Improving Fairness in Information Exposure by Adding Links

Ruben Becker[1], **Gianlorenzo D'Angelo**[2], **Sajjad Ghobadi**[2]
[1]Ca' Foscari University of Venice, Italy
[2]Gran Sasso Science Institute, L'Aquila, Italy
rubensimon.becker@unive.it, gianlorenzo.dangelo@gssi.it, sajjad.ghobadi@gssi.it

Abstract

Fairness in influence maximization has been a very active research topic recently. Most works in this context study the question of how to find seeding strategies (deterministic or probabilistic) such that nodes or communities in the network get their fair share of coverage. Different fairness criteria have been used in this context. All these works assume that the entity that is spreading the information has an inherent interest in spreading the information fairly, otherwise why would they want to use the developed fair algorithms? This assumption may however be flawed in reality – the spreading entity may be purely *efficiency-oriented*. In this paper we propose to study two optimization problems with the goal to modify the network structure by adding links in such a way that efficiency-oriented information spreading becomes *automatically fair*. We study the proposed optimization problems both from a theoretical and experimental perspective, that is, we give several hardness and hardness of approximation results, provide efficient algorithms for some special cases, and more importantly provide heuristics for solving one of the problems in practice. In our experimental study we then first compare the proposed heuristics against each other and establish the most successful one. In a second experiment, we then show that our approach can be very successful in practice. That is, we show that already after adding a few edges to the networks the greedy algorithm that purely maximizes spread surpasses all fairness-tailored algorithms in terms of ex-post fairness. Maybe surprisingly, we even show that our approach achieves ex-post fairness values that are comparable or even better than the ex-ante fairness values of the currently most efficient algorithms that optimize ex-ante fairness.

Introduction

The question of how information spreads through networks has been studied in various research disciplines. In computer science, the so-called influence maximization (IM) paradigm has attracted a lot of attention in the last two decades. The IM problem can be stated as follows. Given a social network $G = (V, E)$ in which information spreads according to some probabilistic model, target a set of at most k seed nodes $S \subseteq V$ in such a way that $\sigma(S)$, the expect number of nodes that receive the information, is maximized (Kempe, Kleinberg, and Tardos 2015).

As online social networks play an essential role in how we acquire information nowadays and as access to information has an important impact on our lives, more recently researchers have started to study the IM framework in the presence of fairness concerns as well. Fairness may be understood w.r.t. individuals or communities, the former being the special case of the latter with singleton communities. Generally, in such works, for a community $C \subseteq V$, one considers the average probability $\sigma_C(S)$ of nodes in C to be reached from S, also called the *community coverage* of C. Then the concern is to choose S in such a way that some fairness criteria on the communities is maximized. The probably most commonly used one is the maxmin (or maximin) criterion. In the most basic setting, studied, e.g., by Fish et al. (2019), the goal is to find a seed set $S \subseteq V$ of size at most k such that the minimum probability that nodes are reached $\min_{v \in V} \sigma_v(S)$ is maximized.

Several articles have been published in this scope and they have shown certain success in finding seeding strategies that lead to fairer outcomes. Nevertheless, they are all based on the assumption that the information spreading entity, i.e., the agent choosing S, has an interest in spreading information in a fair way. This assumption is however rather unrealistic in the real word. Information spreading agents may be, and probably mostly are, purely efficiency-oriented and not particularly interested in choosing fair seeding strategies.

In this work we take a different approach to fairness. We do not rely on the good will of the information spreading entity, but instead modify the underlying social network in such a way as to make efficiency-oriented information spreading automatically fair. The modification of the network may be done by the network owner or any other entity interested in guaranteeing fairness. While different ways of modifying the network are perceivable, we choose the possibly most natural one – we improve the network's connectivity by adding links. Here, we take the rather realistic approach to assume the information spreading entity to be indifferent rather than adversarial towards fairness.

Our Contribution. We formalize this problem as follows. Given a social network $G = (V, E)$, we want to add at most b non-edges $F \subseteq \bar{E} = V^2 \setminus E$ to G in such a way that the minimum community coverage is maximized when information is spread in $G' = (V, E \cup F)$ using a purely efficiency oriented seeding strategy, i.e., a seed set S of size k that maximizes the spread in G', we measure this using the function $\sigma(\cdot, F)$. We call this the $\mathrm{FIM_{AL}}$ problem – fair influence maximization by adding links. We study the complexity of solving $\mathrm{FIM_{AL}}$ (Section The $\mathrm{FIM_{AL}}$ Problem: Making Spread Maximizers Fair) and provide plenty of evidence that solving $\mathrm{FIM_{AL}}$ is challenging, both exactly and approximately. Maybe most importantly, we show that it is unlikely to be able to find an α-approximation to the optimal solution, for any $\alpha \in (0, 1]$, even when having access to an oracle that solves an NP-complete problem. We furthermore show that $\mathrm{FIM_{AL}}$ remains NP-hard for constant b or k (in the latter case even to be approximated).

We thus turn to study a second problem (Section The $\mathrm{FIM^g_{AL}}$ Problem: Towards Fairness in Practice) that is possibly practically better motivated in the first place – the $\mathrm{FIM^g_{AL}}$ problem: Here instead of assuming that the efficiency oriented entity uses maximizing sets to spread information, we assume it to employ the greedy algorithm. This is a quite realistic assumption as the problem of finding a maximizing set is NP-hard, while the greedy algorithm can be used in order to obtain a $1 - 1/e - \varepsilon$-approximation for any $\varepsilon \in (0, 1)$ w.h.p. in $\mathrm{poly}(n, \varepsilon^{-1})$ time, i.e, polynomial time in $n = |V|$ and ε^{-1}. Even more, this approximation guarantee is essentially optimal (Kempe, Kleinberg, and Tardos 2015). Multiple implementations of the greedy algorithm for IM exist (e.g., (Tang, Xiao, and Shi 2014; Tang, Shi, and Xiao 2015)) and they have been shown to be extremely efficient in practice. We observe that, in contrast to $\mathrm{FIM_{AL}}$, the $\mathrm{FIM^g_{AL}}$ problem is polynomial time solvable when b is a constant – exactly in the (unrealistic) case of deterministic instances and up to an arbitrarily small additive error in the probabilistic case. While this highlights the difference between the two problems, the proposed algorithm is essentially a brute-force algorithm and is thus not promising in practice. We complement the finding of this algorithm for the special case of constant k with a lower bound showing that it is NP-hard to provide any approximation algorithm. We then propose a set of algorithms for $\mathrm{FIM^g_{AL}}$ and evaluate them against each other in a first experiment in Section Experiments. We then take the best performing algorithm for $\mathrm{FIM^g_{AL}}$ and, in a second experiment, compare the resulting fairness (i.e., fairness achieved by the greedy algorithm after adding the proposed non-edges to the graph) with competitor algorithms that choose seeds as to optimize fairness. We observe that already after adding very few edges to graphs with thousands of nodes, the fairness achieved by our algorithm outperforms the fairness achieved by the fairness-tailored algorithms. Maybe surprisingly, this even holds for algorithms that optimize ex-ante fairness.

