Fairness in Influence Maximization through Randomization

Ruben Becker[1], Gianlorenzo D'Angelo[1], Sajjad Ghobadi[1], Hugo Gilbert[2]

[1] Gran Sasso Science Institute, L'Aquila, Italy

[2] Université Paris-Dauphine, Université PSL, CNRS, LAMSADE, 75016 Paris, France

ruben.becker@gssi.it, gianlorenzo.dangelo@gssi.it, sajjad.ghobadi@gssi.it, hugo.gilbert@dauphine.psl.eu

Abstract

The influence maximization paradigm has been used by researchers in various fields in order to study how information spreads in social networks. While previously the attention was mostly on efficiency, more recently fairness issues have been taken into account in this scope. In the present paper, we propose to use randomization as a mean for achieving fairness. While this general idea is not new, it has not been applied in the area of information spread in networks.

Similar to previous works like Fish et al. (WWW '19) and Tsang et al. (IJCAI '19), we study the maximin criterion for (group) fairness. By allowing randomized solutions, we introduce two different variants of this problem. While the original deterministic maximin problem has been shown to be inapproximable, interestingly, we show that both probabilistic variants permit approximation algorithms with a constant multiplicative factor of $1 - 1/e$ plus an additive arbitrarily small error due to the simulation of the information spread.

For an experimental study, we provide implementations of our methods and compare the achieved fairness values to existing methods. Non-surprisingly, the ex-ante values, i.e., minimum expected value of an individual (or group) to obtain the information, of the computed probabilistic strategies are significantly larger than the (ex-post) fairness values of previous methods. This confirms that studying fairness via randomization is a worthwhile direction. More surprisingly, we observe that even the ex-post fairness values, i.e., fairness values of sets sampled according to the probabilistic strategies, computed by our routines dominate over the fairness achieved by previous methods on most of the instances tested.

Introduction

The internet has revolutionized the way information spreads through the population. One positive consequence is that important and valuable campaigns can be spread at little cost quite efficiently thanks to news platforms and social media. Such campaigns may be related to HIV prevention (Wilder et al. 2018a; Yadav et al. 2018), public health awareness (Valente and Pumpuang 2007) or financial inclusion (Banerjee et al. 2013). The information spreading process is notably optimized by algorithms that identify key people in the network to act as seed users to initiate the spread of the campaign efficiently. The well known influence maximization problem formalizes this objective (Kempe, Kleinberg, and Tardos 2015): given a network and a probabilistic diffusion model, the task is to find a set of k seed nodes from which the campaign will start to spread, in order to maximize the expected number of reached nodes. The problem has received a tremendous amount of attention (Becker et al. 2020a; Borgs et al. 2014; Borodin et al. 2017; Budak, Agrawal, and El Abbadi 2011; Chen and Teng 2017; Cohen et al. 2014; Tang, Shi, and Xiao 2015; Tang, Xiao, and Shi 2014). However, as the objective function in the influence maximization problem is only concerned with the efficiency of the diffusion process, it does not take into account any fairness criteria. More recently, fairness issues in influence maximization have become a focus of attention for many researchers.

A first sequence of papers has investigated a setting in which several competing players are paying the network's host to influence users in their favor. The goal in these works is to ensure that the host picks seed nodes in a fair way w.r.t. the different players (Chen et al. 2020; Lu et al. 2013; Yu et al. 2017). Another line of research has investigated the fairness of the diffusion process with respect to the vertices, i.e., the users in the network. Indeed when only efficiency is being optimized, some users, or communities, i.e., groups of users, might get an unfairly low coverage (Ali et al. 2019; Fish et al. 2019; Farnad, Babaki, and Gendreau 2020; Khajehnejad et al. 2020; Rahmattalabi et al. 2020; Stoica, Han, and Chaintreau 2020; Tsang et al. 2019). A intuitive criterion to consider here is the maximin criterion. Here, the goal is to choose at most k seed nodes to maximize the minimum probability of a user being reached. When generalized to groups of users or communities, the goal becomes to maximize the minimum expected number of users reached per community. The first problem has been considered by Fish et al. (2019), who showed that the problem is hard to approximate to any constant approximation factor, unless $P = NP$. The second problem has been considered by Tsang et al. (2019). Building on previous work by (Chekuri, Vondrák, and Zenklusen 2010) and (Udwani 2018), the authors designed an algorithm with an asymptotic approximation ratio of $1 - 1/e$ provided that the number of communities is not much larger than k.

In the present paper, we extend these works by studying the impact of randomization on fairness. Our approach is to allow for randomized strategies for choosing seeds rather than to restrict to deterministic strategies, i.e., sets of size k. Indeed, after recalling the necessary technical background related to influence maximization, we introduce two randomized versions of the maximin influence problem. In the first one, we consider strategies that consist of probability distributions over seed sets of size k, we call this the *set-based problem*. In the second case, the *node-based problem*, we consider randomized strategies that pick nodes as seeds with some probability such that the expected size of the resulting seed set is at most k. Such randomized strategies provide certain advantages over deterministic ones. In fact, the use of randomization is a longstanding idea in computational social choice, where it often leads to more tractable results and more expressive solutions via for instance time-sharing mechanisms (David 2013). It is also used to incentivize participation (Aziz, Luo, and Rizkallah 2018) or to workaround impossibility results (Brandl, Brandt, and Seedig 2016). Lastly and closer to our work, using randomization is frequently used to obtain fairer solutions (Aziz, Brandt, and Stursberg 2013; Bogomolnaia and Moulin 2001; Katta and Sethuraman 2006). Indeed, there may be optimization problems for which any deterministic solution is unfair (Machina 1989). In such cases, randomization may help evening things out by considering fairness in expectation, i.e., *ex-ante fairness* rather than *ex-post fairness*. Randomization is both useful for one-shot and for repeated problems. In the former, it provides fairness over opportunities, in the latter it achieves fairness in the long run in a natural way. Lastly, randomization can be used to satisfy the fairness principle of *equal treatment of equals* (Moulin 1991). Despite being an old research topic, the study of randomized solutions is still a hot topic with many open problems (Aziz 2019; Brandt 2019).

