Maximizing Fair Content Spread via Edge Suggestion in Social Networks

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

Sana Ebrahimi  
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%.

PVldb Reference Format:
Ian P. Swift, Sana Ebrahimi, Azade Nova, and Abolfazl Asudeh. Maximizing Fair Content Spread via Edge Suggestion in Social Networks. PVLDB, 15(11): 2692 - 2705, 2022. doi: 10.14778/3551793.3551824

PVldb Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/UIC-InDeXLab/FCS.

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, 76], algorithm or model modification [52, 97, 98], and output post-processing techniques [46, 51, 91] 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, 29, 69, 70, 87, 95]. While similar issues have been reported for cases such as biased advertisement over social networks [82], fairness in content spread becomes critical when related to health-care [94], news and valuable information spread [83], job opportunities [93], 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) [29, 70, 87, 95], 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.

In this paper, 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 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, 92, 96], 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 suggesting edges to achieve a new objective: fairness in content spread.

Existing social networks often use sophisticated algorithms for friendship suggestions. Re-engineering the existing algorithms are costly and perhaps impractical. Instead, we propose a fair content...
spread component that works with any possible friendship suggestion method, taking their output as the set of potential edges and selecting a subset for the suggestion to achieve fairness.

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 to work to consider fairness in this context. Unfortunately, not only is the problem NPhard, but also impossible to approximate in P time unless \( P = NP \). By 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 fairness of its output. We are the first Linear Program designed for content spread with approximation guarantees on fairness. We propose several optimizations that further help our algorithm be efficient in practice. Having to solve an LP, our original algorithm lacks to scale to very large settings. To resolve this issue, we design a scalable algorithm based on dynamically increasing the nodes via sampling. 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.

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, 92, 96, 101] in content spread. We also observed our algorithm achieves comparable results to the Optimal Brute Force method on very small graphs. Our scalable algorithm could scale to within 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. 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 near-zero unfairness of 0.0132, and also had a 400x 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:

- (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 experiments on real data sets. Our experiments verify the effectiveness of our algorithms and 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 a 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, 77, 89]. 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. 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 relationships 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" [96], where only users that are at most within a two-hop distance from a user are candidate for friend suggestion. The intuition behind this model is that users who do not share a friend are unlikely to be friends. 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 [88]. Our work only expects the existence of a candidate edge 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 [86], 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 originality [2, 72], or targetted nodes for optimizations such as influence maximization [54, 86]. In this paper, we assume that, by any of the prior work, content source nodes are selected as a subset of \( V_c \subset V \).

\(^1\)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 of candidate edges

The social network with nodes \( V \) and edges \( E \)

\( E_c \)

\( V_c \)

Content source nodes

\( E_p \)

Suggested candidate edges \( \subseteq E_c \)

\( k \)

Number of edges per node to be suggested

\( g \)

The set of groups

\( g_i \)

The \( i \)-th group in \( g \)

\( N(v_i) \)

Candidate edges \( v_i \in E_c \) incident to the node \( v_i \in V \)

\( p \)

Edge activation probability

\( \varepsilon \)

Content Spread Disparity (Unfairness) – Equation 5

### 2.2 Cascade Model

In social networks, content spread follows a cascade model. Each neighbor recursively shares received info within its own neighborhood, and content propagates through the network to a wider 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 \( v_i \) 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 propagation using IC is known to be \#\text{P}-hard, and accurately estimating the content spread requires a combinatorial large time [19, 20]. Thus, 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 an 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 \( v_i \in V_s \) and a node \( v_j \in V \setminus V_s \), consider a path from \( v_i \) to \( v_j \) such that for every \( e \in E \cup E_p \), the probability of content spread from a source to a destination along the path \( q \) is \( P_r(q) = \prod_{e \in q} P_e \). Let \( \mathcal{Q}_i \) be the set of paths from a content node to the node \( v_i \in V \setminus V_s \). Following MIP, the probability of content reaching \( v_j \) is equal to the probability that the node was reached along the path in \( \mathcal{Q}_i \) with the highest probability. That is, \( \forall v_i \in V \setminus V_s \),

\[
F_{MIP}(v_i, E_p) = \max_{q \in \mathcal{Q}_i} P_r(q)
\]

**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, \( \mathcal{Q}_i \) is the set of paths from a content node to the node \( v_i \in V \setminus V_s \), such that \( \forall q \in \mathcal{Q}_i \) 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

---

\( ^2 \)Unless mentioned otherwise, we simplify the notation \( F_{RMPP}(.) \) to \( F(.) \).

