Universiteit Leiden

Structurally Optimizing the Fairness of Graph Centrality Measures

Cesar Gesta (s2938510)

Supervisors:
Rajat Hazra (MI) & Akrati Saxena (LIACS)

Bachelor Thesis– Wiskunde & Informatica

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

June 30, 2026

Abstract

Centrality measures score the importance of nodes in a network. These scores may reflect structural biases present in the underlying network. This thesis studies the fairness of graph centrality measures from an algorithmic and theoretical perspective. We focus on the degree, eigenvector, Katz and PageRank centrality.

We first review existing notions of algorithmic fairness and fairness notions for centrality measures. We then select two PageRank fairness objectives for fairness optimization: fairness loss and PR-unfairness. Fairness loss measures the mean squared error between the PageRank mass assigned to groups and a target distribution, while PR-unfairness considers whether a protected group receives at least a prescribed amount of PageRank mass. For both fairness objectives, we develop greedy algorithms that improve fairness by applying a limited number of edge operations. Edge operations include edge addition, edge deletion and edge rewiring. Since exhaustive search over all possible graphs is impractical, the algorithms use heuristic candidate generation and evaluate only a subset of promising candidates. Experiments on several real-world networks show that the proposed methods can improve PageRank fairness, although their effectiveness depends on the graph structure and the allowed edge operation types.

We also develop a graphon framework for fairness estimation. We define graphon analogues of fairness notions for centrality measures and study when fairness computed on finite graphs converges to graphon fairness. This gives a limiting interpretation of fairness in large graphs that are sampled from a latent graphon. As an application, we analyze two-block stochastic block model graphons and show that, under suitable assumptions, proportional fairness of degree, eigenvector, Katz and PageRank centrality is equivalent to equality of the two block degrees.

Together, these results show that PageRank fairness can be improved through local structural modifications, and that graphons provide a useful theoretical framework for understanding fairness estimation in large networks.

Contents

Chapter I

Introduction and Preliminaries

In this chapter, we introduce the thesis and provide the required preliminaries for the rest of the thesis.

I.1 Introduction

Networks are ubiquitous in society. At the same time, algorithms are increasingly used in important decision-making processes, often using real-world network data. Generally, such networks are modeled as a graph. Such data may contain structural biases towards (groups of) individuals and it is important that algorithms do not reflect this bias. This motivates the need for algorithms to operate *fairly* on network data. However, what is "fair" is often not universally decided.

An important set of network algorithms are centrality measures. These are algorithms that take a network, modeled as a graph, and produce a real-valued score for all individuals (nodes) in the network; the centrality of each individual. Hence, centrality measures produce a ranking of (sets of) individuals. This ranking may then in turn be used to identify influential individuals or assign resources to individuals or groups. Examples of centrality measures include the degree, eigenvector, Katz and PageRank centrality. PageRank is a particularly widely studied centrality measure that assigns importance to nodes based on the importance of nodes linking to it, together with a base score for each node.

However, when applying a centrality measure to biased network data, the resulting ranking may also be biased. This is problematic when centrality measures are used in decision-making processes. Hence, one may ask whether the centrality is *fair*, for example with respect to a target value, like the population share. This question leads to the study of the fairness of centrality measures.

In addition to optimizing fairness on a fixed observed graph, one may ask whether the observed fairness value is specific to that particular graph, or whether it reflects an underlying random graph model. Graphons provide a mathematical framework for studying such questions, as they can be viewed as limiting objects for sequences of dense graphs.

In this thesis, we focus on the fairness of the degree, eigenvector, Katz and PageRank centrality. Specifically, we develop fairness optimization algorithms for the PageRank centrality, studying two notions of fairness: fairness loss and PR-unfairness. For both notions, we develop greedy optimization methods that aim to improve fairness by modifying the graph through edge operations.

Specifically, we consider edge addition, edge deletion, and edge rewiring under a limited operation budget.

We also study fairness estimation from a graphon perspective. By defining centrality measures and fairness notions for graphons, we can study whether the fairness value computed on a large finite graph converges to a graphon fairness value. This provides a theoretical framework for understanding finite graph fairness convergence. Although classical graphons are best suited to dense graphs, they still provide a useful first theoretical framework. Sparse extensions would require additional tools.

Therefore, this thesis has two main contributions. First, we develop PageRank fairness optimization algorithms. Second, we develop a theoretical graphon-based framework for studying the convergence of finite graph fairness.

I.1.1 Thesis overview

This thesis is divided into several chapters. Chapter I (the current chapter) contains the introduction and the preliminaries needed throughout the remainder of the thesis. Chapter II starts with a literature review of fairness definitions. Then, the developed PageRank fairness optimization algorithms are discussed. The chapter ends with an experimental evaluation of these algorithms. Chapter III develops a graphon framework for fairness estimation. Chapter IV discusses the results and limitations of the thesis, outlines directions for future work and concludes.

