# Fairness of Information Flow in Social Networks

ZEINAB S. JALALI, QILAN CHEN, SHWETHA M. SRIKANTA, and WEIXIANG WANG, Syracuse University
MYUNGHWAN KIM and HEMA RAGHAVAN, Kumo.AI
SUCHETA SOUNDARAJAN, Syracuse University

Social networks form a major parts of people's lives, and individuals often make important life decisions based on information that spreads through these networks. For this reason, it is important to know whether individuals from different protected groups have equal access to information flowing through a network. In this article, we define the `Information Unfairness (IUF)` metric, which quantifies inequality in access to information across protected groups. We then introduce `MinIUF`, an algorithm for reducing inequalities in information flow by adding edges to the network. Finally, we provide an in-depth analysis of information flow with respect to an attribute of interest, such as gender, across different types of networks to evaluate whether the structure of these networks allows groups to equally access information flowing in the network. Moreover, we investigate the causes of unfairness in such networks and how it can be improved.

CCS Concepts: • **Theory of computation** → *Design and analysis of algorithms*; *Graph algorithms analysis*; *Network flows;*

Additional Key Words and Phrases: Social Network Analysis, information flow, information fairness

**ACM Reference format:**
Zeinab S. Jalali, Qilan Chen, Shwetha M. Srikanta, Weixiang Wang, Myunghwan Kim, Hema Raghavan, and Sucheta Soundarajan. 2023. Fairness of Information Flow in Social Networks. *ACM Trans. Knowl. Discov. Data.* 17, 6, Article 79 (February 2023), 26 pages.
https://doi.org/10.1145/3578268



## 1 INTRODUCTION

In professional and other social settings, networks play an important role in people's lives, and communication between individuals can have a significant effect on individuals' decision making [7]. Through such communications, individuals learn about employment, promotion, and award opportunities, as well as make valuable connections to mentors or sponsors, all of which can influence the trajectories of their lives. Moreover, through such networks, individuals learn of new professional ideas, events and, other useful information that can affect their professional success. As such, it is of interest to understand whether information is flowing *fairly* to nodes in social networks.

This work was supported in part by NSF awards #1908048 and #2047224.
Authors' addresses: Z. S. Jalali, Q. Chen, S. M. Srikanta, W. Wang, and S. Soundarajan, Syracuse University, Life Sciences Complex, Syracuse, NY 13210; emails: {zsaghati, qchen55, smanchin, wwang69, susounda}@syr.edu; M. Kim and H. Raghavan, Kumo.AI, 357 Castro St, Suite 200, Mountain View, California 94041; emails: mykim@cs.stanford.edu, hema@kumo.ai.

1556-4681/2023/02-ART79 $15.00
https://doi.org/10.1145/3578268

In particular, depending on the structure of a social network, it is possible that individuals from certain *protected groups* (i.e., those based on so-called protected attributes like race or gender) are deprived of equal access to information spreading in the network. For instance, consider a company's professional network (describing interactions between employees) where White men occupy the majority of central positions, and women and minorities are on the fringes of the network [20]. If such network structure results in greater information flow to members of the advantaged group, then that allows the group to further consolidate power, worsening inequality. Such consequences have been observed in reality: for example, students from poor backgrounds are often unaware of opportunities for attending college and might not become aware of them through their connections, thus perpetuating the cycle of poverty [5].

It is thus important to quantify the extent to which individuals from different attribute groups have equal access to information spreading in the network (as discussed above, we are interested in groups based on sensitive protected attributes like gender or race). Once such unfairness has been detected, efforts can be made to remedy unfairness in information spread.

In this work, we define the **`Information Unfairness (IUF)`** metric, which quantifies inequality in access to information across protected groups. We then introduce `MinIUF`, an algorithm for reducing information unfairness by adding edges to the network. Finally, we provide an in-depth analysis of information flow with respect to an attribute of interest, such as gender, across different types of networks.

The work in this article is based on the earlier work in [19] which was focused on introducing a simpler version of Information Unfairness applied to undirected, unweighted networks with two protected groups. In brief, information unfairness was computed using distances between the means of three distributions: (1) internal flow within group one, (2) flow between groups one and two, and (3) internal flow within group two. These three flow distributions are computed using the *accessibility matrix* $\mathbf{A}_{(k)}$, where each element $a_{i,j}$ describes the expected amount of information that node $u_i$ receives from node $u_j$, measured using the number of walks of length up to $k$ between the two nodes. In this article, we show how to generalize the basic Information Unfairness from [19] in the following ways: (1) Unlike the earlier version, which was designed to handle undirected, unweighted networks, the modified `IUF` can apply to any kind of network. (2) The earlier version was designed to handle networks of two non-overlapping groups. In contrast, the new `IUF` can be used on networks with more than two potentially overlapping groups. (3) In the updated version of `IUF`, the information flow between node pairs is computed more realistically, using either non-backtracking walks or a probability-based model of flow, in contrast to the earlier version that was based on backtracking walks.

We introduce `MinIUF`, an algorithm to reduce information unfairness by adding edges. `MinIUF` is based on *MaxFair*, which we presented in [19]. In brief, *MaxFair* uses a power iteration-like process to compute a score for each pair of unconnected nodes, where the score represents the decrease in information unfairness that would be obtained by connecting those nodes. `MinIUF` is an improved version of *MaxFair*. While *MaxFair* estimates the effect of adding an edge on overall flow to each group, `MinIUF`, makes this estimate more accurate by considering the effect of adding an edge on flows of different length. Finally, we perform a comprehensive analysis of the information unfairness of different complex networks including co-authorship networks, social networks, an email network, and a blog network, and show how to reduce their unfairness.

Our contributions are as follows:

— We propose `IUF`, a generalized version of the Information Unfairness [19] metric, that measures the extent to which individuals from different groups have equal access to the information spreading in the network. `IUF` can be computed using either non-backtracking walks or probability based methods to measure information flow between pairs of nodes.

— We introduce MinIUF, a novel algorithm for reducing IUF by adding a set of edges to a network.
— We perform a detailed experimental analysis of IUF on different networks. We explore the structural properties of these networks that lead to high or low IUF.

The remainder of this article is organized as follows: In Section 2, we discuss relevant related work. In Section 3, we define IUF and show how it can be computed. In Section 4, we propose the MinIUF algorithm for reducing IUF. In Section 5, we perform an extensive analysis on the unfairness of different complex networks. Finally, Section 6 concludes the article.

## 2 RELATED WORK

This work is broadly situated within the realm of "algorithmic fairness", which has attracted a great deal of attention from the algorithmic community in recent years [5]. At a very high level, one primary motivation behind such works is that algorithms should treat individuals from different protected groups equally, where protected groups are those defined based on protected attributes like race or religion (The term *protected group* can encompass both *underprivileged* and *privileged* groups). Most of the existing work on algorithmic fairness has been on machine learning algorithms, including for applications like credit scoring, criminal sentencing, and others [5]. In contrast to most of the work on algorithmic fairness, our primary goal is not to analyze the fairness of an algorithm's output, or design algorithms that are "fair", but rather to understand the effect of a network's structure itself on fair outcomes.

There are some recent works on algorithmic fairness on machine learning approaches related to network data. For instance, Masrour et al., studies algorithmic fairness in node classification and network sampling using machine learning techniques [36], Stocia et al., studies the impact of social recommendations on network fairness by showing the existence of algorithmic glass ceiling in social network [39] and Beilinson et al., introduces clustering, based on fair access to information [6]. In contrast to our work, these works propose methods for fair network analysis, as opposed to studying the fairness of the network structure itself.

Closely related to our work, are works on fairness in influence maximization: Fish et al., and Tsang et al.'s goal is to select seeds in a way that information spread is maximized, while different groups have equal access to the information that is spreading in the network [12, 43] and Wang et al., studies the equality of information access in different dynamic network models and shows the trade-off between efficiency of information access and equality [44]. In contrast, our goal is to measure fair flow of information regardless of the start point, not to select seeds for fair information flow.

Two related concepts to fairness in social networks are echo chambers and homophily in networks. Echo chambers occur when beliefs of members in a group are amplified by other members of that group [4]. Most works on echo chambers are focused on political opinions [13]. Homophily, a measure of segregation of networks, is the tendency of individuals to associate with like minded others. In network science, homophily is often measured by the assortativity coefficient [29]. Such segregation has been observed in professional networks [18], and in societal networks, can lead to educational inequality [31], reduced health outcomes [24], and reduced exposure to advertising [38]. Halberstam and Knight show that members of a majority group can receive more pieces of information than members of a minority group [16] and Karimi, et al., show that as homophily increases, majority nodes have harder time accessing information coming from minority nodes and vice versa [21]. However, we will show later, homophily does not capture the same nuance of information flow as our proposed IUF metric.

In our experiments section, we study unfairness in citation networks with respect to gender, Nettasingh et al.'s work, study fairness and inequalities in the structure of citation networks [28].

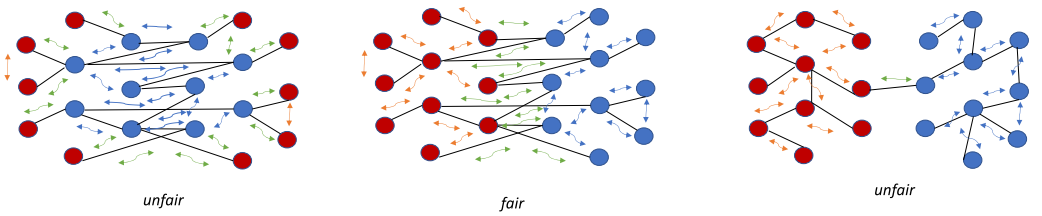



Fig. 1. Three networks with the same number of nodes and edges from each group but different information flow (only selected flows shown).

The final part of our article proposes a method for decreasing unfairness by adding edges. There is some existing work on increasing flow in a network by adding edges [10, 42], but this work has not considered the problem from a fairness perspective.

## 3 INFORMATION UNFAIRNESS (IUF)

In this section, we define the novel IUF (Information UnFairness) metric, which measures the extent to which information flows fairly between protected groups. At a high level, the intuition behind IUF is that we wish to determine whether the structure of the network, including how individuals are positioned in the network, allows groups to benefit equally in terms of their access to information. As an example, Figure 1 shows three networks with the same number of nodes and edges, where half of the nodes are members of the red group and the other half are members of the blue group. In the left-hand network, red nodes have difficulty accessing information starting at other red nodes: in other words, they are isolated from one another. In the right network, red and blue nodes can easily access information starting from their group-mates but have difficulty accessing the information that are from the other group: in other words, they are segregated. In the middle network, both red and blue nodes can easily access the information that starts from anywhere in the network: this network is fair.

### 3.1 Computation

IUF measures whether individuals from different protected groups have different levels of access to the information flowing in the network. Although we do not expect all *individuals* to spread and receive information equally, we consider a network unfair when a *protected group* is disproportionately deprived of the ability to access or spread information.

The symbols used in our computation are showed in Table 1. The input to IUF are:

— Network $G = (V, E)$: a network with $n$ vertices $V = \{v_1, \ldots, v_n\}$ and $m$ edges $E$ with adjacency matrix $\mathbf{M}_{n\times n}$, $0 < m_{ij} \leq 1$ if there is an edge between nodes $v_i$ and $v_j$ and $m_{ij} = 0$ otherwise ($G$ can be directed or undirected, simple or weighted.[1]) $m_{ij}$ denotes the probability that node $v_i$ transmits information directly to node $v_j$.

— A set of $l$ protected groups $P = \{p_1, \ldots, p_l\}$ and protected group membership matrix $\mathbf{PM}_{n\times l}$, where $pm_{ij}$ shows the strength of node $v_i$'s membership in group $p_j$. For each node $v_i$, $\Sigma_{j\in\{1,\ldots,l\}} pm_{ij} = 1$. These protected groups represent the groups of interest based on attributes like race or gender.

[1]Weighted edges can correspond to strengths of connections. For instance, if the input graph has attributes other than the sensitive attribute under study, and one can infer something about the probability of propagation along an edge from these attributes, this information can be encoding into edge weights.

