Maximizing Fair Content Spread via Edge Suggestion in Social Networks

Ian P. Swift
University of Illinois
Chicago
iswift2@uic.edu

Sana Ebrahmi
University of Illinois
Chicago
sebrah7@uic.edu

Azade Nova
Google Brain
azade@google.com

Abolfazl Asudeh
University of Illinois
Chicago
asudeh@uic.edu

ABSTRACT

Content spread inequity is a potential unfairness issue in online social networks, disparately impacting minority groups. In this paper, we view friendship suggestion, a common feature in social network platforms, as an opportunity to achieve an equitable spread of content. In particular, we propose to suggest a subset of potential edges (currently not existing in the network but likely to be accepted) that maximizes content spread while achieving fairness. Instead of re-engineering the existing systems, our proposal builds a fairness wrapper on top of the existing friendship suggestion components.

We prove the problem is NP-hard and inapproximable in polynomial time unless P = NP. Therefore, allowing relaxation of the fairness constraint, we propose an algorithm based on LP-relaxation and randomized rounding with fixed approximation ratios on fairness and content spread. We provide multiple optimizations, further improving the performance of our algorithm in practice. Besides, we propose a scalable algorithm that dynamically adds subsets of nodes, chosen via iterative sampling, and solves smaller problems corresponding to these nodes. Besides theoretical analysis, we conduct comprehensive experiments on real and synthetic data sets. Across different settings, our algorithms found solutions with near-zero unfairness while significantly increasing the content spread. Our scalable algorithm could process a graph with half a million nodes on a single machine, reducing the unfairness to around 0.0004 while lifting content spread by 43%.

KEYWORDS

Algorithmic fairness; Equitable Content Spread

1 INTRODUCTION

Online Social networks have become an inseparable aspect of modern human life and a prominent medium for human interactions

at scale. These networks play an invaluable role in removing the physical communication barriers and enabling the spread of information across the world in real-time. However, like other recent technologies, the benefits of the social networks have not been for free, as they introduce new social challenges and complications that did not exist before. From privacy issues to misinformation spread and even to social media addiction, we keep hearing about the concerns and challenges specific to the nature of social networks.

Machine bias has recently been a focal concern in the social aspects of computer science and (big) data technologies [4, 9], triggering extensive effort, including data pre-processing techniques [15, 30, 50, 75], algorithm or model modification [52, 96, 97], and output post-processing techniques [46, 51, 90] to address these concerns. Social networks are not an exception when it comes to unfairness issues [47, 48, 68].

Inequity in information spread across social networks is one of the potential unfairness challenges, disparately impacting minority groups [13, 69, 70, 86, 94?]. While similar issues have been reported for cases such as biased advertisement over social networks [81], fairness in content spread becomes critical when related to health-care [93], news and valuable information spread [82], job opportunities [92], etc.

Despite its importance, fairness in content spread over social networks has been relatively less studied. Existing works take two directions to address the problem: (i) fairness-aware influence maximization (IM) [70, 86, 94?], which aims to select the content spread seed nodes in a way that a combination of fairness and influence spread is maximized, (ii) [56, 61] which consider fairness in terms of topological properties of the network e.g., promoting inter-group connections in a network in order to alleviate unfairness caused by strong alignment within groups.

This paper takes a different direction to achieve an equitable spread of content in social networks. In particular, we consider friendship suggestion, a popular feature across social network platforms, where a subset of potential edges that currently do not exist in the network is suggested to the users. We view this feature as an opportunity to achieve an equitable spread of content in social networks. That is, instead of selecting a subset of potential edges that only maximize content spread [19, 91, 95], we aim to consider fairness when suggesting the edges. Our proposal is different from the existing work on two angles. First, unlike the works on IM that focus on selecting the fair influential seed nodes with a fixed graph, our focus is on *fair edge selection* with fixed content source nodes. Second, unlike works that aim to directly make the *topology* of the network diverse and independent from the demographic information, we consider a more liberal approach by suggesting edges to achieve a new objective: fairness in content spread.

Existing social networks often use complex and sometimes black-box algorithms for friendship suggestions. Re-engineering the existing algorithms are costly and perhaps impractical. Instead, our proposal adds *a fair content spread component* that interfaces with any possible friendship suggestion method, taking the output of those algorithms as the set of potential edges and selecting a subset for the suggestion to achieve fairness.

Technically speaking, we introduce Fair Content Spread Maximization (FairCS) problem, where given a set of candidate edges, our goal is to select a subset such that: it contains a fixed number of incident edges to each node (k friend suggestions to each node), satisfies fairness (defined on the probability that different demographic groups receive content [11, 12, 61]), and maximizes content spread. To the best of our knowledge, we are the first work to consider fairness in this context. Unfortunately, it turns out, not only is the problem NP-hard, but also not possible to approximate in P time unless P = NP. Therefore, allowing approximation on content spread and relaxation of the fairness constraint, we propose a non-trivial randomized approximation algorithm for the FairCS problem based on LP-relaxation and randomized rounding [63]. Our algorithm provides constant approximation ratios on the content spread and the fairness of its output. We are the first Linear Program designed for content spread with approximation guarantees on fairness. Next, we propose several optimizations that further help our algorithm be efficient in practice. Requiring to solve an LP, our original algorithm lacks to scale to very large settings. To resolve this issue, we make an observation that helps us design a scalable algorithm based on dynamically increasing the nodes via sampling. Intuitively, instead of solving one expensive-to-solve LP that covers the entire problem space, the algorithm iteratively solves problems over subsets of nodes with reasonably small sizes.

In addition to the theoretical analysis, we conduct comprehensive experiments on benchmark data sets. Our experiments confirm that our algorithms not only could find solutions with *near-zero unfairness* but, due to the practical effectiveness of LP-relaxation and randomized techniques, outperformed all baselines from a large breadth of related research [19, 91, 95, 100] in content spread. Additionally, we observed our algorithm achieves comparable results to the Optimal Brute Force method when compared on very small graphs. Moreover, our scalable algorithm could scale to a very large setting with an order of *half a million nodes*, on a single work station in a reasonable time, while none of the baselines we used were able to compute half a million nodes in under 24 hours, confirming the effectiveness of our scalable approach. It generated output with negligible unfairness while significantly increasing content spread. On half a million nodes, we observed a decrease from an initial unfairness of 3.9% to 0.04%, while the content spread increased by 43%. We were able to run our original algorithm on a largest setting of 4000 nodes, achieving a lift of 57.99% on content spread and an *unfairness of less than 0.0001*. Alternatively, our scalable algorithm still achieved a lift of 45.07% and a unfairness of 0.0132, and also had a 489x speedup over our other algorithm. At 4000 nodes, no baseline could produce a higher lift, a lower unfairness, or had a lower runtime than our scalable solution.

Summary of contributions: In summary, our contributions in this paper are as follows:

- (Section 2) We formally introduce the FairCS problem to choose a subset of candidate edges which maximizes content spread while being fair.
- (Section 2.4) We prove the problem is NP-hard, and is inapproximable in polynomial time unless P = NP.
- (Section 3) Approximating on content spread while relaxing the fairness constraint, we provide a randomized approximation algorithm for the problem based on LP-relaxation and randomized rounding, guaranteeing an approximation ratio of $(1 - e^{-1})$ on content spread and $\frac{2e}{1-e^{-1}}$ on fairness.
- (Section 4) We provide multiple optimizations to improve the performance of our algorithm in practice.
- (Section 5) We provide a scalable algorithm that, instead of solving one very-large LP, dynamically adds sampled nodes to the graph allowing the problem to be solved in multiple computationally easier problems.
- (Section 6) Besides theoretical analyses, we conduct comprehensive experiments on real data sets. Our experiments verify the effectiveness of our algorithms and confirm their efficiency, showing that our LP relaxation algorithm outperforms implementations of other algorithms in all but runtime, and that our scalable algorithm produces a fair result of less than 0.05% and does so faster than all implementations even in very large settings.

2 PRELIMINARIES

2.1 Social Networks Model

Network Model: A social network is an abstract model representing relationships between users in form of a graph $G(V, E)$, where V is the set of users, and E is the set of edges, either directed or undirected [27, 76, 88]. Directed relationships represent one way connections such as citations, while undirected relationships represent shared connections such as ongoing correspondence. Our work has targeted both directed and undirected social networks, depending on which application is necessary. We assume the nodes are associated with at least one sensitive attribute, such as race, sex, or age-group used for defining demographic groups, or simply groups[1], as we shall explain in Section 2.3.

Candidate Edges (Friendship): Suggesting a set of *potential relations* that currently do not exist in the social network but a user may accept is an active area of research. Different techniques and algorithms (white or black-box) are used to find such a candidate set. A popular strategy is "*friends of friends*" [95], where only users that are at most within a two-hop distance from a given user are candidates for the friends suggestion. The intuition behind this model is that users who do not share a friend are unlikely to be friends. Besides, more sophisticated methods exist: candidate edges can be determined using an implicit social graph [74], using overlapping ego-nets [28], by augmenting the network and detecting resulting communities [44], or with the addition of a peripheral system by analysis of the lifestyle of potential matches [87]. Our work is agnostic as to what model is used for identifying potential relations, and only expects the existence of an input candidate edge

---

[1]While our motivation in this paper is on fairness over demographic groups, our techniques are not limited to those. In particular, the groups can be defined over any attributes of interest such as occupation or political-affiliation.

set, which can be determined by any of these methods. We refer to the set of candidate edges as E_c where, $\forall e \in E_c, e \notin E$.

Content Nodes: Social networks enable users to share content at an unprecedented scale. Information and marketing [85], user content such as images [18], opinions [17], and even happiness [33] are examples of the content being shared across users in a cascading manner. While often there is a clear source from which the content originated [18, 41, 57], some existing work detects content orignality [2, 72], or targeted nodes for optimizations such as influence maximization [54, 85]. In this paper, we assume that, by any of the prior work, content source nodes are selected as a subset of $V_s \subset V$.

2.2 Cascade Model

In social networks, content spread or information diffusion follows a cascade model. Each neighbor recursively shares received updates within its own neighborhood, and content generated by a user propagates through the network to a wide user population. Specifically, in *independent cascade* (IC) [54] model, the content nodes are active in a timestep t_0. Then at timestep t_{i+1}, each node v_x that was activated in timestep t_i has a probability $P_{x,y}$ of activating each unactivated node v_y for which there is an edge $e_{x,y} \in E$. Computing the expected number of nodes with specific content using IC is known to be #P-hard, and accurately estimating the content spread requires a combinatorially large time [19, 20]. Thus, alternative cascade models such as *Maximum Influence Path* [20], and *Restricted Maximum Probability Path* [19] have been proposed that (i) require polynomial-time computation and (ii) provide a good approximation of IC.

Maximum Influence Path (MIP) [20] takes into consideration the single path from content nodes to each other node for which the path has the maximum propagation probability of reaching the destination node. Given a content node $v_j \in V_s$ and a node $v_i \in V \backslash V_s$, consider a path q from v_j to v_i, such that $\forall e \in q, e \in E \cup E_p$, where $E_p \subseteq E_c$. Given the probability P_e on every edge e, the probability of content spread from a source to a destination along a path q is $Pr(q) = \prod_{e \in q} P_e$. Let Q_i be the set of paths from a content node to the node $v_i \in V \backslash V_s$. Following MIP, the probability of content reaching v_i is equal to the probability that the node was reached along the path in Q_i with the highest probability. That is, $\forall v_i \in V \backslash V_s$:

$$F_{MIP}(v_i, E_p) = \max_{q \in Q_i} Pr(q) \tag{1}$$

Restricted Maximum Probability Path (RMPP) [19] extends MIP by using the included candidate set, such that it considers only paths that include at most one edge from that set. Under RMPP, Q_i is the set of paths from a content node to the node $v_i \in V \backslash V_s$, such that $\forall q \in Q_i$ there exists at most one edge $e \in E_p$. $F_{RMPP}(v_i, E_p)$ is the expected content received by a node v_i considering E_p as the suggested edges[2]. Following prior work [19, 20], we use RMPP to design our randomized algorithm in Section 3.

Edge Suggestion to maximize Content Spread: Considering content nodes and a cascade model for information diffusion, a more sophisticated approach to friend suggestion is maximizing content spread in social networks [19]. Even though candidate set E_c are those friendships that are likely to be accepted by a user [31, 74, 95],

---

[2]Unless mentioned otherwise, we simplify the notation $F_{RMPP}(.)$ to $F(.)$.

Symbol	Description
$G(V, E)$	The social network with nodes V and edges E
E_c	Set of candidate edges $\notin E$
V_s	Content source nodes
E_p	Suggested candidate edges $\subseteq E_c$
k	Number of edges per node to be suggested
$\mathcal{G}$	The set of groups
g_i	The i-th group in $\mathcal{G}$
$\mathcal{N}(v_i)$	Candidate edges $e_j \in E_c$ incident to the node $v_i \in V$
p	Edge activation probability
ε	Content Spread Disparity (Unfairness) – Equation 6

Table 1: Table of Notations

there may be an overwhelming number of candidates for each user. Thus, a subset of k edges is to be selected from the candidate set. We call the chosen set of edges, that is a subset of E_c and has at most k incident edges to each node, the set of *suggested edges* $E_p \subseteq E_c$. In [19], the content spread maximization problem is:

DEFINITION 1 (CONTENT SPREAD MAXIMIZATION (CSM) PROBLEM). *Given a social network $G(V, E)$ and a set of candidate edges E_c, let $\mathcal{E}_p \subseteq 2^{E_c}$ be the collection of subset of edges from E_c such that each set $E_p \in \mathcal{E}_p$ contains at most k incident edges on each node. Find the set $E_p \in \mathcal{E}_p$ that maximizes the content spread.*

$$\max_{E_p \in \mathcal{E}_p} \sum_{v_i \in V} F(v_i, E_p) \tag{2}$$

2.3 Fairness Model

Following the literature on fairness in social networks [11, 12, 61], and other similar settings [1, 3, 5, 45, 71], we found our fairness model based on the notion of **equity**. Consider a social network, where users have sensitive attributes such as gender and race with values such as {female, male} and {black, white} that should be considered for fairness. We identify the demographic groups as the intersection of domain values of the sensitive attributes. For example, {white-male, white-female, black-male, black-female} can be the groups defined as the intersection of race and gender. We use the notation $\mathcal{G} = \{g_1, \ldots, g_c\}$ to denote the set of groups.