Our Contribution. We show that both randomized variants of the maximin influence problems are NP-hard and quantify the loss in efficiency that can be incurred by following our fairness criteria. Thereafter we show that still, in this setting of fairness in influence maximization, randomization leads to a number of advantages. In fact, we prove that the resulting problems can be approximated to within a factor of $1 - 1/e$ (plus an additive $-\varepsilon$ term that is also inherent in the work of Tsang et al. (2019)) even in the case when the number of communities exceeds the number of seed nodes k. This shows that we can circumvent the inapproximability result of Fish et al. (2019) by introducing randomization to the problem. Our algorithms are comparatively simple. For the set-based problem, the problem can be approximated (to within an additive ε term) by a linear program. The downside is that this program is of dimension $\Theta(n^k)$. As the linear program is a covering problem, we are able to show that a multiplicative weights routine that is essentially a black-box application of a method by Young (1995) can be used to obtain the described approximation. This method, as a subroutine, requires an algorithm for an oracle problem. We observe that the oracle problem in our case turns out to be the standard influence maximization problem and thus can be approximated to within a factor of $1 - 1/e$ efficiently both in theory and practice. Although the feasible set to the set-based problem is of dimension $\Theta(n^k)$, the multiplicative weights routine has the nice property that the returned solution is of support linear in n. For the node-based problem, we face a different problem. Here the feasible set is of size n, the problem however is not linear. We show that it is approximated to within a constant factor by a linear program of the same size and thus can be solved in polynomial time. We then evaluate our implementations on random instances and those instances from the work of Tsang et al. (2019) that are publicly available. We compare both the ex-ante and ex-post performance of our techniques with the routines proposed by Tsang et al. (2019) and Fish et al. (2019). We observe that our ex-ante values are superior to the ex-post values of all other algorithms and, maybe surprisingly, also the ex-post values of our algorithms are competitive with or even improve over the ex-post values of previous techniques.

See the full version (Becker et al. 2020b) for all proofs and further details regarding the implementation that have been omitted in this article due to space limitations.

Preliminaries

We consider the classical influence maximization setting where we are given a directed arc-weighted graph $G = (V, A, w)$ with V being the set of n nodes, A the set of arcs, and $w : A \to [0, 1]$ an arc-weight function. In addition we are given an information diffusion model. A broad variety of models can be used as information diffusion model. Two of the most popular models are the *Independent Cascade* (IC) and *Linear Threshold* (LT) models (Kempe, Kleinberg, and Tardos 2015). In both these models, given an initial node set $S \subseteq V$ called *seed nodes*, a spread of influence from the set S is defined as a randomly generated sequence of node sets $(S_t)_{t\in\mathbb{N}}$, where $S_0 = S$ and $S_{t-1} \subseteq S_t$. These sets represent active users, i.e., we say that a node v is *active* at time step t if $v \in S_t$. The sequence converges as soon as $S_{t^*} = S_{t^*+1}$, for some time step $t^* \geq 0$ called the time of quiescence. For a set S, we use the standard notation $\sigma(S) = \mathbb{E}[|S_{t^*}|]$ to denote the expected number of nodes activated at the time of quiescence when running the process with seed nodes S, here the expectation is over the random process of information diffusion that depends on the weights w and moreover on the information diffusion model at hand. Both models are in fact special cases of what is known as the Triggering model. Due to space limitations, we defer the reader to the literature for the precise definitions of the IC and LT models. Here, we proceed by introducing the more general Triggering model. For a node $v \in V$, let N_v denote all in-neighbors of v. In the Triggering model, every node independently picks a *triggering set* $T_v \subseteq N_v$ according to some distribution over subsets of its in-neighbors (usually depending on w). For a possible outcome $L = (T_v)_{v\in V}$ of triggering sets for the nodes in V, let $G_L = (V, A_L)$ denote the sub-graph of G where $A_L = \{(u, v)|v \in V, u \in T_v\}$. We refer to G_L as live-edge graph and to A_L as live edges. In what follows, we denote with $\mathcal{L}$ the random variable that describes the process of generating live-edge graphs, and with L we mean a possible outcome, i.e., value taken by $\mathcal{L}$. We let $\rho_{\mathcal{L}}(S)$ be the set of nodes reachable from S

in $G_{\mathcal{L}}$, then $\sigma(S) = \mathbb{E}_{\mathcal{L}}[|\rho_{\mathcal{L}}(S)|]$. The IC and LT models each occur when triggering sets are chosen in a particular way. In what follows, we assume the Triggering model to be the underlying model describing the information spread. We define $\sigma_v(S) := \Pr_{\mathcal{L}}[v \in \rho_{\mathcal{L}}(S)]$ to be the probability that node v is reached from seed nodes S. Clearly, $\sigma(S) = \mathbb{E}_{\mathcal{L}}[|\rho_{\mathcal{L}}(S)|] = \sum_{v \in V} \Pr_{\mathcal{L}}[v \in \rho_{\mathcal{L}}(S)] = \sum_{v \in V} \sigma_v(S)$. We extend this notation in a natural way, that is, for $C \subseteq V$, we denote by $\sigma_C(S) = \frac{1}{|C|} \cdot \sum_{v \in C} \sigma_v(S)$ the average probability of being reached of nodes in C. Note that $\sigma_v(S) = \sigma_{\{v\}}(S)$ for nodes $v \in V$ and $\sigma(S) = |V| \cdot \sigma_V(S)$ for all nodes V.

For a maximization problem $\max\{F(x) : x \in R\}$ with feasibility region R and objective function $F : R \to \mathbb{R}_{\geq 0}$, and for real values $\alpha \in (0, 1]$ and $\beta \in [0, \infty)$, we say that x is an (α, β)-approximation, if $F(x) \geq \alpha \cdot \mathrm{opt} - \beta$, where opt denotes the optimum value.

Maximin Optimization. The standard objective studied in influence maximization is finding a set S maximizing $\sigma(S)$ under a cardinality constraint $|S| \leq k$ for some integer k. As this objective function does not take into account the fairness of the diffusion process with respect to nodes or communities, Fish et al. (2019) and Tsang et al. (2019), have investigated maximin variants of this objective that can be written as $\max_{S \in \binom{V}{k}} \min_{C \in \mathcal{C}} \sigma_C(S)$, where $\mathcal{C}$ is a set of m different communities $C \subseteq V$ that may not be disjoint and $\binom{V}{k}$ denotes the set of subsets of V of size k. If each node is its own community, this amounts to finding a set of k seed nodes maximizing the minimum probability that a node is reached, which is the problem considered by Fish et al. (2019). We note that this is actually one instance of a broader class of optimization problems that ask to maximize a social welfare function, being the $-\infty$-mean here. Fish et al. (2019) considered the special case where the diffusion model is the Independent Cascade model and in which all arcs have the same probability of diffusion α. They proved that the problem of choosing k seeds such as to maximize $\min_{v \in V} \sigma_v(S)$ is NP-hard to be approximated within a factor better than $O(\alpha)$ and that minimizing the number of seeds to obtain the optimal solution cannot be approximated within a factor $O(\ln n)$. Furthermore, they analysed several natural heuristics which unfortunately exhibit worst-case approximation ratio exponentially small in n.