---

\( ^3 \)The proofs of Lemma 1 and Lemma 3 are included in the appendix.
The expected content received by the two groups should be equal. That is, 
\[ \mathbb{E}[CS_{E}] = \mathbb{E}[CS_{E}]. \]
This happens when only the edges from a vertex-cover of size \( k \) to \( v_{i} \) are selected. In this situation, the expected content received by \( v_{1}, \ldots, v_{n} \) is:

\[
\mathbb{E}[CS_{E}] = \sum_{k=1}^{k} F(v_{i}, E_{p}) + \sum_{i=k+1}^{n} F(v_{i}, E_{p})
= k p + (n - k) p^{2}
\]  

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 \( v_{0} \). This reduces the expected content received by \( g_{2} \) by at least \( p^{2} - p^{3} \).

To resolve this reduction, the valid solution requires to select more than \( k \) edges connected to \( v_{0} \), which contradicts 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 \( P = NP \).

Next, we provide an integer programming formulation for solving the problem exactly. This formulation, though not practical, helps us in designing our solution in § 3.1.

**Warm-up Integer Programming:** The optimal solution to FairCS can be obtained by formulating it as an integer program (IP):

\[
\begin{align*}
\text{maximize} & \quad \sum_{i \in V} x_{i} \\
\text{subject to} & \quad \forall v_{i} \in V, x_{i} \leq \max_{e_{j} \in E_{p}} (y_{j} p_{ij}) \\
& \quad \forall v_{i} \in V, \sum_{e_{j} \in G} y_{j} \leq k \\
& \quad \forall g_{a}, g_{b} \in G, \frac{1}{|g_{a} \cap g_{b}|} \sum_{u \in g_{a}} x_{u} = \frac{1}{|g_{b}|} \sum_{u \in g_{b}} x_{u} \\
& \quad \forall v_{i} \in V, 0 \leq x_{i} \leq 1; \forall e_{j} \in E_{c}, y_{j} \in \{0, 1\}
\end{align*}
\]

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 1 if the edge \( e_{j} \in E_{c} \) is added to \( E_{c} \), 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 \( E_{c} \). 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 \( k \) (\( N(v_{i}) \) 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 size enforce fairness.

Besides the fact the IP is NP-complete, 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. To find an efficient solution for the problem, we allow approximation on 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.](image-url)
Using the above transformation, the objective function can be rewritten as $V_{shortest distance from a content node}$. Furthermore, let $\text{IP}$ gets activated during the cascade process. Then, the probability that content receiving at node $i$ from content nodes in graph $G$, is $\frac{1}{|G|} \sum_{r \in \Gamma_i} x_{ir} = \frac{1}{|G|} \sum_{r \in \Gamma_i} x_{ir} = \frac{1}{|G|} \sum_{r \in \Gamma_i} x_{ir}$. The bottle-neck in the time complexity of Algorithm 1 is solving the LP solver; in step 2 it is $\text{FairCS}$ used for solving $\text{FairCS}$ problem.

and relaxation of the fairness constraint. We define a $(1 + \varepsilon)$-approximation ($\varepsilon \geq 0$) on fairness, as following:

$$\forall g, g' \in G : (1 + \varepsilon)^{-1} \leq \frac{1}{|G|} \sum_{v \in G} F(v, E_p) \leq (1 + \varepsilon)$$

Note that $\varepsilon = 0$ refers to the maximum fairness, and it decreases as $\varepsilon$ increases. Conversely, the disparity of content spread is $\varepsilon$. In the rest of the paper and in our experiments, we refer to $\varepsilon$ 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 LP. 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 $i$, by the weighted sum of a collection of binary variables $x_{ir}$, which is if there exists a path of length at most $r$ from a content node to $i$, and 0 otherwise. Let $p_r$ be the probability that an edge gets activated during the cascade process. Then, the probability that content reaches a node $i$ using a path of length $r$ is $p_r^r$. Given a node $i$, let $r$ 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, $r \geq r_r$, $x_{ir} = 1$, while $r < r_r$, $x_{ir} = 0$. We define the values $\delta_r$, such that $x_{i} = \sum_{r=1}^{r_m} \delta_r x_{ir}$. We shall explain the derivation of $\delta_r$ values in 3.3. Using the above transformation, the objective function can be rewritten as $\sum_{v \in V} S_t = \sum_{r=1}^{r_m} \delta_r \sum_{v \in V} x_{ir}$. 