We summarize our theoretical results for $\mathrm{FIM_{AL}}$ and $\mathrm{FIM^g_{AL}}$ in Table 1 together with references to the respective statements in later sections. Due to space limitation some proofs are deferred to the full version (Becker, D'Angelo, and Ghobadi 2023).

Related Work. There is a rich set of related works in this area, so we are able to summarize only the results most related to our work. Fish et al. (2019) were the first to study the maximin criterion in influence maximization w.r.t. single nodes. Tsang et al. (2019) study the maximin criterion w.r.t. groups. Becker et al. (2022) also consider the maximin criterion for groups, but allow probabilistic seeding strategies. Stoica and Chaintreau (2019) analyze the fairness achieved by standard algorithms for influence maximization.

	general	constant b	constant k
$\mathrm{FIM_{AL}}$	Σ_2^p-hard [Thm. 0.4] Σ_2^p-hard to α-approx. [Thm. 0.5]	NP-hard [Thm. 0.7]	NP-hard to α-approx. [Thm. 0.6]
$\mathrm{FIM^g_{AL}}$	NP-hard to α-approx. [Cor. 0.9]	- poly. time (determ.) [Obs. 0.8] - ε-approx. (prob.) [Lem. 0.11]	NP-hard to α-approx. [Cor. 0.9]

Table 1: Summary of our complexity results. The number α can be any factor in $(0, 1]$.

There are several works in which the authors add links to the network, however they do so with a different objective. Both Castiglioni, Ferraioli, and Gatti (2020) and Corò, D'Angelo, and Velaj (2021) study the problem of adding edges to a graph in order to maximize the influence from a given seed set in different models of diffusion. D'Angelo, Severini, and Velaj (2019) study the problem of adding a set of edges *incident to a given seed set* with the same aim. Wu, Sheldon, and Zilberstein (2015) consider also different intervention actions than just adding edges, e.g., increasing the weights of edges. Khalil, Dilkina, and Song (2014) study both the edge addition and deletion problems in order to maximize/minimize influence in the linear threshold model.

Swift et al. (2022) introduce a problem to suggest a set of edges that contains at most k edges incident to each node to maximize the expected number of reached nodes while satisfying a fairness constraint (reaching each group in the network with the same probability). Garimella et al. (2017) address the problem of recommending a set of edges to minimize the controversy score of the graph. Tong et al. (2012) transform the edge addition/deletion problem to the problem of maximizing/minimizing the eigenvalue of the adjacency matrix. Amelkin and Singh (2019) propose an edge recommendation algorithm to disable an attacker that aims to change the network's opinion by influencing users.

Preliminaries

For an integer k, we denote with $[k]$ the set of integers from 1 to k. We say that an event holds with high probability (w.h.p.), if it holds with probability at least $1 - n^{-\alpha}$ for a constant α that can be made arbitrarily large.

Information Diffusion. Given a directed graph $G = (V, E, w)$ with n nodes V, edge set E, and edge weight function $w : V^2 \to [0, 1]$, we use the *Independent Cascade model* (Kempe, Kleinberg, and Tardos 2015) for describing the random process of information diffusion. For an initial seed set $S \subseteq V$ the spread $\sigma(S)$ from S is the expected number of nodes that are reached from S in a random *live-edge graph* which is constructed as follows. Every node v independently picks a *triggering set* T_v by letting each u in its set of in-neighbors N_v be in T_v independently with probability w_e, where $e = (u, v)$. We then let $E_L := \bigcup_{v \in V}\{(u, v) : u \in T_v\}$ and we call $L = (V, E_L)$ a random live-edge graph. We then define $\rho_L(S)$ as the set of nodes reachable from S in L and the expected number of nodes reached from S is $\sigma(S) := \mathbb{E}[|\rho_L(S)|]$, where the expectation is over the random generation of the live-edge graph L. We furthermore define $\sigma_v(S) := \mathbb{E}[\mathbb{1}_{v \in \rho_L(S)}] = \Pr[v \in \rho_L(S)]$ for every node $v \in V$, i.e., $\sigma_v(S)$ is the probability of v being reached from S. For a set (or group) of nodes $C \subseteq V$, we let $\sigma_C(S) := \frac{1}{|C|}\sum_{v \in C}\sigma_v(S)$ be the group coverage of C.

When the edge probabilities belong to $\{0, 1\}$, we refer to the instance as the *deterministic case*, in this case $\sigma(S)$ is the (deterministic) number of nodes reachable from seeds S in G. In the general case, it is not feasible to compute $\sigma(S)$ via all live-edge graphs L, instead a $(1 \pm \varepsilon)$-approximation of $\sigma(S)$ can be obtained w.h.p. by averaging over $\mathrm{poly}(n, \varepsilon^{-1})$ many live-edge graphs L, see, e.g., Proposition 4.1 in the work of Kempe, Kleinberg, and Tardos (2015). Similarly, $\sigma_v(S)$ (and thus also $\sigma_C(S)$) can be approximated w.h.p., however, only to within an additive error of $\pm\varepsilon$ by averaging over $\mathrm{poly}(n, \varepsilon^{-1})$ many live-edge graphs, see, e.g., Lemma 4.1 in the work of Becker et al. (2022).