Fairness via Randomization

We initiate studying the impact of randomization to increase fairness for influence maximization. We start with a simple example of an influence maximization problem to illustrate the impact of randomization. Consider the graph consisting of two nodes u, v, each forming their own community, connected in both directions by edges $(u, v), (v, u)$ with probabilities $1/2$. Assume that $k = 1$. Then due to symmetry the optimal deterministic strategy is to choose any of the two nodes achieving a minimum probability of being reached of $1/2$ for the non-chosen node. A probabilistic strategy however would be allowed to assign probabilities $1/2$ to both the sets $\{u\}$ and $\{v\}$, achieving a minimum expected probability of $3/4$ for both nodes. While this example seems simplistic and artificial, it shows that the probabilistic strategy may in fact achieve a higher degree of fairness. We consider two different ways of introducing randomness, either via distributions over sets or via distributions over nodes.

Probabilistically Choosing Sets. We relax the maximin problem by allowing for randomized strategies, i.e., feasible solutions in our *set-based probabilistic maximin problem* are not simply sets of size at most k, but rather distributions over sets. Let $\mathcal{S}$ be the set of distributions over sets of size exactly k, i.e., $\mathcal{S} := \{p \in [0, 1]^{\binom{V}{k}} : \mathbb{1}^T p = 1\}$ and let $S \sim p$ denote the random process of sampling S according to the distribution p. One possible way of defining a probabilistic maximin problem would be to consider $\max_{p \in \mathcal{S}} \mathbb{E}_{S \sim p}[\min_{C \in \mathcal{C}} \sigma_C(S)]$. We note however that among the optimal solutions to the above problem, there always also is a deterministic one as any distribution that assigns a probability of 1 to a set in $\mathrm{argmax}_{S \in \binom{V}{k}} \min_{C \in \mathcal{C}} \sigma_C(S)$ and 0 to all other sets is optimal. Certainly, the study of this problem may still be of interest as finding approximation algorithms to it may be easier than for the original maximin problem. Here, we however take a more radical route. That is, we consider the problem

$$\mathrm{opt}_{\mathcal{S}}(G, \mathcal{C}, k) = \max_{p \in \mathcal{S}} \min_{C \in \mathcal{C}} \mathbb{E}_{S \sim p}[\sigma_C(S)],$$

i.e., we reverse the order of the expectation and the minimum over the m communities $\mathcal{C}$. This notion is frequently referred to as ex-ante fairness in the literature (Machina 1989).

Probabilistically Choosing Nodes. An alternative intuitive way of introducing randomness is obtained by considering a maximin problem where feasible solutions are not distributions over sets, but are characterized by probability values for nodes. In this setting, which we call the *node-based probabilistic maximin problem*, we let $\mathcal{X} := \{x \in [0, 1]^n : \mathbb{1}^T x \leq k\}$ be the feasible set and consider the process of randomly generating a set S from x, denoted by $S \sim x$, by letting i be in S independently with probability x_i. In this setting we are thus interested in finding $x \in \mathcal{X}$ that maximizes the minimum expected coverage from S of any community, when S is generated from x as described and the expectation is over this generation. We write this problem as

$$\mathrm{opt}_{\mathcal{X}}(G, \mathcal{C}, k) = \max_{x \in \mathcal{X}} \min_{C \in \mathcal{C}} \mathbb{E}_{S \sim x}[\sigma_C(S)].$$

Extending Set Functions to Vectors. In what follows, we extend set functions to vectors in $\mathcal{S}$ and $\mathcal{X}$ in a straightforward way, i.e., for a set function f, for $p \in \mathcal{S}$, we let $f(p) := \mathbb{E}_{S \sim p}[f(S)]$ and, for $x \in \mathcal{X}$, we let $f(x) := \mathbb{E}_{S \sim x}[f(S)]$.

Relationship between Problems. We illustrate that the set and node-based probabilistic maximin problems may have different optimal values for the same instance. Consider the graph G illustrated in Figure 1. Assume that $\mathcal{C}$ is such that each vertex forms its own community and

$k = 2$. Consider the vector $p \in \mathcal{S}$ with $p(\{1,5\}) = p(\{2,6\}) = 1/2$ and $p(S) = 0$ for all other sets. Then, $\sigma_v(p) = \mathbb{E}_{S\sim p}[\sigma_v(S)] \geq 1/2$ for all nodes v and thus $\mathrm{opt}_{\mathcal{S}}(G,\mathcal{C},2) \geq 1/2$. However, a strategy $x \in \mathcal{X}$ cannot achieve such value. More precisely, the strategy $x \in \mathcal{X}$ with $x_1 = x_2 = x_5 = x_6 = 1/2$ achieves $\sigma_v(x) = 1/2$ for $v \in \{1,2,5,6\}$, $\sigma_3(x) = 3/4$, and $\sigma_4(x) = 3/8$. While this latter value could be increased by making x_4 positive, this is possible only at the cost of reducing x_i below $1/2$ for $i \in \{1,2,5,6\}$, which directly implies $\sigma_i(x) < 1/2$. Hence, $\mathrm{opt}_{\mathcal{X}}(G,\mathcal{C},2) < 1/2$.

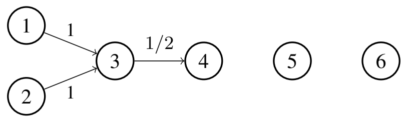

Figure 1: Instance showing that optimal solutions to set and node-based probabilistic maximin problems may differ.

Price of Fairness and Hardness

Price of Group Fairness. The price of group fairness is a quantitative loss measuring the decrease in efficiency that is incurred when we restrict ourselves to solutions respecting a group fairness requirement. In the following, we denote the maximizing solutions to the node and set-based problems by $F_{\mathcal{X}}(G,\mathcal{C},k) = \mathrm{argmax}_{x\in\mathcal{X}} \min_{C\in\mathcal{C}} \mathbb{E}_{S\sim x}[\sigma_C(S)]$ and $F_{\mathcal{S}}(G,\mathcal{C},k) = \mathrm{argmax}_{p\in\mathcal{S}} \min_{C\in\mathcal{C}} \mathbb{E}_{S\sim p}[\sigma_C(S)]$, respectively. Then, the prices of fairness $\mathrm{PoF}_{\mathcal{X}}(G,\mathcal{C},k)$ and $\mathrm{PoF}_{\mathcal{S}}(G,\mathcal{C},k)$ incurred by only considering strategies in $F_{\mathcal{X}}(G,\mathcal{C},k)$ and $F_{\mathcal{S}}(G,\mathcal{C},k)$ respectively are equal to

$$\mathrm{PoF}_{\mathcal{X}}(G,\mathcal{C},k) := \max_{S\in\binom{V}{k}} \sigma(S) \Big/ \max_{x\in F_{\mathcal{X}}(G,\mathcal{C},k)} \sigma(x),$$

$$\mathrm{PoF}_{\mathcal{S}}(G,\mathcal{C},k) := \max_{S\in\binom{V}{k}} \sigma(S) \Big/ \max_{p\in F_{\mathcal{S}}(G,\mathcal{C},k)} \sigma(p).$$

We obtain that for both problems, the price of group fairness can be linear in the graph size.

