algorithmic challenge of selecting a small set of individuals that can serve as early adopters of a new technology and encourage its adoption through word-of-mouth in a social network. As a result of extensive follow-up study to their seminal work, there is now an impressive arsenal of algorithmic and data-mining techniques for maximizing influence in social networks (see e.g. [2]–[10]).

Despite the immense progress made in the past decade, influence maximization algorithms are often designed under idealized assumptions regarding accessibility to the network. Specifically, the guarantees hold for cases where the algorithm can select any node in the network, when in practice it often only has access to a small sample. In marketing applications for example, merchants often apply influence maximization techniques on users who visit their online store, or have engaged in other ways (e.g. subscribe to a mailing list, follow the brand, or installed an application). Similarly, in many other cases, whether due to scale, privacy, or profile-based targeting, influence maximization algorithms are applied on relatively small samples of the network. This naturally raises a concern regarding the performance of current state-of-the-art influence maximization techniques. Since social networks are often characterized by heavy-tailed degree distributions, observing high degree nodes is a rare event. Therefore, if we think of degree as a proxy for influence, for almost all samples – arbitrary subsets of users in the network – application of influence maximization techniques will be ineffective.

It therefore seems compelling to explore alternative approaches that can somehow overcome this barrier. Consider momentarily the possibility of applying influence maximization algorithms on *neighbors* of the sample. Since high degree nodes have (by definition) many neighbors, one can hope such nodes will be connected to the sample. Assuming social networks indeed posses this property, this would imply that influence maximization techniques applied on neighbors of the sample could yield substantially improved results. Indeed, as we now briefly discuss, analytical and empirical evidence support this intuition.

**The friendship paradox and beyond.** First discovered by Feld [11], the *friendship paradox* states that in any network the expected degree of a node is bounded from above by the expected degree of a neighbor. Feld shows this by characterizing the expected degree of a neighbor in terms of the expectation and variance of the degree distribution in the network. Applying this characterization, it is easy to show (see full version for details) that for a power-law graph with parameter $\alpha > 1$ (i.e. where the likelihood of observing a node of degree $d$ is proportional to $d^{-\alpha}$), the ratio between the expected degree of a node and the expected degree of a neighbor is $O(n^{1-\frac{1}{\alpha}})$. Dramatic gaps between degrees of nodes and their friends were observed empirically in various experiments in online social networks [12]. In recent work, Lattanzi and Singer show asymptotic gaps between the total coverage (total number of disjoint neighbors) of random samples of nodes of logarithmic size and the average coverage of their neighbors in generative models of networks with power law degree distributions [13].

### A. Adaptive Seeding

The above discussion implies that while an overwhelmingly large majority of nodes will be ineffective for the purposes of
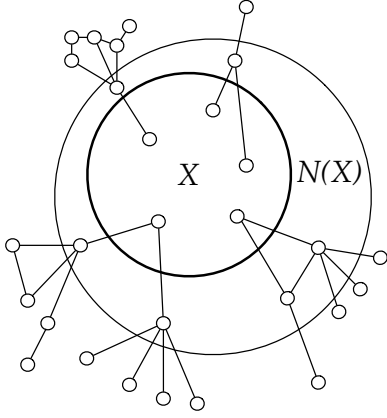
Fig. 1. An illustration of the Adaptive Seeding model. All nodes in $X$ are accessible in the first stage and in the second stage each node in $\mathcal{N}(X)$ becomes accessible with some independent probability when its neighbor in $X$ is seeded.

spreading information, there is a great deal of potential in seeding their *neighbors*. A natural alternative to having the entire budget invested in the sample is an adaptive two-stage seeding process: in the first stage some budget is used to select nodes from the sample to invite their neighbors to join the accessible set, and in the second stage, after some neighbors make themselves accessible to be seeded, the remaining budget is used to select a set of influencers from the (hopefully larger) accessible set. Intuitively, a good policy is one which makes an initial investment that is likely to bring influential nodes to become accessible.

**A model for Adaptive Seeding.** To formalize the notion of adaptive seeding we propose the following model. In the Adaptive Seeding model there is a network $G = (V, E)$, seeding budget $k \in \mathbb{N}$, and an influence function $f : 2^V \to \mathbb{R}_+$ that quantifies the expected number of nodes in the network that are influenced when a subset is seeded. In addition, there is an arbitrary subset $X \subseteq V$ which represents the sample which is initially accessible to be seeded, and for each neighbor $i$ of the sample there is a parameter $p_i \in [0, 1]$ which represents the probability that $i$ becomes accessible (willing to become an influencer in exchange for a unit of the budget) if one of its neighbors in $X$ is seeded[1]. An *adaptive seeding policy* optimizes a two-stage process. In the first stage the policy selects nodes in $X$ into the seeding set $S$. In the second stage, after $S$ is selected, each node $i$ which is a neighbor of $S$ becomes accessible to be seeded independently with probability $p_i$. The goal is to select a subset $S \subseteq X$ of size $t \leq k$ during the first stage s.t. the expected maximal influence achievable in the second stage with the remaining budget $k - t$ is maximal, where the expectation is taken over all possible realizations of neighbors of $S$.

---

[1] If $i \in X \cap \mathcal{N}(X)$, where $\mathcal{N}(X)$ is the set of neighbors of $X$, then trivially $p_i = 1$. We therefore assume without loss of generality that $X \cap \mathcal{N}(X) = \emptyset$.

**Key points about the model.** Before we continue discussing approaches for solving the problem, we highlight several key points. First, when $X = V$ we have the influence maximization problem as formulated by Kempe, Kleinberg, and Tardos and thus our Adaptive Seeding model can be seen as a generalization of their classic influence maximization model. Second, we stress that the process through which nodes in the second stage are recruited is not an influence process. The influence function models a word-of-mouth process that occurs without incentives – no seeding budget is further allocated. The two stage seeding process models the likelihood that a neighbor of $X$ is willing to become an influencer if given a unit of the budget. This can be seen as a bayesian utility model with no externalities. Lastly, while it is tempting to consider policies of more than two stages, empirical evidence leads us to believe that the benefit in influence potential beyond two stages is marginal, and furthermore we conjecture that optimizing a multi-stage process will be computationally hard.

### B. Techniques for Stochastic Optimization

We now turn to discussing stochastic optimization approaches for solving the adaptive seeding problem. As one may expect, the difficulty of the problem largely depends on the influence function we aim to optimize.

**Sample Average Approximation.** A common technique in stochastic optimization is the Sample Average Approximation (SAA) method [14]. The general idea is to sample scenarios that may realize in the second stage, and find a solution which is optimal with respect to the sampled data. In our setting, a combinatorial analogue of the SAA approach can be used when the influence function is simple (e.g. when the function is additive). In such cases we can sample instances from the product distribution over the set of neighbrors and aim to optimize a particular objective function defined over the sampled data. In some special cases, the objective function is submodular, and the celebrated algorithm by Nemhauser et al. [15] can be used to guarantee a $(1 - 1/e - \epsilon)$ approximation for any $\epsilon > 0$. Unfortunately however, the objective function is submodular only for very special cases of influence functions. For the well-studied models such as *Independant Cascade* and *Linear Threshold*, the objective function is *not submodular* and this technique becomes ineffective. We give a detailed description and further discuss this approach in Section IV-A.