Table 1. Notation

| Symbol | Definition |
|---|---|
| $G(V, E)$ | Weighted/simple, undirected/directed attributed graph |
| $V = \{v_i, \ldots, v_n\}$ | Nodes in $G$ |
| $n, m$ | Number of nodes and edges in $G$ |
| $\mathbf{M_{n\times n}}$ | Adjacency matrix of $G$, $m_ij$: probability of $v_j$ receives information from $v_i$ |
| $P = \{p_1, \ldots, _l\}$ | Set of protected groups |
| $\mathbf{PM_{n\times l}}$ | Protected group membership matrix: $pm_{ij}$: probability that node $v_i$ is a member of protected group $p_j$ |
| $k$ | Maximum length cascade considered |
| $\mathbf{S_{n\times n}}$ | Normalization matrix |
| $\mathbf{M}^k$ | Backtracking expected probability matrix: $m^k_{ij}$: probability of information passing through a walk of length $k$ from $v_i$ to $v_j$ |
| $\mathbf{B}^k$ | Non-backtracking expected probability matrix: $b'^k_{ij}$: probability of information passing through a non-backtracking walk of length $k$ from $v_i$ to $v_j$ |
| $\mathbf{A}_{(k)}$, $\mathbf{A'}_{(k)}$ | Accessibility & normalized accessibility matrix |
| $D_{p_ip_j}$ | Joint attribute accessibility distribution of protected groups $p_i$ and $p_j$ |
| $\mathbf{FM}_{l\times l}$ | Flow Mean matrix, $fm_{ij}$: mean of flow from protected group $p_i$ to protected group $p_j$ |

— $k \in N$: the maximum information cascade length. Because most important information has an "expiration date" (e.g., deadline to apply for a job), we do not consider arbitrarily long cascades.
— Normalization matrix $\mathbf{S}_{m\times m}$ (optional) (details are provided in Section 3.2).
— A distance function $Dist(D_1, D_2)$ for computing the distance between two distributions. This distance function could be, e.g., Earth Mover's Distance.

Given this input, IUF is computed as follows (details of the various steps are discussed further below). A fair network will have $IUF$ value close to 0, and higher values indicate greater unfairness.

(1) **Accessibility matrix construction:** Construct the *accessibility matrix* $\mathbf{A}_{(k)}$, where $a_{ij}$ shows the amount of information that is expected to flow from node $v_i$ to node $v_j$ using adjacency matrix $\mathbf{M}$ and maximum cascade length $k$.
(2) **Normalization:** If a normalization matrix $\mathbf{S}$ is provided, we define the normalized accessibility matrix $\mathbf{A}'_{(k)} = \mathbf{A}_{(k)} \oslash \mathbf{S}$. As we discuss later, the normalization matrix can be defined based on density, degree or any other desired properties.
(3) **Characterizing flow between protected groups:** Compute a list of joint attribute accessibility distributions $LD : \{D_{fg} : f, g \in \{1, \ldots, l\}\}$ from $\mathbf{A_{(k)}}$ or $\mathbf{A'_{(k)}}$ that characterize flow between each pair of protected groups $p_f$ and $p_g$, where $\{p_i, p_j\} \in \{p_1, \ldots, p_l\}$.
(4) **Computing the Information Unfairness (IUF):** Using the given distance function, find the distance between each pair of joint attribute accessibility distributions and return the maximum such distance: $IUF = max(\{Dist(D_h, D_z) : D_h, D_z \in LD\})$.

*3.1.1 Accessibility Matrix Construction.* Matrix $\mathbf{A}_{(k)}$ describes the flow of information between each pair of nodes. More formally, $a_{ij}$ is the expected amount of information that node $v_j$ will receive from node $v_i$, and can be computed in accordance with whatever model of information flow one desires. Here, we consider two different ways to construct this matrix. The first method, based on non-backtracking walks, allows one to compute the expected amount of information flowing between each pair of nodes. The second method, which uses the SI contagion model, allows one to compute the probability that information is shared between a pair of nodes. There are various benefits and drawbacks to these methods: The first method is faster to compute and is deterministic, but the second may give results that are of greater interest.

**(1) Non-backtracking Walks:** First, we use the number of walks between two nodes to measure the expected amount of information flowing between those nodes. In a simple network with binary adjacency matrix $\mathbf{M}$, $\mathbf{M}^k$ describes the number of walks of length-$k$ (each element $(i, j)$ in this matrix is the number of length $k$ walks between nodes $v_i$ and $v_j$). In a weighted network, where

elements of $\mathbf{M}$ show the probability of information flow along an edge, each element $m_{ij}^k$ of $\mathbf{M}^k$ is the expected number of times that node $v_j$ hears about a piece of information starting from node $v_j$ that passes along walks of length $k$.

Using this idea, our earlier work in [19] computed the accessibility matrix $\mathbf{A}_{(k)}$ as $\mathbf{A}_{(k)} = \mathbf{M} + \mathbf{M}^2 + \cdots + \mathbf{M}^k$. In this way, $a_{ij}$ shows the expected number of times that node $v_j$ will receive a piece of information starting from node $v_i$ by using walks of length at most $k$ [3]. If $\mathbf{M}$ is invertible, then $\mathbf{A}_{(k)} = (I - \mathbf{M})^{-1}\dot{(}I - \mathbf{M}^{k+1}) - I$. (Note that if $\mathbf{M}$ is not invertible, adding a small amount of error $\epsilon$ to the diagonal elements will make it invertible without significantly affecting the results.)

Note that $\mathbf{M}^k$ contains the number of walks between two nodes of length-$k$ *with* backtracking. However, for purposes of measuring information flow, backtracking walks are less relevant than *non-backtracking* walks. In the information unfairness computation, we are interested to see whether nodes from different groups will learn about a piece of information that is spreading in the network. If a piece of information starts from node $u$ and reaches its neighbor $v$, it is not of interest to see whether node $u$ will hear about that same piece of information immediately from node $v$. By only considering non-backtracking walks, we eliminate such routes while computing the path. Such a restriction has recently been used in many graph applications [2]. In this work, we use the method introduced in [2] for computing non-backtracking walks of length $k$, and compute accessibility matrix as $\mathbf{A}_{(k)} = \mathbf{B}^1 + \mathbf{B}^2 + \cdots + \mathbf{B}^k$. Here, $b_{ij}^k$ shows the expected number of times that node $v_j$ will receive a piece of information starting from node $v_i$ through non-backtracking walks of length at most $k$. $\mathbf{B}^k = \frac{\mathbf{B}_r^k + \mathbf{B}_l^k}{2}$ and $\mathbf{B}_r^k$ and $\mathbf{B}_l^k$ are computed as follows (note that if $G$ is undirected, $\mathbf{B}_r^k = \mathbf{B}_l^k$). Let

$$\mathbf{B}_l^1 = \mathbf{M}, \mathbf{B}_r^1 = \mathbf{M}^T, \mathbf{B}_l^2 = \mathbf{M} \cdot \mathbf{M}^T - \mathbf{D}_1, \mathbf{B}_r^2 = \mathbf{M}^T \cdot \mathbf{M} - \mathbf{D}_2, \mathbf{D}_1 = diag(\mathbf{M} \cdot \mathbf{M}^T), \mathbf{D}_2 = diag(\mathbf{M}^T \cdot \mathbf{M}).$$

For $k > 2$, if $k$ is even:

$$\mathbf{B}_l^k = \mathbf{B}_l^{k-1} \cdot \mathbf{M}^T + \mathbf{B}_l^{k-2} \cdot (\mathbf{I} - \mathbf{D}_1), \mathbf{B}_r^k = \mathbf{B}_r^{k-1} \cdot \mathbf{M} + \mathbf{B}_r^{k-2}\dot{(}\mathbf{I} - \mathbf{D}_2);$$

and if $k$ is odd:

$$\mathbf{B}_l^k = \mathbf{B}_l^{k-1} \cdot \mathbf{M} + \mathbf{B}_l^{k-2}\dot{(}\mathbf{I} - \mathbf{D}_2), \mathbf{B}_r^k = \mathbf{B}_r^{k-1} \cdot \mathbf{M}^T + \mathbf{B}_r^{k-2}\dot{(}\mathbf{I} - \mathbf{D}_1).$$

Finally, after constructing $\mathbf{A}_{(k)}$, we set elements on the diagonal to 0, because the information about whether a node transmitted information to itself is not relevant. Note that because a node may receive multiple cascades from another node, each $a_{ij}$ might be greater than 1. Depending on the application, one may wish to truncate these values at 1 (indicating that a node is expected to receive at least a piece of information at least once).

**(2) SI Contagion-Based Method:** Although the above method is tractable and deterministic, there are more sophisticated models for information flow. Many of these models are based on the **Susceptible-Infected** (**SI**) model of disease spread [27]. One can use such a model to measure information flow between each pair of nodes; however, it is computationally difficult to compute the nodes that will be influenced by a set of seeds, and so extensive simulations are required [45]. This approach is thus best suited for small networks.

The SI model is a classic epidemic model, and is commonly used to simulate information spread. It was originally designed to describe how disease spreads through a population [27]. In this model, every node has two states: susceptible and infected. In each iteration, every infected node will infect its susceptible adjacent nodes with a certain probability. The SI model is an application of the **Independent cascade** (**IC**) model [14], which assumes that the information spreading process begins from an initial active nodes set $S$. At time $t$, let $S_t$ denote the set of activated

nodes. Every node in the set $S_t$ will independently attempt to infect its susceptible neighbors, and will succeed with a specified probability. This process repeats in the next iteration, making the IC model stochastic and progressive [9]. We apply a **Monte Carlo** (**MC**) simulation [23] to estimate the information spreading results. MC methods are typically used in models where the probability of outcomes cannot be determined due to random variable intervention. The core idea of MC simulations is to constantly repeat random sampling to approximate the desired results. The MC simulation in our experiment repeats the information diffusion process for $t$ iterations independently. Using this method, $\mathbf{A}_{(k)}$ is computed as follows:

At each iteration, we compute the spreading probability accessibility matrix **SA** as follow: for each node $v_i$, we set $v_i$ as the only activate node in the initial set $S_0$. Across $k$ iterations, nodes in $S_k$ will activate their neighbors with probability $m_{ij}$, and activated nodes will be added to $S_k + 1$ (each active node can activate its neighbor only once). For each node $u_j$ that is activated during the $k$ steps where $S_0 = u_i$, $sa_{ij} = 1$ and for all non-activated nodes $u_j$, $sa_{ij} = 0$. Finally, we compute $\mathbf{A}_{(k)}$ by taking the average over all computed matrices $\mathbf{SA}_1$ to $\mathbf{SA}_k$. This represents the probability that a node receives a piece of information starting at another node, where transmission probabilities are given by **M**.

*3.1.2 Characterizing Flow Between Protected Groups.* Elements in **A** show the flow between each pair of nodes. However, we are ultimately interested in understanding the flow between the pairs of protected groups. Thus, we define *joint attribute accessibility distributions* $D_{fg}$ for protected groups $p_f$ and $p_g$ where $\{p_f, p_g\} \in P = \{p_1, \ldots, p_l\}$. $D_{fg} = \{a_{ij} \cdot pm_{if} \cdot pm_{jg} : i, j \in \{1, \ldots, n\}\}$. We compute a list of all such distributions as $LD : \{D_{fg} : f, g \in \{1, \ldots, l\}\}$ from **A**.

*3.1.3 Computing* IUF. After computing all the joint accessibility distributions, the Information Unfairness of the network is given by: $IUF = max(\{Dist(D_h, D_z) : D_h, D_z \in LD\})$. Here, *Dist* is a function for computing the distance between two distributions.

Informally, the IUF of a network measures the differences between the joint accessibility distributions. For instance, if there are two protected groups- a minority and a majority- in the network, we want to determine whether information flows equally within the minority group, within the majority group, and between the two groups.

**Choice of Distance Function.** There are a multitude of distance measures for computing the difference between two distributions. Examples include **K–L Divergence** (**KLD**), **Earth Mover's Distance** (**EMD**), distance between **weighted means** (**WM**), distance between **truncated weighted means** (**TWM**), distance between **weighted medians** (**WME**), and so on. Selection of the best distance function depends on the distributions under study. In this work, we consider the distance function as the weighted distance between their means. Note that,by summarizing a distribution into its means, we might lose important information. For comparison, we also experimented with other distance measures, including EMD, WM, TWM, and WME, and saw similar results on the datasets used in this work. (For this work, KLD is not a good choice as it does not consider the distance of the values into account: e.g., using KLD, $Dist([1, 1, \ldots, 1], [2, 2, \ldots, 2]) = Dist([1, 1, \ldots, 1], [3, 3, \ldots, 3])$.)

**Matrix-based Computation of IUF.** If distance between WM is used as the distance function, IUF can be computed directly from matrices **A** and **PM** using matrix multiplications as follows:

- Compute the flow sum matrix $\mathbf{FS}_{l \times l} = \mathbf{PM}^{\mathbf{T}} \cdot \mathbf{A}_{(k)} \cdot \mathbf{PM}$, where $fs_{fg}$ is sum of the flows from all nodes from protected group $p_f$ to all nodes from protected group $p_g$.
- Compute weight matrix $\mathbf{W}_{l \times l} = \mathbf{PM}^{\mathbf{T}} \cdot (\mathbf{M} - Diag(\mathbf{M}) \cdot \mathbf{PM})$, where $w_{fg}$ is the weight matrix used for computing weighted mean of distributions.