Lemma 1. *For any even $n > 0$, there is a graph G with n nodes and a community structure $\mathcal{C}$ such that $\mathrm{PoF}_{\mathcal{X}}(G,\mathcal{C},1) = \mathrm{PoF}_{\mathcal{S}}(G,\mathcal{C},1) = (n+2)/4$, when using the IC model.*

On the positive side we obtain that the price of group fairness is never larger than n/k.

Lemma 2. *For any graph G, community structure $\mathcal{C}$ and number k, $\mathrm{PoF}_{\mathcal{X}}(G,\mathcal{C},k), \mathrm{PoF}_{\mathcal{S}}(G,\mathcal{C},k) \leq n/k$.*

Hardness. Fish et al. (2019) show that the standard maximin problem is NP-hard. We provide an analogous result for the two probabilistic maximin problems.

Theorem 3. *For a directed arc-weighted graph $G = (V,E,w)$ and a value $\alpha \in [0,1]$ it is NP-hard to decide if there is $p \in \mathcal{S}$ with $\min_{v\in V} \mathbb{E}_{S\sim p}[\sigma_v(S)] \geq \alpha$ (resp. $x \in \mathcal{X}$ with $\min_{v\in V} \mathbb{E}_{S\sim x}[\sigma_v(S)] \geq \alpha$) even in the IC model.*

Approximation Algorithms

In this section, we show that there are algorithms that compute $(1-1/e,\varepsilon)$-approximations to both the set-based and the node-based maximin problems. The functions $\sigma_C(p)$ and $\sigma_C(x)$ involved in the optimization problems are not computable exactly in polynomial time (even for a vector p of polynomial support). Even worse, they are not relatively approximable using Chernoff bounds as there is no straightforward absolute lower bound on $\sigma_C(S)$ for sets S of size k and communities $C \in \mathcal{C}$. Using a Hoeffding bound however, we show that the functions $\sigma_v(S)$ for nodes $v \in V$ and $v \in V$ and sets $S \in \binom{V}{k}$ are approximated to within an additive error of $\varepsilon > 0$ by the functions $\tilde{\sigma}_v(S) := \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{v\in\rho_{L_t}(S)}$, where, for $T \in \mathbb{Z}_{\geq 0}$, we let $L_1,\dots,L_T$ denote a set of T live-edge graphs sampled according to the Triggering model. This implies also that $\tilde{\sigma}_v(p) = \mathbb{E}_{S\sim p}[\tilde{\sigma}_v(S)]$ is an absolute ε-approximation of $\sigma_v(p) := \mathbb{E}_{S\sim p}[\sigma_v(S)]$ for any $p \in \mathcal{S}$ and similarly for $\tilde{\sigma}_v(x) = \mathbb{E}_{S\sim x}[\tilde{\sigma}_v(S)]$ for any $x \in \mathcal{X}$. We then get the same result for $\tilde{\sigma}_C(p) := \frac{1}{|C|}\sum_{v\in C}\tilde{\sigma}_v(p)$ for any $p \in \mathcal{S}$ and $C \in \mathcal{C}$ and for $\tilde{\sigma}_C(x) := \frac{1}{|C|}\sum_{v\in C}\tilde{\sigma}_v(x)$ for any $x \in \mathcal{X}$ and $C \in \mathcal{C}$ as these functions are again just averages over other absolute ε-approximations. In summary, we can solve the optimization problem involving the approximations while incurring an additive arbitrarily small error.

Lemma 4. *Let $\delta \in (0,1/2)$ and $\varepsilon \in (0,1)$. Assume that $T \geq 4\varepsilon^{-2} \cdot [(k+1)\cdot\log n + \log\delta^{-1}]$ and that $\tilde{\sigma}_C(\cdot)$ is as above. Let $p \in \mathcal{S}$ be a (α,β)-approximation for $\max_{p\in\mathcal{S}}\min_{C\in\mathcal{C}}\tilde{\sigma}_C(p)$, then p is a $(\alpha,\beta+\varepsilon)$-approximation of $\mathrm{opt}_{\mathcal{S}}(G,\mathcal{C},k)$ with probability at least $1-\delta$. If $T \geq 4\varepsilon^{-2}\cdot[n+\log n+\log\delta^{-1}]$ and $x \in \mathcal{X}$ is a (α,β)-approximation for $\max_{x\in\mathcal{X}}\min_{C\in\mathcal{C}}\tilde{\sigma}_C(p)$, then x is a $(\alpha,\beta+\varepsilon)$-approximation of $\mathrm{opt}_{\mathcal{X}}(G,\mathcal{C},k)$ with probability at least $1-\delta$.*

Probabilistically Choosing Sets

Recall the set-based probabilistic maximin problem $\mathrm{opt}_{\mathcal{S}}(G,\mathcal{C},k) := \max_{p\in\mathcal{S}}\min_{C\in\mathcal{C}}\sigma_C(p)$, where $\sigma_C(p) = \mathbb{E}_{S\sim p}[\sigma_C(S)]$ for $C \in \mathcal{C}$ and $p \in \mathcal{S} := \{p \in [0,1]^{\binom{V}{k}} : \mathbb{1}^T p = 1\}$. In the light of Lemma 4, we focus on finding approximate solutions to $\max_{p\in\mathcal{S}}\min_{C\in\mathcal{C}}\tilde{\sigma}_C(p)$.

Allowing for distributions over sets rather than sets turns the optimization problem at hand, $\max_{p\in\mathcal{S}}\min_{C\in\mathcal{C}}\tilde{\sigma}_C(p)$, into a problem that can be written as a linear program. While the original problem, i.e., the problem of choosing a set maximizing the approximate minimum probability, can be written as an integer linear program using a variable to model a threshold to be maximized. Hence, from an algorithmic point of view, one may think that this makes the problem polynomial time solvable. The caveat is of course that the dimension of $\mathcal{S}$ is large, namely $\Theta(n^k)$, which turns the dimension of the corresponding linear program super-polynomial, at least for super-logarithmic values of k. In this section, we show that, nevertheless, the problem can be approximated to within a constant factor using a specific kind of linear programming algorithm. The essential observation is that the linear program at hand actually is a covering linear program. We will use a result due to Young (1995) that shows that such linear programs can be solved efficiently independent of their dimension under the condition that a certain oracle problem can be solved efficiently.