**Non adaptive policies.** In cases where the objective is complex and adaptive policies cannot be easily found, an alternative approach is to design *non adaptive* policies instead. A non adaptive policy commits on the $k$ nodes it will select in the first and second stage before the scenarios are realized. Non adaptive policies are weaker than adaptive ones since any adaptive policy can simulate a non adaptive policy by ignoring the information on the realizations of the scenario. While such policies are often strictly weaker, it is usually much easier to

design non adaptive policies than adaptive ones. When the ratio between the optimal adaptive policy and non adaptive one (i.e. the *adaptivity gap*) is small, using a non adaptive policy may serve as a reasonable approximation to the optimal adaptive policy. In our case, at a first glance it seems as if non adaptive policies are useless when it comes to problems like the ones we study here: In Section IV-C we use a simple example to show that no non adaptive policy can have a finite approximate ratio, even for very simple influence functions.

**Randomized-and-relaxed policies.** Although non adaptive policies seem hopeless, their relative simplicity is appealing. We therefore explore an alternative concept which we call *randomized-and-relaxed non adaptive policies*: A randomized-and-relaxed non adaptive policy commits to a (possibly randomized) solution for the second stage, a priori to the realization of the scenarios. An optimal randomized-and-relaxed non adaptive strategy selects nodes for the first and second stage s.t. at most $k$ nodes are used *in expectation*. Note that by respecting the budget only in expectation, randomized-and-relaxed non adaptive policies compensate for their a priori commitment, and in particular cannot be simulated through adaptive policies. These concepts are therefore somewhat orthogonal, and their additional strength makes randomized-and-relaxed non adaptive policies useful for solving our problem.

### C. The main result: adaptive policies for the Triggering model

As discussed above the algorithmic challenge arises in attempting to optimize influence functions such as the ones in the Independent Cascade and the Linear Threshold models. These models, as well as others, can be generalized to a broader class known as the Triggering model. In the following section we discuss these models in further depth.

Our main result in this paper shows that any influence function in the Triggering model can be adaptively seeded. That is, we show that there is an algorithm which finds an adaptive policy that is a constant factor approximation to the optimal adaptive policy, for any influence function in this class. The main idea is to design an algorithm which approximates the optimal randomized-and-relaxed non adaptive policy and show that its solution is actually a good approximation to the optimal adaptive policy.

At a high level, we first construct a concave function and formulate our problem as optimizing the concave function under a mix of integral and fractional linear constraints and show that solutions to this objective are an upper bound on adaptive policies. We then present an algorithm which mimics a gradient-ascent process, taking steps in the direction of the densest contributions to the concave objective. Then, somewhat surprisingly, we conclude by showing that our algorithm in fact obtains a constant factor approximation to the optimal *adaptive policy*.

### D. Paper Organization

Following an overview of related work we provide preliminaries in the following section. In Section III we present the main details of the approximation algorithm for adaptive seeding in the Triggering model. We discuss further details in Section IV. Proofs and assisting lemmas omitted from this version can be found in the full version of the paper.

### E. Related Work

The problem of identifying influential users in a social network was first posed by Domingos and Richardson in [16], [17] and has been extensively studied since. In their seminal work, Kempe, Kleinberg and Tardos formulate the *influence maximization problem* which elegantly models the algorithmic challenge of selecting individuals who will maximize adoption through a word-of-mouth effect in a social network [1]. The Adaptive Seeding model we present can be seen as a generalization of their model, and our result is for the influence models formalized in their work. Since their work, there has been ongoing research on characterizing various influence models for the purpose of designing approximation algorithms [4], [5], [9], [18], [19].

Asadpour et al. [20] and Golovin and Krause [21] study a multi stage variant of the influence maximization problem where seeds can be chosen sequentially. These models are substantially different from Adaptive Seeding, as the entire set of possible seeds is available at every stage and the optimization is only over their influence. Hartline et al. [22] and Haghpanah et al. [23] consider revenue maximization of digital goods where the individual's valuation of an item is a function of its adoption by her neighbors. Adaptive Seeding is similar in that seeding neighbors during the second stage depends on having their neighbors seeded as well.

The adaptive seeding model is a stochastic optimization framework (see [24] for a survey). Immorlica et al. [25] study two-stage with recourse minimization problems for which the feasibility constraints are stochastic, and the costs for resources varies between stages. Similar problems and various other techniques were studied by Ravi and Sinha [26], Gupta et al. [27], Srinivasan [28], and Shmoys and Swamy [29], [30]. Our problem is different as it is a maximization problem, and has different structure of constraints between the two stages. Another class of stochastic optimization problems studied are those of maximizing an objective under some budget constraint. One example is the *stochastic knapsack* problem which was first introduced by Kleinberg et al. [31]. Dean et al. [32] studied it in an adaptive scenario, and define the notion of *Adaptivity Gap*. They construct a non-adaptive policy that achieves a constant fraction of the value of the best adaptive policy. Similar ideas were used by Gupta et al. [33] for the *stochastic orienteering* problem. For our problem a standard non-adaptive policy does not provide good guarantees. We are not aware of any literature that studies problems where the decision in the first stage constrain the possible actions on the second stage in addition to the constraint of the joint budget.

## II. PRELIMINARIES

Adaptive Seeding is a two stage process where in the first stage we are given an initial set of nodes $X \subseteq V$ which can be

seeded, and in the second stage each neighbor of the seeded nodes appears with some independent probability $p_j$ and can be seeded. For any set of neighbors $R$ we use $p_R$ to denote its likelihood to be realized, i.e. $p_R = \prod_{i \in R} p_i \prod_{i \notin R}(1 - p_i)$. Given a budget $k \in \mathbb{N}$, the goal of an adaptive policy is to select a subset $S \subseteq X$ of size $t \le k$ in the first stage s.t. the influence function $f : 2^V \to \mathbb{R}_+$ is maximized under the remaining budget, in expectation over all possible realizations of neighbors of $S$. To ease notation down the road, we assume that the influence function is only affected by neighbors of $X$, i.e. $\forall S \subseteq X, f(S) = 0$. Minor adjustments to our proofs and algorithms easily extend to the general case. Using $\mathcal{N}(Q)$ to denote the set of neighbors of $Q \subseteq X$, for any $S \subseteq X$ let $S_R = \operatorname{argmax}\{f(T) : T \subseteq R \cap \mathcal{N}(S), |S| + |T| \le k\}$. The optimal adaptive seeding solution is:

$$S^* := \operatorname{argmax}_{S \subseteq X}\Big\{\sum_R p_R \cdot f(S_R)\Big\}.$$

**Influence functions.** The main result in this paper focuses on maximizing influence as defined by the *Triggering model* which is a generalization of the well-studied *Independent Cascade* and *Linear Threshold* models formulated in [1]. In all cases, for a given set $S$, the value of the influence function $f(S)$ is the expected number of nodes *influenced* by a stochastic process in the network when all nodes in $S$ are influenced initially. In Independent Cascade every edge in the graph is associated with some probability, and an influenced node has a single chance to influence its neighbor with the probability associated with that edge. In the Linear Threshold model every edge is associated with some weight in $[0, 1]$, every node has a threshold chosen uniformly at random in $[0, 1]$ and a node is influenced if the total edge weights of its influenced neighbors exceeds its threshold. The Triggering model is a generalization of these models where every node $v$ chooses independently a random *triggering set* according to some distribution over subsets of its neighbors, and is influenced if any of the nodes in its triggering set is influenced. We assume that we are only given an efficient way to sample realizations of the model and do not assume we have access to a value oracle which given a set $T$ returns its expected influence. In Section IV-A we describe results for other models such as the *Voter model* [19], [34].