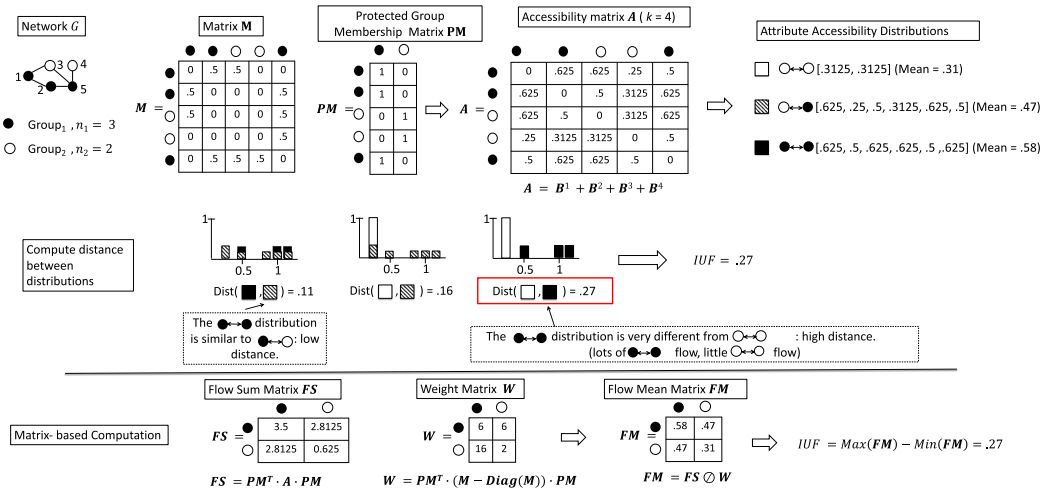



Fig. 2. Overview of information unfairness computation process.

— Compute flow Mean matrix $\mathbf{FM}_{l\times l} = \mathbf{FS} \oslash \mathbf{W}$.
— $IUF = max(\mathbf{FM}) - min(\mathbf{FM})$

An overview of the IUF computation process is given in Figure 2.

*3.1.4 Complexity Analysis and Scalability.* The time complexity of computing the IUF of network $G$ with $n$ nodes depends on how accessibility matrix $\mathbf{A}$ is computed and how distance metrics are defined. If $\mathbf{A}$ is computed using the SI model simulation, the result is not deterministic, and so many simulations may be required for stability. Thus, for our complexity analysis, we focus on analyzing the walk-based models. Moreover, as we used distance between weighted means (WM), we provide analysis for the WM distance metric.

First, for the accessibility matrix computation using non-backtracking walks, $\mathbf{A}_{(k)}$ is computed as $\mathbf{A}_{(k)} = \mathbf{B}^1 + \mathbf{B}^2 + \cdots + \mathbf{B}^k$. For computing $\mathbf{B}^k$ from $\mathbf{B}^{k-1}$, at most 4 matrix-matrix multiplications and 4 matrix additions and subtractions are needed. This takes $O(n^3)$ time, and so the overall computation of $\mathbf{B}^k$ and $\mathbf{A}_{(k)}$ takes $O(kn^3)$ ($\mathbf{B}^1$ to $+\mathbf{B}^{k-1}$ will be computed as part of the process). Note that the time complexity of computing $\mathbf{A}_{(k)}$ for the backtracking version is also $O(kn^3)$, as computing matrix $\mathbf{M}^k$ requires $k$ matrix matrix computation. Thus, the time complexity for either of these methods is $O(kn^3)$. After computing matrix $\mathbf{A}_{(k)}$, as distance metric WM is used, we use the matrix based computation (explained in the previous section) for measuring distance. Computing each of the matrices $\mathbf{FS}_{l\times l}$ and $\mathbf{W}_{l\times l}$ requires two matrix-matrix multiplications, taking $O(ln^2)$ time. Computing $\mathbf{FM}_{l\times l}$ and finding the minimum and maximum elements of this matrix each take $O(l^2)$. Thus, the overall time complexity of computing IUF using walk-based methods is $O(kn^3)$.

As explained, the process of computing IUF using the walk-based computation needs a set of matrix-matrix multiplications. Matrix-matrix multiplications is one of the most vital operations in many applications from computational science, machine-learning, network science and modeling [26], and thus there have been great advances on matrix operations for large matrices in the past few years. From different matrix libraries like LAPACK [34] to hardware accelarators including GPUs [26]. With the use of these techniques, IUF can be quickly computed for large networks.

## 3.2 Normalization Matrix Computation

The accessibility matrix describes the amount of flow between each pair of nodes; however, it may also be of interest to know how this flow compares to what one would expect in a random graph

with some of the same properties as the actual graph. To compare the flow of information in the network to the flow in random networks with the same desired topological properties (i.e., with the same density or degree distribution), the user may provide a normalization matrix **S**. The goal of such normalization is to obtain the specific topological properties that lead to unfairness in information flow because nodes in the network have different structural properties that may have influence on their accessibility score. This is comparable to identifying the causal effect, where different characteristics of treatment groups have influence on the scores used in computing causal effect and propensity score normalization is used to overcome this bias [33]. In Section 3.2, we discuss how to compute **S** based on degree, and density. The purpose of normalization is to compare the information flow in the original network to the expected information in a random network with specific topological properties. To do so, we do element-wise division of accessibility matrix of original network **A** by expected accessibility matrix of random network **S**. In order to compute **S**, one can generate $t$ random networks with the particular properties of interest and compute accessibility matrix for each random network. Then, matrix **S** is the average over all accessibility matrices of $t$ random networks. If $t$ is large enough, $s_{ij}$ is an approximation of the expected flow that starts from node $v_i$ and reaches node $v_j$.

However, generating $t$ networks and computing accessibility matrix for each one, is computationally expensive. In this work, we use two ways of normalization based on (1) density, (2) degree and discuss ways to quickly and effectively estimate **S** using walk based methods (these estimations were first introduced in [19]). In Section 5, we measure the accuracy of our estimations by comparing the estimated values to those generated by generating a large number of random networks.

*3.2.1 Density-Based Normalization.* A drawback in using the unnormalized matrix **A** is that nodes naturally receive more information in a dense graph compared to a sparse graph, so differences between groups are magnified. The IUF of a dense graph may thus be higher than IUF of a sparse graph, but this does not necessarily indicate that the dense graph is less fair than the sparse graph. To compare the IUF of graphs of different densities, it is necessary to normalize $\mathbf{A}_{(k)}$ by density ($\mathbf{S}_{den}$). $\mathbf{S}_{den}$ can be computed by taking the average over accessibility matrices of $t$ random networks with the same density as network $G$. As computing $\mathbf{S}_{den}$ using this process is slow, we estimate $\mathbf{S}_{den}$ as described in [19] by first defining matrix $\mathbf{M}_{den}$ so that all elements are equal to the density of the graph. This value is given by $2m/n^2$, where $m$ and $n$ represent the number of edges and the number of nodes in the network respectively. $\mathbf{M}_{den}$ is an estimation over the average of adjacency matrices of $t$ random graphs with the same number of edges and nodes produces without generating these graphs. After generating $\mathbf{M}_{den}$, $\mathbf{S}_{den}$ can be generated the same way that $\mathbf{A}_{(k)}$ was generated from **M**.

*3.2.2 Degree-Based Normalization.* To see the impact of the number of connections (degrees) of nodes in each protected group on the information unfairness of a network, one can use degree based normalization. For instance, in a collaboration network of STEMM (Science, Technology, Engineering, Math, and Medicine) scientists, it is possible that highest degree nodes are more likely to be senior researchers and thus disproportionately male, as women have only entered the field in large numbers in the past few decades [17]. It is useful to know whether unfairness is due to differences in degree (though such an explanation would not necessarily *excuse* unfairness). Thus, to explore the impact on degree distributions of nodes on unfairness, it is necessary to normalize $\mathbf{A}_{(k)}$ by degree ($\mathbf{S}_{deg}$). As mentioned before, $\mathbf{S}_{deg}$ can be computed by taking the average over accessibility matrices of $t$ random networks with the same degree distributions as network $G$. As computing $\mathbf{S}_{deg}$ using this process is slow, we estimate **S** as introduced in [19] as follows:

We first define matrix $\mathbf{M}_{deg}$ so that $m_{ij} = d_{v_i} d_{v_j}/2m$ where $m$ is the number of edges and $d_{v_x}$ is degree of node $v_x$. Note that this is the same as the normalization used in the modularity metric for characterizing and detecting communities in the network [30]. $\mathbf{M}_{deg}$ is an estimation over the average of adjacency matrices of $t$ random graphs with the same degree distributions without generating these graphs. After generating $\mathbf{M}_{deg}$, $\mathbf{S}_{deg}$ can be generated the same way that $\mathbf{A}$ was generated from $\mathbf{M}$.

### 3.3 Example

Figure 1 depicts three graphs with the same density (20 nodes and 19 edges) and different values of information fairness. For propagation probability of $pp = 0.5$ for all edges in the network and $k = 4$, for each graph, we computed $\mathbf{A}_{(k)}$ using both walk-based and SI-based methods. For the SI-based method, we computed $\mathbf{A}_{(k)}$ by running 1000 simulations and taking the average accessibility matrix over all trials. After computing accessibility matrices, we identified three distributions to characterize flow between different group pairs (red-red, red-blue, and blue-blue node pairs). As all three networks have the same number of nodes and edges, we compare them directly without density normalization.

Using accessibility matrices computed by the walk-based method, for the networks a, b, and c, respectively, the red-red distribution has a mean of $(0.04, 0.12, 0.17)$, the blue-blue distribution has a mean of $(0.18, 0.11, 0.20)$, and the red-blue distribution has a mean of $(0.12, 0.12, 0.08)$. In all networks, there is good flow within the set of blue nodes. The middle graph also has good flow within the set of red nodes and between red and blue nodes. However, the right network has minimal flow between red and blue nodes (but good flow within the set of red nodes), and the left network has minimal flow within the set of red nodes (and good flow between red and blue nodes). $IUF$ is computed by computing the distance between maximum means and minimum means, giving us $0.18 - 0.04 = 0.14$ for the left network, $0.12 - 0.11 = 0.01$ for the middle network and $0.20 - 0.08 = 0.12$ for the left network.

Using accessibility matrices computed by the SI-based method, for the left, middle, and right networks, respectively, the red-red distribution has a mean of $(0.05, 0.14, 0.21)$, the blue-blue distribution has a mean of $(0.23, 0.13, 0.25)$, and the red-blue distribution has a mean of $(0.13, 0.15, 0.10)$. On these networks, the results are close to the results of the walk-based method. The $IUF$ values are then $0.23 - 0.05 = 0.18$ for the left network, $0.15 - 0.13 = 0.02$ for the middle network, and $0.25 - 0.10 = 0.15$ for the right network.

It is clear that the middle network should have much lower information unfairness than the ones on both sides. Both left and right networks are unfair, but for different reasons. The left network suffers from isolation of red nodes (low red-red flow) and the right network suffers from segregation of groups (low red-blue flow). In real contexts, these problems would require different remedies: for instance, in the left network, one should attempt to build connections between members of the red group (e.g., peer group-type connections), while in the right network, more integration between groups is required.

### 3.4 Interpretation

Interpreting `IUF` values is easier after normalization with respect to a null model (e.g., degree- or density-based normalization, as described earlier). Through such normalization, `IUF` value is not affected by different group sizes. Each joint attribute accessibility distribution tells us how information flows from members of one attribute group to members of another attribute group. `IUF` deals with the two most different pairs of attribute groups, and computes the distance between their flow distributions. For instance, in the right network in Figure 1, which is a segregated network, we see good flow between blue nodes (blue group to blue group), good flow between red nodes

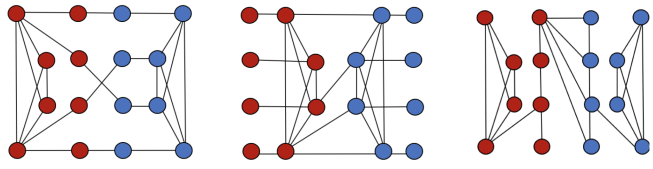



Fig. 3. Graphs with same assortativity coefficient values, but different information unfairness scores [19].

(red group to red group), and weak flow between red and blue nodes (red group to blue group). The greatest distance happens in comparing either (blue-blue to red-blue) or (red-red to red-blue). In the normalization process based on a null model like degree or density we divide each value of the accessibility matrix by the expected value in the random network. Thus, each element $a_{ij}$ after normalization corresponds to the actual amount of flow between $v_i$ and $v_j$, divided by the expected amount of flow between $v_i$ and $v_j$ in a random graph with the same topological properties.