Non-Edges and Spread with Added Edges. We let $\bar{E} := (V \times V) \setminus E$ denote the set of *non-edges* in G. For a set of non-edges $F \subseteq \bar{E}$ and a set of seed nodes $S \subseteq V$, we define $\sigma(S, F)$ as the expected number of nodes reached from S in the graph $G' = (V, E \cup F)$ that results from adding F to G. This is the reason why we have defined the edge weight function also w.r.t. non-edges above. Similarly, for a node $v \in V$, $\sigma_v(S, F)$ is the probability that v is reached from S in G' and, for a community $C \subseteq V$, we define $\sigma_C(S, F)$ to be the average probability of nodes in C being reached from S in G'. We remark that also these functions cannot be computed exactly but only approximated in the same way as their counterparts without added edges.

The $\mathrm{FIM}_{\mathrm{AL}}$ Problem: Making Spread Maximizers Fair

Problem Definition. Consider a directed weighted graph $G = (V, E, w)$ and let $\mathcal{C}$ be a community structure, i.e., m non-empty communities $C \subseteq V$, and let k and b be two integers. For a set of non-edges $F \subseteq \bar{E}$, we define $\mathcal{M}(F, k) := \mathrm{argmax}_{S \subseteq V}\{\sigma(S, F) : |S| \leq k\}$ to be the set of size k maximizers to $\sigma(\cdot, F)$. We are now ready to formally define the $\mathrm{FIM}_{\mathrm{AL}}$ problem motivated above:

$$\max_{F \subseteq \bar{E} : |F| \leq b} \big\{\tau : \min_{C \in \mathcal{C}} \sigma_C(S, F) \geq \tau,\ \forall S \in \mathcal{M}(F, k)\big\}.$$

We denote with $\mathrm{opt}_{\mathrm{AL}}(G, \mathcal{C}, b, k)$ the optimum of $\mathrm{FIM}_{\mathrm{AL}}$. Clearly, our objective in $\mathrm{FIM}_{\mathrm{AL}}$ is to find a set of at most b non-edges $F \subseteq \bar{E}$, that, when added to G, maximizes the minimum community coverage when information is spread in a purely "efficiency-oriented" way, i.e., from a set of at most k seed nodes that is chosen such that the set function $\sigma(\cdot, F)$ is maximized. The motivation behind studying $\mathrm{FIM}_{\mathrm{AL}}$ is to, e.g., as the network owner, change the structure of a social network in such a way that an efficiency-oriented entity that wants to spread information in G automatically spreads information in a more fair way.

In what follows, we give several hardness and hardness of approximation results for $\mathrm{FIM}_{\mathrm{AL}}$. We start by showing that the decision version of the general $\mathrm{FIM}_{\mathrm{AL}}$ problem is Σ_2^p-hard. We even show that it is unlikely that $\mathrm{FIM}_{\mathrm{AL}}$ can be approximated to within any factor. We then turn to special cases of $\mathrm{FIM}_{\mathrm{AL}}$ where either $b = 1$ or $k = 1$ and show that the problem remains NP-hard also in these special cases – for $k = 1$ even hard to approximate to within any factor.

For better comprehensibility, we first note that in the the decision version of $\mathrm{FIM}_{\mathrm{AL}}$, in addition to the graph $G = (V, E)$, the communities $\mathcal{C}$, and the integers b, k, we are given a threshold t and the task is to decide if there exists $F \subseteq \bar{E}$ with $|F| \leq b$ such that for all $S \in \mathcal{M}(F, k)$: $\min_{C \in \mathcal{C}} \sigma_C(S, F) \geq t$.

Σ_2^p-Hardness. We start by recalling the definition of the complexity class Σ_2^p.

Definition 0.1 (Definition 5.1 in (Arora and Barak 2009))**.** *The class Σ_2^p is defined to be the set of all languages L for which there exists a polynomial-time Turing machine M and a polynomial q such that $x \in L$ if and only if $\exists u \in \{0, 1\}^{q(|x|)} : \forall v \in \{0, 1\}^{q(|x|)} : M(x, u, v) = 1$.*[1]

We next introduce the $\Sigma_2\,\mathrm{SAT}$ problem which is Σ_2^p-complete, see, e.g., Exercise 1 in Chapter 5 of the book by Arora and Barak (2009).

Definition 0.2 (Example 5.6 in (Arora and Barak 2009))**.** *Given a boolean expression $\phi(X, Y)$ in 3-CNF with variables $X = (x_1, \ldots, x_\nu)$ and $Y = (y_{\nu+1}, \ldots, y_\mu)$, the $\Sigma_2\,\mathrm{SAT}$ problem entails to decide if $\exists x \forall y : \phi(x, y) = \top$, where $x : X \to \{0, 1\}$ and $y : Y \to \{0, 1\}$ are assignments to the variables X and Y, respectively.*

For ease of presentation, we assume the indices of Y to start at $\nu + 1$, such that indices of X and Y are disjoint. Our goal now is to show that the decision version of $\mathrm{FIM}_{\mathrm{AL}}$ is Σ_2^p-hard. We will describe a reduction from $\Sigma_2\,\mathrm{SAT}$ to the decision version of $\mathrm{FIM}_{\mathrm{AL}}$. We assume that $\phi(X, Y)$ contains m clauses $\phi_1, \ldots, \phi_m$ and for a clause ϕ_r we call $r(s)$, $s \in [3]$, the indices of the three variables corresponding to ϕ_r's three literals (in arbitrary fixed order).

Given an instance of $\Sigma_2\,\mathrm{SAT}$, we create an instance $(G, \mathcal{C}, b, k, t)$ of the decision version of $\mathrm{FIM}_{\mathrm{AL}}$ as follows, see Fig. 1 for an illustration. Fix a constant $M := \mu + \nu + 6m + 1$. The node set V of G consists of

- $U = \{q, P\}$, where $P = p_1, \ldots, p_{M-1}$,
- $V^\exists = \{v_i, \bar{v}_i : i \in [\nu]\}$ and $V^\forall = \{v_j, \bar{v}_j, L_j : j \in [\mu] \setminus [\nu]\}$, where $L_j = l_{j,1}, \ldots, l_{j,M-2}$, and

[1]Equivalently, see, e.g., Theorem 5.12 and Remark 5.16 in the same book, Σ_2^p can be defined as the set of all languages that can be decided by a non-deterministic Turing machine with access to an oracle that solves some NP-complete problem.