**Approximation.** The Max-k-Cover problem can easily be reduced to influence maximzation in the Triggering model, and thus following the result by Feige [35], in this model it is NP-hard to approximate the optimal adaptive seeding policy within a factor better than $1 - 1/e$ assuming $P \ne NP$. We will therefore aim for polynomial-time approximation algorithms and allow for randomization. We say that an algorithm is an $\alpha$-approximation if w.h.p. the ratio between the optimal adaptive policy and the policy returned by the algorithm is at most $\alpha$.

**Randomized-and-relaxed policies.** The main technique we

use in this paper involves designing an algorithm which approximates a randomized-and-relaxed non adaptive version of the above problem. It is important to note that our results do not directly use randomized-and-relaxed non adaptive solutions, but rather this concept guides our approach. We say that a policy is *non adaptive* if already in the first stage, *before the nodes in the second stage realize*, the policy commits on both $S \subseteq X$ and $Q \subseteq \mathcal{N}(S)$ it will select. In Section IV-C we show that the ratio between the expected value of the optimal adaptive policy and the expected value of the optimal non adaptive policy is unbounded. We therefore use *randomized-and-relaxed* non adaptive policies which only commit to the probability they will select each node that appears in the second stage. Such a policy is a set $S \subseteq X$ and a weight $q_i$ for each $i \in \mathcal{N}(S)$ that describes the probability the algorithm will select $i$ if it realizes. The policy's guarantee is to select at most $k$ nodes *in expectation*. The optimal randomized-and-relaxed non adaptive policy is the following optimization problem:

$$\max \quad \sum_{T \subseteq \mathcal{N}(S)} \Big( \prod_{i \in T} p_i q_i \prod_{i \notin T}(1 - p_i q_i) \Big) f(T) \quad (1)$$
$$\text{s.t.} \quad S \subseteq X \quad (2)$$
$$\quad |S| + \sum_{i \in \mathcal{N}(S)} p_i q_i \le k \quad (3)$$

For brevity we will use $\mathcal{F}$ to denote solutions $(S, q)$ that respect (2) and (3) above with the extra property that $q$ has non zero values only on nodes in $\mathcal{N}(S)$. We call the ratio between an optimal adaptive solution and an optimal randomized-and-relaxed non adaptive one the *adaptivity gap* of the problem.

### III. ADAPTIVE POLICIES FOR THE TRIGGERING MODEL

In this section we describe our technique for finding an approximate adaptive policy. An overview of our approach can be summarized as follows.

1) We begin by constructing a concave function which approximates the objective of the randomized-and-relaxed non adaptive problem in the Triggering model. In general, maximizing influence in the Triggering model requires solving an NP-hard problem over exponentially many graphs. The construction converts the objective to a concave relaxation of a succinct coverage problem.

2) We next show that in our problem, the gap between the optimal adaptive solution and the optimal randomized-and-relaxed non adaptive solution is only a constant factor. This implies we can use such policies to approximate an adaptive policy, at the cost of having solutions that respect the budget in expectation.

3) Despite having a concave relaxation, producing good approximations is challenging due to the integral and fractional constraints of the optimization problem. Using the above construction, we use a hill-climbing algorithm which approximates the objective under these constraints. At each iteration the algorithm takes a step in the direction of the densest marginal contribution to the concave objective. The main challenge is in designing subroutines that find these densest contributions

with good guarantees in terms of value and time of convergence.

4) In the final step we argue that in our problem any solution to the concave relaxation can in fact be converted to an adaptive policy, with only a constant loss in the approximation guarantee.

### A. Polynomial concave representation of influence

We now describe a construction of a concave function which approximates the objective of the randomized-and-relaxed non adaptive problem in the Triggering model. We first construct a concave function which approximates the objective though may require exponential representation. We then reduce its representation via sampling to be polynomial sized while maintaining a good approximation with high probability.

**A relaxation of the randomized-and-relaxed objective.** We begin by interpreting influence in the Triggering model as follows. First notice that each realization of the triggering set can be described as a graph, where there is an edge from $u$ to $v$ if and only if $u$ is in the triggering set of $v$. Next, consider a family of graphs $\mathcal{G}$, where the distribution over graphs is given by the probability this graph is the result of a realization of the Triggering model. Given a realized graph $G_i$, the number of nodes a set influences is the number of nodes it can reach in $G_i$. Therefore the influence of a set is the number of nodes it can influence in expectation over all realizations in $\mathcal{G}$. Using $f_G(T)$ to denote the number of nodes reachable from $T \subseteq V$ in a graph $G$ we can write the influence function as:

$$f(T) = \sum_{G_i \in \mathcal{G}} p(G_i) f_{G_i}(T)$$

To optimize the objective in our problem, we will be interested in fractional relaxations. That is, for every realization $G_i$ we wish to evaluate the number of nodes in $G_i$ covered in expectation when selecting nodes with some probability. Let $\mathcal{C}_i(u)$ be the set of all nodes that have a path to $u$ in $G_i$. We will use the multilinear relaxation of the coverage function in $G_i$ defined for every $z \in [0,1]^n$ as:

$$\hat{f}_{G_i}(z) = \sum_{T \subseteq V} \prod_{j \in T} z_j \prod_{j \notin T} (1 - z_j) f_{G_i}(T)$$
$$= \sum_{u \in V} 1 - \prod_{j \in \mathcal{C}_i(u)} (1 - z_j)$$

Let $F(q) = \sum_{G_i \in \mathcal{G}} p(G_i) \hat{f}_{G_i}(p \circ q)$, where $p \circ q$ is the vector which is the element-wise multiplication of $p$ and $q$. The objective of equation (1) can be stated as $\max_{\mathcal{F}} F(q)$. We use the notation $F(q|S)$ to mean the function that gets the value of $F(q)$ when $(S, q) \in \mathcal{F}$ and 0 otherwise.

Although $\hat{f}_{G_i}(\cdot)$ is not concave, it is well known that it can be approximated by a concave function: For any $z \in [0,1]^n$ we have that $L_{G_i}(z) \geq f_{G_i}(z) \geq (1 - 1/e) L_{G_i}(z)$ where $L_{G_i}(z) = \sum_{u \in V} \min\left\{1, \sum_{j \in \mathcal{C}_i(u)} z_j\right\}$. Since concave functions have many desirable properties, we will work with

the concave relaxation of $F$, $\mathcal{L}(q) = \sum_{G_i} p(G_i) L_{G_i}(p \circ q)$. Our objective is to optimize $\mathcal{L}(\cdot|\cdot)$ over the set $\mathcal{F}$ where:

$$\mathcal{L}(q|S) = \sum_{G_i} p(G_i) \sum_{u \in V} \min\left\{1, \sum_{j \in \mathcal{C}_i(u) \cap \mathcal{N}(S)} p_j q_j\right\}.$$

Since there are potentially exponentially many elements in $\mathcal{G}$ we will construct a sampled mean of $\mathcal{L}$ and show it provides a good approximation of $\mathcal{L}$. Let $\hat{\mathcal{L}}(z) = \sum_{G_i} p(G_i) L_{G_i}(z)$. Let $G_1, \ldots, G_N$ be $N$ i.i.d samples of $\mathcal{G}$, and let $\hat{\mathcal{L}}_N(z) = \frac{1}{N} \sum_{i=1}^{N} L_{G_i}(z)$. To show $\mathcal{L}(\cdot)$ can be approximated efficiently we would like to show that for any $z \in [0,1]^n$ the absolute difference between the values of $\hat{\mathcal{L}}(z)$ and $\hat{\mathcal{L}}_N(z)$ is small with high probability.