Note that in a fair network, we do not necessarily expect to see the same flow between nodes $v_i$ and $v_j$ as we see in the random network. However, the reason behind doing this normalization is to see whether nodes from each group as a whole have been harmed or benefited from the structure equally. For instance, if we see the flow between two groups (red-blue) is 50% higher in original network compared to the random network, then in order for the network to be fair, we need to see the flow between two groups (red-red) or (blue-blue) is also 50% higher in the original network compared to the random network. Thus, the distance between two distributions is then relative to the null model.

### 3.5 IUF vs. Assortativity

As a concept, IUF is related to assortativity, which is a measure of homophily. However, because assortativity is a dyadic measure, there are important differences. Figure 3 shows three graphs with the same assortativity coefficient value (0.67) but different values of IUF: graph(a), graph(b), and graph(c) have walk-based IUF values equal to 0.64, 0.15, and 0.57, respectively and SI-based IUF equal to 0.40, 0.12, and 0.35, respectively. The graph(c) has a significantly higher IUF than the graph(b) and slightly lower IUF than the graph(a). It is easy to see that in all graphs, information flows easily between nodes in the same group (red-red and blue-blue); however, information flows much more easily from a blue node to a red node in the graph(b). Assortativity does not capture these differences.

## 4 MINIUF: AN ALGORITHM FOR REDUCING INFORMATION UNFAIRNESS

In certain application domains, it might be possible to reduce the IUF of a network by adding edges. For instance, if a company detects high values of IUF for its professional network, then it can reduce it by adding key employees to a mailing list (increasing flow between employees on that list), planning meetings where specific individuals are present, organizing social groups, and so on. In our earlier work, we formulated the problem and introduced *MaxFair*, an algorithm for adding edges to a network to reduce its Information Unfairness [19]. Here, we describe MinIUF, a new algorithm that improves on *MaxFair* at the task of adding edges to minimize the IUF of the resulting network.

Both MinIUF and *MaxFair* identify a set of candidate edges to add to the network by scoring candidate edges in each iteration. A candidate edge's score describes the amount by which its addition is estimated to decrease IUF. The algorithms then select the highest-scoring edge(s). *MaxFair* uses

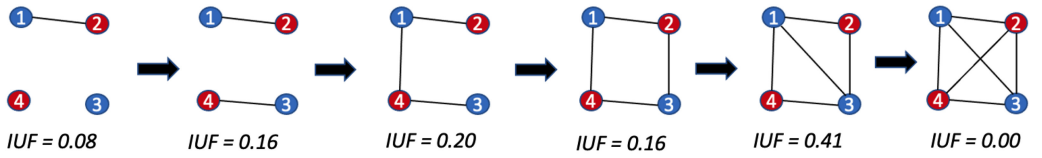



Fig. 4. Adding single edge increases IUF while adding multiple edges decreases IUF.

a power iteration-like process and computes attribute centrality of nodes using accessibility matrix. Each candidate edge (pair of unconnected nodes) is scored using computed attribute centrality. However, *MaxFair*'s process uses only walks of a specific length $k$. In contrast, `MinIUF` considers those walks of length up to $k$ that are affected by addition of an edge.

Note that in some cases, it may be possible to reduce unfairness by removing edges. However, in real networks, removing edges is often much less practical than adding edges. For example, while professional networks may benefit from adding edges as discussed or social networks can use `MinIUF` in a friendship recommendation systems, it is practically difficult to recommend or ask people to *remove* established connections.

### 4.1 Problem Statement

Assume that we are given a network $G$ with adjacency matrix $\mathbf{M}$, cascade length $k$, protected group membership matrix $\mathbf{PM}$, and budget $b$. The goal is to recommend a set of $b$ edges that are not already present in $G$ such that adding those edges to $G$ will minimize the `IUF` of the resulting network.

### 4.2 Challenges

There are several challenges associated with this problem. First, the problem of minimizing `IUF` is not submodular. For example, Figure 4 shows a toy example where adding a single edge will not decrease `IUF`, but adding multiple edges will. Second, it is not easy to estimate the changes in flow after adding a set of edges. Although there are works on characterizing flow in a network using its spectral decomposition properties [8], and works on adding edges to the network such that *overall* flow in the network is maximized [42], these methods cannot be directly used for our problem because (1) they seek to increase the *overall* flow, whereas we seek to increase flow between specific attribute groups and (2) spectral decomposition considers flow where $k$ goes to infinity, whereas we are primarily interested in flow for small values of $k$ (because as mentioned before, high values of $k$ are not practical for information flow in real settings, because information often "expires").

### 4.3 MinIUF Computation

In this section, we introduce `MinIUF`, an algorithm for reducing walk-based `IUF`. The main idea behind this algorithm is that when an edge $(v_i, v_j)$ is added, it can influence cascades of lengths up to $k$ between any pairs of protected groups in the network. Thus, when we want to add an edge, we need to consider its effect on cascades of different length.

The heart of `MinIUF` is a method for scoring candidate edges according to their expected effect on `IUF`. Because this effect can change as edges are added, one would ideally recompute scores after each edge is added; however, because of the associated computational costs, we instead allow for multiple edges to be added before recomputing scores. On our networks, this did not significantly alter outcomes, but if one is beginning with a network that already has a very low `IUF` (and so there is little room for improvement), it may be more important to re-score frequently.

MinIUF consists of $x$ iterations, where each iteration adds a total of $b/x$ edges to the network. In each iteration, MinIUF computes the score for all edges that are not present in $G$ and then selects the $b/x$ highest scoring edges to add to $G$. Scores are computed as follows:

(1) Initialize $counter = 0$:
(2) Let $\mathbf{A}_{(k)}$ be the accessibility matrix corresponding to walks/cascades of up to length $k$. Let $\mathbf{B}^k$ be the matrix of the expected number of times each node receives information from each other node using non-backtracking walks (see Section 3.1.1). Let **PM** be the protected group membership matrix. Define matrices $\mathbf{AC}^k_{n\times l} = \mathbf{A_{(k)}} \cdot \mathbf{PM}$ and $\mathbf{BC}^k_{n\times l} = \mathbf{B^k} \cdot \mathbf{PM}$. Let $pp_{ij}$ be the propagation probability between nodes $u_i$ and $u_j$.
(3) Compute the mean flow matrix $\mathbf{FM}_{l\times l}$ from $\mathbf{A}_{(k)}$ and **PM** as described in Section 3.1.3.
(4) Set $IUF_{before} = max(\mathbf{FM}) - min(\mathbf{FM})$
(5) For each pair of nodes $v_i, v_j$:
   — Initialize $\mathbf{FM'}_{l\times l}$ to a zero matrix.
   — $fm'_{fg} = fm_{fg} + p_{ij}\Sigma_{k'\in\{1,\ldots,k-1\}} ac^{k'}[i,f]\cdot bc^{k-k'-1}[j,g] + ac^{k'}[j,f]\cdot bc^{k-k'-1}[i,g] + ac^{k'}[i,g]\cdot bc^{k-k'-1}[j,f] + ac^{k'}[j,g]\cdot bc^{k-k'-1}[i,f]$
   — $IUF_{after} = max(\mathbf{FM'}) - min(\mathbf{FM'})$.
   — $score(u_i,u_j) = IUF_{after} - IUF_{before}$.
(6) Select top $b/x$ edges with highest score and add them to $G$.
(7) Increment $Counter$, If $counter < x$ go to 2, terminate otherwise.

MinIUF is based on computation of IUF using the flow mean matrix **FM** described in Section 3.1.3. Because $IUF = max(\mathbf{FM}) - min(\mathbf{FM})$, by estimating the changes in matrix **FM** after connecting each pair of nodes, we can select the best edges to add. To compute the changes in matrix **FM** we estimate the changes in flow of information that is caused by new non-backtracking walks of different length at step 4.

The major contributors to MinIUF's running time are recalculation of matrices **FM** and **AM**. Choosing lower values for $x$ increases running time but might affect the performance.

Note that in some cases, certain edges may be impossible to add. In such cases, it is trivial to simply exclude those edges from the process.

Note that when an edge $(u, v)$ is added to the network, it can have an impact on flows of different lengths between many group-pairs. Edge $(u, v)$ might be in the middle of a flow or on either end of a flow. Suppose **FM** is the flow matrix of the initial network and **FM'** is the flow matrix after adding edge $(u, v)$. Then **FM'** can be computed from **FM**, where each element $\mathbf{fm'}_{gm}$ is equal to $\mathbf{fm}_{gm}$ plus all the additional flows that will start from any node from group $p_g$ and reaches any node from group $p_m$ passing through edge $(u, v)$, and vice versa. (Step 5 in the algorithm captures this.)

Thus, if the goal is to find just one pair of nodes at each step and connect them in the network, MinIUF will find the optimal solution at each step. However, when adding multiple edges at once (as may be required for efficiency reasons), the effect of adding an edge to a graph that has already been modified by the addition of some edges may not be the same as the effect of adding that edge to the original graph, and so this process acts as a heuristic.

*4.3.1 Complexity Analysis.* Here, we discuss the time complexity of MinIUF, corresponding to the steps described above. **Step 1:** $O(1)$. **Step 2:** As we showed in Section 3.1.4, computing matrices $\mathbf{B}^1$ to $\mathbf{B}^k$ takes $O(kn^3)$ time, where $n$ is the number of nodes in the network.

After computing $\mathbf{B}^1$ to $\mathbf{B}^k$, $\mathbf{A}_1$ to $\mathbf{A}_k$ is computed using $\mathbf{A}_j = \mathbf{B}^1 + \cdots + \mathbf{B}^j$, which takes $O(n^2)$ time, and then $\mathbf{AC}^1$ to $\mathbf{AC}^k$ and $\mathbf{BC}^1$ to $\mathbf{BC}^k$ are computing using $2k$ matrix-matrix multiplications, taking $O(kn^3)$ time.

Thus, Step 2 takes $O(kn^3)$ time. **Steps 3 and 4:** as we showed in Section 3.1.4, computing **FM** and finding the minimum and maximum elements, and thus the whole process for these two steps, takes $O(ln^2)$ time, where $l$ is the number of classes. **Step 5:** for all $\frac{n(n-1)}{2}$ pairs of nodes, the pairwise flow score is computed. Updating each $l^2$ elements ($u_{fg}$) in matrix **U** takes $O(1)$ time, and finding the minimum and maximum elements in **U** takes $O(l^2)$ time. Thus, the overall time complexity of Step 5 is $O(l^2n^2)$. And the overall process from Step 2 to Step 6 takes $O(l^2n^2 + kn^3)$. As the whole process takes $x$ time. Thus, the overall time complexity of `MinIUF` is $O(x(l^2n^2 + kn^3))$.

Note that although this time complexity might seem high for large networks, as the algorithm is based on matrix operations, in its implementation, we can benefit from advances in hardware and software technologies and run much faster. For example, parallelism techniques using a GPU or clustering can make the process very fast. For instance, in big data analysis, one of the common ways for fast matrix operations is deploying it in parallel on cloud (i.e., Apache Spark on Hadoop) which make data analysis much faster [32].

## 5 EXPERIMENTAL ANALYSIS

Here, we perform a comprehensive experimental analysis on different networks of different types, and show that none of these networks are fair with respect to information flow. Then, we analyze the performance of `MinIUF` in comparison to baseline methods and show that unfairness can be decreased dramatically even with slight changes in the network. We first describe the datasets of our study, then discuss our experimental setup, and end with a discussion of results.

### 5.1 Datasets

In our analysis, we use several different types of networks. First, we analyze the **DBLP** co-authorship networks of articles published in 2015–2019 with respect to gender. Five computer science subfield networks from the **DBLP** co-authorship dataset [41]: Parallel, Graphics, Security, Database, and Datamining).[2] To extract these datasets, for each sub-field, we extract the papers published in the top tier conferences[3] published in 2015 to 2019 and selected the largest connected component.

Second, we analyze regions from the **Pokec** dataset, which represents anonymized social media connections in Slovakia [40]. We sample two regions of this network, one containing all users from the Zilinsky kraj region and the second containing users from the Presovsky kraj region. We separately study these networks with respect to two attributes of interests: "gender" and "language". For the "language" attribute, we divided people into two group: those who can speak English or German and those who can not.

Third, we study the **Enron** email network [37]. This dataset represents email communications between employees of the Enron Corporation. Finally, we study **Polblog**, a dataset representing links between political blogs [1]. Nodes are labeled with the "political view" attribute (liberal or conservative).

Statistics of these datasets are provided in Table 2.