I.2 Preliminaries

In this section, we provide the preliminary mathematical background needed throughout the remainder of the thesis. We first recall basic graph-theoretic definitions, including the edge operations used in the optimization algorithms. We then define centrality measures, and finally discuss graphons. For the latter part, we assume the reader has basic measure theory knowledge.

Definition I.2.1 (Graph)**.** A *graph* is a pair $G = (V, E)$, where V is a set of *nodes* and E is a set of pairs of nodes, called *edges*. If E consists of ordered pairs of nodes we call G *directed* and for an edge between nodes $u, v \in V$ write $(u, v) \in E$. If E consists of unordered pairs of nodes we call G *undirected* and for an edge between nodes $u, v \in V$ write $\{u, v\} \in E$.

Given a graph $G = (V, E)$, a *group* of nodes is any subset $V' \subseteq V$.
If G is *finite*, we often label the nodes as $V = \{1, 2, \ldots, n\}, n \in \mathbb{N}$ and write $|V| = n \in \mathbb{N}, |E| = m \in \mathbb{N}$. Unless otherwise stated, we will always work with finite graphs.
Given a graph $G = (V, E)$ with $|V| = n, V = \{1, 2, \ldots, n\}$, the *adjacency matrix* $A = (A_{ij})_{1 \leq i,j \leq n}$ of G is an $n \times n$ matrix given by

$$A_{ij} = \begin{cases} 1 & \text{if an edge exists between nodes } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases}$$

Unless otherwise stated, graphs in the preliminaries are undirected. However, the PageRank optimization algorithms in Chapter II are applied to directed graphs. Therefore, we define the edge operations used in the optimization part for directed graphs.

Definition I.2.2 (Edge addition)**.** Given a directed graph $G = (V, E)$, nodes $u, v \in V, (u, v) \notin E$, the *edge addition* of (u, v) results in a new graph $G' = (V, E')$ where $E' = E \cup \{(u, v)\}$.

Definition I.2.3 (Edge deletion)**.** Given a directed graph $G = (V, E)$, nodes $u, v \in V, (u, v) \in E$, the *edge deletion* of (u, v) results in a new graph $G' = (V, E')$ where $E' = E \setminus \{(u, v)\}$.

Definition I.2.4 (Edge rewiring)**.** Given a directed graph $G = (V, E)$, distinct nodes $i, j, k \in V$, $(i, j) \in E$, $(i, k) \notin E$, the *edge rewiring* of (i, j) to (i, k) produces a new graph $G' = (V, E')$ where $E' = (E \setminus \{(i, j)\}) \cup \{(i, k)\}$.

Definition I.2.5 (Edge operation)**.** Given a directed graph $G = (V, E)$, we define an *edge operation* to be any operation of edge addition, edge deletion and edge rewiring.

We now continue with the definitions for centrality measures.

Definition I.2.6 (Centrality measure)**.** Given a graph $G = (V, E)$, a *centrality measure* $c\colon V \to \mathbb{R}_{\geq 0}$ is any well-defined function $c\colon V \to \mathbb{R}_{\geq 0}$.

For any group $V' \subseteq V$, we define the *group centrality* as

$$c(V') = \sum_{v \in V'} c(v).$$

For centrality measures $c : V \to \mathbb{R}_{\geq 0}$ that are not (naturally) normalized, we define the *normalized centrality* $\overline{c}\colon V \to [0, 1]$:

$$\overline{c}(v) = \frac{c(v)}{\sum_{u \in V} c(u)}, v \in V.$$

whenever $\sum_{u \in V} c(u) > 0$. For any group $V' \subseteq V$, the normalized *group centrality* is hence

$$\overline{c}(V') = \sum_{v \in V'} \overline{c}(v).$$

and $\overline{c}(V) = 1$.
This will allow us to later extend PageRank-based fairness definitions to the degree, eigenvector and Katz centrality. Formally, a centrality measure is a real-valued function that is defined on the set of nodes V. Intuitively, a centrality measure can be viewed as an algorithm that is run on graphs, producing a real value for each node. This then creates a ranking of the nodes in V. Many centrality measures exist, and many graph properties can be translated to centrality measures [18].

We now define the addressed centrality measures.

Definition I.2.7 (Degree centrality)**.** Given a graph $G = (V, E)$, with $V = \{1, 2, \ldots, n\}$ and adjacency matrix A, the *degree centrality* $c_d\colon V \to \mathbb{R}_{\geq 0}$ is given by

$$c_d(i) = \sum_{j=1}^{n} A_{ij},\ i \in V.$$

If $G = (V, E)$ is directed, we define the *in-degree* and *out-degree* centrality, respectively given by

$$c_d^{\text{in}}\colon V \to \mathbb{R}_{\geq 0},\ c_d^{\text{out}}\colon V \to \mathbb{R}_{\geq 0}$$

where $c_d^{\text{in}}(i) = \sum_{j=1}^{n} A_{ji}, c_d^{\text{out}}(i) = \sum_{j=1}^{n} A_{ij},\ i \in V$. One also writes $d, d^{\text{in}}, d^{\text{out}}$ for respectively $c_d, c_d^{\text{in}}, c_d^{\text{out}}$.