**Lemma 3.1.** *For all $\beta, \epsilon > 0$, all $n \in \mathbb{N}$ and all $N \geq O(1)\frac{n^2}{\epsilon^2}[n \ln(\frac{n^2}{\epsilon}) + \ln(\frac{1}{\beta})]$,*

$$\Pr\{\sup_{x \in [0,1]^n} (|\hat{\mathcal{L}}_N(x) - \hat{\mathcal{L}}(x)|) \geq \epsilon\} \leq \beta.$$

To prove this lemma we need to show the function is Lipschitz continuous and to bound the moment generating function of a related random variable. We can then use a uniform exponential convergence bound to get the result. We defer details to the full version of the paper.

**Efficient representation.** Given the above, we can now describe the construction of our objective function. For any desired precision $\epsilon > 0$, and any desired confidence $\beta > 0$ we first sample $N = O(1)\frac{n^2}{\epsilon^2}[n \ln(\frac{n^2}{\epsilon}) + \ln(\frac{1}{\beta})]$ graphs i.i.d from $\mathcal{G}$ by following the distribution defined by the Triggering model. Using these samples we will create the following bipartite graph. Given the set $X$, for every $v \in \mathcal{N}(X)$ and $u \in V$ we will add a copy $u_i$ of $u$ to the set $\mathcal{U}$ and edge $(v, u_i)$ to its edge set $\mathcal{E}$ if and only if there is a path from $v$ to $u$ in $G_i$, for all $i \in 1, \ldots, N$. For every $u \in \mathcal{U}$ let $\mathcal{C}(u)$ be the set of vertices that cover $u$ in the bipartite graph. For a given $S \subseteq X$ we define our objective function to be:

$$L(q|S) = \frac{1}{N} \sum_{u \in \mathcal{U}} \min\{1, \sum_{i \in \mathcal{C}(u) \cap \mathcal{N}(S)} p_i q_i\}$$

**Corollary 3.2.** *For any $\epsilon > 0$, any $\beta > 0$, and all $0 \leq \alpha \leq 1$, $L(\cdot|\cdot)$ can be constructed in polynomial time such that an $\alpha$-approximation algorithm for $L(\cdot|\cdot)$ is a $(\alpha - \epsilon)$-approximation algorithm for $\mathcal{L}(\cdot|\cdot)$.*

### B. The adaptivity gap

We now prove a bound on the adaptivity gap which justifies the use of an approximation to a randomized-and-relaxed policy. We use $OPT_A, OPT_{NA}$ to denote the expected values of the optimal adaptive and randomized-and-relaxed non adaptive policies, respectively. We use $OPT_{\mathcal{L}}$ to denote the optimal solution for the $\mathcal{L}(\cdot|\cdot)$ function as defined above. Recall that randomized-and-relaxed non adaptive policies compensate for committing on nodes in advance by respecting the cardinality constraint in expectation.

**Lemma 3.3.** $OPT_A \leq OPT_{\mathcal{L}} \leq \left(e/(e-1)\right)OPT_{NA}$.

*Proof:* We will prove that $OPT_A \leq OPT_{\mathcal{L}}$ and since $(1 - 1/e)\mathcal{L}(q|S) \leq F(q|S) \leq L(q|S)$ for any $(S,q) \in \mathcal{F}$ we know that $OPT_{\mathcal{L}} \leq (e/(e-1))OPT_{NA}$. We will use $OPT_{\mathcal{L}(\cdot|S)}$ to denote the optimal solution in $\mathcal{L}(\cdot|\cdot)$ when $S$ is selected in the first stage. We will consider the objective of maximizing the expected number of nodes in the Triggering model as coverage over an exponentially large universe $\mathcal{U}$, as we discussed above.[2]

Let $S \subseteq X$ represent the set of $k - t$ nodes selected by the optimal adaptive policy in the first stage, and let $R_1, \ldots, R_m$ be all possible realizations of nodes in $\mathcal{N}(S)$. First, we claim that the optimal adaptive policy can be interpreted as a feasible solution in $\mathcal{L}(\cdot|S)$. In the optimal adaptive policy, in each realization there are $t$ nodes which are allocated. Let $\ell = |\mathcal{N}(S)|$, and let $\{y_1, \ldots, y_\ell\}$ be the set of all nodes in $\mathcal{N}(S)$. For each realization $R_i$ we define the function $\alpha_i : \mathcal{N}(S) \to [0,1]$ as:

$$\alpha_i(y_j) = \begin{cases} p(R_i) & \text{if } y_j \text{ is allocated in } R_i \\ 0 & \text{otherwise} \end{cases}$$

where $p(R_i)$ is the probability that a realization $R_i$ occurs. Now let $\alpha(y_j) = \sum_{i=1}^m \alpha_i(y_j)$, and let $p(y_j)$ be the probability that $y_j$ is realized. Clearly, $\alpha(y_j) \leq p(y_j)$ and therefore there is some $q_j \leq 1$ s.t. we can write $\alpha(y_j) = p(y_j)q_j$. Since the solution is feasible in each realization $R_i$ we know that $\sum_{j=1}^\ell \alpha_i(y_j) \leq t \cdot P(R_i)$. As all realizations sum to 1:

$$\sum_{i=1}^m \sum_{j=1}^n \alpha_i(y_j) \leq t \iff \sum_{j=1}^n \sum_{i=1}^m \alpha_i(y_j) \leq t$$
$$\iff \sum_{j=1}^n \alpha(y_j) \leq t$$
$$\iff \sum_{j=1}^n p(y_j)q_j \leq t.$$

We can therefore consider $(S,q)$ as a feasible solution to $\mathcal{L}(\cdot|S)$ that corresponds with the optimal adaptive policy. We will next show that the value of this solution yields a higher value in $\mathcal{L}(\cdot|S)$ than the expected value of $OPT_A$.

Let $u$ be an element in the universe, and let $\mathcal{C}(u)$ be the nodes that cover $u$. Fix an arbitrary ordering $\sigma$ on the nodes s.t. $\sigma(y_1) < \sigma(y_2) < \ldots < \sigma(y_\ell)$. We say that $u$ is owned by $y_j \in \mathcal{C}(u)$ in the realization $R_i$ if $y_j$ is a member of the optimal adaptive solution in that realization and there is no node $y_r \in \mathcal{C}(u)$ s.t. $\sigma(y_r) < \sigma(y_j)$ which is also part of the optimal adaptive solution in that realization. Similarly as above, we define:

$$\alpha_i^u(y_j) := \begin{cases} p(R_i) & \text{if } y_j \text{ owns } u \text{ in } R_i \\ 0 & \text{otherwise} \end{cases}$$

And we define $\alpha^u(y_j) := \sum_{i=1}^m \alpha_i^u(y_j)$. Stated in these terms we have that:

$$OPT_A = \sum_{u \in \mathcal{U}} \min\{1, \sum_{j \in \mathcal{C}(u)} \alpha^u(y_j)\}$$