**Inferring Gender:** We infer gender of the authors in co-authorship networks and email network using Gender API[4] which has shown to perform the best among different competing libraries for inferring gender [35]. (Our earlier work in [19] used the Genderize library, which has significantly greater uncertainty associated with many names, particularly those of non-Western origin [35].)

[2] Available at https://www.aminer.cn/citation/v11.
[3] https://webdocs.cs.ualberta.ca/~zaiane/htmldocs/ConfRanking.html.
[4] https://gender-api.com/en/.

Table 2. Dataset Statistics

| Name | Description | #nodes | #edges | Assort | Attributes Group1, Group2 | Mean Degree Group1, Group2 |
|---|---|---|---|---|---|---|
| **Parallel** | Co-authorship Network | 1,251 | 4,356 | 0.06 | Men (82%), Women (18%) | Men (7.12), Women (6.25) |
| **Graphics** | Co-authorship Network | 3,525 | 10,399 | 0.06 | Men (71%), Women (29%) | Men (6.08), Women (6.26) |
| **Security** | Co-authorship Network | 1,962 | 5,976 | 0.07 | Men (80%), Women (20%) | Men (5.90), Women (5.67) |
| **Databases** | Co-authorship Network | 3,185 | 9,386 | 0.10 | Men (68%), Women (32%) | Men (5.91), Women (5.84) |
| **Data Mining** | Co-authorship Network | 2,272 | 7,643 | 0.08 | Men (66%), Women (34%) | Men (6.70), Women (6.69) |
| **Pokec-pl** | Social Network | 3,312 | 22,707 | 0.07 | Speak (84%), Don't Speak (16%) | Speak (14.0), Don't Speak (12.1) |
| **Pokec-pg** | Social Network | 3,312 | 22,707 | 0.09 | Men (50%), Women (50%) | Men (13.8), Women (13.7) |
| **Pokec-zl** | Social Network | 3,018 | 23,470 | 0.03 | Speak (85%), Don't Speak ((15%) | Speak ((16.0), Don't Speak ((13.0) |
| **Pokec-zg** | Social Network | 3,018 | 23,470 | 0.03 | Men (51%), Women (49%) | Men (15.8), Women (15.3) |
| **Enron** | Email network | 144 | 1,344 | 0.03 | Men (76%), Women (24%) | Men (18.6), Women (19.0) |
| **Polblog** | Blog Directories network | 1,224 | 16,715 | 0.81 | Liberal (48%), Conservative (52%) | Liberal (27.5), Conservative (27.1) |

For each name, Gender API provides a gender and corresponding probability (for most names, Gender API predicted a greater than 90% accuracy). The Gender API database contains 6,084,389 validated names from 189 countries and 191 languages [35]. This database is created from publicly reachable government and social media sources, allowing for high accuracy in predicting gender. (In contrast, genderize.io, which we used in our previous work [19], only supports 79 countries and 89 languages). In our datasets, we observed that gender-API had much greater ability to associate genders to Asian and South African names than did genderize.io. A detailed comparison of the usage of various gender identification API's is presented in [35].

### 5.2 Results

In this section, we provide four sets of experiments. In this set of experiments, we consider various Propagation Probabilities $pp$ and generate adjacency matrix $\mathbf{M}$ based on $pp$. $m_{ij} = pp$ if there is an edge between vertices $v_i$ and $v_j$ and $m_{ij} = 0$ otherwise (we assume that each node has the same $pp$ value).

**The IUF of different Networks**

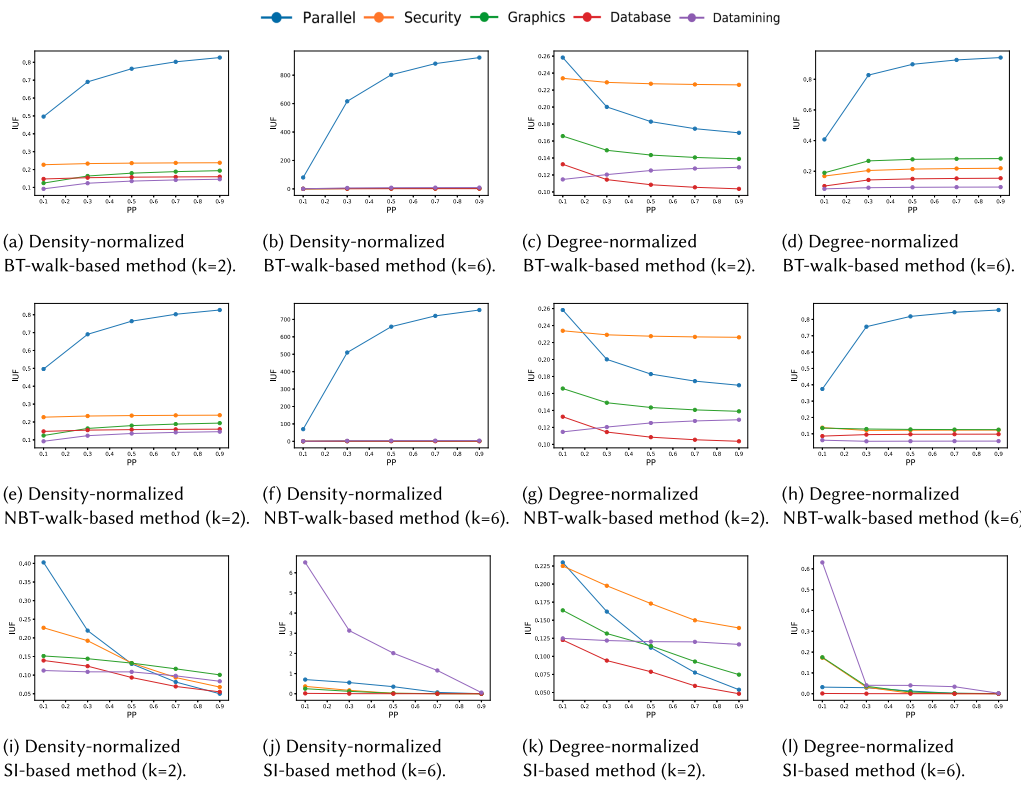
In the first experiment, we compare the IUF of different networks normalizing with respect to degree and density. We compare the five co-authorship networks to one another in one plot, and present results on other networks separately.

For co-authorship networks, we assign each node to the gender considered most likely by the Gender API library (later, in Section 5.2, we account for the probability that a node belongs to each gender).

We consider $k \in \{2, 4, 6, 10\}$ and $pp \in \{0.1, 0.3, 0.5, 0.9\}$ (we assume that each node has the same $pp$ value). We considered a maximum cascade length of 10, because cascades longer than this are not common in practice [25]. We compute both the walk-based and SI-based values of IUF. Note that these two values aim to measure two different things: in the walk-based method, one is measuring the *amount* of information flowing between two nodes, while in the SI-based method, one is measuring *whether* information is expected to flow between two nodes. The choice of method thus depends on which of these two objectives is most important.

Figures 5 and 6 show results for $k \in \{2, 6\}$ and $pp \in \{0.1, 0.3, 0.5, 0.9\}$ for the SI-based, **non-backtracking** (**NBT**) walk-based methods, and **backtracking** (**BT**) walk-based methods (our earlier version, presented in [19]). Results for $k \in \{4, 10\}$ are not shown, but were similar.

First, consider the results shown in the first two columns of these figures, which show results when normalizing by density, allowing us to compare networks of different sizes. In the walk-based method, as $pp$—the propagation probability—increases (for sufficiently large $k$), unfairness generally increases. This is because IUF is computed using the powers of $pp$ and of the adjacency matrix, so as $pp$ gets larger, differences between pairwise group flows are magnified. In contrast,



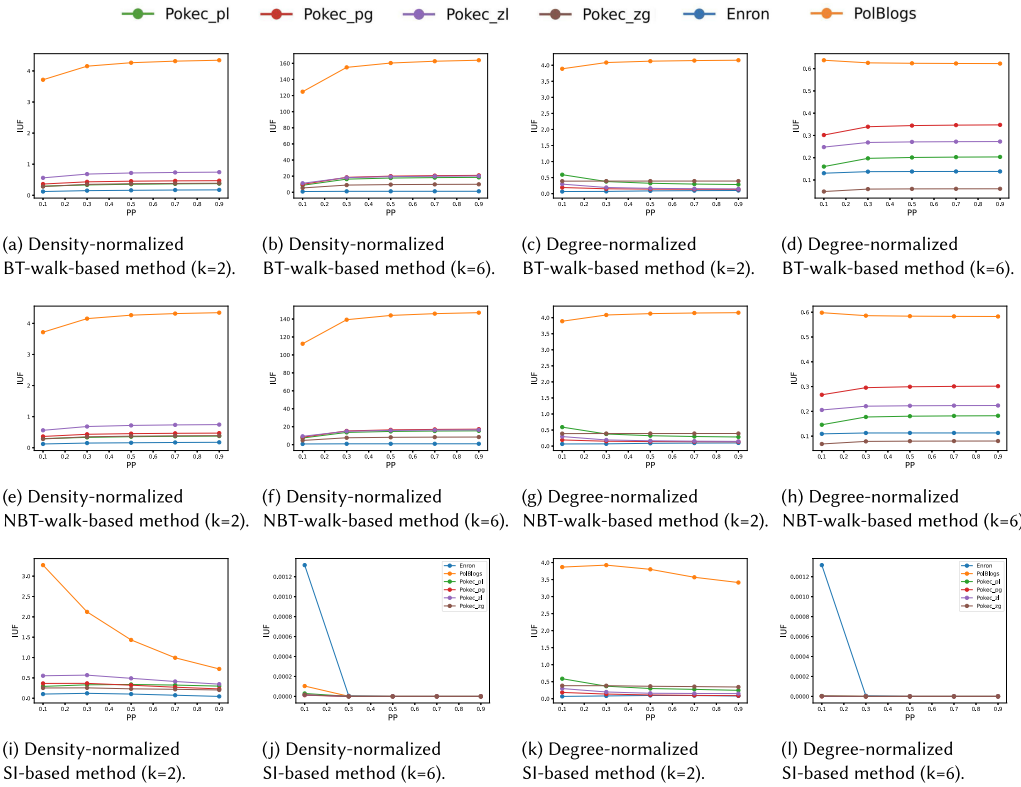

(a) Density-normalized BT-walk-based method (k=2).
(b) Density-normalized BT-walk-based method (k=6).
(c) Degree-normalized BT-walk-based method (k=2).
(d) Degree-normalized BT-walk-based method (k=6).
(e) Density-normalized NBT-walk-based method (k=2).
(f) Density-normalized NBT-walk-based method (k=6).
(g) Degree-normalized NBT-walk-based method (k=2).
(h) Degree-normalized NBT-walk-based method (k=6).
(i) Density-normalized SI-based method (k=2).
(j) Density-normalized SI-based method (k=6).
(k) Degree-normalized SI-based method (k=2).
(l) Degree-normalized SI-based method (k=6).

Fig. 5. Information unfairness results for **DBLP** subfields.

in the SI-based method, as $pp$ increases, unfairness decreases, because information spreads farther from its source, thus reducing any unfairness caused by segregation.

As $k$ increases, unfairness increases for both SI- and walk-based methods. This is because of the combinatorial explosion in the amount of flow between nodes that are near each other: because of this, as $k$ increases, the amount of flow between nodes that are near each other dramatically increases, and if nodes have any preference for connecting to others in their group, IUF will also rapidly increase. Next, using degree normalization allows us to investigate the extent to which unfairness is due to differences in degree. By examining the $y$-axes in the figures of the last two columns, we see that the values are generally lower than the $y$-axes in the figures of the first two columns indicating that much of the unfairness is due to differences in group degrees. Of the considered networks, when using the walk-based method in the co-authorship networks, the Parallel Processing network is by far the least fair. When investigating the reason for this behavior, we discovered that this network contains a very large clique, which has 38 men and only 1 woman. (There are several large cliques; this is the largest.) Due to the combinatorial explosion of walks discussed above this clique will be responsible for a huge amount of flow between men. When we removed the clique from the network the unnormalized IUF value decreased from 1,409 to 33 (for $k = 6, pp = 0.4$, with similar results at other parameter values).

The results for non-backtracking and backtracking walk-based methods are very close with slight differences. Thus, for the rest of the experiments, we include only the results for NBT walk-based methods.




(a) Density-normalized BT-walk-based method (k=2).
(b) Density-normalized BT-walk-based method (k=6).
(c) Degree-normalized BT-walk-based method (k=2).
(d) Degree-normalized BT-walk-based method (k=6).
(e) Density-normalized NBT-walk-based method (k=2).
(f) Density-normalized NBT-walk-based method (k=6).
(g) Degree-normalized NBT-walk-based method (k=2).
(h) Degree-normalized NBT-walk-based method (k=6).
(i) Density-normalized SI-based method (k=2).
(j) Density-normalized SI-based method (k=6).
(k) Degree-normalized SI-based method (k=2).
(l) Degree-normalized SI-based method (k=6).