Next, for every node $i$, and a distance $r$, we precompute the set $S_{ir}$, as the set of edges that if added to $E_p$, the distance to $i$ 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 $y_i$ is a binary attribute that is 1 when the edge $e_i \in E_p$ gets selected. Similarly, $x_{ir}$ is 1 only if there exists a path of length $r$ to $i$ 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 $\ast = \{x_{ir}^* \mid V_i \in V, r \in [1, r_m]\}$ and $\ast = \{y_i^* \mid e_i \in E_c\}$ 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 $y^* = \{y_i^* \mid e_i \in E_p\}$ into binary values $y^* = \{y_i^* \mid e_i \in E_c\}$. An edge $e_i \in E_c$ is selected for suggestion if $y_i^* = 1$. We follow a randomized rounding for rounding $y^*$ to $y^*$, using $y^*$ as the probability to select edges for suggestion. That is, the probability to suggest edge $e_i \in E_p$ is $P(y_i^* = 1) = y_i^*$. To do so, Algorithm 2 draws a number uniformly at random from the range $[0, 1]$. If the generated sample is less than $y_i^*$, $e_i$ is added to $E_p$, i.e., $y_i^* = 1$. 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 $O(|E_p|)$). Since the rounding is randomized, it is reasonable to repeat the rounding step many times to boost the success probability. Let $\text{iter}_{\text{rm}}$ be the number of times the rounding is repeated and let $E_{P_1}, \cdots, E_{P_{\text{iter}_{\text{rm}}}}$ 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 one found, and returns the one that maximizes the content spread (Lines 5-10 of Algorithm 2).

### 3.3 Constructing LP Inputs

For every node $i$, and a distance $r$, we need to precompute the set $S_{ir}$, of edges in $E_c$, that makes $i$ at most $r$ hops away from a content node. Again, under RMMR, a path from a content node to $i$ can contain at most one edge from $E_c$. Hence, to compute the sets $S = \{S_{ir} \mid i \in V \setminus V_s, r \leq r_m\}$, we compute the shortest distance from any content node to every node in the graph with edge $e_i \in E_p$. That is, $V_i e_i E_p$, we compute all-pair shortest paths from content nodes in graph $G(V, E_\cup \{e_i\})$. Suppose $d_{ij}$ is the shortest path length from content nodes to a node $i$, in graph $G(V, E \cup \{e_i\})$. Then $e_i$ is
3.4 Approximation Analysis

Lemma 3. The expected number of selected edges by Algorithm 1 per each node \( \nu_i \in V \) 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 \(\delta c_r\) on fairness.

Proof. Let \( \{ x_{ir}^* \mid \nu_i \in V, r \in [1, r_m]\} \) and \( \{ y_{ir}^* \mid e_i \in E_c \} \) be the optimal values of LP, shown in Figure 2. Besides, let \( \{ x_{ir} \mid \nu_i \in V, r \in [1, r_m]\} \) and \( \{ y_{ir} \mid e_i \in E_c \} \) be the approximation values for the FAIRCS problem. Let \( v \) be the approximation ratio on the content spread.

\[
\mathbb{E} \left[ x_{ir}^* \right] = \Pr \left( x_{ir}^* = 1 \right) = 1 - \Pr \left( x_{ir}^* = 0 \right) = 1 - \prod_{e_i \in S_{ir}} (1 - y_{ir}^*) \geq 1 - e^{-\sum_{e_i \in S_{ir}} y_{ir}^*} \geq 1 - e^{-v} x_{ir}^* \geq (1 - e^{-1}) x_{ir}^* \tag{7}
\]

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

\[
\mathbb{E}[APR] = \mathbb{E} \left[ \sum_{r=1}^{r_m} \sum_{\nu_i \in V} x_{ir}^* \right] = \sum_{r=1}^{r_m} \sum_{\nu_i \in V} \mathbb{E} \left[ x_{ir}^* \right] \\
\geq \sum_{r=1}^{r_m} \sum_{\nu_i \in V} (1 - e^{-1}) x_{ir}^* = (1 - e^{-1}) \sum_{r=1}^{r_m} \sum_{\nu_i \in V} x_{ir}^* \tag{8}
\]

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

\[
\begin{align*}
\sum_{r=1}^{r_m} \sum_{\nu_i \in V} x_{ir}^* &\geq \text{OPT} \\
\mathbb{E}[APR] &\geq (1 - e^{-1}) \text{OPT}
\end{align*}
\]

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

Looking at Figure 2, if \( x_{ir}^* = 1 \), then \( x_{ir}^* \) is the optimal content spread. Since the optimal solution for OPTLP ratio of relaxation algorithm (Algorithm 1) for solving it satisfies the approximation ratio of \((1 - e^{-1})\), let \( \nu_i \) be a shortest path of length \( r \) to \( \nu_i \). Then \( S_{ir}^\nu = \nu_i \).