It is easy to see that for every $u$ and every $y_j$ we have that $\alpha^u(y_j) \leq \alpha(y_j)$ which implies:

$$OPT_A = \sum_{u \in \mathcal{U}} \min\{1, \sum_{j \in \mathcal{C}(u)} \alpha^u(y_j)\}$$
$$\leq \sum_{u \in \mathcal{U}} \min\{1, \sum_{j \in \mathcal{C}(u)} p(y_j)q_j\}$$
$$\leq OPT_{\mathcal{L}(\cdot|S)}$$

Therefore, in expectation, the value of the optimal adaptive solution is no greater than the value of its corresponding solution in $L(\cdot|S)$, which is a lower bound on $OPT_{\mathcal{L}}$. ∎

### C. The non adaptive algorithm

Following the construction of Section III-A, finding a non adaptive policy reduces to solving the optimization problem:[3]

$$\max_{\mathcal{F}} L(q|S).$$

Recall that a feasible solution $(S,q) \in \mathcal{F}$ is a pair $S \subseteq X$ and a vector $q$, s.t. $|S| + \sum_{i \in \mathcal{N}(S)} p_i q_i \leq k$ and $q$ has non zero values only for elements in $\mathcal{N}(S)$. We now describe the approximation algorithm for this optimization problem and discuss its main components. The main procedure of the algorithm finds the densest contribution to $L(q|S)$ – the node in $X$ and allocation of a fraction of budget $\xi$ on its neighbors that will have the largest marginal contribution per spend to $L(q|S)$ – and seeds a node $x \in X$ (i.e. adds $x$ to $S$) if it hadn't already been seeded. We use $x \vee y$ to denote the element-wise maximum operator on $x, y \in [0,1]^n$. Omitted proofs and lemmas can be found in the full paper.

| **NONADAPTIVESEEDING** |
| --- |
| **input:** budget $k$ |
| initalize: $S = \emptyset, q = \vec{0}$ |
| 1. while $|S| + \sum_i p_i q_i \leq k$ |
| $\quad k' \longleftarrow k - (|S| + \sum_i p_i q_i)$ |
| $\quad (x, q') \longleftarrow$ FINDDENSESTCONTRIBUTION$(S, q, k')$ |
| $\quad (S, q) \longleftarrow (S \cup \{x\}, q \vee q')$ |
| 2. $(S', q') \longleftarrow \arg\max_x$ DENSE$(x, \vec{0}, k)$ |
| 3. If $(L(q'|S') > L(q|S))$ then $(S, q) \longleftarrow (S', q')$ |
| **return:** $(S, q)$ |

For every $(T, r) \in \mathcal{F}$ the *marginal density* of $(T, r)$ with respect to $(S, q)$ is the ratio between the marginal contribution of adding $(T, r)$ to $(S, q)$ and its marginal cost. The algorithm iteratively seeks for $(x, r') \in \mathcal{F}$ with the largest marginal density that can be added to the current solution. Before we

---

[2]Note that for the sake of this proof we are not concerned about the size of the input and recall that we have an efficient manner to approximate the objective function within any desirable accuracy.

[3]We ignore the factor of $\frac{1}{N}$ in $L(q|S)$ throughout this section since it does not affect the solution the algorithm obtains.

continue to describe the procedures for computing densities, we justify this strategy using the following lemma.

**Lemma 3.4.** *Let* $(S,q),(T,r) \in \mathcal{F}$. *There exist* $(x,r') \in \mathcal{F}$ *such that the marginal density of* $(x,r')$ *with respect to* $(S,q)$ *is at least the marginal density of* $(T,r)$ *with respect to* $(S,q)$.

**Finding the densest contributions.** Computing the densest marginal contribution depends on whether or not the node has already been seeded. For nodes which are not already seeded by the algorithm there is an additional cost of 1 for including the node in the solution, which is amortized over the contribution of its neighbors. For such cases, we use the procedure $\text{DENSE}(x,q,\xi)$ to find the change to the vector $q$ of $\mathcal{N}(x)$ that adds the most value using a budget of $\xi$. The procedure returns both the density value and the values of the $q'$ variables. The DENSE procedure is simply an LP, and we defer its formal description to the full version. The procedure is summarized in the figure below:[4]

---
**DENSESTUNSEEDED**
**input:** $x \in X \setminus S$, current solution $(S,q)$, budget $t$
   for all $i \in \{0,1,\ldots,\lfloor \log_{(1+\delta)}(t) \rfloor\}$
      $d(i), \bar{q}(i) \longleftarrow \text{DENSE}(x,q,(1+\delta)^i)$
   $j = \text{argmax}_j d(j)$
**return:** $\bar{q}(j)$

---

**Lemma 3.5.** *For any* $0 < \delta < 1$, *and for all* $x \in X \setminus S$, *the above algorithm is a* $(1+\delta)$-*approximation of the densest marginal contribution that adds* $x$. *It runs in time* $poly(\frac{\log(k)}{\delta}, n, N)$.

For nodes which are already seeded, the penalty for their inclusion is already accounted for, and their marginal density only depends on their neighbors. For any such node $x \in X$, the function $L(\cdot|x)$ is concave and similar to the continuous greedy heuristic in [36] it can be shown that the maximal marginal density can be found coordinate-wise. A concave function $g(\cdot)$ is concave in every coordinate, and for any concave function $g_i(\cdot)$ defined over the interval $[0,a]$, the function $g_i(z)/z$ is always non-increasing in $z$ and thus its maximum is at $z=0$. So, for general concave functions, this won't suffice since we seek marginal contributions that are strictly positive. Note however that for the special case where $g_i(\cdot)$ is linear on the interval, $g_i(z)/z$ is constant and achieves its maximum at $a > 0$. Fortunately, while our function is not linear, it is piecewise linear. We denote $\Delta_i L(\xi,q|S)$ to be the marginal contribution to $L$ in coordinate $i$ when adding $\xi$ given a solution $S, q$:

$$\Delta_i L(\xi,q|S) = \sum_{u:i \in \mathcal{C}(u)} \min \left\{ 1, \sum_{j \in \mathcal{N}(S) \setminus \{i\}} p_j q_j + p_i(q_i + \xi) \right\}$$
$$- \sum_{u:i \in \mathcal{C}(u)} \min \left\{ 1, \sum_{j \in \mathcal{N}(S)} p_j q_j \right\}$$

[4]More precisely, DENSESTUNSEEDED also checks the point $\text{Dense}(x,q,t)$.

The function $\Delta_i L(\cdot,q|S)$ is piecewise linear and strictly positive when there exists some $u : i \in \mathcal{C}(u)$ for which $1 - \sum_{i \in \mathcal{C}(u)} p_i q_i > 0$. For each $i \in \mathcal{N}(S)$ we can therefore take the maximal $\xi > 0$ for which the function remains linear, and have the desired property. In particular, using $\mathcal{U}^-$ to denote the set of all elements $u$ in the universe for which $1 - \sum_{i \in \mathcal{C}(u)} p_i q_i > 0$ this is:

$$\xi_i^* = \min_{u \in \mathcal{U}^- : i \in \mathcal{C}(u)} \left\{ (1 - \sum_{i \in \mathcal{C}(u)} p_j q_j)/p_i \right\}$$

We denote $\widehat{\mathcal{N}}(x) = \{j | j \in \mathcal{N}(x), q(j) < 1\}$ and summarize the procedure below:

---
**DENSESTSEEDED**
**input:** $x \in S$, current solution $(S,q)$, budget $t$
   $i = \text{argmax}_{j \in \widehat{\mathcal{N}}(x)} \big| \{u \in \mathcal{U}^- \mid j \in \mathcal{C}(u)\} \big|$
   $\xi_i^* = \min_{u \in \mathcal{U}^- : i \in \mathcal{C}(u)} \{(1 - \sum_{i \in \mathcal{C}(u)} p_j q_j)/p_i\}$
   $q_i' = q_i + \min \left\{ \xi_i^*, 1 - q_i, t/p_i \right\}$
**return:** $q' = (q_{-i}, q_i')$

---

**Lemma 3.6.** *For all* $x \in X$, DENSESTSEEDED *finds the densest marginal contribution using* $x$, *and runs in time* $poly(n,N)$. *In addition, the number of calls to the procedure is at most* $O(nN)$.