Young's Algorithm. Young (1995) gives algorithms for solving packing and covering linear programs. A covering problem in the sense of Young is of the following form: Let $P \subseteq \mathbb{R}^\nu$ be a convex set and let $f : P \to \mathbb{R}^\mu$ be a μ-dimensional linear function over P. Assume that $0 \leq f_j(x) \leq \omega$ for all $j \in [\mu]$ and $x \in P$, where ω is the width of P w.r.t. f. The *covering problem* consists of computing $\lambda^* := \max_{x \in P} \min_{j \in [\mu]} f_j(x)$, when $f_j(x) \geq 0$ for all $x \in P$.

Theorem 5 ((Young 1995))**.** *Let $\eta \in (0,1)$ and assume that there is an oracle that, given a non-negative vector $z \in \mathbb{R}^\mu$ returns $x \in P$ and $f(x)$ satisfying $\sum_{j \in [m]} z_j f_j(x) \geq \alpha \cdot \max_{x \in P}\{\sum_{j \in [m]} z_j f_j(x)\}$ for some constant $\alpha \leq 1$, then there is an algorithm that computes $x \in P$ with $\min_{j \in [\mu]} f_j(x) \geq \alpha(1-\eta) \cdot \lambda^*$ in $O(\omega \eta^{-2} \log \mu / \lambda^*)$ iterations in each of which it does $O(\mu)$ work and calls the oracle once. The output x is the arithmetic mean of the vectors returned by the oracle.*

Set-Based Problem via Young's Algorithm. Clearly $\tilde{\sigma}_C$ is a linear function in p, namely $\tilde{\sigma}_C(p) = \sum_{S \in \binom{V}{k}} p_S \tilde{\sigma}_C(S)$ and thus the problem $\max_{p \in \mathcal{S}} \min_{C \in \mathcal{C}} \tilde{\sigma}_C(p)$ takes exactly the form of a covering problem in the sense of Young with $\nu = \binom{n}{k}$, $\mu = m = |\mathcal{C}|$, $P = \mathcal{S}$, and $\omega = 1$. Hence, we can compute a $(1-1/e, 0)$-approximation for $\max_{p \in \mathcal{S}} \min_{C \in \mathcal{C}} \tilde{\sigma}_C(p)$, if we provide an oracle with approximation factor $1 - 1/e$.

Let us now take a closer look at the requirements of Theorem 5 in terms of the oracle problem. Given a non-negative vector $z \in \mathbb{R}^m$, the oracle is required to return $p \in \mathcal{S}$ and $\tilde{\sigma}_C(p)$ for $C \in \mathcal{C}$ such that $\sum_{C \in \mathcal{C}} z_C \tilde{\sigma}_C(p) \geq \alpha \cdot \max_{p \in \mathcal{S}}\{\sum_{C \in \mathcal{C}} z_C \tilde{\sigma}_C(p)\}$ for some $\alpha \leq 1$. Note that, by linearity of expectation, $\sum_{C \in \mathcal{C}} z_C \tilde{\sigma}_C(p)$ is equal to

$$\mathbb{E}_{S \sim p}\Big[\sum_{C \in \mathcal{C}} z_C \cdot \frac{1}{|C|} \sum_{v \in C} \tilde{\sigma}_v(S)\Big] = \mathbb{E}_{S \sim p}\Big[\sum_{v \in V} \omega_v \cdot \tilde{\sigma}_v(S)\Big],$$

where $\omega_v := \sum_{C \in \mathcal{C}: v \in C} z_C / |C|$. We observe that this is a weighted average over sets $S \in \binom{V}{k}$ of the values $\sigma^\omega(S) := \sum_{v \in V} \omega_v \cdot \tilde{\sigma}_v(S)$ and hence $\max_{p \in \mathcal{S}}\{\sum_{C \in \mathcal{C}} z_C \tilde{\sigma}_C(p)\}$ is attained by a vector that assigns 1 to a set that maximizes $\tilde{\sigma}^\omega(\cdot)$ over all sets in $\binom{V}{k}$ and 0 to all other sets. Hence solutions to the oracle problem can be obtained by exact or approximate solutions to the problem of maximizing the set function $\tilde{\sigma}^\omega(S)$ with respect to a cardinality constraint $|S| \leq k$. The crucial observation here is that $\tilde{\sigma}^\omega(S)$ is submodular and monotone and thus can be approximated within a factor of $1 - 1/e$ using the greedy algorithm. The submodularity property is evident as $\tilde{\sigma}^\omega$ is an approximation (obtained via sampling through the Hoeffding bound) to the weighted influence function $\sigma^\omega(S) := \sum_{v \in V} \omega_v \cdot \sigma_v(S)$. Hence we get the following theorem.

Theorem 6. *Let $\delta \in (0, \frac{1}{2})$ and $\varepsilon \in (0,1)$. There is a polynomial time algorithm that, with probability at least $1-\delta$, computes $p \in \mathcal{S}$ s.t. $\min_{C \in \mathcal{C}} \sigma_C(p) \geq (1 - \frac{1}{e}) \operatorname{opt}_{\mathcal{S}}(G, \mathcal{C}, k) - \varepsilon$. Moreover, the support of p is $O(\varepsilon^{-2} n \log m/k)$.*

Probabilistically Choosing Nodes

We turn to the node-based problem $\operatorname{opt}_{\mathcal{X}}(G, \mathcal{C}, k) := \max_{x \in \mathcal{X}} \min_{C \in \mathcal{C}} \sigma_C(x)$, where $\sigma_C(x) = \mathbb{E}_{S \sim x}[\sigma_C(S)]$ for $C \in \mathcal{C}$ and $x \in \mathcal{X} := \{x \in [0,1]^n : \mathbb{1}^T x \leq k\}$. Recall that $S \sim x$ denotes the random process of independently letting $i \in V$ be in S with probability x_i. Analogous to the set-based problem, in what follows, we thus have to argue how to get a good approximation algorithm for the problem As for the set-based problem, we use Lemma 4 and thus focus on finding good approximations to $\max_{x \in \mathcal{X}} \min_{C \in \mathcal{C}} \tilde{\sigma}_C(x)$. We first observe that Theorem II.5 from (Chekuri, Vondrák, and Zenklusen 2010) in combination with a binary search on a threshold can be used in order to get a a $(1-1/e, 0)$-approximation for that problem. In what follows we give a more direct derivation of such an approximation. The approach fundamentally differs from the set-based problem.

Node-based Problem via LP. In particular, the problem here is not linear as, for given x, the probability to sample $S \in 2^V$ is equal to $\prod_{i \in S} x_i \prod_{i \notin S}(1 - x_i)$. We argue however that the problem can be constantly approximated by an LP.