Proof. For cases where \( r = r_m \), the probability that a node \( \nu_i \) receives content can be written as

\[
\sum_{r=1}^{r_m} \delta_{S} \delta_{S_{ir}} = \delta_{S_{m-1}}. \quad \text{Hence, } \delta_{S_{m-1}} = p^{r_m}.
\]

Similarly, \( \delta_{S} \delta_{S_{ir}} \) can be written as

\[
\sum_{r=1}^{r_m} \delta_{S} \delta_{S_{ir}} = \sum_{r=1}^{r_m} \delta_{S} + \sum_{r=1}^{r_m} \delta_{S_{ir}} = p^d.
\]

In summary,

\[
\delta_{S} = \begin{cases} p^{r_m} & r = r_m \\ p^d - p^{r+1} & r < r_m \end{cases}
\]

\[
\mathbb{E}[APR] \geq (1 - e^{-1}) \mathbb{E} \left[ x_{ir}^* \right] \geq (1 - e^{-1}) \text{OPT}
\]

We now use the lower-bound found in Equation 9 and the upper-bound in Equation 12, for every group \( g_a, g_b \in G \).

From the fairness constraint in the LP formulation (the third constraints Figure 2), we know that

\[
\sum_{r=1}^{r_m} \sum_{\nu_i \in V} x_{ir}^* = \sum_{r=1}^{r_m} \sum_{\nu_i \in V} x_{ir}^* \\
\leq \sum_{r=1}^{r_m} \sum_{\nu_i \in V} 2e x_{ir}^* = 2e \sum_{r=1}^{r_m} \sum_{\nu_i \in V} x_{ir}^* \tag{12}
\]

Using the upper-bound and the lower-bound in Equations 9 and 12, for every group \( g_a, g_b \in G \), we know:

\[
\frac{1}{|g_a|} \sum_{r=1}^{r_m} \sum_{\nu_i \in g_a} x_{ir}^* \geq (1 - e^{-1}) \frac{1}{|g_a|} \sum_{r=1}^{r_m} \sum_{\nu_i \in g_a} x_{ir}^* \\
\text{and} \quad \frac{1}{|g_a|} \sum_{r=1}^{r_m} \sum_{\nu_i \in g_a} x_{ir}^* \leq 2e \frac{1}{|g_a|} \sum_{r=1}^{r_m} \sum_{\nu_i \in g_a} x_{ir}^* \\
\]

As a result, for all \( g_a, g_b \in G \):

\[
\frac{1 - e^{-1}}{2e} \leq \frac{1}{|g_b|} \sum_{r=1}^{r_m} \sum_{\nu_i \in g_b} x_{ir}^* \leq \frac{2e}{1 - e^{-1}}
\]
4 PRACTICAL OPTIMIZATIONS

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 excluding $e_j$, then if $e_j$ is in the chosen set $E_p$, including $e'_j$ does not change the maximum probability of these paths. 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 $E_c$. 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, finds a solution relatively efficiently with theoretical guarantees on accuracy. 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 $E ← E ∪ E_p$ and $E_c ← E_c \setminus E_p$. By moving $E_p$ from $E_c$ to $E$, in the next iteration $e'_j$ can provide additional content to nodes, even under RMPP. To maintain k expected incident edges in this modification, we only find $k'_j$ edges for a node $v_i$, determined by the number of incident edges to $v_i$ 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. As we demonstrate in our experiments in Section 6, our LP-relaxation algorithm is not scalable for such settings. 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 is based on the observation that the contribution of a path to the content spread exponentially reduces as its length increases, and the nodes that are far from the content nodes have a minimal impact in the content spread. 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. 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.
We now propose algorithm 