**Approximation guarantee.** Since we can find the densest contributions within any desired approximation ratio, we can analyze the approximation ratio in an analogous manner to that of Nemhauser et. al [15]. The main component involves comparing the density of each iteration's contribution to the density of the optimal solution. See full version for the proof.

**Theorem 3.7.** *For any* $\epsilon > 0$, NONADAPTIVESEEDING *is a* $(\frac{1-e^{-1}}{2-e^{-1}} - \epsilon)$-*approximation algorithm for* $\max_{\mathcal{F}} L(q|S)$, *and runs in* $poly(n, N, \frac{\log(k)}{\epsilon})$ *time.*

Together with the adaptivity gap this implies the following.

**Corollary 3.8.** *For any* $\epsilon > 0$ *and any* $\beta > 0$ *there is an algorithm that finds a randomized-and-relaxed non adaptive policy with expected value* $(\frac{(1-e^{-1})^2}{2-e^{-1}} - \epsilon)OPT_A$ *with probability* $1 - \beta$ *in time* $poly(n, \ln(\frac{1}{\beta}), \frac{\log(k)}{\epsilon})$.

*D. The main result*

The approximation guarantee on the randomized-and-relaxed policy is conditioned on respecting the budget constraint in expectation during the second stage. We now show that any randomized-and-relaxed policy can also be interpreted as an adaptive policy with an approximate value.

**Lemma 3.9.** *For any randomized-and-relaxed policy* $(S,q) \in \mathcal{F}$ *such that* $t = k - |S| > 1$, *there exist an adaptive policy with value* $\geq (\frac{e-1}{4e} - \frac{1}{t})\mathcal{L}(q|S)$.

*Proof:* We consider the adaptive policy that selects the same set $S$ in the first stage. Consider a random process which

selects each node $j \in \mathcal{N}(S)$ with probability $q_j/c$, for some constant $c$, into a set $\hat{T}$ if it is realized, and define the set $T$ as:

$$T = \begin{cases} \hat{T} & \text{if } |\hat{T}| \leq t \\ \emptyset & \text{otherwise} \end{cases}$$

The policy runs the proccess and chooses $T$ as the set on the second day. Thus, for every realization of nodes in the second stage $T$ is a feasible solution, and showing it has a high expected value over all possible realizations will conclude our proof. We denote the value of this strategy $A(S)$. We will consider the objective of maximizing the expected number of nodes in the Triggering model as coverage over an exponentially large universe $\mathcal{U}$, as discussed above. We say that a node $u \in \mathcal{U}$ is covered by $R$ if $\mathcal{C}(u) \cap R \neq \emptyset$ and denote this event as $\mathcal{C}_u^R$. First observe that the probability that $u$ is covered in $\hat{T}$ is:

$$\Pr[\mathcal{C}_u^{\hat{T}}] = 1 - \prod_{j \in \mathcal{C}(u) \cap \mathcal{N}(S)} (1 - \frac{p_j q_j}{c})$$

$$\geq \frac{1}{c}\left(1 - \frac{1}{e}\right) \min\{1, \sum_{j \in \mathcal{C}(u) \cap \mathcal{N}(S)} p_j q_j\}$$

We next bound the probability that $u$ is covered in $T$ conditioned on $u$ being covered in $\hat{T}$. Note that since there may be several nodes covering $u$ in $\hat{T}$, conditioning on $u$ being covered by $\hat{T}$ increases the likelihood of all nodes covering $u$ to be in $\hat{T}$. To avoid this and work with mutually exclusive events we define for some ordering of the elements in $\mathcal{C}_S(u) = \mathcal{C}(u) \cap \mathcal{N}(S)$ the event $A_j$ to be the event that $j \in \hat{T}$ and $i \notin \hat{T}$ for every $\{i \in \mathcal{C}_S(u) \setminus \{j\} : i > j\}$. Since the randomized and relaxed policy selects $t$ nodes in expectation and the probability of each node in $\mathcal{N}(S)$ to be selected by the random process is $p_j q_j/c$, the expected cardinality of any set is bounded by $t/c$. By Markov's inequality we have that:

$$\Pr\left[T = \emptyset \,\Big|\, \mathcal{C}_u^{\hat{T}}, A_j\right] \leq \Pr\left[|\hat{T} \setminus \{j\}| \geq t - 1 \Big| \mathcal{C}_u^{\hat{T}}, A_j\right]$$
$$\leq \frac{1}{t-1}\left(\mathrm{E}[|\hat{T} \setminus \{j\}| \,\Big|\, \mathcal{C}_u^{\hat{T}}, A_j]\right)$$
$$\leq \frac{t}{c(t-1)}$$

We therefore have:

$$\Pr[\mathcal{C}_u^T] = \Pr\left[\mathcal{C}_u^T \,\Big|\, \mathcal{C}_u^{\hat{T}}\right] \cdot \Pr\left[\mathcal{C}_u^{\hat{T}}\right]$$
$$= \Pr\left[T \neq \emptyset \,\Big|\, \mathcal{C}_u^{\hat{T}}\right] \cdot \Pr\left[\mathcal{C}_u^{\hat{T}}\right]$$
$$= \Pr\left[\mathcal{C}_u^{\hat{T}}\right]\left(\sum_{j \in \mathcal{C}_S(u)} \Pr\left[T \neq \emptyset \,\Big|\, \mathcal{C}_u^{\hat{T}}, A_j\right] Pr[\mathcal{C}_u^{\hat{T}}, A_j]\right)$$
$$\geq \Pr\left[\mathcal{C}_u^{\hat{T}}\right]\left(\sum_{j \in \mathcal{C}_S(u)} \left(1 - \frac{t}{c(t-1)}\right) \Pr[\mathcal{C}_u^{\hat{T}}, A_j]\right)$$
$$= \left(\frac{c(t-1)-t}{c^2(t-1)}\right)\left(\frac{e-1}{e}\right)\left(\min\{1, \sum_{j \in \mathcal{C}_S(u)} p_j q_j\}\right)$$

The last step is justified by the fact all events $A_j$ are mutually exclusive and must cover the entire probability space when $u$ is covered by $\hat{T}$. It is easy to verify that if $t > 1$ the optimal value of $c$ is $\frac{2t}{t-1}$. Thus, by linearity of expectation:

$$A(S) = \sum_{u \in U} Pr[\mathcal{C}_u^T] \geq \left(\frac{t-1}{4t}\right)\left(\frac{e-1}{e}\right)\left(\min\{1, \sum_{j \in \mathcal{C}_S(u)} q_j\}\right)$$
$$\geq \left(\frac{e-1}{4e} - \frac{1}{t}\right)\mathcal{L}(q|S)$$