For a live-edge graph L and a node $v \in V$, what is the probability of sampling a set S that can reach v in L, i.e., what is $q_v(L, x) := \Pr_{S \sim x}[v \in \rho_L(S)]$? It is the opposite event of not sampling any node that can reach v in L, hence $q_v(L, x) = 1 - \prod_{i \in V : v \in \rho_L(i)}(1 - x_i)$ and this is approximated by the function $p_v(L, x) := \min\{1, \sum_{i \in V : v \in \rho_L(i)} x_i\}$:

Observation 7. *For any live-edge graph L, node $v \in V$, and $x \in \mathcal{X}$, it holds that $q_v(L, x) \in [(1 - \frac{1}{e}) \cdot p_v(L, x), p_v(L, x)]$.*

Defining $\lambda_v(x) := \frac{1}{T} \sum_{t=1}^T p_v(L_t, x)$ and analogously $\lambda_C(x) := \frac{1}{|C|} \sum_{v \in C} \lambda_v(x)$ directly yields the following lemma.

Lemma 8. *Let $x \in \mathcal{X}$ be an optimal solution to $\max_{x \in \mathcal{X}} \min_{C \in \mathcal{C}} \lambda_C(x)$, then x is a $(1 - 1/e, 0)$-approximation to $\max_{x \in \mathcal{X}} \min_{C \in \mathcal{C}} \tilde{\sigma}_C(x)$.*

Together with Lemma 4 and the fact that the above problem can be solved by an LP we get the following result.

Theorem 9. *Let $\delta \in (0, \frac{1}{2})$ and $\varepsilon \in (0,1)$. There is a polynomial time algorithm that, with probability at least $1 - \delta$, computes $x \in \mathcal{X}$ s.t. $\min_{C \in \mathcal{C}} \sigma_C(x) \geq (1 - \frac{1}{e}) \operatorname{opt}_{\mathcal{X}}(G, \mathcal{C}, k) - \varepsilon$.*

Experiments

This section reports on an experimental study on the probabilistic maximin problems. We provide implementations of multiplicative-weight routines for both the set-based and the node-based problems. The routine for the set-based problem is the one described above. For the node-based problem, an implementation of the LP-based algorithm does not seem promising as it requires solving a large LP. Instead, we propose a heuristic approach that is again based on a multiplicative-weight routine. The essential observation is that the optimization problem $\max_{x \in \mathcal{X}} \min_{C \in \mathcal{C}} \lambda_C(x)$ from Lemma 8 is again a covering LP and thus can be solved using a multiplicative-weight routine again. In this case however, the oracle problem turns out to be the LP-relaxation of the standard influence maximization problem and thus we are again faced with a linear program of a similar form. This is where our approach becomes heuristic, we propose to again use the greedy algorithm for influence maximization in order to obtain feasible solutions for this LP. While this comes without any guarantee on approximation ratio, it is very efficient in practice and in fact yields very similar results to the algorithm for the set-based problem. We stress that our implementations, at this point, have to be considered proof-of-concept implementations and are not tuned for run-time efficiency at all. In fact, we implement all routines and subroutines on our own in Python and refrain from using the very efficient implementations of the greedy algorithm for influence maximization that exist in the literature for reasons of simplicity. Our implementation also does not use any parallelization. While it is easily perceivable that this could lead to big run-time improvements (both for the greedy algorithm and for the multiplicative-weight routines (Young 2001)), this is out of the scope of the work reported here. We rather focus on measuring the fairness achieved by our methods and compare it to results of the algorithms proposed by Fish et al. (2019) and Tsang et al. (2019).

Experimental Setting. Just like Fish et al. (2019) and Tsang et al. (2019), we use the IC model as diffusion model and a constant number of live-edge graphs for simulating the information spread. We explored both setting edge weights uniformly at random and setting them to a constant. This does not seem to have a big impact on the results; we report results from both choices while specifying the choice for each experiment. Each datapoint in our plots is the result of averaging over 25 experiments, 5 runs on each of 5 graphs generated according to the respective graph model. Error-bars in our plots indicate 95-% confidence intervals. For the multiplicative weight routine we choose the $\eta = 0.1$.

We report both ex-ante and ex-post fairness values for our methods (for short, we use *ea* and *ep*). These have the following precise meaning. After computing probabilistic strategies p or x for the set-based and node-based problem, the ex-ante values correspond to the objective values $\min_{C \in \mathcal{C}} \mathbb{E}_{S \sim p}[\sigma_C(S)]$ for the set-based and $\min_{C \in \mathcal{C}} \mathbb{E}_{S \sim x}[\sigma_C(S)]$ for the node-based problem. The ex-post values on the other side are obtained by sampling a set S according to the probabilistic strategy p or x respectively and then reporting the value $\min_{C \in \mathcal{C}} \sigma_C(S)$. In our evaluation, we also compare to the uniform distribution for the node-based problem, i.e., the distribution that uniformly selects every node as a seed with probability k/n, we call this "method" `uf_node_based` in our plots. We report also both ex-ante and ex-post values for the method of Tsang et al. (2019), since, at the core, their algorithm works with the multilinear extension and thus also computes a continuous solution $x \in \mathbb{R}^n$, i.e., a feasible solution to the node-based problem. Hence for their method we report both the value $\min_{C \in \mathcal{C}} \mathbb{E}_{S \sim x}[\sigma_C(S)]$ as ex-ante value and a value $\min_{C \in \mathcal{C}} \sigma_C(S)$ as ex-post value, where S is computed by swap rounding from x as described in their paper.

Results. We evaluate our implementations on random instances and those instances from the work of (Tsang et al. 2019) that are publicly available. As random instances we choose graphs generated according to the Barabasi-Albert as well as by the block-stochastic graph model. We also explore different community structures. The instance generators are available within the implementation provided.

For the Barabasi-Albert model, we choose the parameter $m = 2$, i.e., every newly introduced node during the generation of the graph is connected to two previously existing nodes. We explored the following community structures: (1) singleton communities: every node is his own community. (2) BFS community structure: For a predefined number of communities m, we iteratively grow communities of size n/m by BFS from a random source node (once there are no more reachable nodes but the community is still not of size n/m, we pick a new random source, until every node is in one of the m communities. In the plots reported here we chose $m = k$ as in this case the algorithm of Tsang et al. satisfies its approximation guarantee. This way of choosing the communities results in a rather connected community structure. (3) Random imbalanced community structure: we randomly assign nodes to one of 4 communities, the community sizes are fixed to $4n/10, 3n/10, 2n/10, n/10$. This results in rather unconnected community structures. The results are reported in Figure 2. We can see that the ex-ante values of our methods dominate over all other values. Furthermore, we can see that particularly in the last plot, where the community structure is less simplistic, even the ex-post values of our methods are higher than the ones of all competitors.