Fig. 6. Information unfairness results for different networks.

**Statistical Significance Testing**

To investigate whether the results obtained so far are statistically significant, we perform hypothesis testing, which quantifies whether a result is likely due to random chance or to some factor of interest by computing a $p$ value in order to support or reject the null hypothesis. The smaller the $p$ value, the stronger the evidence that one should reject the null hypothesis. To perform these tests, we generate 1,000 random graphs that preserve some aspect of the original graph, and compare the IUF in these random graphs to the IUF in the observed graph. The $p$ value is then the fraction of random graphs that generated an IUF greater than the actual IUF. If the $p$ value is greater than 0.05, then the IUF of the original graph can be considered to not be statistically significant (i.e., may be explained by that particular aspect).

For the density-based significance test, we generate random graphs with the same number of nodes and edges as the original graph, using a slightly modified Erdös–Renyi model that returns a graph with exactly the desired number of edges [11]. For the degree-based significance test, the random networks have the same degree distribution as the original network. In addition to density and degree-based significance test, we computed attribute-based significance test in which we compare the IUF of the original network with IUF of networks with the same topology and protected group sizes, but in which the protected attributes are assigned to nodes at random. Our analysis is based on the idea that in network settings, small sets of nodes may have a very large effect on overall unfairness: for example, cliques and hub nodes can dramatically affect information flow. If there are few such structures, it is possible that even when attributes are assigned at random, there is a non-negligible probability that members of the same protected

Table 3. Percentage of Graphs that have Information Unfairness Higher than the Original Graph

| Dataset | Density Normalization | Degree Normalization | Attribute Normalization |
|---|---|---|---|
| **Parallel** | 0.0% | 0.0% | 11.2% |
| **Security** | 0.0% | 0.1% | 28.4% |
| **Graphics** | 0.0% | 0.2% | 43.3% |
| **Database** | 0.0% | 0.1% | 53.6% |
| **Data Mining** | 0.0% | 0.1% | 18.3% |
| **Pokec-pl** | 0.0% | 0.0% | 1.0% |
| **Pokec-pg** | 0.0% | 0.0% | 0.1% |
| **Pokec-zl** | 0.0% | 0.0% | 0.1% |
| **Pokec-zg** | 0.0% | 0.0% | 3.8% |
| **Enron** | 0.0% | 0.0% | 36.3% |
| **Polblog** | 0.0% | 0.0% | 0.0% |

group are—purely by coincidence—are assigned to a disproportionate number of such "important' nodes. For the attribute-based significance test, the topology of the random networks is the same as the original network (the same nodes and edges) but attributes in the graph are shuffled.

Table 3 shows the fraction of random networks with IUF is greater than the IUF of original network for the walk-based method using $k = 4$ and $pp = 0.5$. (This procedure is too slow for the SI-based model, so we used the faster, more tractable non-backtracking walk-based method).

From Table 3, we can observe that the results are statistically significant when accounting for density and degree ($p$ values are 0 or close to 0 for density and degree, respectively). This shows that IUF cannot be explained entirely by the density and degree of the graph.

However, in many cases, we observe a large fraction of graphs having higher IUF than the original when we randomly assign attributes to nodes. This result is very interesting, because it shows even when shuffling the attributes randomly, there is a very good chance that the IUF could be at least as high as that in the observed graph! This indicates that there is something inherent in the graph's topology, not considering attribute distribution, that makes unfairness very likely to occur. As discussed before, this may be due to cliques (as in the Parallel network) or a very skewed degree distribution.

**Analysis of Joint Attribute Accessibility Distributions**

In order to further drill down into why unfairness occurs in different networks, we compare the mean value for different joint attribute accessibility distributions (Group1-Group1, Group1-Group2, and Group2-Group2: because the network is undirected, the Group2-Group1 distribution is the same as Group1-Group2 distribution) using both the walk- and SI-based methods. Table 4 shows these results for $k = 4$ and $pp = 0.5$ (the pattern for other values were similar) and density based normalization (allowing us to compare networks of different sizes).

Interestingly, in studying fairness with respect to gender, in co-authorship networks, only in Parallel Processing network is the Group1-Group1 (Men–Men) flow higher than the Group2-Group2 (Women–Women) flow! For the Parallel Processing co-authorship network, as discussed before, this is largely due to the presence of a large, almost entirely male clique in the Parallel Processing network and because, as shown in Table 2, men in this network have a substantially higher mean degree than women. In contrast, for the other networks, Women–Women flow is slightly greater than Men-Men flow. In other networks, however, the Group1-Group1 (Men–Men) flow is always higher than the Group2-Group2 (Women–Women) flow.

As the mean values for joint distributions in many cases were close to each other, to determine whether the differences between means are meaningful, we perform the $t$-test for the means of each pair of joint attribute distributions. We use the $ttest - ind$ function from the Python SciPy library, which performs a two-sided test for the null hypothesis that two independent samples have

Table 4. Mean Value for Different Joint Attribute Accessibility Distribution: Group1-Group1, Group1-Group2, Group2-Group2

| | Walk | | | SI | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Group1-Group1** | **Group1-Group2** | **Group2-Group2** | **Group1-Group1** | **Group1-Group2** | **Group2-Group2** |
| **Parallel** | 26.6 | 6.63 | 4.05 | 0.21 | 0.20 | 0.21 |
| **Security** | 2.26 | 2.45 | 2.71 | 0.40 | 0.42 | 0.48 |
| **Graphics** | 2.64 | 2.69 | 3.18 | 0.66 | 0.67 | 0.73 |
| **Database** | 3.07 | 3.23 | 3.52 | 0.47 | 0.48 | 0.52 |
| **Data Mining** | 4.39 | 4.71 | 5.04 | 0.80 | 0.82 | 0.85 |
| **Pokec-pl** | 7.94 | 6.19 | 5.45 | 0.58 | 0.48 | 0.41 |
| **Pokec-pg** | 8.98 | 7.19 | 6.17 | 0.59 | 0.54 | 0.51 |
| **Pokec-zl** | 7.55 | 5.69 | 4.28 | 0.44 | 0.40 | 0.36 |
| **Pokec-zg** | 7.80 | 6.89 | 6.34 | 0.44 | 0.43 | 0.42 |
| **Enron** | 2.28 | 2.01 | 1.77 | 0.02 | 0.02 | 0.02 |
| **Polblog** | 33.07 | 9.43 | 21.25 | 0.03 | 0.03 | 0.03 |

Table 5. T-test Results for Different Pairs of Joint Attribute Accessibility Distribution: Group1-Group1 to Group1-Group2 (g1-g1 to g1-g2), Group1-Group1 to Group2-Group2 (g1-g1 to g2-g2), and Group1-Group2 to Group2-Group2 (g1-g2 to g2-g2)

| | Walk | | | SI | | |
|---|---|---|---|---|---|---|
| **Dataset** | **g1-g1 to g1-g2** | **g1-g1 to g2-g2** | **g1-g2 to g2-g2** | **g1-g1 to g1-g2** | **g1-g1 to g2-g2** | **g1-g2 to g2-g2** |
| **Parallel** | 0.00 | 0.00 | 0.26 | 0.13 | 0.09 | 0.03 |
| **Security** | 0.00 | 0.00 | 0.08 | 0.00 | 0.00 | 0.00 |
| **Graphics** | 0.14 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **Database** | 0.06 | 0.02 | 0.16 | 0.20 | 0.00 | 0.00 |
| **Data Mining** | 0.00 | 0.00 | 0.05 | 0.00 | 0.00 | 0.06 |
| **Pokec-pl** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **Pokec-pg** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **Pokec-zl** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **Pokec-zg** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **Enron** | 0.00 | 0.00 | 0.01 | 0.13 | 0.01 | 0.07 |
| **Polblog** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

identical average values. Table 5 shows the $p$ value for this test between each pair of joint attribute distributions using density-based normalization. The results show that in all cases the $p$-value for comparing Men–Men and Women–Women flow is below 0.05, indicating statistical significance.

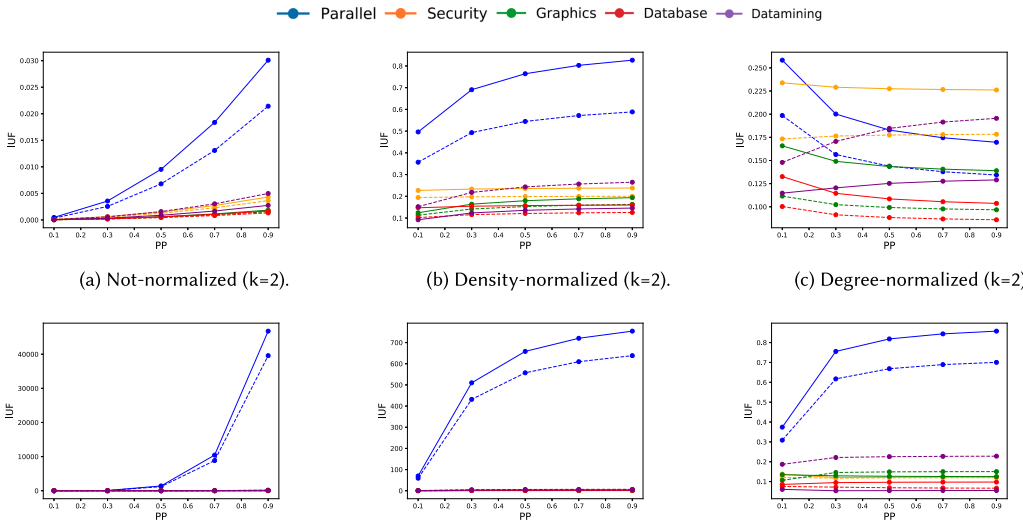
**Accounting for Error Associated with Inferring Gender**

Recall that in our co-authorship datasets, we inferred gender using the Gender API library. This library associates a probability or confidence with each inference. In order to explore the account for these probabilities, we compute a version of IUF in which the group membership matrix is non-binary, allowing an individual to partially belong to both groups (weights sum to one). We compare the resulting IUF to the IUF values computed before, using a binary group membership matrix.

Figure 7 shows the results for $k = 2$ and $k = 6$ for walk-based method, with solid lines showing the original IUF values, and dashed lines showing the recomputed IUF values. In most cases IUF does not change dramatically (and certainly, patterns stay the same). In the recomputed version of IUF, unfairness generally decreases. This is because, in some sense, group memberships are 'flattened out', reducing disparities between pairwise group flows. For example, in the Parallel Processing dataset (which has the greatest decrease), the large clique that was responsible for high Men–Men flow now has some Women–Women flow associated with that clique.

**Validating Normalization**

One important aspect of our IUF computation is the normalization step. Recall that in this step, to compare the flow between each pair of nodes in the actual network to the flow between the




(a) Not-normalized (k=2). (b) Density-normalized (k=2). (c) Degree-normalized (k=2).

(d) Not-normalized (k=6). (e) Density-normalized (k=6). (f) Degree-normalized (k=6).

Fig. 7. Information unfairness results for **DBLP** subfields. Dashed lines show information unfairness value for weighted version.

same nodes expected in a random graph sharing some properties with the original network (e.g., density or degree distribution), we perform element-wise division of accessibility matrix of original network **A** by the expected accessibility matrix of random network **S**.

However, properly computing matrix **S** as the average of all possible accessibility matrices is computationally infeasible (except in certain very limited cases), as one would need to generate every graph with the desired properties, and then compute its accessibility matrix. Instead, we computed a single accessibility matrix corresponding to the average of the adjacency matrices for all possible graphs. The distinction here is subtle but important, and so it is useful to understand the extent to which our simplification changes the outcome.

In this section, we compare IUF results computed using our approximation to the results obtained by normalizing using the average accessibility matrix of 1,000 random graphs. Table 6 shows results for for $k = 4$ and $pp = 0.5$ ("Estimated IU" represents the original results and "Actual IU" shows the recomputed values), and while there are some differences, values are generally similar.

### 5.3 Discussion

Observation 1 (All networks exhibit some degree of unfairness). *All networks exhibit non-zero* IUF*, regardless of normalization. However, this unfairness occurs due to different reasons.*

When studying fairness with respect to gender attribute, in most of the co-authorship networks (except for Parallel Processing network), Women–Women flow is the highest. Exploring the sociological causes for such behavior is outside the scope of this paper; one possibility is that efforts to build mentorship and other networks among women have been successful. In the Parallel Processing network and in other datasets (Pokec and Enron) on the other hand, women are disadvantaged: they receive less information than men both from men and from other women.