∎

**Theorem 3.10.** *For any $\epsilon > 0$ and any $\beta > 0$ there is an algorithm that finds an adaptive policy with expected value $\left(\frac{(1-e^{-1})^2}{4(2-e^{-1})} - \epsilon\right)OPT_A$ with probability $1 - \beta$ in time $poly(n, N, \frac{log(k)}{\epsilon})$.*

*Proof:* The algorithm first optimizes $L(\cdot)$ as described in Section III-C. Let $(S, q)$ be the result of the algorithm, $\epsilon_1 = \epsilon_2 = \epsilon_3 = \frac{\epsilon}{3}$, and let $t = k - |S|$. Assume that $t > 1/\epsilon_3$. We can use the same adaptive policy as in the previous lemma. We know that with probability at least $1 - \beta$:

$$A(S) \geq \left(\frac{e-1}{4e} - \epsilon_3\right)\mathcal{L}(q|S)$$
$$\geq \left(\frac{e-1}{4e} - \epsilon_3\right)\left(\frac{1-e^{-1}}{2-e^{-1}} - \epsilon_1 - \epsilon_2\right)OPT_\mathcal{L}$$
$$\geq \left(\left(\frac{e-1}{4e}\right)\left(\frac{1-e^{-1}}{2-e^{-1}}\right) - \epsilon_1 - \epsilon_2 - \epsilon_3\right)OPT_\mathcal{L}$$
$$\geq \left(\left(\frac{e-1}{4e}\right)\left(\frac{1-e^{-1}}{2-e^{-1}}\right) - \epsilon\right)OPT_A$$

where the first inequality is due to Lemma 3.9. The second inequality is due to Theorem 3.7 which gives us the approximation ratio for $L(\cdot)$ and Corollary 3.2 that tells us that we get a similar approximation ratio for $\mathcal{L}(\cdot)$. The last inequality is due to the adaptivity gap shown in Lemma 3.3.

If $t < 1/\epsilon_3$ then we can iteratively remove from $S$ the $c = 2\lceil\frac{1}{\epsilon_3}\rceil$ elements (and the elements of $\mathcal{N}(S)$ that are covered only by these elements) that contribute the least to the value of the solution. Let $S', q'$ be the result of this procedure. Notice that $t' \geq 2/\epsilon_3$. As we removed the elements with the least value we know that $L(q'|S') \geq (1 - \frac{c}{|S|})L(q|S)$ and $A(S) \geq (\frac{e-1}{4e} - \frac{\epsilon_3}{2})(1 - \frac{c}{|S|})\mathcal{L}(q|S)$. It is easy to check that if $k > \frac{2}{\epsilon_3}\lceil\frac{1}{\epsilon_3}\rceil$ we get the same approximation ratio. Otherwise we find an optimal adaptive policy by brute force search. ∎

## IV. Simpler Approaches

We conclude by briefly discussing a simple approach applicable in special cases, and examples that justify our approach for adaptive seeding in the Triggering model.

### A. Additive and symmetric influence functions

In the introduction we discussed an analogue of Sample Average Approximation (SAA) as a simple technique which nearly obtains a $1 - 1/e$ approximation for adaptive seeding

in some special cases. For a network $G = (V, E)$ given an influence function $f : 2^V \rightarrow \mathbb{R}_+$ consider the following function $F_{H,t} : 2^X \rightarrow \mathbb{R}_+$ defined for a subgraph $H$ and parameter $t \in \mathbb{N}$ as:

$$F_{H,t}(S) := \max_{T \subseteq \mathcal{N}(S) : |T| \leq t} f(S \cup T)$$

That is, the function is the maximal value that can be obtained from of the set $S$ and $t$ elements from the neighbors of $S$ in $H$. Recall that a function $F : 2^X \rightarrow \mathbb{R}_+$ is submodular if for all $j \in X$ we have that $F(S \cup \{j\}) - F(S) \geq F(T \cup \{j\}) - F(T), \forall S \subseteq T$. In cases when the influence function is such that $F_{H,t}(\cdot)$ is guaranteed to be a monotone submodular function for all $t \leq k$ and all graphs $H$, we can use the following simple technique to approximate the optimal solution of the two stage problem. The two stage problem can be stated as the following optimization problem:

$$\max \left\{ \sum_{H \in \mathcal{G}} p_H F_{H,t}(S) : S \subseteq X, \ |S| + t \leq k \right\}$$

In the above formulation we use $\mathcal{G}$ denote the set of all possible realizations and $p_H$ is the probability of the bipartite graph $H = (X \cup R, E_{X,R})$ to realize where $R \subseteq \mathcal{N}(X)$ and $E_{X,R}$ is the set of edges between $X$ and $R$ in the network $G = (V, E)$. For any fixed $t \leq k - 1$ the function $F_t(S) = \sum_{H \in \mathcal{G}} p_H F_{H,t}(S)$ is a monotone submodular function since it is a weighted sum of monotone submodular functions. Such a function can be well approximated via sampling polynomially many graphs, and the celebrated greedy algorithm for submodular maximization by Nemhauser et al. [15] guarantees to find a $(1 - 1/e - \epsilon)$-approximation for any $\epsilon > 0$. We can therefore iterate over all $t \in \{0, \ldots, k-1\}$ and return the solution $S \subseteq X$ with the maximal value, and thus obtain a $(1 - 1/e - \epsilon)$-approximation for the two stage problem.

We now briefly discuss two special cases of influence functions for which $F_{H,t}(\cdot)$ is monotone submodular.

**Additive functions.** We say that a function is additive when $f(S) = \sum_{i \in S} w_i$ for some fixed $w_i, i \in [n]$. The well studied *Voter model* [34], for example, is an additive influence function [19]. For such functions $F_{H,t}(\cdot)$ as defined above is monotone submodular. The intuition is easy. For any $S \subseteq X$ these functions take the highest valued nodes in $\mathcal{N}(S)$. Adding a node $i \in X$ to $S$ introduces no more candidates to the top $t$ valued nodes than adding $i$ to any $S' \subset S$. Details and proof are deferred to the full version.

**Theorem 4.1.** *For any additive influence function $f(\cdot)$, the function $F_t(\cdot)$ is monotone submodular.*

**Symmetric submodular functions.** We say that a function is *symmetric submodular* when $f(T) = \sum_{i \leq |T|} r_i$ where $r_i \in \mathcal{R}_+$ is some fixed set that respects $r_1 \geq r_2 \geq \ldots \geq r_n$. It is easy to see that for this kind of influence function the function
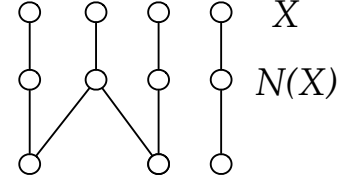


Fig. 2. An example of a graph where the two stage adaptive seeding is not submodular when the influence function is a cover function. From left to right the nodes in $X$ are $A, B, C, D$ as discussed below.

$F_{H,t}(\cdot)$ is also submodular. The intuition here is even simpler. As these are cardinality functions, it is clear that adding a new element to a larger set introduces less new neighbors. In addition the marginal contribution of each new neighbor decreases as the number of neighbors grows. We defer the exact details of the proof to the full version.