Algorithm 4

Algorithm 4: Scalable Algorithm (LP-SCALE)

| Input: | $G(V, E)$, $V_s$, $E_c$, $k$, $G$; Iter. | Output: | suggested edges $E_p$ |
|---|---|---|---|
| 1: | excluded← {}; $E_p$ ← {}; $y_{w,e} : k'_i ← k$ |
| 2: | while ($G' (V', E') ← ForestFireSample (G(V, E), V_s, Iter)$) $\neq G$ do |
| 3: | add $E_p$ to $G' (V', E')$ |
| 4: | $E'_c ← E_p$ on $G'$ ← excluded |
| 5: | add $E'_c$ to excluded |
| 6: | $S' ←$ ShortestDistances($G' (V', E'), E_c$) // Algorithm 5 in Appendix |
| 7: | $(x^*, y^*) ←$ Solve the LP in Figure 2 |
| 8: | $\text{add ROUND (} y^*, G' (V', E'), k', V_s, g\text{)}$ to $E'_p$ // Algorithm 2 |
| 9: | for $v_i \in V$ do $k'_i ← \max(k - ||e_{u,v} \in E_p|u = i|), 0$ |
| 10: | return $E_p$ |

impact on content spread earlier on, while making less meaningful decisions later.

5.1 LP-SCALE Algorithm

We now propose algorithm LP-SCALE (Algorithm 4) for solving the FAIRCS at scale. The algorithm uses 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. 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 $k'$ also get updated accordingly. 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 $E'_c$, using the increased information of the new nodes in $V''$. 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 with low disparity and high content spread.

6 EXPERIMENTS

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, using [75]). 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 ($\varepsilon$ in Equation 5) as the % difference between the two groups’ size weighted content spread, or $\varepsilon \times 100$.

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, our algorithms work (almost) equally well for other cascade models. Confirming that we observed consistent results for RMPP, MIP, and IC, we report our evaluations using MIP to demonstrate practical performance of our work beyond RMPP. The change in Content Spread is reported as the percent increase in MIP Content Spread:

$$\sum_{v_i \in V} \frac{F_{MIP}(v_i, E_p) - F_{MIP}(v_i, \{\})}{F_{MIP}(v_i, \{\})} \times 100 \quad (14)$$

Baselines:

- **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 included in the appendix.
- **IRFA** [92]. 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 pre-processing step, we used [85] to remove cycles from the graph.
- **SpGreedy** [101]. Measures the relationship between demographic groups via the opinion model properties polarization (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** [96]. 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:

- **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 $v_i$. 
6.1 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 7), 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. Algorithm 3 is run both with and without the fairness constraint, with \( k = 5, p = 0.5, |V| = 500 \), an initial disparity between 30% and 35%, and with 20 trials across each of a varying \( |V_n| \). The resulting average lift and average disparity are recorded. The algorithm without the fairness constraint achieves 1.25x on lift. 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 demonstrated as a loss in overall lift for the result of near zero disparity 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. 7 graphs with \( |V| = 40 \) are sampled from the Antelope Valley dataset, such that Brute Force can achieve 0 disparity through edge selection , using 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 graphs and the average and standard deviation were reported and compared against the results of 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 \( |V_n| \). For each setting, 20 trials were run, with \( k = 3, p = 0.5 \) and \( |V_n| = 3 \) 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 Continuous 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 \( k = 3, p = 0.5 \) for each setting, and \( |V_n| = 5 \) for sizes up to and including 10k nodes, while \( |V_n| = 10 \) 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 included 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 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
Comparing Lift (%), Disparity (%), and Runtime (sec), on 500 node Antelope Valley data sets, of LP-Advanced and all baselines, varying |V_k|, using 20 trials per setting. Both FoF and IGC were considered for candidate edge selection.