Existing work highlights unfairness in social networks as different groups receive content with different probabilities [11]. This creates a mandate for improving social networks to promote equity. We consider the notion of group fairness in social networks to assure that users from different groups receive content in an equitable manner. We define fairness in content spread as Definition 2.

DEFINITION 2 (FAIR CONTENT SPREAD). *The content spread in a social network is fair, if for each pair of groups g_i and g_j in $\mathcal{G}$, the expected total content received by group g_i proportional to its size is equal to the one by g_j proportional to its size. That is, $\forall g_i, g_j \in \mathcal{G}$:*

$$\frac{1}{|g_i|} \sum_{v_\ell \in g_i} F(v_\ell, E_p) = \frac{1}{|g_j|} \sum_{v_\ell \in g_j} F(v_\ell, E_p) \tag{3}$$

Accordingly, we introduce the following problem that maximizes fair content spread.

DEFINITION 3 (FAIR CONTENT SPREAD MAXIMIZATION (FAIRCS) PROBLEM). *Given a social network $G(V, E)$, a set of candidate edges E_c, and the demographic groups $\mathcal{G} = \{g_1, \cdots, g_c\}$, let $\mathcal{E}_p \subseteq 2^{E_c}$ be the collection of subset of edges from E_c such that each set $E_p \in \mathcal{E}_p$ contains at most k incident edges on each node and E_p satisfies fairness in content spread (Definition 2). Find the set $E_p \in \mathcal{E}_p$ that maximizes the content spread.*
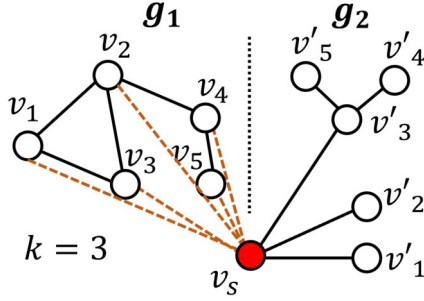
Figure 1: (Proof of Theorem 2) Illustration of a vertex-cover instance (left graph, $k = 3$) to an instance of FAIRCS problem. Candidate edges E_c are specified in dashed brown.

2.4 Problem complexity and inapproximability

In this section, we study the complexity of the FAIRCS problem, its inapproximability, and examine an exact integer programming formulation. Note that existing work on content spread maximization [19, 91] heavily rely on the submodularity of the problem to propose approximation algorithms. However, under fairness constrain we prove that even *finding a valid solution* is NP-complete.

LEMMA 1. *The Fair Content Spread Maximization (*FAIRCS*) Problem is* NP*-hard.*[3]

A major difference between the content spread maximization (CSM) and the FAIRCS problem is that for any given problem instance, CSM always has a valid solution, but FAIRCS problem may not even have a valid solution for a particular problem instance. More importantly, in proof of Theorem 2 we show it is NP-complete to determine if an instance of FAIRCS problem has a valid solution, even when there are only two groups g_1, g_2. It consequently proves the inapproximability result for FAIRCS problem as in Theorem 2. Following this negative result, none of the existing approximation algorithms for the CSM problem (including the greedy approach) works for FAIRCS.

THEOREM 2. *There exists no polynomial approximation algorithm for the* FAIRCS *problem, even when there are only two groups* g_1 *and* g_2, *unless* P = NP.

PROOF. In the following, we prove that an approximation algorithm for the FAIRCS problem provides an exact solution for (the decision version of) the Vertex-Cover problem (VC). As a result, since VC is NP-complete [24], the polynomial approximation algorithm for FAIRCS does not exist, unless P = NP.

Consider an instance of the VC problem, where given the graph $G(V, E)$, a value k, and $|V| = n$, the objective is to determine if there exists a subset of k vertices that are at least one endpoint of every edge in E. For example, in Figure 1, let the (induced sub-)graph with vertices v_1 to v_5 and the edges between them (the left graph) be the input to a VC problem. Given $k = 3$, the answer to the VC is yes, since there are 3 vertices (e.g. $\{v_1, v_2, v_5\}$) that includes at least one endpoint of every edge of the graph.

We transform a given instance of the VC to an instance of FAIRCS problem over the graph $G''(V'', E'')$ as following:

---

³ The proofs of Lemma 1 and Lemma 3 are provided in the appendix.

- Add V to V'' and E to E''. Also, for each vertex $v \in V$ add a vertex v' to V''.
- Let $\mathcal{G} = \{g_1, g_2\}$, where every vertex $v \in V$ belongs to g_1 and every vertex v' belongs to g_2.
- Add a single content node v_s to V_s. Add k edges between $v'_1, \cdots, v'_k$ and v_s (in Figure 1, connected v'_1, v'_2, v'_3 to v_s). Add $(n - k)$ edges between $v'_{k+1}, \cdots, v'_n$ and v'_k (in Figure 1, connected v'_4, v'_5 to v'_3). Add n potential edges between $v_1, \cdots, v_n$ and v_s to E_c.

Now, as shown below, using $k'' = k$ as the max incident edges suggested to each node, an approximation solution for the FAIRCS problem on G'', provides an exact solution for VC on G.

To see why, consider the universe of valid solutions for this instance of the FAIRCS problem. A solution is valid if it selects at most k incident edges to each node and satisfies fairness constraints *but is not necessarily optimal* (i.e., it may not have the maximum content spread). As we shall show in Equations 4 and 5, in order to satisfy the fairness constraints in the constructed FAIRCS problem, any valid solution (optimal or approximate) *must* select k edges between the nodes of a vertex-cover (of size k) and v_s. The expected content received by $v'_1, \cdots, v'_n$ is:

$$\mathbb{E}\left[CS_{g_2}\right] = \sum_{v'_i} F(v'_i, E_p) = \sum_{i=1}^{k} F(v'_i, E_p) + \sum_{i=k+1}^{n} F(v'_i, E_p)$$
$$= k\,p + (n - k)\,p^2 \tag{4}$$

where p is the edge activation probability. To satisfy the fairness constraint, the expected content received by the two groups should be equal. That is, $\mathbb{E}\left[CS_{g_1}\right] = \mathbb{E}\left[CS_{g_2}\right]$. This happens when only the edges from a vertex-cover of size k to v_s are selected. In this situation, the expected content received by $v_1, \cdots, v_n$ is:

$$\mathbb{E}\left[CS_{g_1}\right] = \sum_{v_i} F(v_i, E_p) = \sum_{(v_i, v_s) \in E_p} F(v_i, E_p) + \sum_{(v_i, v_s) \notin E_p} F(v_i, E_p)$$
$$= k\,p + (n - k)\,p^2 \tag{5}$$

So far, we proved that every VC of size k corresponds to a valid solution. Next, we prove by contradiction that every valid solution also corresponds to a VC of size k. Suppose there exists a valid solution that does not correspond to a VC of size k. In this situation, there should exist at least one node in g_1 (one of the end-nodes of an uncovered edge) with a distance of at least 3 to v_s. This reduces the expected content received by g_1 by at least $p^2 - p^3$. To resolve this reduction, the valid solution requires to select more than k edges connected to v_s, which has contradiction with the requirement that at most k incident edges on each node should be selected. The one-to-one mapping between the VC of size k and the valid solutions complete our proof: a polynomial approximation algorithm, finding a valid solution for FAIRCS, solves (the decision version of) VC; hence, such an algorithm does not exist unless P = NP. □

Having provided the complexity and inapproximability results for the FAIRCS problem, we next provide an integer programming formulation for solving the problem exactly. This formulation, though not practical, helps us in designing our solution in § 3.1.

A Warm-up Integer programming: The optimal solution to FAIRCS can be obtained by formulating it as an integer program (IP). The IP formulation of FAIRCS appears as follows.

$$\text{maximize} \quad \sum_{v_i \in V} x_i$$

$$\text{subject to} \quad \forall v_i \in V, \; x_i \leq \max_{e_i \in E_p} (y_j p_{ij})$$

$$\forall v_i \in V, \; \sum_{e_j \in \mathcal{N}(v_i)} y_j \leq k$$

$$\forall g_a, g_b \in \mathcal{G}, \; \frac{1}{|g_a|} \sum_{\ell \in g_a} x_\ell = \frac{1}{|g_b|} \sum_{\ell \in g_b} x_\ell$$

$$\forall v_i \in V, \; 0 \leq x_i \leq 1; \; \forall e_i \in E_c, \; y_j \in \{0, 1\}$$

In the above formulation, $x_i \in [0, 1]$ is the probability that a node v_i receives content, and the goal is to maximize this for all nodes. y_j is a binary variable that is 1 if the edge $e_j \in E_c$ is added to E_p, 0 otherwise. p_{ij} is a pre-computed probability for each edge e_j and node v_i, indicating the likelihood that v_i receives content, if only e_j is added to E_p. The first set of constraints computes x_i following the RMPP cascade model (§ 2.2). The second constraints enforce that selected edges incident to each node v_i should not be more than k ($\mathcal{N}(v_i)$ is the candidate edges $e_j \in E_c$ incident to the node v_i). Finally, the weighted sum of nodes activation is the overall content spread, and the weighted sum of node activation values between groups proportional to group size enforce fairness.

Besides the fact the IP is NP-complete, the *max* function in the right-hand side of the first constraints adds further complications to this solution. The max function in the IP constraints is not immediately solvable by any of the IP solvers and needs to be converted into a compatible mathematical format, which turns out to make the solution double exponential [16].

3 THE RANDOMIZED APPROXIMATION ALGORITHM

Theorem 2 proves the inapproximability of the FAIRCS problem. Therefore, to find an efficient solution for the problem, we allow approximation on content spread and relaxation of the fairness constraint. We define a $(1 + \varepsilon)$-approximation ($\varepsilon \geq 0$) on fairness, as following:

$$\forall g_i, g_j \in \mathcal{G} \; : \; (1 + \varepsilon)^{-1} \leq \frac{\frac{1}{|g_i|} \sum_{v_\ell \in g_i} F(v_\ell, E_p)}{\frac{1}{|g_j|} \sum_{v_\ell \in g_j} F(v_\ell, E_p)} \leq (1 + \varepsilon) \quad (6)$$

Note that $\varepsilon = 0$ refers to the maximum fairness, and it decreases as ε increases. Conversely, the disparity of content spread is equivalent to ε. Therefore, in the rest of the paper and in our experiments, we refer to ε as *disparity* or *unfairness*. Our algorithm is LP-relaxation followed by randomized rounding. The first step towards developing the algorithm is to formulate the problem as IP. At a high level, the algorithm has three steps (Algorithm 1): step 1 computes the values and necessary inputs for the LP solver; in step 2 it forms an LP-relaxation and solves it; finally, in step 3 it randomly rounds the output of the solver into an approximation solution for the FAIRCS problem.

3.1 LP-relaxation

Instead of the IP formulation discussed in 2.4, we consider an alternative formulation inspired from [43] for the LP-relaxation. A major change in the IP formulation is that we replace the variable x_i, representing the probability of content receiving at node v_i, by the weighted sum of a collection of *binary* variables x_{ir},

---

Algorithm 1 LP-relaxation (LP-APPROX)

Input : $G(V, E), V_s, E_c, k, \mathcal{G}$; **Output** : suggested edges E_p
1: $\mathcal{S}' \leftarrow \text{SHORTESTDISTANCES}(G(V, E), E_c)$ // Algorithm 5
   // $S_{ir} = \{e_j \; | \; (e_j, d_{ij}) \in S_i' \in \mathcal{S}', \; d_{ij} \leq r\}$
2: $\langle \mathbf{x}^*, \mathbf{y}^* \rangle \leftarrow$ Solve the LP in Figure 2
3: **return** ROUND $(\mathbf{y}^*, G(V, E), k, V_s, \mathcal{G})$ // Algorithm 2

$$\text{maximize} \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} x_{ir}$$

$$\text{subject to}$$

$$\forall v_i \in V, r \in [1, r_m], \; x_{ir} \leq \sum_{e_j \in S_{ir}} y_j$$

$$\forall v_i \in V, \; \sum_{e_j \in \mathcal{N}(v_i)} y_j \leq k$$

$$\forall g_a, g_b \in \mathcal{G}, \; \frac{1}{|g_a|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in g_a} x_{\ell r} = \frac{1}{|g_b|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in g_b} x_{\ell r}$$

$$\forall v_i \in V, r \in [1, r_m], \; 0 \leq x_{ir} \leq 1$$

$$\forall e_j \in E_c, \; 0 \leq y_j \leq 1$$

Figure 2: LP-relaxation (relaxed $x_{ir} \in \{0, 1\}$ and $y_j \in \{0, 1\}$) used for solving FAIRCS **problem.**

which is 1 if there exists a path of length at most r from a content node to v_i, and 0 otherwise. Let p be the probability that an edge gets activated during the cascade process. Then, the probability that content reaches a node v_i using a path of length r is p^r. Given a node v_i, let ℓ be its shortest distance from a content node. Furthermore, let r_m be the maximum distance for which all nodes $V \setminus V_s$ are reachable from a content node. First, $\forall r \geq \ell$, $x_{ir} = 1$, while $\forall r < \ell, x_{ir} = 0$. We define the values δ_r, such that $x_i = \sum_{r=1}^{r_m} \delta_r x_{ir}$. We shall explain the derivation of δ_r values in 3.3. Using the above transformation, the objective function can be rewritten as $\sum_{v_i \in V} x_i = \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} x_{ir}$.

Next, for every node v_i and a distance r, we precompute the set S_{ir}, as the set of edges that if added to E_p, the distance to v_i from a content node will be at most r. Introducing the new variables, the LP-relaxation formulation is shown in Figure 2. Recall that in the IP formulation y_j is a binary attribute that is 1 when the edge $e_j \in E_c$ gets selected. Similarly, x_{ir} is 1 only if there exists a path of length r to v_i from a content node. That is, if at least one of the edges in S_{ir} has been selected. This is enforced by the first set of constraints in the LP formulation. The second constraints enforce k incident edges to each node, while the third one enforces fairness.

After constructing the LP as in Figure 2, the algorithm finds its optimal values $\mathbf{x}^* = \{x_{ir}^* \; | \; \forall v_i \in V, r \in [1, r_m]\}$ and $\mathbf{y}^* = \{y_j^* \; | \; e_j \in E_c\}$ and passes those to step 3 of Algorithm 1 for rounding, as we shall explain next in 3.2.