Observation 2 (As *pp* increases, walk-based IUF tends to increase, and SI-based IUF tends to decrease). *This occurs because these two different ways of measuring information flow are measuring different things. When the accessibility matrix is computed using walks, we are measuring*

Table 6. Comparison Results of Actual vs. Estimated Normalization

| Dataset | Density | | Degree | |
|---|---|---|---|---|
| | **Estimated IU** | **Actual IU** | **Estimated IU** | **Actual IU** |
| **Parallel** | 22.60 | 22.16 | 0.10 | 0.12 |
| **Security** | 0.46 | 0.47 | 0.11 | 0.14 |
| **Graphics** | 0.54 | 0.43 | 0.15 | 0.17 |
| **Database** | 0.45 | 0.38 | 0.08 | 0.10 |
| **Data Mining** | 0.65 | 0.61 | 0.04 | 0.04 |
| **Pokec-pl** | 2.49 | 2.51 | 0.08 | 0.09 |
| **Pokec-pg** | 2.80 | 2.83 | 0.18 | 0.20 |
| **Pokec-zl** | 3.27 | 3.28 | 0.16 | 0.18 |
| **Pokec-zg** | 1.46 | 1.48 | 0.04 | 0.04 |
| **Enron** | 0.51 | 0.49 | 0.10 | 0.10 |
| **Polblog** | 23.64 | 23.76 | 0.85 | 0.89 |

*the amount of information flowing between pairs of nodes, while when computed using an SI-type contagion, we are measuring whether information is expected to flow between each pair of nodes. In the former method, we observe a combinatorial explosion, where as pp increases, the amount of flow between nodes that are near one another increases dramatically. If nodes tend to connect with others with the same attribute, then this further increases unfairness. In contrast, with the SI-based method, as pp increases, information travels farther from its source, decreasing unfairness.*

Note that there are some exceptions to this: for example, in the degree-normalized version, for low $k$ values, IUF *decreases* as $pp$ increases. For large $k$, cascades are able to travel farther from the originating node, and the local effects of homophily are diminished. Note, though, that this can only happen for large $pp$ (because at small $pp$, even if $k$ allows for long cascades, in practice very little information will travel far from the source). However, even for large $pp$, this effect is countered by the combinatorial explosion of cascades (walks) that stay in the local neighborhood of the originating node. For each network, we observe some "balance point" between $k$ and $pp$ where cascades can grow long enough to overcome the local effects of homophily, but are not so long as to encounter such combinatorial explosion. With such cascades, information unfairness decreases as $pp$ increases.

Observation 3 (Degree account for a large part of the networks' information unfairness). *The density-normalized* IUF *values, which account only for the size of the graph, are much higher than* IUF *values computed when normalizing with respect to degree. respectively.*

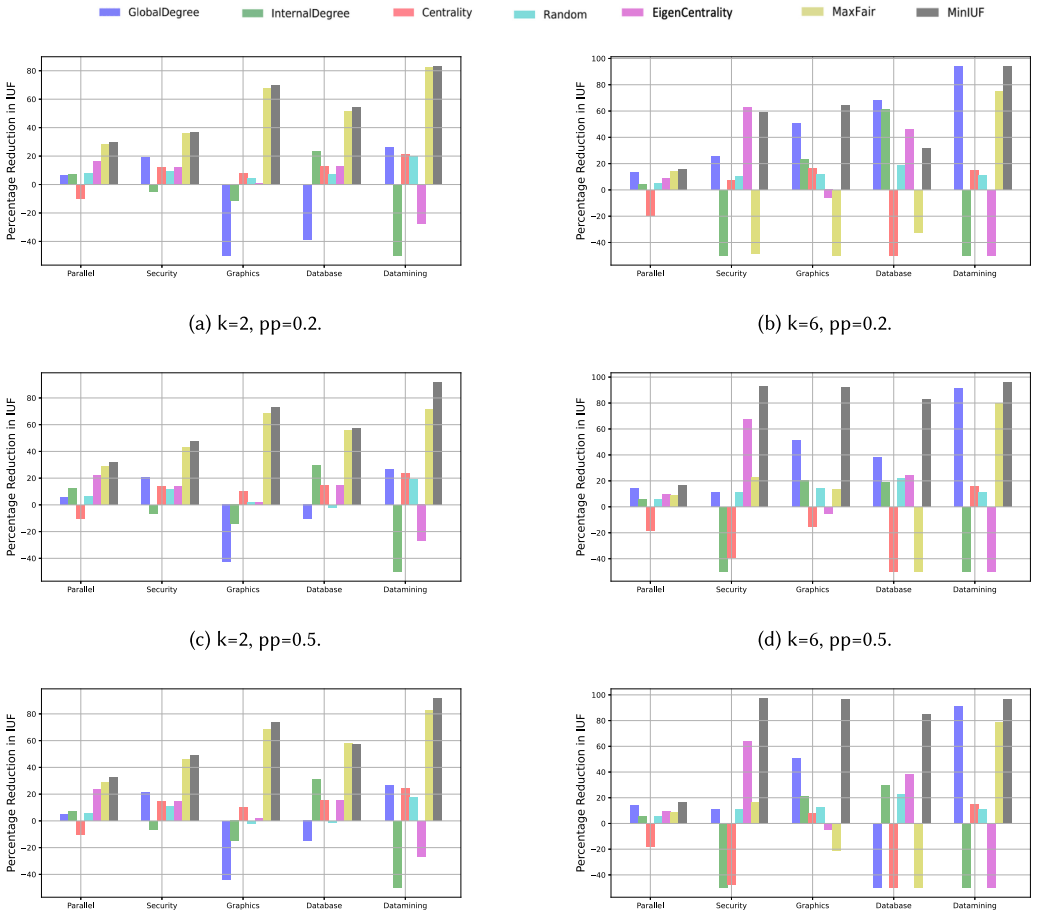
*5.3.1 Performance of* MinIUF. In this section, we first compare the performance of MinIUF at decreasing IUF compared to baseline methods, and then we show the trade-off between accuracy and running time for MinIUF.

**Comparison Results**

In this section, we evaluate the performance of MinIUF against six baseline methods: *MaxFair* (our earlier method for the same problem), *AttributeCentrality*, *GlobalDegree*, *InternalDegree*, *Centrality*, and *Random*.

All of the baseline methods use the same approach as MinIUF in the sense that they score candidate edges and, in each iteration, at the highest-scoring edge(s):

(1) *MaxFair* computes scores using the attribute-based centrality vector from the accessibility matrix, as described in [19]. Attribute centrality matrix $\mathbf{AM}_{n\times l} = \mathbf{PM} + \mathbf{A}_{(k)} \cdot \mathbf{PM}$. Here, $\mathbf{PM}$ is the protected group membership matrix and $\mathbf{A}$ is accessibility matrix. Then, $score(u_i, u_j) = \Sigma_{fg} \mathbf{S}_{fg} \cdot (am_{if} \cdot am_{jg} + am_{ig} \cdot am_{jf})$, where $\mathbf{S}_{fg} = mean(\mathbf{FM} - \mathbf{FM}_{fg})$ and $\mathbf{FM}$ is flow matrix and $f, g \in \{1, \ldots, l\}$.




(a) k=2, pp=0.2.

(b) k=6, pp=0.2.

(c) k=2, pp=0.5.

(d) k=6, pp=0.5.

(e) k=2, pp=0.7.

(f) k=6, pp=0.7.

Fig. 8. Percentage improvement on unfairness reduction on co-authorship datasets.

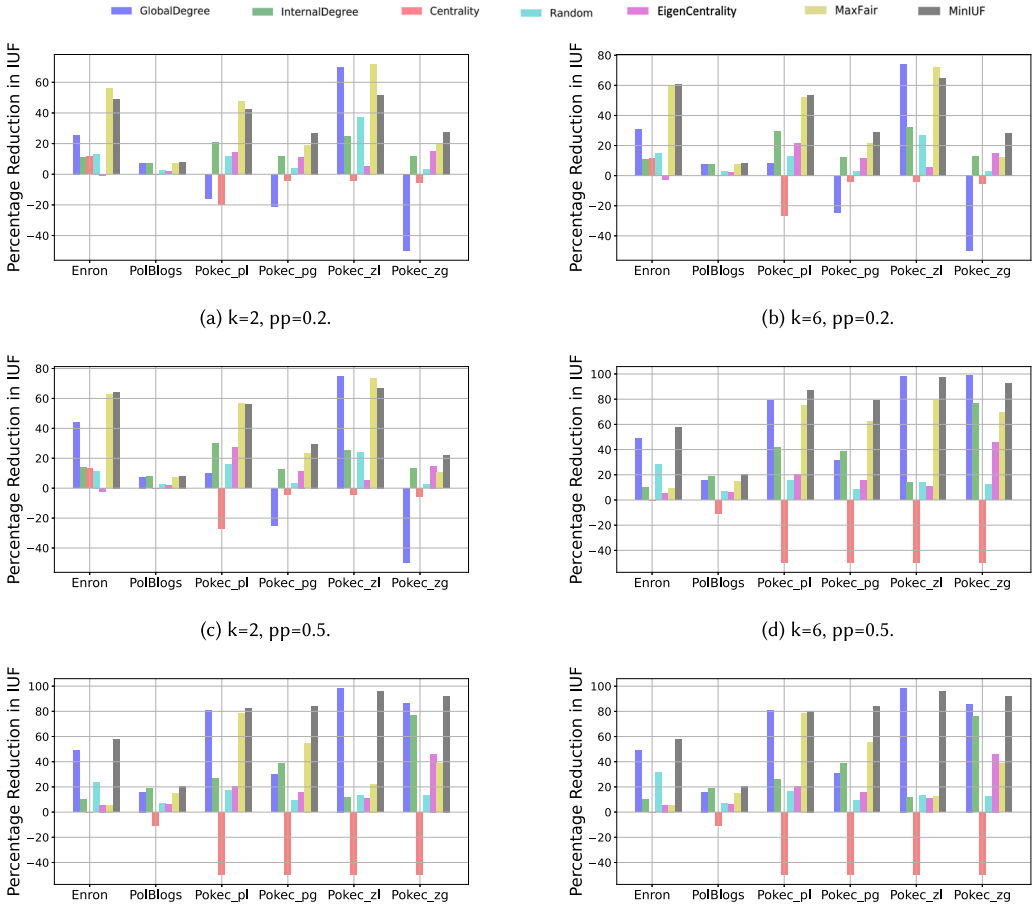
(2) *EigenCentrality* uses the same approach as *MaxFair*, except that it uses attribute-based eigen centrality matrix (**EM**) instead of attribute-based centrality. $\mathbf{EM}_{n\times l} = \mathbf{SM} \cdot \mathbf{PM}$. Here, **SM** is Sum Matrix, an estimation of sum over walks of different length $k$ based on Katz-style centrality [22] which is the basis of eigen-vector centrality. $\mathbf{SM} = \mathbf{M}+\mathbf{M}^2+\ldots = (I-\alpha\mathbf{M})^{-1}-I$, where $0 \leq \alpha \leq \rho(\mathbf{M})^{-1} - I$ and $\rho(\mathbf{M})$ is the spectral radius of **M**. By multiplying **M** by $\alpha$ we ensure that the geometric series generated by **M** converges [15].

(3) *InternalDegree* uses the same approach as *MaxFair*, except that it uses the degree of each node with respect to each group $C_f$ instead of attribute centrality.

(4) *GlobalDegree* and *Random* first identify the joint attribute accessibility distribution $D_{fg}$ that has the lowest mean. *GlobalDegree* selects two nodes from protected groups $P_f$ and $P_g$ that are not connected and have the highest degree product. Random selects two nodes from protected groups $P_f$ and $P_g$ that are not connected at random.

(5) Finally, *Centrality* selects the node pairs with the highest eigenvector centrality product.

Note that the latter 4 methods were used as baselines in [19].




(a) k=2, pp=0.2.

(b) k=6, pp=0.2.

(c) k=2, pp=0.5.

(d) k=6, pp=0.5.

(e) k=2, pp=0.7.

(f) k=6, pp=0.7.

Fig. 9. Percentage improvement on unfairness reduction for different networks.

For all networks, we set $b = 0.01 \times |E|$ where $E$ is the set of edges in the network and set $x = 10$. Figures 8 and 9 shows results for $k \in \{2, 6\}$ and $pp \in \{0.2, 0.5, 0.8\}$ normalized by density, allowing us to evaluate the performance of MinIUF across different parameters. The results show that in almost all cases, MinIUF outperforms baseline methods. Moreover, adding only a few edges (1% of the existing edges in the network) can decrease unfairness dramatically: for $k = 2$ the unfairness increases by almost 40% and for or $k = 6$ the unfairness increases by 80%–90% in most cases.