**Theorem 4.2.** *For any Symmetric Submodular influence function $f(\cdot)$, the function $F_t(\cdot)$ is monotone submodular.*

These functions can capture cases where the exact structure of the network is not known, and thus the exact influence value for each node is unknown. In such cases, it seems reasonable to maximize the number of seeds assuming that additional seeds have decreasing marginal contributions.

### B. $F_t$ is not submodular for Cover functions

As we now show the approach we describe above is inapplicable to the Triggering model. We now show that $F_t$ is not monotone submodular when $f$ is a Cover function, which is a special case of the Triggering model. Recall that a function $f : 2^V \rightarrow \mathbb{R}_+$ is a Cover function if there are some sets $T_1, T_2, \ldots, T_{|V|}$ of some universe and for every $S \subseteq V$ we have that $f(S) = |\cup_{i \in S} T_i|$.

Consider the example in Figure 2, where all nodes in $\mathcal{N}(X)$ realize with probability 1 if any of their parents is seeded. Consider $F_2$. It is easy to see that $F_2(\{A, C\}) = 2$ and also $F_2(\{A, C, D\}) = 2$. On the other hand, $F_2(\{A, B, C\}) = 2$ and also $F_2(\{A, B, C, D\}) = 3$. This implies that adding $D$ to the larger set $\{A, B, C\}$ increases the value of the function by more than when adding it to the set $\{A, C\}$. This is a contradiction to the assumption that $F_2$ is submodular.

### C. Unbounded adaptivity gap

We now show that we cannot hope to use non-relaxed non adaptive policies to approximate an adaptive policy in adaptive seeding. We show that the ratio between the expected payoff of an adaptive policy and a non-relaxed non adaptive policy can be unbounded (even for very simple influence functions).

Consider the instance in Figure 3. Let the budget be 2 and the influence function be the number of seeded nodes in $\mathcal{N}(X)$ (this is an additive function where each item's value is 1). The optimal adaptive policy here is trivial. It chooses the single node in $X$, and in any realization where there is some node in $\mathcal{N}(X)$ that realized it chooses any one of those nodes. The expected payoff of this solution goes to $(1 - 1/e)$ as $n$ grows.
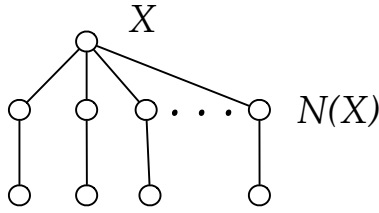
Fig. 3. The unbounded adaptivity gap. When each node in $\mathcal{N}(X)$ appears with probability $1/n$ the adaptivity gap is unbounded.

On the other hand, any non-relaxed non adaptive policy needs to commit on a node in $\mathcal{N}(X)$ in advance. No matter which node is chosen the expected value of this policy goes to 0 as $n$ grows, and thus the ratio between the two expected payoffs goes to $\infty$ as $n$ grows.

## Acknowledgment

## References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 137–146.

[2] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," in *ACM Conference on Electronic Commerce*, 2006, pp. 228–237.

[3] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance, "Cost-effective outbreak detection in networks," in *KDD*, 2007, pp. 420–429.

[4] E. Mossel and S. Roch, "On the submodularity of influence in social networks," in *STOC*, 2007, pp. 128–134.

[5] N. Chen, "On the approximability of influence in social networks," in *SODA*, 2008, pp. 1029–1037.

[6] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *KDD*, 2010, pp. 1019–1028.

[7] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *WSDM*, 2011.

[8] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen, "Sparsication of influence networks," in *KDD*, 2011.

[9] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Influence maximization in social networks: Towards an optimal algorithmic solution," *arXiv preprint arXiv:1212.0884*, 2012.

[10] I. Abraham, S. Chechik, D. Kempe, and A. Slivkins, "Low-distortion inference of latent similarities from a multiplex social network," in *SODA*, 2013, pp. 1853–1872.

[11] S. L. Feld, "Why your friends have more friends than you do," *American Journal of Sociology*, pp. 1464–1477, 1991.

[12] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, "The anatomy of the facebook social graph," *CoRR*, vol. abs/1111.4503, 2011.

[13] S. Lattanzi and Y. Singer, "The power of sampling neighbors in social networks," *Working paper*.

[14] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 479–502, 2002.

[15] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions ii," *Math. Programming Study 8*, pp. 73–87, 1978.

[16] P. Domingos and M. Richardson, "Mining the network value of customers," in *KDD*, 2001, pp. 57–66.

[17] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *KDD*, 2002, pp. 61–70.

[18] A. R. Karlin, D. Kempe, and T. Tamir, "Beyond VCG: Frugality of truthful mechanisms," in *FOCS*, 2005, pp. 615–626.

[19] E. Even-Dar and A. Shapira, "A note on maximizing the spread of influence in social networks," *Inf. Process. Lett.*, vol. 111, no. 4, pp. 184–187, 2011.

[20] A. Asadpour, H. Nazerzadeh, and A. Saberi, "Stochastic submodular maximization," in *Internet and Network Economics*. Springer, 2008, pp. 477–489.

[21] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *Journal of Artificial Intelligence Research*, vol. 42, no. 1, pp. 427–486, 2011.

[22] J. D. Hartline, V. S. Mirrokni, and M. Sundararajan, "Optimal marketing strategies over social networks," in *WWW*, 2008, pp. 189–198.

[23] N. Haghpanah, N. Immorlica, V. S. Mirrokni, and K. Munagala, "Optimal auctions with positive network externalities," in *ACM Conference on Electronic Commerce*, 2011, pp. 11–20.

[24] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. Society for Industrial Mathematics, 2009, vol. 9.

[25] N. Immorlica, D. Karger, M. Minkoff, and V. S. Mirrokni, "On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems," in *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2004, pp. 691–700.

[26] R. Ravi and A. Sinha, "Hedging uncertainty: Approximation algorithms for stochastic optimization problems," in *Integer programming and combinatorial optimization*. Springer, 2004, pp. 101–115.

[27] A. Gupta, M. Pál, R. Ravi, and A. Sinha, "Boosted sampling: approximation algorithms for stochastic optimization," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM, 2004, pp. 417–426.

[28] A. Srinivasan, "Approximation algorithms for stochastic and risk-averse optimization," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1305–1313.

[29] D. B. Shmoys and C. Swamy, "Stochastic optimization is (almost) as easy as deterministic optimization," in *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*. IEEE, 2004, pp. 228–237.

[30] ——, "An approximation scheme for stochastic linear programming and its application to stochastic integer programs," *Journal of the ACM (JACM)*, vol. 53, no. 6, pp. 978–1012, 2006.

[31] J. Kleinberg, Y. Rabani, and É. Tardos, "Allocating bandwidth for bursty connections," *SIAM Journal on Computing*, vol. 30, no. 1, pp. 191–217, 2000.

[32] B. C. Dean, M. X. Goemans, and J. Vondrdk, "Approximating the stochastic knapsack problem: The benefit of adaptivity," in *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*. IEEE, 2004, pp. 208–217.

[33] A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi, "Approximation algorithms for stochastic orienteering," in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2012, pp. 1522–1538.

[34] R. A. Holley and T. M. Liggett, "Ergodic theorems for weakly interacting infinite systems and the voter model," *The annals of probability*, pp. 643–663, 1975.

[35] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[36] L. A. Wolsey, "Maximising real-valued submodular functions: Primal and dual heuristics for location problems," *Mathematics of Operations Research*, vol. 7, no. 3, pp. 410–425, 1982.