3.2 Randomized Rounding

Next step after solving the LP in Figure 2 is to round the continuous values for the edge selection variables $\mathbf{y}^* = \{y_j^* \; | \; e_j \in E_c\}$ into binary values $\mathbf{y}^+ = \{y_j^+ \; | \; e_j \in E_c\}$. An edge $e_j \in E_c$ is selected for suggestion if $y_j^+ = 1$. We follow a randomized rounding for rounding

Algorithm 2 Randomized Rounding (ROUND)

Input : $\mathbf{y}^*, G(V,E), V_s, \mathcal{G}$; **Output** : selected edges E_p

1: $sol \leftarrow \{\}$
2: **for** $i \leftarrow 1$ to $iter_m$ **do**
3:     **for** $e_j \in E_c$ **do**
4:         **if** RAND$(0,1) \leq y_j^*$ **then** $y_j^+ = 1$, i.e., add e_j to E_{p_i}
5:     $CS \leftarrow \sum_{v_j \in V} F(v_j, E_{p_i})$ // content spread
6:     $\mathcal{F} \leftarrow \max_{g_i, g_j \in \mathcal{G}} \left(\frac{\frac{1}{|g_i|} \sum_{v_\ell \in g_i} F(v_\ell, E_{p_i})}{\frac{1}{|g_j|} \sum_{v_\ell \in g_j} F(v_\ell, E_{p_i})} \right)$ // fairness
7:     add $\langle E_{p_i}, \mathcal{F}, CS \rangle$ to sol
8: $\mathcal{F}_{min} \leftarrow \min_{\mathcal{F}}(sol)$
9: candidates $\leftarrow \{\langle E_{p_i}, \mathcal{F}, CS \rangle \in sol \mid \mathcal{F} - \mathcal{F}_{min} \leq \epsilon\}$
10: **return** argmax$_{CS}$ (candidates)

$\mathbf{y}^*$ to $\mathbf{y}^+$, using $\mathbf{y}^*$ as the probability to select edges for suggestion. That is, the probability to suggest edge $e_j \in E_p$ is $P(y_j^+ = 1) = y_j^*$. To do so, Algorithm 2 draws a number uniformly at random from the range $[0, 1]$. If the generated sample is less than y_j^*, e_j is added to E_p i.e. $y_j^+ = 1$. As we shall study in 3.4, following this rounding approach, the expected number of edges selected per each node is k, and the algorithm guarantees constant approximation ratios both on fairness and the content spread.

The bottle-neck in the time complexity of Algorithm 1 is solving the LP, i.e., the second step. On the other hand, the rounding of the variables is fast (rounding the values of y^* to y^+ is in $O(|E_c|)$). Therefore, since the rounding is randomized, it is reasonable to repeat the rounding step many times to boost the success probability. Let $iter_m$ be the number of times the rounding is repeated and let $E_{p_1} \cdots, E_{p_{iter_m}}$ be the results obtained over different repetition of randomized rounding. From these, the algorithm considers the ones that have fairness close to the most-fair found solution, and from those returns the one that maximizes the content spread (Lines 5-10 of Algorithm 2). In Section 6, we empirically show that in practice this approach finds near-optimal solutions.

3.3 Constructing LP inputs

In Algorithm 1, the first step before solving the LP is to compute the values and construct the sets that are needed to form the LP.

For every node v_i and a distance r, we need to precompute the set S_{ir}, of edges in E_c, that makes v_i at most r hops away from a content node. Again, under RMMP, a path from a content node to v_i can contain at most one edge from E_c. Hence, to compute the sets $S = \{S_{ir} \mid v_i \in V \backslash V_s, r \leq r_m\}$, we compute the shortest distance from any content node to every node in the graph with edge $e_j \in E_c$. That is, $\forall_{e_j \in E_c}$, we compute all-pair shortest paths from content nodes in graph $G(V, E \cup \{e_j\})$. Suppose d_{ij} is the shortest path length from content nodes to a node v_i in graph $G(V, E \cup \{e_j\})$. Then e_j is added to all sets S_{ir} where $r \geq d_{ij}$, because e_j enables a path with length $\geq d_{ij}$ to v_i. Instead of repeating e_j in all sets $S_{ir}, r \geq d_{ij}$, one can save space by defining the intermediate sets $S' = \{S'_{ir} \mid v_i \in V \backslash V_s, d_{ij} = r\}$, where S'_{ir} contains the edges in E_p that enable a shortest path of length r to v_i. Then $S_{ir} = \cup_{r' \leq r} S'_{ir'}$.

Lastly, in order to form the LP, we need to compute the values of δ_r, such that $\sum_{r=1}^{r_m} \delta_r x_{ir}$ is equal to x_i, the probability that a node v_i receives content. Let d be the length of the shortest path from a content node to a node v_i, and let p be the activation probability

for each edge. Then, the probability that it receives content (using RMPP) is $x_i = p^d$. We want to compute the values of δ_r, in a way that for every shortest-path length d, $\sum_{r=1}^{r_m} \delta_r x_{ir} = p^d$. For cases where $d = r_m$, $x_{ir_m} = 1$ and $x_{ir} = 0, \forall r < r_m$. As a result: $\sum_{r=1}^{r_m} \delta_r x_{ir} = \delta_{r_m}$. Hence, $\delta_{r_m} = p^{r_m}$.

When $d = r_m - 1$, $\sum_{r=1}^{r_m} \delta_r x_{ir} = \delta_{r_{m-1}} + \delta_{r_m} = p^{r_m - 1}$. Therefore, $\delta_{r_{m-1}} = p^{r_m - 1} - p^{r_m}$.

Similarly, $\forall r < r_m$, $\sum_{r=1}^{r_m} \delta_r x_{ir}$ can be written as

$$\sum_{r=1}^{r_m} \delta_r x_{ir} = \sum_{r=d}^{r_m} \delta_r = \delta_d + \sum_{r=d+1}^{r_m} \delta_r = p^d$$

$$\sum_{r=d+1}^{r_m} \delta_r = p^{d+1} \Rightarrow \delta_d = p^d - p^{d+1}$$

In summary, $\qquad \delta_r = \begin{cases} p^{r_m} & r = r_m \\ p^r - p^{r+1} & r < r_m \end{cases}$ (7)

3.4 Approximation Analysis

LEMMA 3. *The expected number of selected edges by Algorithm 1 per each node $v_i \in V$ is k.*

THEOREM 4. *If FAIRCS problem has at least one valid solution, then the LP-relaxation algorithm (Algorithm 1) for solving it satisfies the approximation ratio of $(1 - e^{-1})$ on the content spread and $\frac{2e}{1 - e^{-1}}$ on fairness.*

PROOF. Let $\{x_{ir}^* \mid \forall v_i \in V, r \in [1, r_m]\}$ and $\{y_j^* \mid e_i \in E_c\}$ be the optimal values of LP, shown in Figure 2. Besides, let $\{x_{ir}^+ \mid \forall v_i \in V, r \in [1, r_m]\}$ and $\{y_j^+ \mid e_i \in E_c\}$ be the approximation values for the FAIRCS problem. Let us first compute the approximation ratio on the content spread.

$$\mathbb{E}\left[x_{ir}^+\right] = Pr(x_{ir}^+ = 1) = 1 - Pr(x_{ir}^+ = 0)$$
$$= 1 - \prod_{e_j \in S_{ir}} (1 - y_j^*) \geq 1 - \prod_{e_j \in S_{ir}} e^{-y_j^*} = 1 - e^{-\sum_{e_j \in S_{ir}} y_j^*}$$
$$\geq 1 - e^{-x_{ij}^*} \geq (1 - e^{-1})x_{ij}^* \quad (8)$$

Let APR be the content spread of the approximation algorithm output and let OPT be the optimal content spread.

$$\mathbb{E}[\text{APR}] = \mathbb{E}\left[\sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} x_{ir}^+ \right] = \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} \mathbb{E}\left[x_{ir}^+\right]$$
$$\geq \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} (1 - e^{-1})x_{ij}^* = (1 - e^{-1}) \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} x_{ij}^* \quad (9)$$

Since the optimal solution for LP is an upper-bound on the IP solution (optimal content spread),

$$\sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} x_{ij}^* \geq \text{OPT} \quad (10)$$

$$\Rightarrow \mathbb{E}[\text{APR}] \geq (1 - e^{-1}) \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} x_{ij}^* \geq (1 - e^{-1}) \text{ OPT}$$

$$\Rightarrow \frac{\mathbb{E}[\text{APR}]}{\text{OPT}} \geq (1 - e^{-1}) \quad (11)$$

Next, we prove the approximation ratio on fairness. Inspired from [3], we first provide an upper bound on $\mathbb{E}[x_{ir}^+]$ that bounds the value of fairness across different groups.

Looking at Figure 2, if $\sum_{e_j \in S_{ir}} y_j \geq 1$, then $x_{ir}^* = 1$. Hence, (case I) $\mathbb{E}[x_{ir}^+] \leq 1 = x_{ir}^*$.

For cases where $\sum_{e_j \in S_{ir}} y_j < 1$, we know $x_{ir}^* = \sum_{e_j \in S_{ir}} y_j$. Now, if $\exists\, e_j \in S_{ir}$, such that $y_j^* \leq \frac{1}{2e}$, then $x_{ir}^* \geq \frac{1}{2e}$. Therefore, (case II) $\mathbb{E}[x_{ir}^+] \leq 1 \leq 2e x_{ir}^*$.

The only case left is when $\forall e_j \in S_{ir}$, $y_j^* \leq \frac{1}{2e}$. Since for this case, $y_j^* \in [0, 1]$, using the second-order Taylor series expansion of $e^{-2y_j^*}$, we know:
$$e^{-2y_j^*} \leq 1 - 2y_j^* + \frac{e}{2}(2y_j^*)^2 \leq 1 - y_j^*$$

Therefore (case III),
$$\mathbb{E}\left[x_{ir}^+\right] = Pr(x_{ir}^+ = 1) = 1 - Pr(x_{ir}^+ = 0)$$
$$= 1 - \prod_{e_j \in S_{ir}} (1 - y_j^*) \leq 1 - \prod_{e_j \in S_{ir}} e^{-2y_j^*} = 1 - e^{-\sum_{e_j \in S_{ir}} 2y_j^*}$$
$$\leq 1 - e^{-2x_{ir}^*} \leq 1 - (1 - 2x_{ir}^*) = 2x_{ir}^* \quad (12)$$

From cases I, II, and III, we can conclude that $\mathbb{E}\left[x_{ir}^+\right] \leq 2e x_{ir}^*$. As a result,
$$\mathbb{E}\left[\sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} x_{ir}^+\right] = \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} \mathbb{E}\left[x_{ir}^+\right]$$
$$\leq \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} 2e\, x_{ij}^* = 2e \sum_{r=1}^{r_m} \delta_r \sum_{v_i \in V} x_{ij}^* \quad (13)$$

We now use the lower-bound found in Equation 10 and the upper-bound in Equation 13 to bound the unfairness (Equation 6) between any pair of groups $\mathbf{g}_a, \mathbf{g}_b \in \mathcal{G}$.

From the fairness constraint in the LP formulation (the third constraints Figure 2), we know that
$$\frac{1}{|\mathbf{g}_a|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in \mathbf{g}_a} x_{\ell r}^* = \frac{1}{|\mathbf{g}_b|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in \mathbf{g}_b} x_{\ell r}^*$$

Using the upper-bound and the lower-bound in Equations 10 and 13, for every group $\mathbf{g}_a \in \mathcal{G}$, we know:
$$\frac{1}{|\mathbf{g}_a|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in \mathbf{g}_a} x_{\ell r}^+ \geq (1 - e^{-1}) \frac{1}{|\mathbf{g}_a|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in \mathbf{g}_a} x_{\ell r}^*$$
and
$$\frac{1}{|\mathbf{g}_a|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in \mathbf{g}_a} x_{\ell r}^+ \leq 2e \frac{1}{|\mathbf{g}_a|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in \mathbf{g}_a} x_{\ell r}^*$$

As a result, for all $\mathbf{g}_a, \mathbf{g}_b \in \mathcal{G}$,
$$\frac{1 - e^{-1}}{2e} \leq \frac{\frac{1}{|\mathbf{g}_a|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in \mathbf{g}_a} x_{\ell r}^+}{\frac{1}{|\mathbf{g}_b|} \sum_{r=1}^{r_m} \delta_r \sum_{v_\ell \in \mathbf{g}_b} x_{\ell r}^+} \leq \frac{2e}{1 - e^{-1}} \quad (14)$$

$\square$

4 PRACTICAL OPTIMIZATIONS

In this section, we introduce further optimizations to improve the performance of our algorithm in practice.

LP-Advanced algorithm (Extension beyond RMPP): Our approximation algorithm is designed using RMPP as the cascade model. In RMPP, the content spread to a node is the *maximum* probability of a path from a content node to this node containing at most one edge from E_c. This means, if by including edge e_j' the content received by each node is less than or equal to the content received by each node by including e_j, then if e_j is in the chosen set E_p, including e_j' does not change the maximum probability of these

Algorithm 3 LP-Advanced

Input : $G(V, E), V_s, E_c, k, \mathcal{G}, cutoff$; **Output** : suggested edges E_p
1: $sol \leftarrow \{\}$; $\forall_{v_i \in V}\ k_i' \leftarrow k$; Copy E_c to E_c^*
2: **repeat**
3:     $E_p \leftarrow$ LP-Approx$(G(V, E), V_s, E_c^*, k', \mathcal{G})$ // Using MIP for rounding
4:     $sol \leftarrow sol \cup E_p$; $E \leftarrow E \cup E_p$
5:     $\forall_{v_i \in V}\ k_i' \leftarrow \max(k - |\{e_{u,v} \in sol \mid u == i \mid\mid v == i\}|, 0)$
6:     $E_c^* \leftarrow \{e_{u,v} \in E_c^* \setminus E_p \mid k_u' > 0\ \&\&\ k_v' > 0\}$
7: **until** $|E_p| == 0$ OR a small ratio of E_c left
8: **return** sol

paths. Consequently, under *RMPP*, the edge e_j' will not be included if e_j is included. Alternatively, under the MIP and IC models, the content spread to a node is determined by the path with the highest probability (for MIP) or the probability over all paths (for IC) such that the paths contain *any number* of edges from E_c. Consequently, after adding e_j, a path that includes both e_j and e_j' may have an additional effect under MIP and IC when having no effect under RMPP. As discussed in Section 3, Algorithm 1, using RMPP, has been proven to find a solution relatively efficient with theoretical guarantees on accuracy. To expand on this in order to find edges that would be included in MIP and IC but not RMPP, we propose to *rerun* Algorithm 1 multiple times, allowing it to first calculate a result E_p, then redefine $E \leftarrow E \cup E_p$ and $E_c \leftarrow E_c \setminus E_p$. By moving E_p from E_c to E, in the next iteration e_j' can provide addition content to nodes, even under RMPP. To mantain k expected incident edges in this modification, we only find k_i' edges for a node v_i, determined by the number of incident edges to v_i in E_p from all previous iterations This process is shown in Algorithm 3.