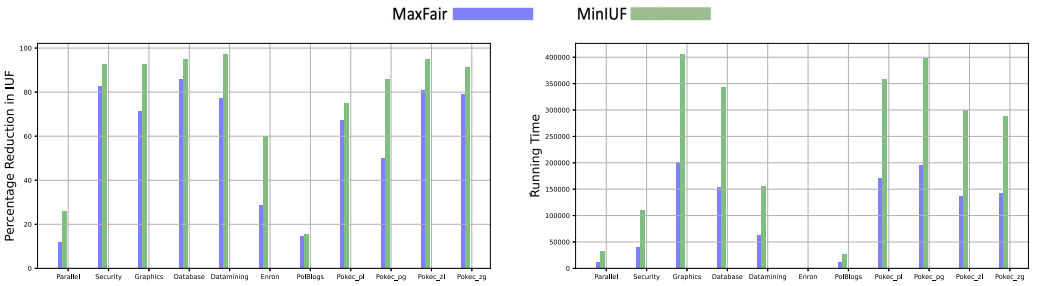
**Running Time**

The main contributors to MinIUF's running time is the score computation and re-computation. In the above experiments, we set $x$ (the number of re-computations) to be 10. Now, we consider $x \in \{4, 8, 16, 32\}$, and show the percentage improvement on IUF and the corresponding running time in seconds. Table 7 shows the results for $k = 4$ and $pp = 0.5$ (Experiments run on a 2020 MacBook Pro with Apple M1 processor). On these networks, setting $x \geq 8$ appears to be sufficient, with no major improvements in performance for larger values of $x$.

Next, we compare the running time of MinIUF with MaxFair. As the results in Figures 8 and 9 show, in most cases, MinIUF outperforms *MaxFair* with respect to reducing information unfairness.

Table 7. Percentage Improvement (PI) over $IUF$ and its Corresponding Running Time in Seconds for Different $x$ Values

| Dataset | | x = 4 | x = 8 | x = 16 | x = 32 |
|---|---|---|---|---|---|
| **Parallel** | **Percentage Improvement** | 24% | 26% | 25% | 25% |
| | **Running Time** | 147 | 326 | 1,015 | 2,034 |
| **Graphics** | **Percentage Improvement** | 0% | 93% | 96% | 96% |
| | **Running Time** | 2,585 | 4,059 | 8,484 | 15,510 |
| **Security** | **Percentage Improvement** | 55% | 93% | 93% | 93% |
| | **Running Time** | 557 | 1,097 | 2,318 | 6,878 |
| **Data Mining** | **Percentage Improvement** | 57% | 97% | 98% | 98% |
| | **Running Time** | 960 | 1,548 | 3306 | 6,287 |
| **Database** | **Percentage Improvement** | 14% | 85% | 94% | 94% |
| | **Running Time** | 2,173 | 3,434 | 6801 | 15,586 |
| **Pokec-pl** | **Percentage Improvement** | 67% | 75% | 80 % | 83% |
| | **Running Time** | 1995 | 3,592 | 6252 | 11,847 |
| **Pokec-pg** | **Percentage Improvement** | 76% | 86% | 87% | 86% |
| | **Running Time** | 2,284 | 3,979 | 6363 | 12,184 |
| **Pokec-zl** | **Percentage Improvement** | 93% | 95% | 96% | 96% |
| | **Running Time** | 1,832 | 2992 | 5266 | 9,740 |
| **Pokec-zg** | **Percentage Improvement** | 53% | 91% | 92% | 93% |
| | **Running Time** | 1,826 | 2,877 | 5105 | 7,867 |
| **Enron** | **Percentage Improvement** | 64% | 60% | 60% | 60% |
| | **Running Time** | 2.5 | 7.2 | 7.1 | 7.0 |
| **Polblog** | **Percentage Improvement** | 16% | 16% | 16% | 16% |
| | **Running Time** | 135 | 266 | 731 | 1,410 |




(a) Improvement.

(b) Running Time.

Fig. 10. Running time versus accuracy of MinIUF for different $x$ values.

Asymptotically, the big-O time complexity for *MaxFair* is the same as `MinIUF`, because constructing the attribute centrality matrix takes $O(kn^3)$ and computing scores for each pairs of nodes takes $O(l^2n^2)$, resulting in an overall time complexity is $O(x(kn^3+l^2n^2))$. However, the absolute number of operations needed for `MinIUF` is more than `MaxFair`, and so it is slower in practice. `MaxFair` performs well, though not as well as `MinIUF`, and so may be a reasonable choice if running time is an issue. Figure 10 shows the trade-off between running time and percentage reduction in IUF for different datasets ($k = 4$, $pp = 0.5$, $x = 8$ and normalized by density). As the results show, while `MaxFair` is faster, `MinIUF` performs better.

## 6 CONCLUSION AND FUTURE WORK

In this work, we introduced `IUF`, a generalized version of the `Information Unfairness` from [19], which measures the extent to which information flows to individuals from different protected

groups equally. Next, we showed how to reduce unfairness of these networks by adding a specified number of edges using the new MinIUF algorithm. We performed an in depth analysis of different networks, and analyzed the causes of unfairness in these networks. We showed that MinIUF is capable of reducing $IUF$ dramatically, and that it outperforms baseline methods under varied conditions. In future work, we would like to extend our work to study fair distribution of correct information with the presence of misinformation and disinformation in the network.

## REFERENCES


[1] Lada A. Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 US election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*. 36–43.

[2] Francesca Arrigo, Desmond J. Higham, and Vanni Noferini. 2019. Non-backtracking alternating walks. *SIAM Journal on Applied Mathematics* 79, 3 (2019), 781–801.

[3] Abhijit Banerjee, Arun G. Chandrasekhar, Esther Duflo, and Matthew O. Jackson. 2014. *Gossip: Identifying Central Individuals in a Social Network*. Technical Report. National Bureau of Economic Research.

[4] Pablo Barberá, John T. Jost, Jonathan Nagler, Joshua A. Tucker, and Richard Bonneau. 2015. Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological Science* 26, 10 (2015), 1531–1542.

[5] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2017. Fairness in machine learning. *NIPS Tutorial* 1 (2017).

[6] Hannah C. Beilinson, Nasanbayar Ulzii-Orshikh, Ashkan Bashardoust, Sorelle A. Friedler, Carlos E Scheidegger, and Suresh Venkatasubramanian. 2020. Clustering via information access in a network. arXiv:2010.12611. Retrieved from https://arxiv.org/abs/2010.12611.

[7] Daniel J. Brass, Kenneth D. Butterfield, and Bruce C. Skaggs. 1998. Relationships and unethical behavior: A social network perspective. *Academy of Management Review* 23, 1 (1998), 14–31.

[8] Giulia Cencetti and Federico Battiston. 2019. Diffusive behavior of multiplex networks. *New Journal of Physics* 21, 3 (2019), 035006.

[9] Biao Chang, Tong Xu, Qi Liu, and En-Hong Chen. 2018. Study on information diffusion analysis in social networks and its applications. *International Journal of Automation and Computing* 15, 4 (2018), 377–401.

[10] Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. 2019. Recommending links through influence maximization. *Theoretical Computer Science* 764 (2019), 30–41.

[11] Paul Erdős and Alfréd Rényi. 1960. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5, 1 (1960), 17–60.

[12] Benjamin Fish, Ashkan Bashardoust, Danah boyd, Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2019. Gaps in information access in social networks. In *Proceedings of the WWW*.

[13] 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.

[14] Jacob Goldenberg, Barak Libai, and Eitan Muller. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters* 12, 3 (2001), 211–223.

[15] Peter Grindrod, Desmond J Higham, and Vanni Noferini. 2018. The deformed graph Laplacian and its applications to network centrality analysis. *SIAM Journal on Matrix Analysis and Applications* 39, 1 (2018), 310–341.

[16] Yosh Halberstam and Brian Knight. 2016. Homophily, group size, and the diffusion of political information in social networks: Evidence from Twitter. *Journal of Public Economics* 143 (2016), 73–88.

[17] Luke Holman, Devi Stuart-Fox, and Cindy E. Hauser. 2018. The gender gap in science: How long until women are equally represented? *PLoS Biology* 16, 4 (2018), e2004956.

[18] Herminia Ibarra. 1997. Paving an alternative route: Gender differences in managerial networks. *Social Psychology Quarterly* 60, 1 (1997), 91–102.

[19] Zeinab S. Jalali, Weixiang Wang, Myunghwan Kim, Hema Raghavan, and Sucheta Soundarajan. 2020. On the information unfairness of social networks. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 613–521.

[20] S. Jones. 2017. White men account for 72% of corporate leadership at 16 of the Fortune 500 companies. *Fortune Magazine* (2017).

[21] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. 2018. Homophily influences ranking of minorities in social networks. *Scientific Reports* 8, 1 (2018), 1–12.

[22] Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1 (1953), 39–43.

[23] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 137–146.

[24] Min-Ah Lee and Kenneth F Ferraro. 2007. Neighborhood residential segregation and physical health among Hispanic Americans: Good, bad, or benign? *Journal of Health and Social Behavior* 48, 2 (2007), 131–148.
[25] Jure Leskovec, Lada A Adamic, and Bernardo A. Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)* 1, 1 (2007), 5.
[26] Weifeng Liu and Brian Vinter. 2014. An efficient GPU general sparse matrix-matrix multiplication for irregular data. In *Proceedings of the 2014 IEEE 28th International Parallel and Distributed Processing Symposium*. IEEE, 370–381.
[27] Jyoti Sunil More and Chelpa Lingam. 2019. A SI model for social media influencer maximization. *Applied Computing and Informatics* 15, 2 (2019), 102–108.
[28] Buddhika Nettasinghe, Nazanin Alipourfard, Vikram Krishnamurthy, and Kristina Lerman. 2021. Emergence of structural inequalities in scientific citation networks. arXiv:2103.10944. Retrieved from https://arxiv.org/abs/2103.10944.
[29] Mark E. J. Newman. 2003. Mixing patterns in networks. *Physical Review E* 67, 2 (2003), 026126.
[30] Mark E. J. Newman. 2006. Modularity and community structure in networks. *PNAS* 103, 23 (2006), 8577–8582.
[31] Gary Orfield and Chungmei Lee. 2005. Why segregation matters: Poverty and educational inequality. *Civil Rights Project at Harvard University* (2005).
[32] Jorge L. Reyes-Ortiz, Luca Oneto, and Davide Anguita. 2015. Big data analytics in the cloud: Spark on hadoop vs mpi/openmp on beowulf. *Procedia Computer Science* 53 (2015), 121–130.
[33] Paul R. Rosenbaum and Donald B. Rubin. 1984. Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American Statistical Association* 79, 387 (1984), 516–524.
[34] Yousef Saad. 2011. *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. Vol. 66. Siam.
[35] Lucía Santamaría and Helena Mihaljević. 2018. Comparison and benchmark of name-to-gender inference services. *PeerJ Computer Science* 4 (2018), e156.
[36] Farzan Masrour Shalmani. 2021. *Fairness in Social Network Analysis: Measures and Algorithms*. Ph. D. Dissertation. Michigan State University.
[37] Jitesh Shetty and Jafar Adibi. 2004. The Enron email dataset database schema and brief statistical report. *Information Sciences Institute Technical Report, University of Southern California* 4, 1 (2004), 120–128.
[38] Eithel M. Simpson, Thelma Snuggs, Tim Christiansen, and Kelli E Simples. 2000. Race, homophily, and purchase intentions and the black consumer. *Psychology & Marketing* 17, 10 (2000), 877–889.
[39] Ana-Andreea Stoica, Christopher Riederer, and Augustin Chaintreau. 2018. Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity. In *Proceedings of the WWW*. 923–932.
[40] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, Vol. 1. Present Day Trends of Innovations Lamza Poland.
[41] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the KDD*. 990–998.
[42] Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, and Christos Faloutsos. 2012. Gelling, and melting, large graphs by edge manipulation. In *Proceedings of the CIKM*.
[43] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. 2019. Group-fairness in influence maximization. In *Proceedings of the IJCAI*.
[44] Xindi Wang, Onur Varol, and Tina Eliassi-Rad. 2021. Information access equality on network generative models. *Applied Network Science* 7, 54 (2022).
[45] Fangcao Xu, Bruce Desmarais, and Donna Peuquet. 2020. STAND: A spatio-temporal algorithm for network diffusion simulation. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoSpatial Simulation*. 20–29.

Received 1 September 2021; revised 16 August 2022; accepted 9 December 2022