We generate Block Stochastic graph instances to further explore how the connectivity of the community structure influences the performance of the different approaches. We fix the number of nodes to 120, the number of communities to 6 and their sizes to $4n/12, 3n/12, 2n/12, n/12, n/12, n/12$. We then choose a parameter p that we vary from 0.03 to 0.27 in steps of 0.03 and create a sequence of instances where the probability of an edge within a community is p and between communities $0.3 - p$. The results are reported in Figure 3. Again the ex-ante values of our methods dominate all others. Among the ex-post values, greedy performs best with our method second. There seems to be a trend that for higher p, i.e., communities are better connected within each other than between each other, there is a bigger advantage for ex-ante fairness over ex-post fairness algorithms.

We conclude with the instances used by Tsang et al. (2019). These are synthetic networks introduced by

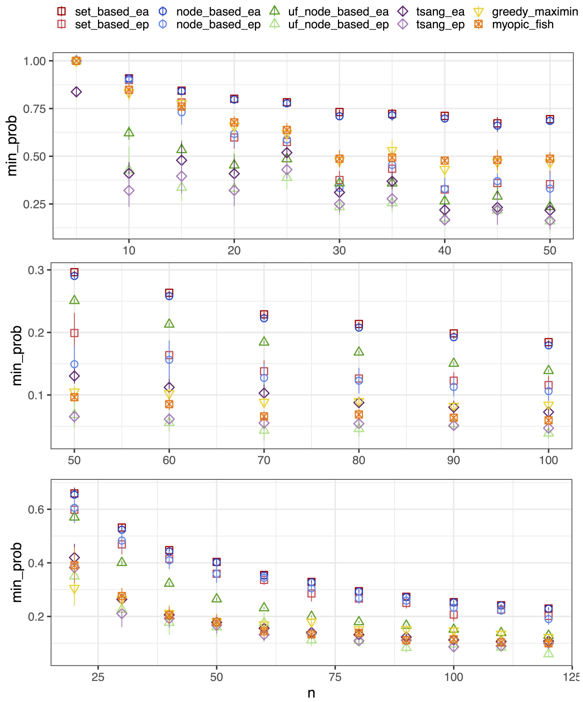

Figure 2: Barabasi Albert instances: (1) Singleton community structure, edge weights uniformly at random, $k = 10$, n increasing from 5 to 50 in steps of 5. (2) BFS community structure, edge weights constant 0.1, $k = 10$, n increasing from 50 to 100 in steps of 10. (3) Random imbalanced community structure, edge weights constant 0.1, $k = 10$, n increasing from 20 to 120 in steps of 10.

Wilder et al. (2018b) in order to analyze the effects of health interventions. Each of the 500 nodes in these networks has some attributes (region, ethnicity, age, gender, status) and more similar nodes are more likely to share an edge. The attributes induce communities and we test, as proposed by Tsang et al. (2019), all algorithms w.r.t. group fairness of the communities induced by some of those attributes. Figure 4 shows the results. Again the ex-ante values of our methods dominate over all other values in both experiments reported. In the first experiment (communities induced by gender) the ex-post values of our algorithms and of the algorithm of Tsang et al. are very close to their respective ex-ante values. All algorithms are close to each other and the achieved minimum probability of the communities is rather large. For the second experiment (communities induced by region), the achieved minimum probability of all communities is much smaller. Again, our algorithm performs best among all algorithms in the ex-ante values and best in the ex-post values as well. In this case, there is a large advantage for ex-ante over ex-post values (even for the uniform distribution).

Conclusions

We have presented new results on the problem of determining key seed nodes in order to influence the users of a social network in an efficient and fair manner. Notably, we have designed approximation algorithms achieving a constant multiplicative factor of $1 - 1/e$ (plus an additive arbitrarily small error) for the objective of maximizing the maximin influence received by a community. We achieved this by using randomized strategies, thus enlarging the solution set and enabling us to find fairer solutions ex-ante. Our detailed experimental study confirms the increase in ex-ante fairness achieved over previous methods (Fish et al. 2019; Tsang et al. 2019), indicating that randomness as source of fairness in influence maximization is very promising to be further explored. Interestingly, our study shows that even the ex-post fairness achieved by our methods frequently outperforms the fairness achieved by other tested methods.

Several future work directions are conceivable. Improving our approximation guarantees or providing approximation hardness results seems challenging. Moreover, efficiently engineering our methods could yield a big practical impact. Lastly, using randomization to increase fairness for influence maximization may be used for other fairness criteria as, e.g., the group rational criterion of Tsang et al. (2019).

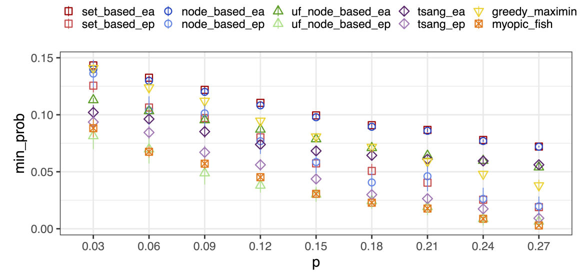

Figure 3: Block stochastic graphs: edge weights 0.05, $k = 6$, $n = 120$, p increasing from 0.03 to 0.27 in steps of 0.03.

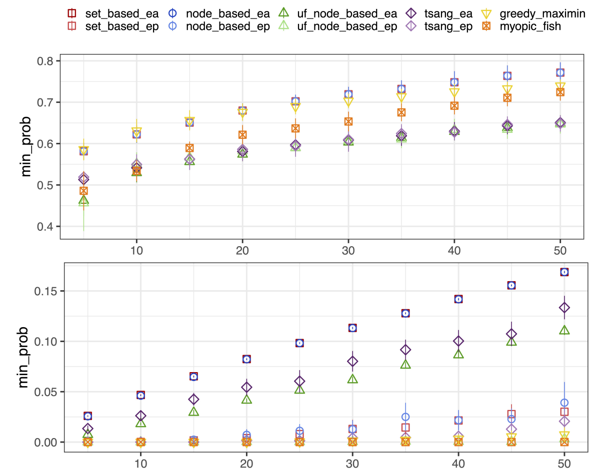

Figure 4: Instances of Tsang et al. (2019): (1) Community structure induced by attribute gender, edge weights u.a.r., k increasing from 5 to 50 in steps of 5. (2) Community structure induced by attribute region, edge weights 0.1, k increasing from 5 to 50 in steps of 5.

Acknowledgments

This work was partially supported by the Italian MIUR PRIN 2017 Project "ALGADIMAR" Algorithms, Games, and Digital Markets.

References

Ali, J.; Babaei, M.; Chakraborty, A.; Mirzasoleiman, B.; Gummadi, K. P.; and Singla, A. 2019. On the Fairness of Time-Critical Influence Maximization in Social Networks. *arXiv preprint arXiv:1905.06618* .

Aziz, H. 2019. A probabilistic approach to voting, allocation, matching, and coalition formation. In *The Future of Economic Design*, 45–50. Springer.