Additional optimizations to limit the variables being solved in an Linear Program and to dynamically update the graph are included in the appendix.

5 SCALABILITY

Though polynomial, solving LP problems is time-taking, and usually impractical for large settings with tens of thousands or more variables. Similarly, as we demonstrate in our experiments in Section 6, our LP-relaxation algorithm is not scalable for such settings. Thus, we introduce a different approach for large settings and, instead of solving one inefficient LP, use a heuristic to break down the problem into *multiple instances of* LP *with reasonably small sizes*.

Our extension for large-scale settings is based on the observation that *the contribution of a path to the content spread exponentially reduces as its length increases*. As a result, the nodes that are far from the content nodes have a minimal impact in the content spread. Therefore, solving the problem for the high-impact nodes can be viewed as an approximation heuristic for solving the full problem. On the other hand, only considering the impact of a small portion of the graph on the overall problem, can significantly reduce the size of LP, remarkably increasing the algorithms efficiency. Using this observation, we propose an iterative approach traversing the graph that, at a high level, starts from a local neighborhood around the content nodes, forms a reasonably small subgraph in the neighborhood, and solves it using the LP-relaxation algorithm. Then we gradually add nodes, branching away for the content nodes, and

solving the overall problem by making changes to just those nodes using the same LP-relaxation technique.

Subgraphs: Solving the FAIRCS problem using Algorithm 1 or Algorithm 3, for a candidate edge set E_c, and each vertex whose expected content is improved by an edge in E_c, $V^* \subset V$, requires solving a LP of size $n = \Omega(|V^*| + |E_c|)$. Instead of solving a single LP with n variables, taking $O(n^{2.37})$ time using [23], we aim to solve a series of b smaller problems of size $\bar{n} << n$, so that the runtime is improved to $O(b \times \bar{n}^{2.37})$.

Specifically, we design a method to *incrementally* add nodes to the graph such that processing the graph in stages provides a meaningful result while limiting computation time. To do this, we start by dividing the graph (as we shall discuss later) into b subsets of nodes, each containing $\frac{N}{b}$ nodes. In each iteration i, we consider the set of nodes V' defined as the union of all the subsets $1..i$. This means with each iteration our graph grows, until approaching the full graph, allowing us to isolate portions of the graph and converges on an overall solution. Also, in iteration i, we consider *new* candidate edges E'_c, incident to both a node from iteration i as well as another node which can be from iteration i or any other iteration in V'. This has a few effects. First, since some nodes are processed earlier, they may have reached k edges in an earlier iteration and can not add additional candidate edges, limiting the processing which occurs in later iterations. Next, since each candidate edge that is considered is only processed once, the total sum of edges processed is less than the number of candidate edges, $|E_c| \geq \sum |E'_c|$. It follows that each iteration the candidate set E'_c has less than or equal to $\frac{|E_c|}{b}$ candidate edges on average. Finally, the nodes receiving an improvement in expected content $V'^* \subset V'$ is localized to the ones reachable by candidate edges from the nodes in the new iteration and thus V'^* is also much smaller than both V' and V. Hence, at each iteration we process a much smaller *subgraph* limited by the current nodes V' and the set of edges available in the current iteration E'_c, equal in size to $\bar{n} = \Omega(|V'^*| + |E'_c|)$, which is then repeated over a total of b iterations.

When processing a subgraph, a decision is made to add or not add an edge $e'_j \in E'_c$ to E_p. If this decision were made on a full graph, the choice to add or not add an edge would be dependent on 1) the impact of the edge on all nodes in V, as well as 2) all other candidate edges in E_c it could choose instead. In order for the heuristic to be effective, V' must somewhat mimic the structure of V, and the method of producing subgraphs must provide a variety of options, with the options believed to be more impactful being considered first.

Forest Fire Sampling: In order to generate the subsets described in the previous section, we need a way to find portions of the graph which are representative of the graph as a whole, and provide choices with a higher impact earlier. To do so, starting from the content nodes V_s, we perform sampling using a traversal based method. There are multiple intuitions as to why this is a good choice. First, the content received by each node $v_i \in V \setminus V_s$ is based on its path from a content node $v_j \in V_s$. By having the graph that always includes the nodes in V_s, we can find new paths $v_j \rightarrow v_i$ which have a shorter distance than the initial distance to a node v_i. Second, the impact of an edge $e_{u,v}$ is based on the nodes reachable by paths from

---

Algorithm 4 Scalable Algorithm (LP-SCALE)

Input : $G(V, E), V_s, E_c, k, \mathcal{G}$, Iter; **Output** : suggested edges E_p

1: excluded← {}; $E_p \leftarrow$ {}; $\forall_{v_i \in V}$: $k'_i \leftarrow k$
2: **while** $(G'(V', E') \leftarrow$ForestFireSample$(G(V, E), V_s, $Iter$)) \neq G$ **do**
3:     add E_p to $G'(V', E')$
4:     $E'_c \leftarrow E_c$ on G' − excluded
5:     add E'_c to excluded
6:     $S' \leftarrow$ SHORTESTDISTANCES$(G'(V', E'), E'_c)$ // Algorithm 5
7:     $\langle \mathbf{x}^*, \mathbf{y}^* \rangle \leftarrow$ Solve the LP in Figure 2
8:     add ROUND $(\mathbf{y}^*, G'(V', E'), \mathbf{k}', V_s, \mathcal{G})$ to E'_p // Algorithm 2
9:     **for** $v_i \in V$ **do** $k'_i \leftarrow \max(k - |\{e_{u,v} \in E_p | u = i\}|, 0)$
10: **return** E_p

---

v, where the paths continue to improve content, so analyzing connected components allowed us to make more meaningful decisions. A traversal based approach supplies such connected components.

We use the Forest Fire algorithm [58], the de-facto graph sampling method that maintains the underlying properties of the graph. This method of sampling randomly generates a subset of a node's neighbors in the graph. In doing so, the subgraphs being sampled would include both a large number of neighbors for each node, as well as a significant depth, meaning a new candidate edge $e_{u,v} \in E'_c$ which ended at a node in the current subgraph $v \in V'$ would have information about many of the *additional* nodes whose content spread was updated by including edge $e_{u,v}$. Additionally, since the branching factor is large in a given sample, a node u might have several candidate edges, $e_{u,v_1}, e_{u,v_2}, ...$ such that it could make opinionated choices about which edges to select. Finally, the forest fire method will produce more prominent nodes closer to the source earlier than less prominent nodes further away. This means that we will be able to make the decisions which are more likely to have a higher impact on content spread earlier on, while making less meaningful decisions later.

5.1 LP-SCALE **Algorithm**

We now propose our algorithm LP-SCALE (Algorithm 4) for solving the FAIRCS at scale. The algorithm works by using a Forest Fire sampling to achieve nodes close to the content nodes, and then finds the subproblem on that induced subgraph consisting of the candidate edges on that graph E'_c and the nodes improved by those edges V'^* as described above. Based on this problem, an instance of LP-Advanced is solved for the first edges to be added to E_p based on the current understanding of the graph $G'(V', E')$. For every node $v_i \in V$, let k'_i be the number of candidate edges remained to be suggested to it. Initially, $k'_i = k$ since no edge has been added to E_p. After adding the new candidate edges to E_p, values of k'_i in vector $\mathbf{k}'$ also get updated accordingly. Additionally, the new shortest distances to all nodes from source nodes and the corresponding content spread disparity is updated dynamically without recalculating the entire graph. The graph is expanded finding nodes that are either less connected or further from the content source nodes V_s, meaning the redefined candidate set E'_c contains edges that are generally less important than the previous iteration, but more important than later iterations. The algorithm then expands upon E_p with a subset of the edges in E'_c, now using the increased information of the new nodes in V'. As described before, this process

repeats over b iterations, adding $\frac{N}{b}$ nodes and an average $\frac{|E_c|}{b}$ each iteration. Keeping the size of each LP instances relatively small in each iteration, our algorithm scales to large setting while producing results with low disparity and high content spread.

6 EXPERIMENTS

6.1 Experiments Setup

The experiments were conducted using a single work station with a Core i9 Intel X-series 3.5 GHz processor and 128 GB of DDR4 memory, running Linux Ubuntu. The algorithms were implemented in Python 3. For evaluation purposes, we generated samples from two data sets: a collection of Antelope Valley synthetic networks to model obesity prevention intervention, as well as a real data set from the Pokec social network.

Antelope Valley Synthetic Networks Data Set: We use the Antelope Valley Synthetic Networks from Farnadi et al.'s work on Fair Influence Maximization [29]. We used twenty of the synthetic networks. Each of these networks initially contained 500 nodes and between 1576-1697 edges. Each node in the network has a sensitive attribute of gender with two classes.

Pokec Social Network Data Set: We used the Pokec Social Network data set from the SNAP's network data sets [59]. The Pokec data set represents real data on the Pokec social network, and it contains 1.63 million nodes and 30.6 million edges. Each node in the network contains a sensitive attribute gender.

Evaluation Plan and Performance Measures: We evaluate our algorithms LP-Approx (Algorithm 1), LP-Advanced (Algorithm 3), and LP-SCALE (Algorithm 4). The metrics we focus on are (1) the run time of the algorithm, (2) the disparity between groups, and (3) the change in content spread. We report the runtime in seconds, we report disparity (ε in Equation 6) as the % difference between the two groups' size weighted content spread, or $\varepsilon \times 100$.

As explained in Section 2, computing the content spread using IC is known to be #P-hard [19, 20] and, hence MIP [20], and later RMPP [19], have been proposed alternative measures to approximate it. Even though the theoretical analyses in this paper have been carried out using RMPP, in practice our algorithms work (almost) equally good for other cascade models. Confirming that we observed consistent results for RMPP, MIP, and IC, we report our evaluations using MIP to demonstrate the practical performance of our algorithm beyond RMPP. The change in Content Spread is reported as the percent increase in MIP Content Spread:

$$\sum_{v_i \in V} \frac{F_{MIP}(v_i, E_p) - F_{MIP}(v_i, \{\})}{F_{MIP}(v_i, \{\})} \times 100 \qquad (15)$$

Baselines: In order to fully demonstrate the effectiveness of our method, we compare against several baseline algorithms. All baselines have been adapted to choose k edges per node if their original algorithm did not:

- *Brute Force*. Checks all valid combinations of edges from the candidate set, choosing the combination which maximizes content spread, constrained to k edges per node, and perfect fairness between demographic groups.
- *Continuous Greedy* [19]. Each candidate edge is assigned a weight starting at 0. Over 500 iterations, 10 sampled graphs are generated with the edges included with the probability of their weights.

Each edge is then added to the sampled graphs. A value based on it's lift on the sample is assigned to the edge. A matching of the highest valued edges is made, and then those edges have their weights increased. After all iterations the edges with the highest weights are chosen as the solution. Optimizations made based on our dynamic algorithms are included in the appendix.

- *IRFA* [91]. Greedily chooses edges based on their Influence Ranking (IR) or effect on down stream nodes, and the improvement in probability of that influence ranking based on adding the edge. It then performs a fast adjustment (FA) to recalculate probabilities and influence rankings. The input to this algorithm should be a Directed Acyclic Graph. As a preprocessing step, we used [84] to remove cycles from the graph.
- *SpGreedy* [100]. Measures the relationship between demographic groups via the opinion model properties *polarizaiton* (groups being radically separated) and *disagreement* (members of different groups neighboring each other). The fairness of the network is then attempted to be increased by greedily choosing edges which minimize these properties. The opinions of one demographic group are given as 1 and for the other are -1.
- *ACR-FoF* [95]. The algebraic connectivity of including an edge and the number of shared friends are combined to score each edge. The highest scored edges are greedily chosen to k per node. This algorithm require an undirected graph, so for this baseline, in our experiments we ignored the edge directions.

Candidate Edge Selection: A candidate edge set is one of the inputs to our algorithm and our contribution is agnostic to what model is used for identifying such a candidate set. We considered following types of friend suggestion to empirically show how our algorithm performs over different inputs.

- *Friend of Friend (FoF)* - All friends of friends (not already friends) are considered as possible recommendations.
- *Intersecting Group Count (IGC)* [74]. For each node $v_i \in V$, a rating for a potential candidate edge is given to each other node equal to the number of groups that node shares with friends of v_i. Each node on the graph's neighborhood is considered as a group. If a node is not in the same group as any friend it does not have a rating. The 1/3 highest rated nodes are included in the candidate set of suggestions for v_i.

6.2 Experiment Results

The Cost of Fairness: Algorithm 3 uses LP-Relaxation and randomized rounding for solving FairCS. Alternatively, if the LP did not contain the fairness constraint (Definition 2), it would generate an approximate solution to CSM (Definition 1). Fairness is a desirable property of an algorithm, but it comes with "The Cost of Fairness (CoF)". The CoF of FairCS is demonstrated in Figure 3. The left-axis and solid lines in the figure show the average lift, while the disparity is shown in the right-axis with dashed lines. In this figure, Algorithm 3 is run both with and without the fairness constraint, with $k = 5$, $p = .5$, $|V| = 500$, an initial disparity between 30% and 35%, and with 20 trials across each of a varying $|V_s|$. The resulting average lift and average disparity are recorded. As shown the algorithm without the fairness constraint achieves 1.25x on lift. Alternatively the algorithm with the fairness constraint reduces disparity by over 98% in all cases while the algorithm without it

reduces it by less than 60%. The "Cost of Fairness" is the demonstrated as a loss in overall lift for the result of near zero fairness between demographic groups.

Demonstration of Near Optimality: Algorithm 1, LP-Approx, is demonstrated to be nearly optimal in both overall content spread and disparity, by comparison with the optimal Brute Force baseline. For this demonstration, 7 graphs with $|V| = 40$ are sampled from the Antelope Valley dataset, such that Brute Force can achieve 0 disparity through edge selection , while considering 1 content node, with $k = 2$, $p = 0.5$, and an initial disparity of $95.6 − 98.9\%$. The LP-Approx algorithm was run ten times on each of the 40 node graphs and the average and standard deviation of these runs were reported and compared against the results of running Brute Force.

Figure 4 demonstrates the near optimal disparity results. As shown, for all but two problem instances, the LP-Approx was able to achieve optimal disparity ($\varepsilon = 0$). Even in the worst case, the average disparity across experiments never exceeded 0.006.