| Candidate | | v | V_k | Initial | LP - Advanced (ours) | Continuous Greedy | IRFA | SpGreedy | ACR - FoF |
|-----------|---|---|---|---|---|---|---|---|---|---|
| FoF | 3 | 34.4±0.3 | 82.3±1.4 | 0.3±0.5 | 3.5±0.9 | 51.6±4.2 | 25.8±3.6 | 30.4±6.9 | 29±9.1 | 29.2±8.6 | 8.3±1.8 | 20.1±9.0 | 21.1±6.1 | 11.1±10.7 | 27.1±9.0 | 25.4±7.1 | 1.6±0.7 |
| | 6 | 34.4±0.3 | 70.9±1.3 | 0.2±0.8 | 10.8±3.5 | 49.6±8.1 | 22.7±6.9 | 26.6±4.0 | 22.7±8.8 | 26.1±6.9 | 6.8±1.9 | 17.8±5.4 | 21.5±3.0 | 33.2±6.0 | 23.4±3.3 | 25.0±3.4 | 1.9±0.8 |
| | 9 | 33.8±0.0 | 61.9±6.2 | 0.2±0.4 | 10.8±3.5 | 49.6±8.1 | 22.7±6.9 | 26.6±4.0 | 22.7±8.8 | 26.1±6.9 | 6.8±1.9 | 17.8±5.4 | 21.5±3.0 | 33.2±6.0 | 23.4±3.3 | 25.0±3.4 | 1.9±0.8 |
| | 12 | 33.8±0.0 | 53.4±7.9 | 0.6±1.2 | 10.8±3.5 | 49.6±8.1 | 22.7±6.9 | 26.6±4.0 | 22.7±8.8 | 26.1±6.9 | 6.8±1.9 | 17.8±5.4 | 21.5±3.0 | 33.2±6.0 | 23.4±3.3 | 25.0±3.4 | 1.9±0.8 |
| | 3 | 34.2±0.3 | 94.0±20.5 | 0.6±1.3 | 18.7±4.4 | 61.9±13.5 | 21.6±10.3 | 33.1±10.0 | 28±9.9 | 27.5±9.2 | 0.4±1.1 | 3.5±11.6 | 20.0±7.1 | 14.7±16.0 | 20.6±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 | 0.4±0.7 | 19.9±5.9 | 19.8±5.9 | 31.2±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±0.1 | 13.4±5.0 | 40.9±5.4 | 20.2±4.2 | 26.7±8.0 | 20.8±4.1 | 26.0±6.3 | 5.2±1.0 | 6.7±5.3 | 22.2±5.1 | 33.5±8.7 | 18.1±5.2 | 25.9±4.0 | 2.2±1.1 |
| | 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 | 5.2±1.0 | 6.7±5.3 | 22.2±5.1 | 33.5±8.7 | 18.1±5.2 | 25.9±4.0 | 2.2±1.1 |

potential edges that once added, maximize the content spread [19, 81, 92, 96]. 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, 88]. 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 targeted to the content spread capabilities of the entire network as opposed to a given set of sources [96]. Recent works in content spread have optimized the process by considering only the influence ranking of a single node [92]. 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 could be calculated using matrix multiplication [14]. Several authors built upon this idea establishing calculation of polarization and other metrics using the matrix multiplication 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, 101]. The main difference between these works and ours is that these works
target conflict between groups on the social network, while our targets equitable reception of content between groups, making it suitable 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, 63] and data management [4, 6, 7, 36, 49, 67, 78–80, 84, 90, 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, access to equitable information has raised questions. As a result, a related line of research started focusing on 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 maximization 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 less in influence over additional influence [49], a new multi-objective model that considers a fair influence spread among different groups [87, 95], 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**

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 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%.

To study the problem of edge suggestion without considering the recommendation acceptance probabilities. While a simple resolution may be to recommend a higher number of edges, addressing this problem requires new techniques we consider 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.

**9 ETHICS STATEMENT**

While this paper studies friendship suggestions in social networks that maximizes content spread while achieving fairness, all experiments performed herein are based on publicly available datasets in academic settings. We would like to note that this approach has not been integrated in any real world product, service, or platform. As a practical constraint, the assumption of the availability of user profile information may not be realistic due to the fact that by design none of the social networks operating today collects user demographic data. This paper is based on the fact that unfair spread of data still exists and it aims to address balanced data flow among different sets of nodes/users. Moreover, we highlight that adding such features to social networks requires rigorous regulation to avoid exploiting such features and cause ethical or societal implications. For example, even though this method is designed to bring benefits to users and maximize fairness, one
may change the system setup to do the opposite. We recognize that such ethical concerns are far more complex and need policymakers to look into how platforms can detect such misusage. We hope our work could help effect a positive change in this direction.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (NSF 2107290) and the Google research scholar award. We also thank Dr. Golnoosh Farnadi for sharing their dataset with us.