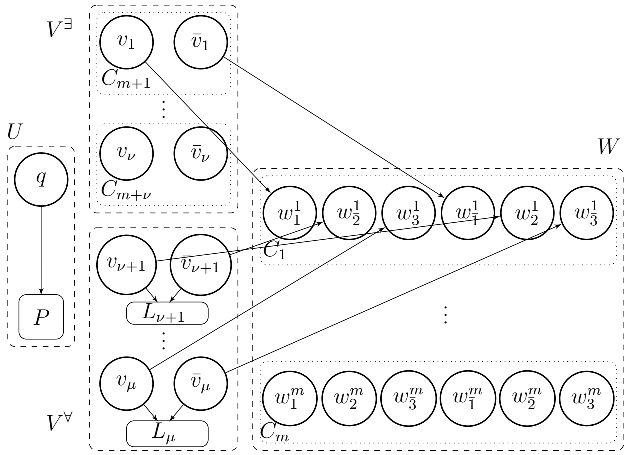

Figure 1: Construction of G from a Σ_2 SAT instance. Only the edges to the nodes corresponding to the first clause ϕ_1 are drawn. All drawn edges have weight 1. The only edges that are not in G are the ones from q to $V^{\exists}$.

- $W = \{w_1^r, w_{\bar{1}}^r, w_2^r, w_{\bar{2}}^r, w_3^r, w_{\bar{3}}^r : r \in [m]\}$.

The edge set E consists of

- $E^{\text{var}} := \{(v_{r(s)}, w_s^r), (\bar{v}_{r(s)}, w_{\bar{s}}^r) : s \in [3], r \in [m]\}$,
- E^L that consists of all edges from the nodes $v_j, \bar{v}_j$ to all nodes $v \in L_j$, for $j \in [\mu] \setminus [\nu]$,
- E^P that consists of edges from q to all nodes in P, and
- $Z := V^2 \setminus (E^{\text{var}} \cup E^L \cup E^P \cup E(q, V^{\exists}))$, where $E(q, V^{\exists}) := \{(q, v) : v \in V^{\exists}\}$.

We note that as a result $\bar{E} = E(q, V^{\exists})$. The edge weight function is defined as $w_e = 0$ for all edges $e \in Z$ and $w_e = 1$ otherwise. The community structure $\mathcal{C}$ consists of: (1) communities $C_1, \dots, C_m$, where each C_r is of cardinality 3 and for $s \in [3]$, $w_s^r \in C_r$ if $x_{r(s)} \in \phi_r$ (or $y_{r(s)} \in \phi_r$) and $w_{\bar{s}}^r \in C_r$ if $\bar{x}_{r(s)} \in \phi_r$ (or $\bar{y}_{r(s)} \in \phi_r$); and (2) communities $C_{m+1}, \dots, C_{m+\nu}$, with $C_{m+i} = \{v_i, \bar{v}_i\}$ for each $i \in [\nu]$. We set $k = \mu + 1$, $b = \nu$ and $t = 1/3$.

Our goal is now to show that the Σ_2 SAT instance is a yes-instance if and only if the constructed FIM_{AL} instance is. We first need the following lemma.

Lemma 0.3. *Let $F \subseteq \bar{E} = E(q, V^{\exists})$ with $|F| \le \nu$. It holds that $S \in \mathcal{M}(F, \mu+1)$ if and only if $q \in S$ and $S \cap \{v_j, \bar{v}_j\} \neq \emptyset$ for all $j \in [\mu] \setminus [\nu]$.*

We are now ready to prove the theorem.

Theorem 0.4. *The decision version of FIM_{AL} is Σ_2^p-hard even in the deterministic case.*

Proof. We show that the Σ_2 SAT instance is a yes-instance if and only if the constructed FIM_{AL} instance is.

($\Rightarrow$) Assume that the Σ_2 SAT instance is a yes-instance, i.e, there exists an assignments x to the variables X such that for all assignment y to the variables Y, it holds that $\phi(x, y) = \top$. We will now show that there exists $F \subseteq \bar{E}$ with $|F| \le \nu$ such that for all $S \in \mathcal{M}(F, \mu + 1)$, it holds that $\min_{C \in \mathcal{C}} \sigma_C(S, F) \ge 1/3$. Let $F \subseteq \bar{E} = E(q, V^{\exists})$ be equal to the set of edges from q to $V^{\exists}$ that correspond to the assignment x. Now, let $S \in \mathcal{M}(F, \mu + 1)$ be arbitrary. It then follows using Lemma 0.3 that $S = \{q\} \cup \dot{S}$, where $\dot{S}$ corresponds to an assignment y of Y. As $\phi(x, y) = \top$ it follows that, for every clause ϕ_r at least one literal is true, thus for every community C_r with $r \in [m]$, at least one node $w \in C_r$ is reached and hence $\sigma_{C_r}(S, F) \ge 1/3$. For communities C_i with $i \in [m+1, m+\nu]$, we obtain that $\sigma_{C_i}(S, F) \ge 1/2$, as F corresponds to an assignment and S contains q according to Lemma 0.3.

($\Leftarrow$) Now, assume that the FIM_{AL} instance admits a solution $F \subseteq \bar{E}$ with $|F| \le \nu$ such that for all $S \in \mathcal{M}(F, \mu+1)$, it holds that $\min_{C \in \mathcal{C}} \sigma_C(S, F) > 0$. Notice that $\sigma_C(S, F) > 0$ for every $S \in \mathcal{M}(F, \mu + 1)$ together with Lemma 0.3 implies that F consists of a set of edges to $V^{\exists}$ that corresponds to an assignment. Let now y be an arbitrary assignment to Y and let S be the set containing q and all nodes from $V^{\forall}$ that correspond to the assignment y. Again using Lemma 0.3 it follows that $S \in \mathcal{M}(F, \mu + 1)$ and thus $\sigma_{C_r}(S, F) > 0$ for all $r \in [m]$. This means that at least one node in every community C_i is reached or equivalently at least one literal in every clause ϕ_r is true in the assignments x and y. It follows that $\phi(x, y) = \top$. ☐

From the same reduction, we can even conclude that it is unlikely to find an arbitrary approximation to FIM_{AL} as shown in the next theorem. The class Δ_2^p is the class of all languages decided by polynomial-time Turing Machines that have access to an oracle for some NP-complete problem. It is widely believed that Σ_2^p and Δ_2^p are distinct (see Section 17.2 in (Papadimitriou 1994)).