Figure 5 demonstrates the near optimal lift results. For the previously mentioned disparity results, LP-Approx achieved optimal lift in every case except two. It should be noted that in some cases LP-Approx strictly dominates the optimal solution in both disparity and lift. This is because occasionally in rounding $> k$ edges are selected, as only the expected number of edges for each node is k.

Comparison with Various Baseline Methods: Algorithm 3 (LP-Advanced), is demonstrated to achieve with both methods of candidate edge selection a higher content spread and lower disparity than all baselines. These results are demonstrated on both the 500 node Antelope Valley datasets, and on 1000 node samples of the Pokec dataset, with sources selected to be between $30 − 35\%$ and $20 − 25\%$ respectively. Experiments were done varying k, p, and $|V_s|$. For each setting, 20 trials were run, with $k = 3$, $p = 0.5$ and $|V_s| = 3$ when not being varied (varying k and varying p are included in the appendix.)

In Table 2, baseline methods and LP-Advanced are compared on 500 nodes. The bold elements contain the best values for each property on each setting. LP-Advanced had the highest lift and lowest disparity compared against all baselines in all instances. It however was not the fastest, where ACR-FoF had the lowest runtime (Figure 6 demonstrates the comparison between LP-Advanced and ACR-FoF w.r.t. runtime). Even though we are 10x slower than ACR-FoF, we had 3-4x greater content spread lift, and while they reduced disparity by at most 33% we reduced it by 99%! The baseline with the next highest lift was Continuous Greedy (Figure 7 demonstrates the comparison between LP-Advanced and Continuous Greedy w.r.t. lift), and the next lowest disparity was closely between Continuous Greedy and SpGreedy (Figure 8 demonstrates comparison between LP-Advanced and both these methods w.r.t. disparity). Additionally, at 500 nodes, LP-Advanced had a shorter runtime than both Continuous Greedy and SpGreedy.

For 1000 node graphs, LP-Advanced is compared against baselines in Table 3. Again, the best results are shown in bold. LP-Advanced still achieved the highest lift and lowest disparity, and in these instances IRFA had the lowest runtime (Figure 9 demonstrates the comparison between LP-Advanced and IRFA w.r.t. runtime) While we are 10x slower than IRFA we produced 3x greater lift in content spread, and they achieved only a 25% improvement in disparity compared to our 95% improvement. Continuous Greedy

produced the next highest lift. (Figure 10 demonstrates the comparison between LP-Advanced and Continuous Greedy w.r.t. lift), and the next lowest disparity was between SpGreedy and Continous Greedy (Figure 11 demonstrates comparison between LP-Advanced and both these methods w.r.t. disparity). Additionally, at 1000 nodes, LP-Advanced runs faster than only SpGreedy.

Scalability: For these experiments, Pokec was sampled at 500, 1000, 2000, and 4000, 10k, 20k, 50k, 200k, and 500k nodes, with $k = 3$, and $p = 0.5$ for each setting, and $|V_s| = 5$ for sizes up to and including 10k nodes, while $|V_s| = 10$ for 20k nodes and greater. For up to and including 4k nodes, results are averaged over 5 trials. Due to runtime constraints, for 10k nodes and higher, results are from one trial. FoF candidate edges were used. The baseline methods were compared with both LP-Advanced as well as LP-SCALE. If an algorithm took greater than 24 hours, results were not reported. Additionally, IRFA is not reported over 20k due to constraints with generating the necessary DAG from the initial graph.

The results are shown in Table 4 (best results in bold). This table shows that while LP-Advanced is able to be run, it produces both the highest lift and lowest disparity (except once our scalable algorithm produced higher lift). LP-Advanced produced 1.5x as much content spread as the next closest baseline, Continuous Greedy. SpGreedy achieved the next best reduction in disparity of 50%, and LP-Advanced *reduced the disparity by at least 99.9%*. In all other cases, the next best algorithm is LP-SCALE, producing only marginally less lift and slightly higher disparity. For 500 and 1000 nodes, IRFA had a lower runtime than LP-SCALE. However, LP-SCALE had 2x the lift of IRFA and while IRFA reduced disparity by 25%, LP-SCALE reduced disparity by 96%. For 2000 nodes and higher, LP-SCALE always resulted in the shortest runtime. Additionally, when LP-Advanced cannot be run, LP-SCALE always produces the lowest disparity among all methods. LP-SCALE produced the highest lift at 10k, 20k, and 50k nodes, followed by Continuous Greedy. Then for two settings, 100k nodes and 200k nodes, Continuous Greedy produced a higher lift than LP-SCALE. On the same settings Continuous Greedy only reduced the disparity by at most 25%, whereas LP-SCALE reduced the disparity by at least 98%. Since all other methods failed to run within runtime constraints at 500k nodes, LP-SCALE is the defacto best option. LP-Advanced, LP-SCALE, and Continuous Greedy are compared w.r.t. lift, disparity, runtime and in Figures 12, 13, and 14 respectively.

Additional experiment results on configuring LP-SCALE are provided in the appendix.

7 RELATED WORK

Propagation in networks has been extensively studied in the literature [25, 73]. There are two main approaches on propagation optimization, 1. (influence maximization) the orthogonal problem of selecting nodes to maximize propagation, and also, 2. (content spread) targeting the topology of the network and selecting edges to recommend to users to maximize propagation.

Influence Maximization: The problem of finding a set of most influential (seed) nodes in a graph is called Influence Maximization(IM) [8, 25, 60]. The first discrete optimization definition of Influence Maximization was proposed in [54], where relying on sub-modularity of the Influence function, the author proposed a
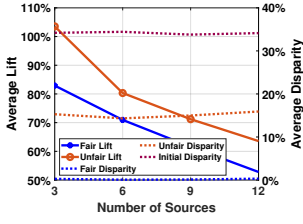
Figure 3: Alg. 3: w/ and w/o the fairness constraint, comparing lift and disparity
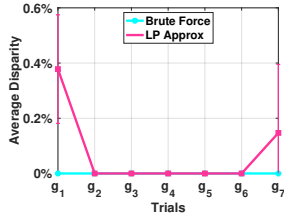

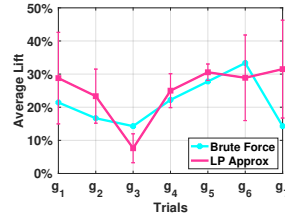**Figure 4: demonstration of near optimal disparity of LP-Approx**

Figure 5: demonstration of near optimal lift of LP-Approx
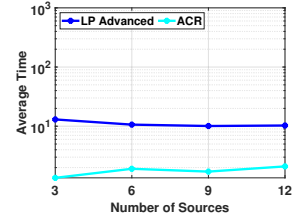

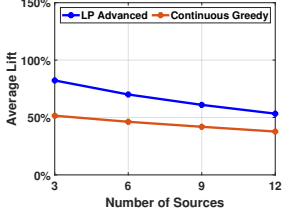**Figure 6: Runtime with varying sources at 500 nodes**

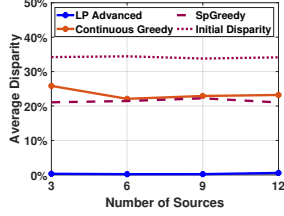**Figure 7: Lift with varying sources at 500 nodes**

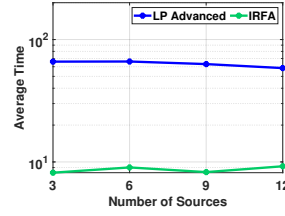**Figure 8: Disparity with varying sources at 500 nodes**

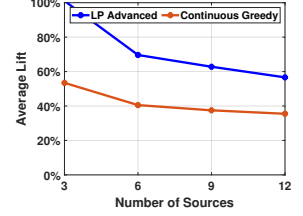**Figure 9: Runtime with varying sources at 1000 nodes**

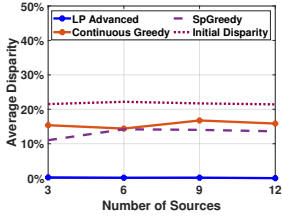**Figure 10: Lift with varying sources at 1000 nodes**

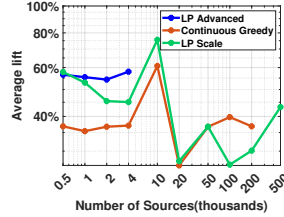**Figure 11: Disparity with varying sources at 1000 nodes**
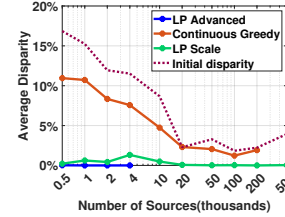

Figure 12: Lift at scale

Figure 13: Disparity at scale
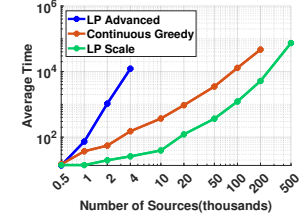
Figure 14: Runtime at scale

Candidate Edges	$|V_s|$	Initial Disparity	LP - Advanced (ours)			Continuous Greedy			IRFA			SpGreedy			ACR - FoF		
			Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time
FoF	3	34.2±0.3	**82.3±14.4**	**0.3±0.5**	13.0±13.9	51.6±8.2	25.8±5.6	33.0±8.9	25±9.1	29.2±8.6	4.3±1.6	20.1±9.0	21.1±6.1	41.1±10.0	27.1±9.0	25.0±7.1	**1.3±0.7**
	6	34.4±0.5	**70.0±11.3**	**0.2±0.8**	10.6±2.6	46.2±7.3	22.1±6.9	26.6±4.0	22.7±5.8	26.1±6.9	3.6±1.6	17.8±5.4	21.5±5.0	43.3±6.9	24.4±4.3	25.0±5.4	**1.9±0.8**
	9	33.8±0.6	**61.0±6.2**	**0.2±0.4**	10.1±3.5	41.9±5.1	22.9±6.2	25.5±6.9	21.0±5.3	27.5±5.0	4.0±1.4	16.5±3.0	22.2±3.7	41.4±7.7	21.2±4.4	27.4 ±4.7	**1.7±0.8**
	12	34.1±0.5	**53.4±7.9**	**0.6±1.7**	10.3±3.4	37.8±5.7	23.2±5.6	23.3±6.5	20.1±5.0	26.8±3.9	3.9±1.8	16.7±3.0	21.0±4.4	45.4±9.5	20.2±4.3	25.0±4.5	**2.1±1.1**
IGC	3	34.2±0.3	**94.0±20.5**	**0.6±1.5**	18.7±4.8	61.9±13.5	21.0±10.1	35.1±10.1	28.0±9.9	27.5±9.2	3.0±1.2	23.6±11.6	20.0±7.1	54.7±16.6	20.0±6.3	26.3±7.2	**2.5±1.0**
	6	34.4±0.5	**77.0±13.5**	**0.1±0.2**	11.7±2.9	51.2±8.4	16.7±8.0	24.1±6.5	21.0±5.9	27.5±6.1	3.0±0.7	19.9±5.9	19.8±5.9	51.2±9.9	18.2±5.3	27.9±5.8	**1.9±0.6**
	9	33.8±0.6	**65.8±9.5**	**0.1±0.3**	17.4±15.8	46.7±7.6	20.0±7.4	24.1±5.8	21.7±5.1	27.4±4.7	2.8±0.7	19.1±4.9	21.7±5.4	48.0±11.0	19.8±4.2	26.9±6.4	**2.2±1.0**
	12	34.1±0.5	**56.0±7.7**	**0.1±0.1**	13.4±5.0	40.9±5.4	20.2±4.2	26.7±8.0	20.8±4.1	26.0±6.3	3.2±1.0	16.7±5.3	22.2±5.1	53.3±8.7	18.19±5.2	25.9±4.0	**2.2±1.1**

Table 2: Comparing Lift (%), Disparity (%), and Runtime (sec), on 500 node Antelope Valley data sets, of LP-Advanced and all baselines, varying $|V_s|$, using 20 trials per setting. Both FoF and IGC were considered for candidate edge selection. $k = 3$, $p = 0.5$

greedy algorithm for the problem. Furthermore, practical optimizations have been proposed [20, 58] to address the scalability issue of the greedy algorithm.

Content Spread: Unlike the IM research that finds a set of seed *nodes*, the objective of the Content Spread Maximization problem is to find a set of *potential edges* that once added, maximize the content spread [19, 80, 91, 95]. The candidate set of edges can be considered an input into this problem for many methods, allowing it to wrap various friendship suggestion methods.[28, 44, 74, 87]. With the introduction of the RMPP cascade model, [19] proposes the Continuous Greedy approximation algorithm, achieving a content spread 90 times higher than other heuristics and greedy strategies. Content spread algorithm have also been derived to target the content spread capabilities of the entire network as opposed to a given set

of sources [95]. Recent works in content spread have optimized the process by considering only the influence ranking of a single node[91]. Our paper falls under this category of Content Spread algorithms, adding fairness as a constraint to the problem. However, as shown by Theorem 2 and demonstrated in our experiments, existing approaches fail to extend for our problem setting.