REFERENCES

[1] Chiara Accinelli, Barbara Catania, Giovanna Guerini, and Simone Minoti. 2021. The impact of reweeting on coverage constraint satisfaction. In EDBT/ICDT Workshops.
[2] Eytan Adar and Lada A Adamic. 2005. Tracking information epidemics in blogspace. In W3C, IEEE, 207–214.
[3] Abolfazl Asudeh, Tanya Berger-Wolf, Bhaskar DasGupta, and Anastasios Sidiropoulos. 2020. Maximizing coverage while ensuring fairness: a tale of conflicting objectives. 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), 3443–3448.
[5] Abolfazl Asudeh, HV Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing fair ranking schemes. In SIGMOD. 1259–1266.
[6] Abolfazl Asudeh, Zhongjun Jin, and HV Jagadish. 2019. Assessing and remedying coverage for a given dataset. In ICDE. IEEE, 554–563.
[7] Abolfazl Asudeh, Nima Shabazz, Zhongjun Jin, and HV Jagadish. 2021. Identifying insufficient Data Coverage for Ordinal Continuous-Valued Attributes. In SIGMOD. 129–141.
[8] Suman Banerjee, Mamata Jernamani, and Dilip Kumar Pratihar. 2020. A survey on influence maximization in a social network. KAES 62, 9 (2020), 3417–3455.
[9] Golnoosh Farnadi, Behrouz Babaki, and Michel Gendreau. 2020. A Unify- ing 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/10.1145/3366422.3383555
[10] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In SIGKDD. 259–268.
[11] He Feng and Xueming Qian. 2013. Recommendation via user’s personality and social contextual. In CIKM. 1523–1524.
[12] 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.
[13] 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).
[14] Noah E Friedkin and Eugene C Johnsen. 1990. Social influence and opinions. Journal of Mathematical Sociology 15, 3-4 (1990), 193–206.
[15] Suman Banerjee, Mamata Jernamani, and Dilip Kumar Pratihar. 2020. A survey on influence maximization in a social network. KAES 62, 9 (2020), 3417–3455.
[16] Abolfazl Asudeh, HV Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing fair ranking schemes. In SIGMOD. 1259–1266.
[17] Dumitru-Clementin Cercel and Stefan Trausan-Matu. 2014. Opinion propagation within social networks: Evidence on gender inequalities from Mali. Journal of Development Economics 106, 2 (2014), 174–192.
[18] Ruben Becker, Gianluca D’Angel, Sajjad Ghobadi, and Hugo Gilbert. 2022. Fair ranking schemes for the recommendation problem: A survey. Information 13, 8 (2022), 359–367.
[19] Kiran Garmella, Giammarco 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.
[20] Shay Gershtein, Tova Milo, Brit Youngham, and Gal Zevry. 2018. DM-balanced: influence maximization under balance constraints. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 1919–1922.
[21] Flavio Calmon, Dennis Wei, Bhaskar Sundaravadhan, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. In NIPS. 3992–4001.
[22] Evren Güney. 2019. An efficient linear programming based method for the influence maximization problem in social networks. Information Sciences 503 (2019), 589–605.
[23] Sainyam Galhotra, Karthikeyan Shanmugam, Prasanna Sattigeri, Kush R Varshney, and Rachel Bellamy. 2022. Causal Feature Selection for Algorithmic Fairness. (2022).
[24] Ruben Becker, Gianluca D’Angel, Sajjad Ghobadi, and Hugo Gilbert. 2022. Fairness in influence maximization through randomization. Journal of Artificial Intelligence Research 73 (2021), 1251–1283.
[25] Lauren Barocas. 2020. Moritze Hart, and Arvind Narayanan. 2019. Fairness and Machine Learning. fairmlbook.org. http://www.fairmlbook.org.
[26] Massimiliano Caramia and Paolo Dell’Olmo. 2008. Multi-objective management of freight logistics. Springer.
[27] Evrem Güney. 2019. An efficient linear programming based method for the influence maximization problem in social networks. Information Sciences 503 (2019), 589–605.
[28] 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).
[29] Noah E Friedkin and Eugene C Johnsen. 1990. Social influence and opinions. Journal of Mathematical Sociology 15, 3-4 (1990), 193–206.
[30] Sainyam Galhotra, Giammarco 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.
[31] Evren Güney. 2019. An efficient linear programming based method for the influence maximization problem in social networks. Information Sciences 503 (2019), 589–605.
[32] Shyam Gershstein, Tova Milo, Brit Youngham, and Gal Zevry. 2018. DM-balanced: influence maximization under balance constraints. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 1919–1922.
[33] Evren Güney. 2019. An efficient linear programming based method for the influence maximization problem in social networks. Information Sciences 503 (2019), 589–605.
[34] Sainyam Galhotra, Giammarco 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.
[35] Shyam Gershstein, Tova Milo, Brit Youngham, and Gal Zevry. 2018. DM-balanced: influence maximization under balance constraints. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 1919–1922.
[36] Evren Güney. 2019. An efficient linear programming based method for the influence maximization problem in social networks. Information Sciences 503 (2019), 589–605.
[37] Sainyam Galhotra, Giammarco 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.
Algorithm 5 ShortestDistances