Theorem 0.5. *Let $\alpha \in (0, 1]$. If computing an α-approximation to FIM_{AL} is in Δ_2^p, then $\Sigma_2^p = \Delta_2^p$.*

Still Hard Special Cases. While we have shown above that the general problem is Σ_2^p-hard, we will now show that not even in the apparently simple case where $k = 1$, we can hope to find any approximation unless $\text{P} = \text{NP}$.

Theorem 0.6. *For any, $\alpha \in (0, 1]$, it is NP-hard to approximate FIM_{AL} to within a factor of α, even in the deterministic case and if $k = 1$.*

A natural next question is whether the problem remains hard also if $b = 1$. We show that this is the case:

Theorem 0.7. *The decision version of FIM_{AL} is NP-hard even in the deterministic case and if $b = 1$.*

The $\text{FIM}^{\text{g}}_{\text{AL}}$ Problem: Towards Fairness in Practice

Problem Definition. We have seen a lot of evidence above that FIM_{AL} is intractable. We thus continue by proposing an alternative problem that not only turns out to be more computationally tractable, but also is possibly practically better motivated in the first place in the following sense: The problem of finding a set of at most k nodes that maximizes $\sigma(\cdot, F)$ is however an NP-hard optimization problem and thus it is unrealistic to assume the entity to spread information using a maximizing set. Instead what is frequently used in practice for the computation of an efficient seed set is the greedy algorithm. In fact, the choice of the greedy algorithm is also well-founded in theory, as, for a fixed set of non-edges F, the set function $\sigma(\cdot, F)$ is monotone and submodular and thus one is guaranteed to achieve an essentially optimal approximation factor of $1 - 1/e - \varepsilon$ for any $\varepsilon > 0$, see the work of Kempe, Kleinberg, and Tardos (2015). Hence, an optimization problem that is practically better motivated than $\mathrm{FIM}_{\mathrm{AL}}$, assumes that the efficiency-oriented entity, in order to spread information, uses the greedy algorithm for computing the seed set. The greedy algorithm for $\sigma(\cdot, F)$ is however a randomized algorithm, as it relies on simulating information spread using a polynomial number of live-edge graphs (or reverse reachable (RR) sets, depending on the implementation). It becomes thus necessary that we consider the output of the algorithm to be a distribution over seed sets of size k, rather than just a single set. For a set of non-edges $F \subseteq \bar{E}$ and an integer k, let us denote this distribution with $p(F, k)$. We then define the $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ problem as:

$$\max_{F \subseteq \bar{E}: |F| \leq b} \left\{\tau : \mathbb{E}_{S \sim p(F,k)}[\sigma_C(S, F)] \geq \tau \; \forall\, C \in \mathcal{C}\right\}.$$

Intuitively, our goal in the optimization problem $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ is to find a set of at most b non-edges $F \subseteq \bar{E}$, that, when added to G, maximizes the minimum community coverage (in expectation) when information is spread using the greedy algorithm – a quite realistic assumption. We assume the greedy algorithm to break ties arbitrarily, but consistently.

Here, we do not assume to have access to $p(F, k)$, not even for one set F, as it would generally require exponential space to be encoded. Instead, we assume to have access to the greedy algorithm in an oracle fashion, i.e., for a given set F, we can call the greedy algorithm on $\sigma(\cdot, F)$ with budget k and get a set S. One can then show using an easy Hoeffding bound argument, see below, that $\mathbb{E}_{S \sim p(F,k)}[\sigma_C(S, F)]$ can be approximated arbitrarily well w.h.p. for every F.

It is also worth mentioning that our approach can be extended to a setting where we want to be fair w.r.t. multiple implementations of the greedy algorithm or even more generally to multiple implementations of multiple algorithms (different from the greedy algorithm). This can be achieved as follows. Assume that $(p_i)_{i \in [N]}$ are a priori-likelihoods of using one of N different algorithms and assume $p^i(F, k)$ to reflect the probability distribution of seed sets corresponding to algorithm i. Then the distribution with $p_S(F, k) := \sum_i p_i \cdot p^i_S(F, k)$ for $S \subseteq V$ reflects the distribution over seed sets resulting from using all N algorithms. The only condition here, for our algorithmic results below to keep working, is that the algorithms are polynomial time.

Polynomiality of Deterministic Case with Constant b. We now first observe that in the deterministic case with constant b, it is simple to solve the problem exactly in polynomial time, simply by going through all at most $\binom{n^2-m}{b} \leq n^{2b}$ possible sets of non-edges F, computing the deterministic set S_F that the greedy algorithm outputs for maximizing $\sigma(\cdot, F)$, and checking what is the value $\tau_F = \min_{C \in \mathcal{C}} \sigma_C(S_F, F)$. Then return the set F that achieves the maximum τ_F. Although this seems trivial, we notice that such an approach cannot work for $\mathrm{FIM}_{\mathrm{AL}}$, for which we showed that the problem remains NP-hard in the deterministic case even if $b = 1$, see Theorem 0.7.

Observation 0.8. *There is a polynomial time algorithm to compute an optimal solution to $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ in the deterministic case when b is constant.*

Hardness. In the language of parameterized complexity, Observation 0.8 shows that the deterministic $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ problem belongs to the class XP when parameterized by b. A natural question is therefore whether there exists an FPT algorithm that solves or approximates $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ in deterministic instances. In fact, already Theorem 0.6 answers negatively to this question as the proof shows a polynomial-time reduction from the SET COVER problem to the deterministic case of $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ in which b is equal to the size of a set cover κ. As SET COVER is W[2]-hard w.r.t. κ, $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ does not admit an FPT algorithm w.r.t. b, even in the deterministic case, unless $\mathrm{W}[2] = \mathrm{FPT}$. Moreover, under the same condition, no parameterized α-approximation algorithm exists since the optimum of a $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ instance is strictly positive if and only if there exists a set cover of size κ.