Social Network Polarization and Opinion Formation: Groups in the network often take opposing opinions, and measuring the structure of the network in relation to these opinions is the problem of opinion formation [21, 34, 40], and polarization [22, 38, 42]. A popular approach for this line of work is based on the Friedkin-Johnsen model [34]. In this model, individuals have an innate opinion, and then based on that and the topology of the network, they establish an expressed opinion. Bindel et al. recognized that this opinion

Candidate Edges	$|V_s|$	Initial Disparity	LP - Advanced (ours)			Continuous Greedy			IRFA			SpGreedy			ACR - FoF		
			Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time
FoF	3	21.5±1.2	**101±40.8**	**0.2±0.6**	66.0±18.0	53.5±16.2	15.4±4.1	51.1±11.1	30.9±16.1	16.4±4.2	**8.2±2.6**	30.8±12.8	11.0±2.9	290±94.5	36.8±29.5	16.2±3.9	19.2±14.4
	6	22.2±1.4	**69.6±21.5**	**0.1±0.5**	66.2±19.3	40.5±9.5	14.4±3.4	43.7±9.1	24.8±13.0	18.3±3.2	**9.0±3.9**	20.1±8.2	14.2±2.5	281±82.0	19.8±12.1	18.3±3.9	19.0±11.7
	9	21.7±1.3	**62.8±13.6**	**0.2±0.3**	63.0±11.2	37.5±6.1	16.8±3.2	42.7±11.5	20.7±5.5	18.8±3.3	**8.3±3.7**	18.2±3.8	14.0±3.0	257±46.6	19.7±6.6	18.0±3.0	13.7±4.5
	12	21.4±1.1	**56.6±9.6**	**0.0±0.0**	58.5±11.9	35.5±5.4	15.9±2.7	40.1±7.6	20.1±3.6	18.3±2.7	**9.2±3.9**	16.9±3.4	13.6±1.9	275±95.3	17.1±6.4	18.0±2.5	17.1±15.5
IGC	3	21.5±1.2	**109±59.8**	**1.4±2.6**	124±69.1	67.0±43.0	13.2±4.3	63.5±19.2	31.8±18.5	16.4±4.1	**8.8±3.7**	38.6±37.8	11.8±4.9	345±118	23.7±16.3	17.6±3.4	45.1±37.0
	6	22.2±1.4	**70.1±30.2**	**1.2±2.2**	102±52.3	45.9±20.5	13.7±3.8	54.1±15.8	26.5±16.1	18.5±3.2	**8.3±3.1**	21.8±14.4	14.6±3.9	336±108	16.7±10.4	19.0±3.1	36.7±32.0
	9	21.7±1.2	**62.5±16.9**	**0.8±1.8**	89.6±29.9	41.7±11.7	14.9±2.6	51.1±16.0	21.5±6.6	18.1±3.4	**9.5±4.0**	20.2±8.1	14.7±3.0	306±77.0	15.4±7.6	19.4±3.2	26.3±16.7
	12	21.4±1.1	**56.6±14.8**	**0.8±1.4**	84.3±35.7	37.7±9.5	14.7±2.5	47.8±17.6	21.0±5.2	18.0±2.6	**10.3±3.8**	19.7±8.4	14.6±2.1	312±114	14.8±7.4	19.4±2.7	32.8±40.1

Table 3: Comparing Lift (%), Disparity (%), and Runtime (sec), on 1000 node Pokec samples, of LP-Advanced and all baselines, varying $|V_s|$. 20 trials are used at each setting. Both Friend of Friend and Intersecting Group Count are reported. $k = 3$, $p = 0.5$

Nodes	Initial Disparity	LP-Advanded (ours)			LP-SCALE (ours)			Continous Greedy			IRFA			SpGreedy			ACR-FoF		
		Lift	Disparity	Runtime	Lift	Disparity	Runtime	Lift	Disparity	Runtime	Lift	Disparity	Runtme	Lift	Disparity	Runtime	Lift	Disparity	Runtime
500	16.6±0.1	56.5±15.2	**0.0±0.0**	13.7±4.4	**57.8±14.2**	0.2±0.2	14.1±5.5	36.9±11.7	11.0±4.6	14.9±2.3	19.1±2.8	14.0±7.8	**3.1±0.8**	20.8±7.4	9.0±5.0	83.9±16.1	16.7±8.7	14.3±5.6	6.7±3.6
1000	15.2±0.1	**55.4±13.7**	**0.0±0.0**	73.4±13.8	53.0±12.9	0.6±0.4	14.0±4.3	35.4±8.1	10.7±3.4	37.0±9.6	20.2±5.5	12.6±5.3	**13.2±2.5**	15.6±5.4	9.8±3.7	332±70.0	11.1±4.4	13.7±4.8	40.4±18.9
2000	11.9±0.0	**54.3±12.5**	**0.0±0.0**	1054±346	45.4±9.3	0.4±0.4	**19.7±6.8**	36.7±8.1	8.3±2.0	55.2±9.2	16.2±6.8	10.6±2.6	74.7±13.1	14.2±5.8	7.2±3.4	1597±369	10.9±5.4	10.9±3.01	285±141
4000	11.5±0.0	**58.0±13.8**	**0.0±0.0**	1.2e4±3776	45.1±7.7	1.3±0.7	**26.0±8.9**	37.1±8.7	7.6±1.0	151±34.1	15.2±8.7	9.3±1.3	337±44.2	12.34±4.2	6.8±1.6	7391±1267	10.8±4.5	11.1±1.4	1355±490
10k	8.67	-	-	-	**75.6**	**0.5**	**39.2**	60.8	4.7	370	17.4	8.4	442	16.4	3.4	4.9e4	16.9	7.0	4754
20k	2.3	-	-	-	**27.6**	**0.1**	**122**	26.6	2.3	943	6.1	3.4	1.2e4	-	-	-	-	-	-
50k	3.3	-	-	-	**36.7**	**0.0**	**365**	36.7	2.1	3498	-	-	-	-	-	-	-	-	-
100k	1.8	-	-	-	26.8	**0.0**	**1222**	39.8	1.2	1.3e4	-	-	-	-	-	-	-	-	-
200k	2.2	-	-	-	30.1	**0**	**5160**	37	1.9	4.7e4	-	-	-	-	-	-	-	-	-
500k	3.9	-	-	-	**43.3**	**0.0**	**7.4e4**	-	-	-	-	-	-	-	-	-	-	-	-

Table 4: Scaling results comparing Lift (in %), Disparity (in %), and Runtime (in seconds), while varying graph size from 500 nodes to 500k nodes. $k = 3$, $p = 0.5$ and Candidate Edges are FoF for all settings. 5 trials were used up to 400k nodes, then only 1 trial is used. $|V_s| = 5$ for 500-10k nodes, and $|V_s| = 10$ for 20k+ nodes. If an algorithm took longer than 24 hours, results were not reported, and IRFA cannot be run greater than 20k due to preprocessing constraints.

could be calculated using matrix multiplication [14]. Several authors built upon this idea establishing calculation of polarization and other metrics using this technique [22, 62, 64]. Related to our work, several authors used the method of edge suggestion in relation to network opinions in order to alter polarization and other metrics [37, 61, 100]. The main difference between these works and ours is that these works target conflict between groups on the social network, while ours targets equitable reception of content between groups, making it suited for solving FairCS.

Fairness: Machine bias and fairness in data science has become timely across different research communities, especially in ML [9, 10, 15, 26, 30, 35, 50, 52, 53, 65] and data management [4, 6, 7, 36, 49, 66, 67, 77–79, 83, 89, 98, 99]. While extensive research focused on the scalability and efficiency in this literature, recently fairness started to come into the picture in propagation optimization. More recently, equitable access to information has raised questions. As a result, a related line of research started with proposing fairness-aware frameworks. As shown in [32], the structure of the network can cause individuals from minority groups to have less access to the content across the network. In influence maximization, while works like [32] uses a maxmin formulation to impose fairness in Influence Maximization problem, other works address fairness by defining fairness constraints [12, 29, 43], equivalent objective functions that encounter fairness notions [70], assure that a certain group is influenced in addition to overall influence [39], a new multi-objective model that considers a fair influence spread among different groups [86, 94], or even a Mixed Integer linear Programming model which can satisfy a variety of fairness constraints [29]. Also, ML community has combined the adversarial networks with fairness concepts in order to design fairness aware methods in both link prediction [61] and node selection [55]. Still, to the best of our

knowledge, this paper is the first to consider fairness in the Content Spread approach to propagation optimization.

8 CONCLUSION AND FUTURE WORK
In this paper, we proposed the problem of edge (friendship) suggestion for maximizing the fair content spread. We showed the problem is NP-hard and proved its inapproximability in polynomial time. Then allowing approximation both on fairness and on the objective value, we designed a randomized LP-relaxation algorithm with fixed approximation ratios. We proposed practical optimizations and designed a scalable algorithm that dynamically adds the solutions of sub-problems with reasonably small sizes to the graph, updating the solution incrementally. We conducted comprehensive experiments on real and synthetic data sets to evaluate our algorithms. Our results confirm the effectiveness of our algorithms, being able to find solutions with near-zero unfairness while significantly increasing the content spread. Our scalable algorithm could scale up to *half a million nodes* in a reasonable time, reducing unfairness down to around 0.0004 while increasing the content spread by 43%.

In this paper, we study the problem of edge suggestion without considering the recommendation acceptance probabilities. While a simple resolution to this may be to recommend a higher number of edges, carefully addressing this problem requires new techniques that we consider as part of our future work. Additionally, in future works, we will explore alternative scenarios, such as determining fair content spread when all nodes produce content and the content is of a variety of topics.

REFERENCES

[1] Chiara Accinelli, Barbara Catania, Giovanna Guerrini, and Simone Minisi. 2021. The impact of rewriting on coverage constraint satisfaction.. In *EDBT/ICDT Workshops*.

[2] Eytan Adar and Lada A Adamic. 2005. Tracking information epidemics in blogspace. In *WI*. IEEE, 207–214.

[3] Abolfazl Asudeh, Tanya Berger-Wolf, Bhaskar DasGupta, and Anastasios Sidiropoulos. 2020. Maximizing coverage while ensuring fairness: a tale of conflicting objective. *CoRR*, abs/2007.08069 (2020).

[4] Abolfazl Asudeh and HV Jagadish. 2020. Fairly evaluating and scoring items in a data set. *PVLDB* 13, 12 (2020), 3445–3448.

[5] Abolfazl Asudeh, HV Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing fair ranking schemes. In *SIMGOD*. 1259–1276.

[6] Abolfazl Asudeh, Zhongjun Jin, and HV Jagadish. 2019. Assessing and remedying coverage for a given dataset. In *ICDE*. IEEE, 554–565.

[7] Abolfazl Asudeh, Nima Shahbazi, Zhongjun Jin, and HV Jagadish. 2021. Identifying Insufficient Data Coverage for Ordinal Continuous-Valued Attributes. In *SIGMOD*. 129–141.

[8] Suman Banerjee, Mamata Jenamani, and Dilip Kumar Pratihar. 2020. A survey on influence maximization in a social network. *KAIS* 62, 9 (2020), 3417–3455.

[9] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning*. fairmlbook.org. http://www.fairmlbook.org.

[10] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.* 104 (2016), 671.

[11] Lori Beaman and Andrew Dillon. 2018. Diffusion of agricultural information within social networks: Evidence on gender inequalities from Mali. *Journal of Development Economics* 133 (2018), 147–161.

[12] Ruben Becker, Gianlorenzo D'Angelo, Sajjad Ghobadi, and Hugo Gilbert. 2022. Fairness in influence maximization through randomization. *Journal of Artificial Intelligence Research* 73 (2022), 1251–1283.

[13] Hannah Beilinson. 2020. *Fairness and Information Access Clustering in Social Networks*. Ph.D. Dissertation.

[14] David Bindel, Jon Kleinberg, and Sigal Oren. 2015. How bad is forming your own opinion? *Games and Economic Behavior* 92 (2015), 248–265.

[15] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. In *NIPS*. 3992–4001.

[16] Massimiliano Caramia and Paolo Dell'Olmo. 2008. *Multi-objective management in freight logistics*. Springer.

[17] Dumitru-Clementin Cercel and Stefan Trausan-Matu. 2014. Opinion propagation in online social networks: A survey. In *Proceedings of the 4th international conference on web intelligence, mining and semantics (WIMS14)*. 1–10.

[18] Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. 2009. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*. 721–730.

[19] Vineet Chaoji, Sayan Ranu, Rajeev Rastogi, and Rushi Bhatt. 2012. Recommendations to boost content spread in social networks. In *Proceedings of the 21st international conference on World Wide Web*. 529–538.

[20] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*. 1029–1038.

[21] Xi Chen, Jefrey Lijffijt, and Tijl De Bie. 2018. Quantifying and minimizing risk of conflict in social networks. In *KDD*. 1197–1205.

[22] Uthsav Chitra and Christopher Musco. 2020. Analyzing the impact of filter bubbles on social network polarization. In *WSDM*. 115–123.

[23] Michael B Cohen, Yin Tat Lee, and Zhao Song. 2021. Solving linear programs in the current matrix multiplication time. *JACM* 68, 1 (2021), 1–39.

[24] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to algorithms*. MIT press.

[25] Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *KDD*. 57–66.

[26] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *ITCS*. 214–226.

[27] David Easley and Jon Kleinberg. 2010. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge university press.

[28] Alessandro Epasto, Silvio Lattanzi, Vahab Mirrokni, Ismail Oner Sebe, Ahmed Taei, and Sunita Verma. 2015. Ego-net community mining applied to friend suggestion. *PVLDB* 9, 4 (2015), 324–335.

[29] Golnoosh Farnadi, Behrouz Babaki, and Michel Gendreau. 2020. A Unifying Framework for Fairness-Aware Influence Maximization. In *Companion Proceedings of the Web Conference 2020* (Taipei, Taiwan) (*WWW '20*). Association for Computing Machinery, New York, NY, USA, 714–722. https://doi.org/10.1145/3366424.3383555

[30] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *SIGKDD*. 259–268.

[31] He Feng and Xueming Qian. 2013. Recommendation via user's personality and social contextual. In *CIKM*. 1521–1524.

[32] Benjamin Fish, Ashkan Bashardoust, Danah Boyd, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2019. Gaps in Information Access in Social Networks?. In *WWW*. 480–490.

[33] James H Fowler and Nicholas A Christakis. 2008. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study. *Bmj* 337 (2008).

[34] Noah E Friedkin and Eugene C Johnsen. 1990. Social influence and opinions. *Journal of Mathematical Sociology* 15, 3-4 (1990), 193–206.

[35] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*. 329–338.

[36] Sainyam Galhotra, Karthikeyan Shanmugam, Prasanna Sattigeri, Kush R Varshney, Rachel Bellamy, Kuntal Dey, et al. 2022. Causal Feature Selection for Algorithmic Fairness. (2022).

[37] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2017. Reducing controversy by connecting opposing views. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 81–90.

[38] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 world wide web conference*. 913–922.

[39] Shay Gershtein, Tova Milo, Brit Youngmann, and Gal Zeevi. 2018. IM balanced: influence maximization under balance constraints. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1919–1922.

[40] Aristides Gionis, Evimaria Terzi, and Panayiotis Tsaparas. 2013. Opinion maximization in social networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 387–395.

[41] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. 2004. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*. 491–501.

[42] Pedro Guerra, Wagner Meira Jr, Claire Cardie, and Robert Kleinberg. 2013. A measure of polarization on social media networks based on community boundaries. In *AAAI*, Vol. 7. 215–224.

[43] Evren Güney. 2019. An efficient linear programming based method for the influence maximization problem in social networks. *Information Sciences* 503 (2019), 589–605.

[44] Md Hamid, Md Naser, Md Hasan, Hasan Mahmud, et al. 2014. A cohesion-based friend-recommendation system. *Social Network Analysis and Mining* 4, 1 (2014), 1–11.