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

1: $d \leftarrow$ multi-src-shortest-distances ($G(V, E)$)
2: for $v_i \in V \setminus V_s$ do $S'_i \leftarrow \{d_i\}$
3: for $e_j \in E_c$ do
4: let $e_j = (v_i, v_j)$ where $d_i < d_j$
5: if $d_i < d_j + 1$ then continue
6: $d_{ij} \leftarrow d_j + 1$; add $(v_i, d_{ij})$ to queue
7: while queue is not empty do
8: remove $(v_i, d_{ij})$ from queue
9: add $(e_j, d_{ij})$ to $S'_i$
10: for $v_j \in$ edges$(v_i)$ do
11: if $d_{ij} + 1 < d_j$ then
12: $d_{ij} \leftarrow d_j + 1$
13: add $(v_j, d_{ij})$ to queue
14: return $S' = \{S'_i \mid v_i \in V \setminus V_s\}$

APPENDIX

A PROOFS

Lemma 1. 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 $V'_c$, and the value $k'$ of 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 $G = G' \cup V'' \cup E''$.

- Let $S = \{v_i, v'_j\}$, where every node $v'_i \in V'$ belongs to $g_i$ and every node $v''_i \in V''$ belongs to $g_j$.

- Set $V_s = V'' \cup \{v''_i \mid v'_i \in V_s\}$. 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 $G, V_s, G, k$, and the decision value $T$ is yes.

Lemma 3. 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

$$\mathbb{E} \left( \sum_{e_j \in N(v_i)} y'_j \right) = \sum_{e_j \in N(v_i)} Pr(y'_j) = \sum_{e_j \in N(v_i)} y'_j = k$$
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. $p = 0.5$, $|V_i| = 3$

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 2 | 10.5 | 2.5 | 90.5 |
| 3 | 11.5 | 3.5 | 100 |
| 4 | 12.5 | 4.5 | 110 |

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 5 | 13.5 | 5.5 | 120 |
| 6 | 14.5 | 6.5 | 130 |
| 7 | 15.5 | 7.5 | 140 |

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 8 | 16.5 | 8.5 | 150 |
| 9 | 17.5 | 9.5 | 160 |
| 10 | 18.5 | 10.5 | 170 |

Table 6: Comparing Lift (in %), Disparity (in %), and Runtime (in seconds), on 1000 node Pokoc 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. $p = 0.5$, $|V_i| = 3$

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 2 | 10.5 | 2.5 | 90.5 |
| 3 | 11.5 | 3.5 | 100 |
| 4 | 12.5 | 4.5 | 110 |

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 5 | 13.5 | 5.5 | 120 |
| 6 | 14.5 | 6.5 | 130 |
| 7 | 15.5 | 7.5 | 140 |

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 8 | 16.5 | 8.5 | 150 |
| 9 | 17.5 | 9.5 | 160 |
| 10 | 18.5 | 10.5 | 170 |

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. $k = 3$, $|V_i| = 3$

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 2 | 10.5 | 2.5 | 90.5 |
| 3 | 11.5 | 3.5 | 100 |
| 4 | 12.5 | 4.5 | 110 |

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 5 | 13.5 | 5.5 | 120 |
| 6 | 14.5 | 6.5 | 130 |
| 7 | 15.5 | 7.5 | 140 |

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 8 | 16.5 | 8.5 | 150 |
| 9 | 17.5 | 9.5 | 160 |
| 10 | 18.5 | 10.5 | 170 |

Table 8: Comparing Lift (in %), Disparity (in %), and Runtime (in seconds), on 1000 node Pokoc 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. $k = 3$, $|V_i| = 3$

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 2 | 10.5 | 2.5 | 90.5 |
| 3 | 11.5 | 3.5 | 100 |
| 4 | 12.5 | 4.5 | 110 |

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 5 | 13.5 | 5.5 | 120 |
| 6 | 14.5 | 6.5 | 130 |
| 7 | 15.5 | 7.5 | 140 |

| K | Lift | Disparity | Runtime |
|---|------|-----------|----------|
| 8 | 16.5 | 8.5 | 150 |
| 9 | 17.5 | 9.5 | 160 |
| 10 | 18.5 | 10.5 | 170 |

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