Aziz, H.; Brandt, F.; and Stursberg, P. 2013. On popular random assignments. In *International Symposium on Algorithmic Game Theory*, 183–194. Springer.

Aziz, H.; Luo, P.; and Rizkallah, C. 2018. Rank maximal equal contribution: A probabilistic social choice function. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Banerjee, A.; Chandrasekhar, A. G.; Duflo, E.; and Jackson, M. O. 2013. The diffusion of microfinance. *Science* 341(6144): 1236498.

Becker, R.; Coro, F.; D'Angelo, G.; and Gilbert, H. 2020a. Balancing Spreads of Influence in a Social Network. In *Proceedings of the Thirty-Fourth AAAI conference on artificial intelligence, AAAI 2020, New York, USA, February 7-12,2020*.

Becker, R.; D'Angelo, G.; Ghobadi, S.; and Gilbert, H. 2020b. Fairness in Influence Maximization through Randomization. *arXiv preprint arXiv:2010.03438* .

Bogomolnaia, A.; and Moulin, H. 2001. A new solution to the random assignment problem. *Journal of Economic theory* 100(2): 295–328.

Borgs, C.; Brautbar, M.; Chayes, J. T.; and Lucier, B. 2014. Maximizing Social Influence in Nearly Optimal Time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, 946–957.

Borodin, A.; Braverman, M.; Lucier, B.; and Oren, J. 2017. Strategyproof mechanisms for competitive influence in networks. *Algorithmica* 78(2): 425–452.

Brandl, F.; Brandt, F.; and Seedig, H. G. 2016. Consistent probabilistic social choice. *Econometrica* 84(5): 1839–1880.

Brandt, F. 2019. Collective Choice Lotteries. In *The Future of Economic Design*, 51–56. Springer.

Budak, C.; Agrawal, D.; and El Abbadi, A. 2011. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, 665–674.

Chekuri, C.; Vondrák, J.; and Zenklusen, R. 2010. Dependent Randomized Rounding via Exchange Properties of Combinatorial Structures. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, 575–584.

Chen, W.; and Teng, S.-H. 2017. Interplay between social influence and network centrality: a comparative study on shapley centrality and single-node-influence centrality. In *Proceedings of the 26th international conference on world wide web (WWW)*, 967–976.

Chen, Y.; Wang, W.; Feng, J.; Lu, Y.; and Gong, X. 2020. Maximizing multiple influences and fair seed allocation on multilayer social networks. *Plos one* 15(3): e0229201.

Cohen, E.; Delling, D.; Pajor, T.; and Werneck, R. F. 2014. Sketch-based Influence Maximization and Computation: Scaling up with Guarantees. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, 629–638.

David, M. 2013. *Algorithmics of matching under preferences*, volume 2. World Scientific.

Farnad, G.; Babaki, B.; and Gendreau, M. 2020. A Unifying Framework for Fairness-Aware Influence Maximization. In *Companion Proceedings of the Web Conference 2020*, 714–722.

Fish, B.; Bashardoust, A.; Boyd, D.; Friedler, S.; Scheidegger, C.; and Venkatasubramanian, S. 2019. Gaps in Information Access in Social Networks? In *The World Wide Web Conference*, 480–490. ACM.

Katta, A.-K.; and Sethuraman, J. 2006. A solution to the random assignment problem on the full preference domain. *Journal of Economic theory* 131(1): 231–250.

Kempe, D.; Kleinberg, J. M.; and Tardos, É. 2015. Maximizing the Spread of Influence through a Social Network. *Theory of Computing* 11: 105–147.

Khajehnejad, M.; Rezaei, A. A.; Babaei, M.; Hoffmann, J.; Jalili, M.; and Weller, A. 2020. Adversarial Graph Embeddings for Fair Influence Maximization over Social Networks. *arXiv preprint arXiv:2005.04074* .

Lu, W.; Bonchi, F.; Goyal, A.; and Lakshmanan, L. V. 2013. The bang for the buck: fair competitive viral marketing from the host perspective. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 928–936.

Machina, M. J. 1989. Dynamic consistency and non-expected utility models of choice under uncertainty. *Journal of Economic Literature* 27(4): 1622–1668.

Moulin, H. 1991. *Axioms of cooperative decision making*. 15. Cambridge University Press.

Rahmattalabi, A.; Jabbari, S.; Lakkaraju, H.; Vayanos, P.; Rice, E.; and Tambe, M. 2020. Fair Influence Maximization: A Welfare Optimization Approach. *arXiv preprint arXiv:2006.07906* .

Stoica, A.-A.; Han, J. X.; and Chaintreau, A. 2020. Seeding Network Influence in Biased Networks and the Benefits of Diversity. In *WWW2020*.

Tang, Y.; Shi, Y.; and Xiao, X. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, 1539–1554.

Tang, Y.; Xiao, X.; and Shi, Y. 2014. Influence maximization: near-optimal time complexity meets practical efficiency. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, 75–86.

Tsang, A.; Wilder, B.; Rice, E.; Tambe, M.; and Zick, Y. 2019. Group-Fairness in Influence Maximization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 5997–6005.

Udwani, R. 2018. Multi-objective Maximization of Monotone Submodular Functions with Cardinality Constraint. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, 9513–9524.

Valente, T. W.; and Pumpuang, P. 2007. Identifying opinion leaders to promote behavior change. *Health education & behavior* 34(6): 881–896.

Wilder, B.; Onasch-Vera, L.; Hudson, J.; Luna, J.; Wilson, N.; Petering, R.; Woo, D.; Tambe, M.; and Rice, E. 2018a. End-to-end influence maximization in the field. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 1414–1422. International Foundation for Autonomous Agents and Multiagent Systems.

Wilder, B.; Ou, H.; de la Haye, K.; and Tambe, M. 2018b. Optimizing Network Structure for Preventative Health. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, 841–849.

Yadav, A.; Wilder, B.; Rice, E.; Petering, R.; Craddock, J.; Yoshioka-Maxwell, A.; Hemler, M.; Onasch-Vera, L.; Tambe, M.; and Woo, D. 2018. Bridging the Gap Between Theory and Practice in Influence Maximization: Raising Awareness about HIV among Homeless Youth. In *IJCAI*, 5399–5403.

Young, N. E. 1995. Randomized Rounding Without Solving the Linear Program. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1995. San Francisco, California, USA*, 170–178.

Young, N. E. 2001. Sequential and Parallel Algorithms for Mixed Packing and Covering. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, 538–546.

Yu, Y.; Jia, J.; Li, D.; and Zhu, Y. 2017. Fair Multi-influence Maximization in Competitive Social Networks. In *International Conference on Wireless Algorithms, Systems, and Applications*, 253–265. Springer.