[45] Anikó Hannák, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. 2017. Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr. In *CSCW*. 1914–1933.

[46] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[47] Eduardo Hargreaves, Claudio Agosti, Daniel Menasché, Giovanni Neglia, Alexandre Reiffers-Masson, and Eitan Altman. 2019. Fairness in online social network timelines: Measurements, models and mechanism design. *Performance Evaluation* 129 (2019), 15–39.

[48] Eduardo Martins Hargreaves and Daniel Sadoc Menasché. 2020. Filters for Social Media Timelines: Models, Biases, Fairness and Implications. In *Anais Estendidos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC, 153–160.

[49] Maliha Tashfia Islam, Anna Fariha, Alexandra Meliou, and Babak Salimi. 2022. Through the Data Management Lens: Experimental Analysis and Evaluation of Fair Classification. In *Proceedings of the 2022 International Conference on Management of Data*. 232–246.

[50] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2012), 1–33.

[51] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *2010 IEEE International Conference on Data Mining*. IEEE, 869–874.

[52] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *ECML PKDD*. Springer, 35–50.

[53] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *ICML*. 2564–2572.

[54] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 137–146.

[55] Moein Khajehnejad, Ahmad Asgharian Rezaei, Mahmoudreza Babaei, Jessica Hoffmann, Mahdi Jalili, and Adrian Weller. 2021. Adversarial graph embeddings for fair influence maximization over social networks. In *Proceedings of the*

Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. 4306–4312.

[56] Charlotte Laclau, Ievgen Redko, Manvi Choudhary, and Christine Largeron. 2021. All of the Fairness for Edge Prediction with Optimal Transport. In *AISTATS.* PMLR, 1774–1782.

[57] Kristina Lerman and Laurie Jones. 2006. Social browsing on flickr. *CoRR, abs/0612047* (2006).

[58] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.* 177–187.

[59] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[60] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. 2018. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1852–1872.

[61] Farzan Masrour, Tyler Wilson, Heng Yan, Pang-Ning Tan, and Abdol Esfahanian. 2020. Bursting the filter bubble: Fairness-aware network link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 841–848.

[62] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. 2017. Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery* 31, 5 (2017), 1480–1505.

[63] Rajeev Motwani and Prabhakar Raghavan. 1995. *Randomized algorithms.* Cambridge university press.

[64] Cameron Musco, Christopher Musco, and Charalampos E Tsourakakis. 2018. Minimizing polarization and disagreement in social networks. In *Proceedings of the 2018 World Wide Web Conference.* 369–378.

[65] Arvind Narayanan. 2018. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*, Vol. 1170. 3.

[66] Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish. 2021. Tailoring Data Source Distributions for Fairness-aware Data Integration. *PVLDB* 14, 11 (2021).

[67] Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish. 2022. Responsible Data Integration: Next-generation Challenges. In *Proceedings of the 2022 International Conference on Management of Data.* 2458–2464.

[68] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. 2019. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data* 2 (2019), 13.

[69] Alexandra Petri. 2020. Opinion: Let's make social media fair and balanced. The Washington Post.

[70] Aida Rahmattalabi, Shahin Jabbari, Himabindu Lakkaraju, Phebe Vayanos, Max Izenberg, Ryan Brown, Eric Rice, and Milind Tambe. 2021. Fair influence maximization: A welfare optimization approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11630–11638.

[71] Aida Rahmattalabi, Phebe Vayanos, Anthony Fulginiti, Eric Rice, Bryan Wilder, Amulya Yadav, and Milind Tambe. 2019. Exploring algorithmic fairness in robust graph covering problems. *Advances in Neural Information Processing Systems* 32 (2019).

[72] Praboda Rajapaksha, Reza Farahbakhsh, and Noël Crespi. 2017. Identifying content originator in social networks. In *GLOBECOM 2017-2017 IEEE Global Communications Conference.* IEEE, 1–6.

[73] Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.* 61–70.

[74] Maayan Roth, Assaf Ben-David, David Deutscher, Guy Flysher, Ilan Horn, Ari Leichtberg, Naty Leiser, Yossi Matias, and Ron Merom. 2010. Suggesting friends using the implicit social graph. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* 233–242.

[75] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional fairness: Causal database repair for algorithmic fairness. In *SIGMOD.* 793–810.

[76] John Scott and Peter J Carrington. 2011. *The SAGE handbook of social network analysis.* SAGE publications.

[77] Nihar B Shah and Zachary Lipton. 2020. SIGMOD 2020 Tutorial on Fairness and Bias in Peer Review and Other Sociotechnical Intelligent Systems. In *SIGMOD.* 2637–2640.

[78] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and HV Jagadish. 2022. A Survey on Techniques for Identifying and Resolving Representation Bias in Data. *CoRR, abs/2203.11852* (2022).

[79] Suraj Shetiya, Ian Swift, Abolfazl Asudeh, and Gautam Das. 2022. Fairness-Aware Range Queries for Selecting Unbiased Data. *ICDE* (2022).

[80] Nitai B Silva, Ren Tsang, George DC Cavalcanti, and Jyh Tsang. 2010. A graph-based friend recommendation system using genetic algorithm. In *IEEE congress on evolutionary computation.* IEEE, 1–7.

[81] Till Speicher, Muhammad Ali, Giridhari Venkatadri, Filipe Nunes Ribeiro, George Arvanitakis, Fabrício Benevenuto, Krishna P Gummadi, Patrick Loiseau, and Alan Mislove. 2018. Potential for discrimination in online targeted advertising. In *ACM FAccT.* PMLR, 5–19.

[82] Shani Starinsky. 2021. From Books to Facebook: How Social Media Became the Biggest Publisher of our Time. *Publishing Research Quarterly* (2021), 1–14.

[83] Julia Stoyanovich, Serge Abiteboul, Bill Howe, HV Jagadish, and Sebastian Schelter. 2022. Responsible data management. *Commun. ACM* 65, 6 (2022), 64–74.

[84] Jiankai Sun, Deepak Ajwani, Patrick K Nicholson, Alessandra Sala, and Srinivasan Parthasarathy. 2017. Breaking cycles in noisy hierarchies. In *Proceedings of the 2017 ACM on Web Science Conference.* 151–160.

[85] Shaojie Tang, Jing Yuan, Xufei Mao, Xiang-Yang Li, Wei Chen, and Guojun Dai. 2011. Relationship classification in large scale online social networks and its impact on information propagation. In *2011 Proceedings IEEE INFOCOM.* IEEE, 2291–2299.

[86] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. 2019. Group-fairness in influence maximization. *CoRR, abs/1903.00967* (2019).

[87] Zhibo Wang, Jilong Liao, Qing Cao, Hairong Qi, and Zhi Wang. 2014. Friendbook: a semantic-based friend recommendation system for social networks. *IEEE transactions on mobile computing* 14, 3 (2014), 538–551.

[88] Stanley Wasserman, Katherine Faust, et al. 1994. Social network analysis: Methods and applications. (1994).

[89] Dong Wei, Md Mouinul Islam, Baruch Schieber, and Senjuti Basu Roy. 2022. Rank Aggregation with Proportionate Fairness. In *Proceedings of the 2022 International Conference on Management of Data.* 262–275.

[90] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. 2017. Learning non-discriminatory predictors. In *Conference on Learning Theory.* PMLR, 1920–1953.

[91] Wenguo Yang, Jianmin Ma, Yi Li, Ruidong Yan, Jing Yuan, Weili Wu, and Deying Li. 2019. Marginal gains to maximize content spread in social networks. *IEEE Transactions on Computational Social Systems* 6, 3 (2019), 479–490.

[92] Zahi K Yaseen and Yussef Marwan. 2016. The influence of social media on recruitment and selection process in SMEs. *Journal of Small Business and Entrepreneurship Development* 4, 2 (2016), 21–27.

[93] Naohiro Yonemoto, Yoshitaka Kawashima, Kaori Endo, and Mitsuhiko Yamada. 2019. Gatekeeper training for suicidal behaviors: a systematic review. *Journal of affective disorders* 246 (2019), 506–514.

[94] Ying Yu, Jinglan Jia, Deying Li, and Yuqing Zhu. 2017. Fair multi-influence maximization in competitive social networks. In *WASA.* Springer, 253–265.

[95] Zhi Yu, Can Wang, Jiajun Bu, Xin Wang, Yue Wu, and Chun Chen. 2015. Friend recommendation with content spread enhancement in social networks. *Information Sciences* 309 (2015), 102–118.

[96] M. B. Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2015. Fairness constraints: Mechanisms for fair classification. *CoRR, abs/1507.05259* (2015).

[97] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International conference on machine learning.* PMLR, 325–333.

[98] Hantian Zhang, Xu Chu, Abolfazl Asudeh, and Shamkant B Navathe. 2021. OmniFair: A Declarative System for Model-Agnostic Group Fairness in Machine Learning. In *SIGMOD.* 2076–2088.

[99] Hantian Zhang, Nima Shahbazi, Xu Chu, and Abolfazl Asudeh. 2021. FairRover: explorative model building for fair and responsible machine learning. In *Proceedings of the Fifth Workshop on Data Management for End-To-End Machine Learning.* 1–10.

[100] Liwang Zhu, Qi Bao, and Zhongzhi Zhang. 2021. Minimizing Polarization and Disagreement in Social Networks via Link Recommendation. *Advances in Neural Information Processing Systems* 34 (2021).

Algorithm 5 SHORTESTDISTANCES

Input: $G(V, E)$, V_s E_c
Output: Collection of shortest distances S'

1: $\mathbf{d} \leftarrow$ MULTI-SRC-SHORTEST-DISTANCES $(G(V, E), V_s)$
2: **for** $v_i \in V \backslash V_s$ **do** $S'_i \leftarrow \{d_i\}$
3: **for** $e_j \in E_c$ **do**
4:     let $e_j = (v_\ell, v_i)$ where $d_\ell \leq d_i$
5:     **if** $d_i \leq d_\ell + 1$ **then continue**
6:     $d_{ij} \leftarrow d_\ell + 1$; add $\langle v_i, d_{ij} \rangle$ to queue
7:     **while** queue is not empty **do**
8:         remove $\langle v_i, d_{ij} \rangle$ from queue
9:         add (e_j, d_{ij}) to S'_i
10:        **for** $v_{i'} \in edges(v_i)$ **do**
11:            **if** $d_{ij} + 1 < d_{i'}$ **then**
12:                $d_{i'j} \leftarrow d_{ij} + 1$
13:                add $\langle v_{i'}, d_{i'j} \rangle$ to queue
    return $S' = \{S'_i \mid v_i \in V \backslash V_s\}$

APPENDIX

A PROOFS

LEMMA1. *The Fair Content Spread Maximization (*FAIRCS*) Problem is NP-hard.*

PROOF. To prove that FAIRCS is NP-hard, we use the reduction from the content spread problem (CSM) as in Definition 1, which is known to be NP-complete [19]. Consider an instance of the CSM problem (Definition 1), on the graph $G'(V', E')$, the set of candidate edges E'_c, the content nodes V'_s, and the value k' on the number of incident edges selected for each node $v'_i \in V'$. Also, let the decision value for content spread be T'. We generate an instance of FAIRCS problem from the CSM instance as follows:

- Let $G''(V'', E'')$ be a duplicate of $G'(V', E')$. That is, for each node $v' \in V'$, there exists $v'' \in V''$, and for each edge $(v'_i, v'_j) \in E'$, there exists an edge $(v''_i, v''_j) \in E''$. We define the graph G as the union of G' and G''. That is $G(V' \cup V'', E' \cup E'')$.
- Let $\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2\}$, where every node $v' \in V'$ belongs to $\mathbf{g}_1$ and every node $v'' \in V''$ belongs to $\mathbf{g}_2$.
- Set $V_s = V'_s \cup \{v''_i \mid v'_i \in V'_s\}$. Set $k = k'$ and $T = 2T'$.

Now the answer to the CSM problem is yes (i.e., there exists a selection of E'_p such that satisfies the content spread of T') iff the answer to the FAIRCS problem with input G, V_s, $\mathcal{G}$, k, and the decision value T is yes. □

LEMMA3. *The expected number of selected edges by Algorithm 1 per each node $v_i \in V$ is k.*

PROOF. Following the rounding step, the probability of selecting an edge $e_j \in E_c$ is $Pr(y_j^+) = y_j^*$. The expected number of edges incident to a node v_i is

$$\mathbb{E}\left[\sum_{e_j \in \mathcal{N}(v_i)} y_j^+\right] = \sum_{e_j \in \mathcal{N}(v_i)} Pr(y_j^+) = \sum_{e_j \in \mathcal{N}(v_i)} y_j^* = k$$

□

B PRACTICAL OPTIMIZATIONS

Variable Reduction: The number of variables in the LP formulation directly impacts its run time. To make our algorithm efficient we prune variables which would have no impact on the output. Let r_i be the shortest distance from content nodes to a node v_i without including any edges from E_c. In our LP formulation in Figure 2, for every length of path of $r \geq r_i$, there exists a path which satisfies variables x_{ir}. That is, $\forall r \geq r_i$, $x_{ir} = 1$. All such variables can be replaced with the constant 1, both in the objective function, as well as the fairness enforcing constraints. Additionally, for cases where the set of edges which reduce the distance to a node v_i to at most r is empty, there is no edge selection for which x_{ir} is > 0. In such cases, we replace x_{ir} by a constant zero, removing the corresponding unnecessary variables. In the friend of friend edge suggestion model [95], in particular, this reduces the order of the variables to consider. Since an edge in this model can only improve the distance by one, there is only one x_{ir} variable for each node for which we are unsure of the final value. Consequently the number of variables in LP reduces from $O(r_m |V| + |E_c|)$ down to $O(|V| + |E_c|)$.

Computing the shortest paths: Computing the shortest paths of an entire graph is relatively inefficient, taking $\Theta(|V| + |E|)$ time. Our algorithm needs to compute the effect of each edge e_j in E_c on the graph when added to the graph independently. However, when only one or more edges are added to a graph, there is a deterministic subset of nodes $V' \subset V$ for which the shortest distances improve. Algorithm 5 calculates only the changes in the distance for the nodes in V'. The basic idea is that the search starts at the new edge or edges, and performs a search, not following an edge if traversing over it does not improve the distance to the destination. If the union of all the outbound edges of all nodes in V' is called $E_{V'}$, then this algorithm runs in $\Theta(|V'| + |E_{V'}|)$. For the clear theoretical improvement, we used this algorithm in our implementation.