A natural next question is what happens for general b, but with $k = 1$. The problem remains hard in this case. Consider the instance constructed in the reduction in Theorem 0.6. As $k = 1$ and as the instance is deterministic, it is clear that the greedy algorithm, for any set $F \subseteq \bar{E}$ of non-edges, simply computes a maximizing set of cardinality 1. Hence the following statement can be shown in the same way as in the proof of Theorem 0.6: there exists a set cover $\mathcal{S}$ of size at most κ if and only if there exists a set of non-edges $F \subseteq \bar{E}$ with $|F| \leq b$, such that $\min_{C \in \mathcal{C}} \mathbb{E}_{S \sim p(F,k)}\, \sigma_C(S, F)] \geq 1$. This yields the following corollary to Theorem 0.6.

Corollary 0.9. *For any $\alpha \in (0, 1]$, it is* NP*-hard to approximate the $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ problem to within a factor of α, even in the deterministic case and if $k = 1$.*

As mentioned above, we will see below that $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ for general constant b turns out to be arbitrarily well approximable. To prove this, we first turn back to the question of approximating $\mathbb{E}_{S \sim p(F,k)}[\sigma_C(S, F)]$ for a fixed F.

Approximating $p(F, k)$. As mentioned above, we do not assume access to $p(F, k)$, instead we show that, using the greedy algorithm in an oracle fashion, we can approximate $\mathbb{E}_{S \sim p(F,k)}[\sigma_C(S, F)]$ arbitrarily well using a Hoeffding bound. We first recall that already σ_C cannot be evaluated exactly but has to be approximated using $\mathrm{poly}(n, \varepsilon^{-1})$ many samples of live-edge graphs.

Lemma 0.10. *Given an instance $(G, \mathcal{C}, b, k)$ of $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ with constant b, one can in $\mathrm{poly}(n, m, \varepsilon^{-1})$ time, compute functions f_C such that, w.h.p., $|f_C(F) - \mathbb{E}_{S \sim p(F,k)}[\sigma_C(S, F)]| \leq \varepsilon$ for all $C \in \mathcal{C}$ and $F \subseteq \bar{E}$ with $|F| \leq b$. Here $m = |C|$.*

General Approximation for Constant b. The above lemma enables us to provide a polynomial time algorithm for $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ when b is constant that finds a set $F \subseteq \bar{E}$ that is ε-close to optimal (in an additive sense) w.h.p. After proving the above lemma, the idea is simple and similar to the deterministic case: Again, go through all at most n^{2b}

possible sets of non-edges F, compute $\varepsilon/2$-approximations $(f_C(F))_{C\in\mathcal{C}}$ as in Lemma 0.10, and return the set with maximum value $\tau_F = \min_{C\in\mathcal{C}} f_C(F)$. This set is an additive ε-approximation of the maximizing set F^* (using the approximation guarantee once for F and once for F^*).

Lemma 0.11. *Let $\varepsilon \in (0,1)$, there is a polynomial time algorithm to compute an additive ε-approximation to the optimal solution of $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ when b is constant.*

Practical Algorithms. For the case with general budget b, recall that the problem is in-approximable unless $\mathrm{P} = \mathrm{NP}$ according to Corollary 0.9. We still propose several algorithms in this paragraph that perform well in practice as we will show later on. All our algorithms are of a greedy flavour and based on restricting to the evaluation of increments of non-edges that seem promising to improve fairness. In the following, we give the informal description of the methods.

grdy_al. The algorithm that, starting with $F = \emptyset$, in b iterations, chooses the non-edge e into F that maximizes the increment $\min_C \mathbb{E}_{S\sim p(F,k)}[\sigma(S, F \cup \{e\})] - \min_C \mathbb{E}_{S\sim p(F,k)}[\sigma(S, F)]$. For efficiency we restrict to evaluate only non-edges that are (1) incident to S_p, the union over all sets with positive support in $p(F,k)$, and (2) are inter-community edges. Note that at the beginning of each iteration, we recompute $p(F,k)$ as F changes.

to_minC_infl. The algorithm that, starting from the empty set $F = \emptyset$, adds the non-edge $e = (u,v) \in \bar{E} \setminus F$ to F that connects a node from S_p with a node that maximizes $f(e) := \Pr_{S\sim p(F,k)}[u \in S] \cdot w_e \cdot \mathbb{E}_{S\sim p(F,k)}[\sigma_{\bar{C}}(S \cup \{v\}, F)]$, where $\bar{C}$ is the community of minimum coverage. The rationale being to choose the non-edge that connects a seed node with a node that has large influence in the community $\bar{C}$ taking into account both the probability that u is a seed and the edge weight w_e.

to_minC_min. The algorithm that, starting from the empty set, adds a non-edge to the node $\bar{v}$ with minimum probability of being reached in the community that currently suffers the smallest community coverage. Among all these non-edges we choose the non-edge $(u, \bar{v})$ that maximizes the product $\Pr_{S\sim q}[u \in S] \cdot w_{(u,\bar{v})}$.

We also use two techniques in order to speed up our implementations: (1) a pruning technique for **grdy_al** and (2) a way to update RR sets rather than recompute them from scratch after adding edges.

Experiments

In this section, we report on two experiments involving the $\mathrm{FIM}^{\mathrm{g}}_{\mathrm{AL}}$ problem. In the first experiment, we compare the algorithms presented above in terms of quality and running time. In a second experiment, we evaluate the best performing algorithm against other fairness-tailored seeding algorithms. We show, for several settings, that already adding just a few edges can lead to a situation where purely efficiency-oriented information spreading becomes automatically fair.[2] We proceed by describing the experimental setup. We proceed by describing the experimental setup.