C EXTENSIVE EXPERIMENT RESULTS

LP-SCALE, **choice of hyper-parameter**: In this experiment we examine the choice of Nodes per Iteration (NpI) on the LP-SCALE algorithm. In each experiment, we ran LP-SCALE on 10,000 nodes, with 5 content nodes, $p = 0.5$, and observed over different values of k from 2-5. We vary NpI between 200, 400, 600, and 800. For each experiment we run 5 trials and report the average. The initial average disparity across these 5 trials was $\sim 7.84\%$.

As demonstrated in Figure 15, as the NpI increases, the trend is that LP-SCALE achieves a lower disparity. Equivalently, as demonstrated in Figure 16, when the NpI increases, LP-SCALE also produces a higher lift. It is thus obvious that higher NPI results in a more correct output. To balance this, as shown in 17, as the NpI increases, so does the time required to produce a selection. This demonstrates that the increased lift and fairness the edge selection for higher NpI comes at a cost of increased runtime.

Candidate Edges	k	Initial Disparity	LP - Advanced			Continuous Greedy			IRFA			SpGreedy			ACR - FoF		
			Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time
FoF	2	34.2±0.34	**66.08±10.76**	**0.14±0.51**	9.9±3.09	45.71±9.04	25.71±9.34	28.83±8.47	15.15±6.4	28.9±6.81	2.62±1.07	11.73±3.29	24.11±5.15	29.88±8.38	19.65±8.78	28.14±6.53	**1.57±0.81**
	3	34.2±0.34	**84.04±13.64**	**0.08±0.21**	13.94±15.24	51.68±8.54	25.2±6.36	29.01±8.13	23.03±7.4	29.92±9.21	3.88±1.3	20.11±9.03	21.08±6.1	40.73±9.68	27.06±9.0	24.98±7.13	**1.77±0.91**
	4	34.2±0.34	**94.23±15.54**	**0.33±0.9**	11.36±4.08	56.41±8.87	22.81±5.83	32.37±8.15	32.2±10.31	27.21±11.12	4.9±2.55	25.82±9.66	19.41±6.91	53.07±9.45	33.59±9.47	23.5 2±8.13	**1.86±1.02**
	5	34.2±0.34	**105.28±18.91**	**1.68±3.66**	13.28±7.07	60.12±7.92	24.26±7.27	34.2±10.36	37.96±10.26	25.07±11.33	6.81±2.65	32.17±11.14	17.13±7.8	56.92±13.45	40.53±10.65	22.54±7.69	**2.05±1.16**
IGC	2	34.2±0.34	**78.69±17.11**	**0.77±2.04**	14.23±4.64	54.1±13.78	22.17±10.03	31.4±11.12	19.51±7.3	30.1±6.8	2.1±0.93	16.95±10.81	23.11±7.53	35.37±9.83	13.91±4.66	29.54 ±5.4	**2.04±0.98**
	3	34.2±0.34	**94.26±21.55**	**0.55±1.46**	15.65±5.94	60.92±13.1	23.51±11.15	32.7±11.51	27.23±10.76	28.21±8.44	2.83±1.07	23.62±11.58	19.89±7.09	50.76±13.59	20.04±6.29	26.28±7.18	**2.5±1.35**
	4	34.2±0.34	**108.58±22.96**	**0.52±1.04**	18.23±6.74	67.73±14.55	21.2±9.8	35.37±10.47	27.07±8.12	26.34±9.41	3.91±1.11	31.76±15.67	17.71±7.79	62.24±8.7	27.55±9.3	25. 27±8.49	**2.17±0.62**
	5	34.2±0.34	**116.99±23.29**	**0.81±1.75**	16.06±3.83	72.8±14.04	20.84±12.02	33.46±9.66	34.83±12.1	24.85±8.42	5.73±1.27	37.0±13.79	16.19±6.47	87.31±17.68	33.3±10.32	22.91±7.73	**2.3±0.95**

Table 5: (Best results bold) Comparing Lift (in %), Disparity (in %), and Runtime (in seconds), on 500 node Antelope Valley data sets, of LP-Advanced and all baselines, varying k. 20 trials are used at each setting. Both Friend of Friend and Intersecting Group Count are reported. $p = 0.5$, $|V_s| = 3$

Candidate Edges	k	Initial Disparity	LP - Advanced			Continuous Greedy			IRFA			SpGreedy			ACR - FoF		
			Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time
FoF	2	21.52±1.2	**84.6±31.66**	**0.4±1.3**	71.0±15.51	46.81±16.56	14.81±3.8	56.53±16.14	23.4±13.69	17.16±3.66	**6.02±1.95**	21.05±10.86	12.81±3.17	210.27±63.42	27.41±23.96	1 7.31±4.17	21.09±18.5
	3	21.52±1.2	**100.4±40.76**	**0.23±0.65**	74.85±22.83	51.97±16.02	14.7±3.18	52.77±12.37	33.18±20.1	17.05±4.31	**8.08±3.0**	30.78±12.77	11.03±2.93	311.1±90.71	36.79±29.53	16.19±3.87	20.97±17.07
	4	21.52±1.2	**109.93±38.03**	**0.54±1.24**	72.04±21.03	56.23±15.54	15.37±3.5	56.59±16.73	39.83±21.99	16.29±4.23	**13.31±6.23**	37.67±20.37	10.28±2.78	400.14±121.0	43.93±32.43	15.29±3.65	21.41±14.44
	5	21.52±1.2	**123.06±44.94**	**0.49±1.07**	66.04±17.58	58.71±15.22	14.53±3.18	52.47±11.13	47.65±23.39	15.26±2.88	**14.09±5.63**	49.91±28.26	9.51±2.34	489.08±166.75	53.84±41.19	14.6±3.88	21.64±19.15
IGC	2	21.52±1.2	**94.23±57.62**	**2.52±5.01**	95.87±41.96	61.88±36.96	12.93±5.29	49.06±18.41	24.96±16.3	17.98±3.39	**5.28±2.33**	22.6±19.52	13.74±4.73	198.21±91.44	19.69±18.18	18.2±4.6	33.63±31.67
	3	21.52±1.2	**109.88±60.69**	**1.14±2.19**	141.6±61.67	66.32±41.25	13.16±3.58	67.38±23.32	30.32±17.49	17.04±4.23	**8.64±3.57**	38.56±37.75	11.75±4.9	377.39±113.57	23.65±16.25	17.55±3.44	40.32±33.28
	4	21.52±1.2	**121.99±64.27**	**1.35±2.35**	137.22±76.26	73.2±45.57	13.01±3.68	66.66±20.64	36.55±19.95	14.92±4.3	**14.37±5.74**	48.81±46.7	11.18±4.59	490.52±134.02	32.0±19.68	15.81±4.2	37.7±32.0
	5	21.52±1.2	**133.82±72.74**	**0.88±1.91**	133.89±71.11	72.9±44.3	13.21±3.77	66.84±20.57	43.76±28.86	16.17±4.68	**15.43±4.31**	56.26±47.34	10.09±5.01	589.39±175.17	40.94±29.59	15.57±3.91	44.88±40.27

Table 6: Comparing Lift (in %), Disparity (in %), and Runtime (in seconds), on 1000 node Pokec samples, of LP-Advanced and all baselines, varying k. 20 trials are used at each setting. Both Friend of Friend and Intersecting Group Count are reported. $p = 0.5$, $|V_s| = 3$

Candidate Edges	p	Initial Disparity	LP - Advanced			Continuous Greedy			IRFA			SpGreedy			ACR - FoF		
			Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time
FoF	0.3	71.83±23.64	**87.46±14.25**	**1.81±4.97**	12.71±5.56	60.41±10.65	46.4±16.75	31.68±9.88	28.87±7.05	56.29±21.74	3.96±1.35	21.47±6.74	44.72±13.83	43.04±10.62	25.95±9.29	54.46±22.14	**1.88±0.91**
	0.5	34.2±0.34	**80.25±12.1**	**0.42±1.0**	15.57±17.33	51.94±8.82	25.58±4.99	29.44±7.96	25.48±10.53	26.92±8.62	3.55±1.59	20.11±9.03	21.08±6.1	42.01±9.7	27.06±9.0	24.98±7.13	**2.18±0.95**
	0.7	14.62±3.93	**50.37±9.14**	**0.08±0.21**	9.95±4.9	29.13±4.37	11.52±4.13	32.32±9.24	16.27±5.24	13.11±6.05	3.84±1.53	9.26±4.06	36.4±8.88	19.64±5.85	10.45±4.09	**1.51±0.69**	
	0.9	4.07±2.09	**15.1±2.76**	**0.48±0.85**	14.84±16.96	8.7±1.08	3.66±2.1	25.53±11.39	6. 11±2.13	3.41±2.18	3.39±1.47	4.01±1.68	2.84±1.84	28.07±9.99	8.34±2.21	2.82±1.1 9	**1.09±0.82**
IGC	0.3	71.83±23.64	**86.45±17.63**	**2.18±4.88**	14.17±3.49	68.84±13.63	41.86±16.89	37.06±10.2	31.18±13.43	54.57±24.55	2.99±1.01	26.65±11.2	41.46±14.73	54.23±11.65	20.83±7.33	57.12±21.12	**2.57±0.99**
	0.5	34.2±0.34	**83.69±13.9**	**0.39±1.0**	16.69±20.23	60.19±10.61	20.44±8.15	29.03±7.98	26.15±10.93	30.29±8.95	2.91±0.7	23.62±11.58	19.89±7.09	52.18±10.99	20.04±6.29	26.28±7.18	**2.25±0.67**
	0.7	14.62±3.93	**49.24±7.84**	**0.56±1.5**	10.77±2.95	35.79±7.26	10.4±5.15	35.62±9.39	13.58±3.14	12.53±4.43	3.91±1.01	15.33±8.89	8.38±3.9	51.63±9.15	13.84±4.33	10.95±4.06	**1.98±0.69**
	0.9	4.07±2.09	**15.21±2.56**	**0.34±0.77**	24.08±29.55	11.07±2.1	3.33±2.05	29.7±8.09	4 .76±2.15	3.48±2.06	2.81±1.01	4.57±2.0	3.01±2.03	47.51±9.84	5.72±2.04	3.01±1.6 6	**2.38±0.99**

Table 7: Comparing Lift (in %), Disparity (in %), and Runtime (in seconds), on 500 node Antelope Valley data sets, of LP-Advanced and all baselines, varying p. 20 trials are used at each setting. Both Friend of Friend and Intersecting Group Count are reported. $k = 3$, $|V_s| = 3$

Candidate Edges	p	Initial Disparity	LP - Advanced			Continuous Greedy			IRFA			SpGreedy			ACR - FoF		
			Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time	Lift	Disparity	Time
FoF	0.3	62.92±25.82	**93.55±36.44**	**9.62±23.26**	70.64±22.78	61.65±17.82	37.91±17.06	48.45±9.44	40.53±18.3	41.84±17.59	**8.44±3.57**	31.88±13.2	35.46±20.56	283.91±112.87	33.8±25.25	46.94±15.32	21.58±19.6
	0.5	21.52±1.2	**97.67±34.85**	**0.13±0.53**	68.58±20.51	53.58±16.69	14.86±3.84	54.17±9.07	33.16±17.94	18.09±4.0	**9.46±4.31**	30.78±12.77	11.03±2.93	293.93±103.49	36.79±29.53	16.19±3.87	23.0±20.99
	0.7	8.43±2.46	**58.03±20.8**	**0.08±0.34**	74.43±32.97	29.24±7.95	6.17±2.21	49.63±8.95	20.31±11.63	6.59±2.21	**9.55±3.46**	21.07±13.59	4.58±1.37	283.88±98.09	23.08±17.19	6 .46±2.79	18.93±15.25
	0.9	2.16±1.03	**16.51±4.75**	**0.06±0.18**	47.52±15.84	8.5±1.8	1.71±0.85	35.08±10.8	5. 25±2.31	1.75±0.91	**5.65±2.54**	6.03±2.38	1.26±0.56	185.19±59.9	7.36±4.86	1.69±0. 94	13.16±10.99
IGC	0.3	62.92±25.82	**96.05±37.42**	**9.85±23.4**	70.01±19.33	72.25±32.56	33.4±12.95	62.88±18.65	38.81±19.8	43.53±19.17	**8.68±3.06**	36.62±27.76	32.47±16.73	351.33±132.05	24.82±17.68	48.78±20.72	41.8±40.85
	0.5	21.52±1.2	**100.23±40.08**	**0.13±0.55**	73.25±17.7	65.76±41.34	13.62±4.49	64.79±16.75	31.56±21.74	16.58±3.62	**9.89±3.73**	38.56±37.75	11.75±4.9	359.06±120.53	23.65±16.25	17.55±3.44	39.87±35.98
	0.7	8.43±2.46	**57.62±20.63**	**0.06±0.21**	74.58±28.19	37.52±20.75	5.57±2.37	58.01±17.58	19.43±11.0	6.69±2.64	**9.3±3.25**	22.75±21.01	5.07±2.85	332.06±130.84	14.55±9.57	7.17±2.63	39.41±32.5
	0.9	2.16±1.03	**16.59±4.79**	**0.09±0.2**	42.45±12.17	10.52±4.75	1.66±0.84	37.92±14.07	6.03±2.53	1.7±0.89	**5.65±2.62**	6.9±4.65	1.45±0.88	197.04±79.45	4.66±2.68	1.89± 0.96	21.83±19.2

Table 8: Comparing Lift (in %), Disparity (in %), and Runtime (in seconds), on 1000 node Pokec samples, of LP-Advanced and all baselines, varying p. 20 trials are used at each setting. Both Friend of Friend and Intersecting Group Count are reported. $k = 3$, $|V_s| = 3$
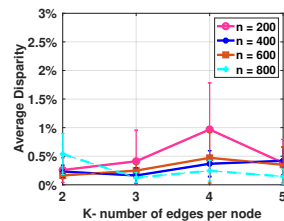

Figure 15: Disparity of LP-SCALE **for diff. # nodes per iteration**

Figure 16: The lift of the LP-SCALE **for diff. # nodes per iteration**
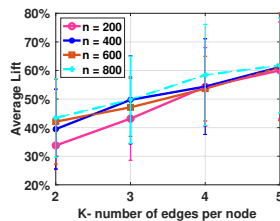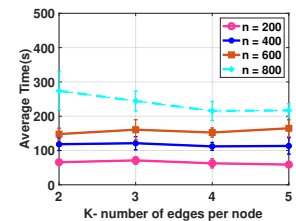

Figure 17: The runtime of the LP-SCALE **for diff. # nodes per iteration**