[2]https://github.com/sajjad-ghobadi/fair_adding_links.git

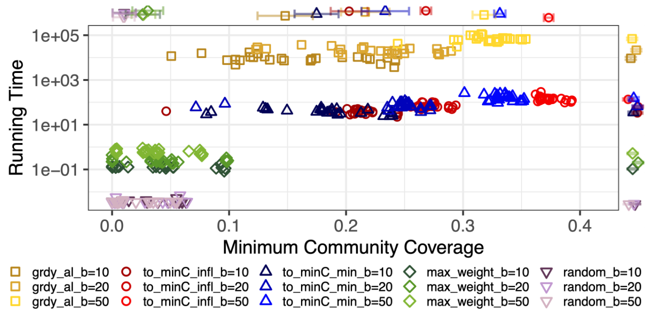

Figure 2: Results Experiment 1: Synthetic instances ($k = 25$, $n = 500$, communities induced by gender and region).

Experimental Setting. In our experiments we use random, synthetic and real world instances. (1) Random instances are generated using the Barabasi-Albert model connecting newly added nodes to two existing nodes. (2) The synthetic instances are the ones used by Tsang et al. (2019). (3) We use similar real world instances as Fish et al. (2019). For random and synthetic instances we select edge weights uniformly at random in the interval $[0, 0.4]$, and in the interval $[0, 0.2]$ for the real world instances (other than `youtube`). For `youtube`, we choose the edge weights from the interval $[0, 0.1]$. We choose the non-edge weights uniformly at random from the interval $[0, 1]$. We consider different community structures: (1) Singleton communities: each node has its own community. (2) BFS communities: for every $i \in [m]$, we generate a community C_i of size n/m using a breadth first search from a random source node (we continue this process if the size of a community is less than n/m). (3) Community structures given for the synthetic networks and some of the real world networks.

We repeat each algorithm 5 times per graph. For random and synthetic instances, we average in addition over 5 graphs, resulting in 25 runs per algorithm. The error-bars in our plots represent 95% confidence intervals. We use the TIM implementation for IM by Tang, Xiao, and Shi (2014) in order to implement the greedy algorithm for IM. The algorithms **grdy_al**, **to_minC_infl**, and **to_minC_min** are implemented in `C++` and were compiled with g++ 7.5.0. All experiments were executed on a compute server running Ubuntu 16.04.5 LTS with 24 Intel(R) Xeon(R) CPU E5-2643 3.40GHz cores and a total of 128 GB RAM.

Experiment 1. In addition to the three algorithms described in the previous section, we evaluate the following two base lines: **random**: the algorithm that chooses b non-edges uniformly at random, and **max_weight**: the algorithm that chooses the b non-edges of maximal weight. The results can be found in Fig. 2 for the synthetic instances. We observe that, despite the pruning approach, **grdy_al**'s running time is the worst. Furthermore, the fairness that it achieves is worse than the one of **to_minC_infl**. We thus exclude **grdy_al** from further experiments. In Fig. 3, we can see the results for the real world instance `ca-GrQc`. We observe that the running times of both algorithms **to_minC_infl** and **to_minC_min** are comparable, while

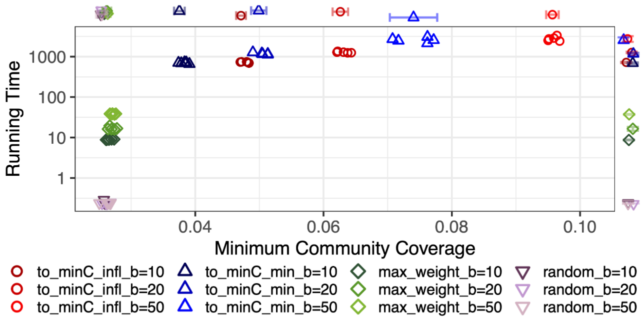

Figure 3: Results Experiment 1: `ca-GrQc` with BFS communities ($m = 10$), $k = 20$.

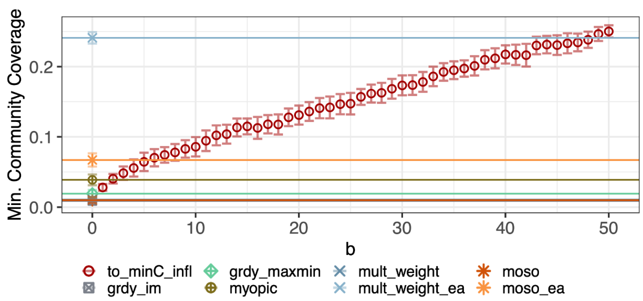

Figure 4: Results Experiment 2: Random instances ($k = 25$, $n = 200$, singleton communities).

to_minC_infl achieves better values of fairness. We thus choose **to_minC_infl** as the best performing algorithm as a result of this experiment.

Experiment 2. The goal of the second experiment is to analyze how many links we need to add in order to make the standard greedy algorithm for IM satisfy similar or better fairness guarantees than fairness-tailored algorithms. To this end, we compare our method **to_minC_infl** with the following competitors: **grdy_im**, the standard greedy algorithm for IM (we use the implementation of Tang, Xiao, and Shi (2014)) serves mainly as a baseline; **grdy_maxmin**, the greedy algorithm that iteratively selects k seed nodes to maximize the minimum community coverage; **myopic**, a simple heuristic proposed by Fish et al. (2019) that iteratively chooses the node with minimum probability of being reached as seed; **mult_weight**, the multiplicative weights routine for the set-based problem of Becker et al. (2022); **moso**, a multi-objective submodular optimization approach proposed by Tsang et al. (2019) (we choose gurobi as solver (Gurobi Optimization, LLC 2022)).

We note that the algorithms **mult_weight** and **moso** are designed to compute distributions over seed sets and nodes, respectively, and thus they can be used to obtain both ex-ante and ex-post fairness guarantees. We defer the reader to the work of Becker et al. (2022) for details regarding probabilistic seeding and ex-ante guarantees. Hence, for these two algorithms we include both there ex-post and ex-ante values in our evaluations. It is worth pointing out that is much easier (especially in settings with many communities)

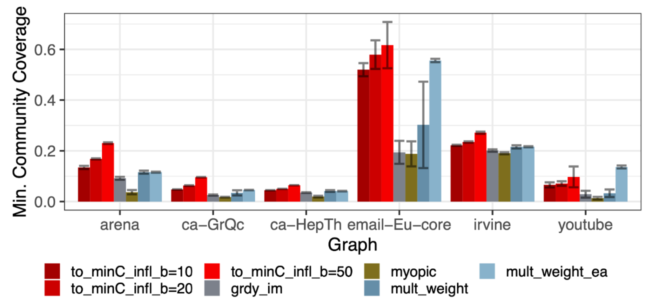

Figure 5: Results Experiment 2: Real world graphs with BFS communities ($m = 10$) for `arena`, `ca-GrQc`, `ca-HepTh`, `irvine` and real communities for `email-Eu-core` and `youtube`, $k = 20$.

to achieve good values ex-ante rather than ex-post.

We show the results for the random instances in Fig. 4. Already for small values of b, i.e., after adding just a few edges, our algorithm surpass all ex-post fairness values of the competitors. Even better and maybe surprisingly, our algorithm also achieves ex-post values higher than the ex-ante values of **mult_weight** and **moso**. We exclude the algorithms **grdy_maxmin** and **moso** from experiments with the real world instance as they perform the worst in terms of running time. We turn to the real world instances, see Fig. 5, on which we evaluate our algorithm for three fixed values of $b = 10, 20, 50$. We observe that after adding only 50 edges, we obtain the best (or comparable to the best) fairness values on all instances.

Conclusion

We studied two optimization problems with the goal of adding links to a social network such as to make purely efficiency-oriented information spreading automatically fair. In the first problem FIM_{AL}, our goal is to add at most b non-edges F to the graph such that the minimum community coverage $\sigma_C(S, F)$ is maximized w.r.t. maximizing sets S of size at most k to spread information. We showed several hardness and hardness of approximation results for FIM_{AL}. Maybe most importantly, the decision version of FIM_{AL} is Σ_2^p-hard even in the deterministic case and remains NP-hard even if $b = 1$ or $k = 1$ (in the latter case even to approximate within any factor). We thus proposed to study a second optimization problem $\text{FIM}^{\text{g}}_{\text{AL}}$ that entails to add at most b non-edges F to the graph such that the minimum expected community coverage is maximized when information is spread using the greedy algorithm for influence maximization. As we observed, also this problem remains NP-hard to approximate to within any factor if $k = 1$. On the other hand, in contrast to FIM_{AL}, $\text{FIM}^{\text{g}}_{\text{AL}}$ becomes polynomial time $-\varepsilon$-approximable if b is a constant. We then proposed several heuristics for $\text{FIM}^{\text{g}}_{\text{AL}}$ and evaluated them in an experimental study. Lastly, we conducted an experiment showing that the greedy algorithm for IM achieves similar or even better levels of fairness than fairness-tailored algorithms already after adding a few edges proposed by our algorithm.

Acknowledgments

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

References

Amelkin, V.; and Singh, A. K. 2019. Fighting Opinion Control in Social Networks via Link Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 677–685. ACM.

Arora, S.; and Barak, B. 2009. *Computational Complexity - A Modern Approach*. Cambridge University Press. ISBN 978-0-521-42426-4.

Becker, R.; D'Angelo, G.; and Ghobadi, S. 2023. Improving Fairness in Information Exposure by Adding Links. *CoRR*, abs/2302.13112.

Becker, R.; D'Angelo, G.; Ghobadi, S.; and Gilbert, H. 2022. Fairness in Influence Maximization through Randomization. *J. Artif. Intell. Res.*, 73: 1251–1283.

Castiglioni, M.; Ferraioli, D.; and Gatti, N. 2020. Election Control in Social Networks via Edge Addition or Removal. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 1878–1885. AAAI Press.

Coro, F.; D'Angelo, G.; and Velaj, Y. 2021. Link Recommendation for Social Influence Maximization. *ACM Trans. Knowl. Discov. Data*, 15(6): 94:1–94:23.

D'Angelo, G.; Severini, L.; and Velaj, Y. 2019. Recommending links through influence maximization. *Theor. Comput. Sci.*, 764: 30–41.

Fish, B.; Bashardoust, A.; danah boyd; Friedler, S. A.; Scheidegger, C.; and Venkatasubramanian, S. 2019. Gaps in Information Access in Social Networks? In *WWW2019*, 480–490. ACM.

Garimella, K.; Morales, G. D. F.; Gionis, A.; and Mathioudakis, M. 2017. Reducing Controversy by Connecting Opposing Views. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, 81–90. ACM.

Gurobi Optimization, LLC. 2022. Gurobi Optimizer Reference Manual. https://www.gurobi.com. Accessed: 2022-04-17.

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

Khalil, E. B.; Dilkina, B.; and Song, L. 2014. Scalable diffusion-aware optimization of network topology. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 1226–1235. ACM.

Papadimitriou, C. H. 1994. *Computational complexity*. Addison-Wesley.

Stoica, A.; and Chaintreau, A. 2019. Fairness in Social Influence Maximization. In *FATES2019 – WWW2019 Companion*, 569–574. ACM.

Swift, I. P.; Ebrahimi, S.; Nova, A.; and Asudeh, A. 2022. Maximizing Fair Content Spread via Edge Suggestion in Social Networks. *Proc. VLDB Endow.*, 15(11): 2692–2705.

Tang, Y.; Shi, Y.; and Xiao, X. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *SIGMOD2015*, 1539–1554.

Tang, Y.; Xiao, X.; and Shi, Y. 2014. Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD2014*, 75–86.

Tong, H.; Prakash, B. A.; Eliassi-Rad, T.; Faloutsos, M.; and Faloutsos, C. 2012. Gelling, and melting, large graphs by edge manipulation. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, 245–254. ACM.

Tsang, A.; Wilder, B.; Rice, E.; Tambe, M.; and Zick, Y. 2019. Group-Fairness in Influence Maximization. In *IJCAI2019*, 5997–6005.

Wu, X.; Sheldon, D.; and Zilberstein, S. 2015. Efficient algorithms to optimize diffusion processes under the independent cascade model. *NIPS Work. on Networks in the Social and Information Sciences*, 1(1).