Given the degree, in-degree and out degree centrality $c_d, c_d^{\text{in}}, c_d^{\text{out}}$ we define the *degree*, *in-degree* and *out-degree* matrix respectively as D, D^{in}, D^{out} as

$$D_{ij} = \begin{cases} c_d(i), & i = j, \\ 0 & \text{otherwise.} \end{cases}$$

analogously for D^{in}, D^{out}.
The degree centrality counts the number of neighbors a node has: the number of people an individual is connected to. Nodes with higher degree are ranked higher and can thus be viewed as important.

Definition I.2.8 (Eigenvector centrality)**.** Given a graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$, adjacency matrix A and $\lambda_{\max} > 0$ the largest eigenvalue of A, the *eigenvector centrality* $c_{ev} \colon V \to \mathbb{R}_{\geq 0}$ is given by

$$c_{ev}(i) = \frac{1}{\lambda_{\max}} \sum_{j=1}^{n} A_{ij} c_{ev}(j), \ i \in V.$$

By writing $v = (c_{ev}(1), c_{ev}(2), \ldots, c_{ev}(n))$, the eigenvector centrality is a solution of the equation

$$Av = \lambda_{\max} v.$$

as shown in [13]. The centrality scores hence form an *eigenvector* of the eigenvalue $\lambda_{\max}$. If G is connected and undirected, the Perron–Frobenius theorem gives a unique eigenvector associated to $\lambda_{\max}$ having strictly positive entries after normalization [16].

The eigenvector centrality can be interpreted as assigning high centrality to nodes adjacent to other nodes with high centrality. In contrast to the degree centrality, the centrality of a node does not only depend on the number of neighbors, but also on their centrality scores. The next and final two addressed centrality measures also rely on this recursive idea.

Definition I.2.9 (Katz centrality)**.** Given a graph $G = (V, E), V = \{1, 2, \ldots, n\}$ with adjacency matrix A. Let $\lambda_{\max}$ be the largest eigenvalue of A, and suppose $\lambda_{\max} > 0$. With parameter $\alpha \in \left(0, \frac{1}{\lambda_{\max}}\right)$, the *Katz centrality* $c_{\text{Katz}} \colon V \to \mathbb{R}_{\geq 0}$ is given by

$$c_{\text{Katz}}(i) = \sum_{k=0}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ij}, i \in V.$$

Let $\lambda_{\max}$ be the largest eigenvalue of A. As shown in [13], the centrality values c_{Katz} converge only when $\alpha < \frac{1}{\lambda_{\max}}$.

Intuitively, the Katz centrality of a node $i \in V$ is recursively defined by the centrality of its neighbors with importance of neighbors being weighted by a parameter α.

Definition I.2.10 (PageRank centrality)**.** Let $G = (V, E)$ be a directed graph with $V = \{1, 2, \ldots, n\}$ and let $\beta \in (0, 1)$. Let P be the *transition matrix* of G, and let $v = (v_1, \ldots, v_n)$ be a *restart vector*

satisfying $v_i \geq 0$ for all $i \in V$ and $\sum_{i=1}^{n} v_i = 1$.
The *PageRank centrality* $c_{PR} \colon V \to \mathbb{R}_{\geq 0}$ is defined by the vector $p \in \mathbb{R}^n$ satisfying

$$p^T = \beta \left(p^T P\right) + (1 - \beta) v^T.$$

The value $1 - \beta \in (0, 1)$ is called the *restart probability*. By writing $p = (p_1, \ldots, p_n)$ we define $c_{PR}(i) = p_i$ for every $i \in V$.

Usually, one takes $\beta = 0.85$ and chooses the uniform restart vector $v = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$. The transition matrix P is commonly defined by

$$P_{ij} = \begin{cases} \frac{A_{ij}}{\sum_{k=1}^{n} A_{ik}}, & \text{if } \sum_{k=1}^{n} A_{ik} > 0, \\ v_j & \text{otherwise.} \end{cases}$$

Here, A denotes the adjacency matrix of G. One may also refer to the PageRank parameters collectively as $(P, 1 - \beta, v)$.
Component-wise, the PageRank equation is given by

$$c_{PR}(i) = \beta \sum_{j=1}^{n} c_{PR}(j) P_{ji} + (1 - \beta) v_i, \ i \in V.$$

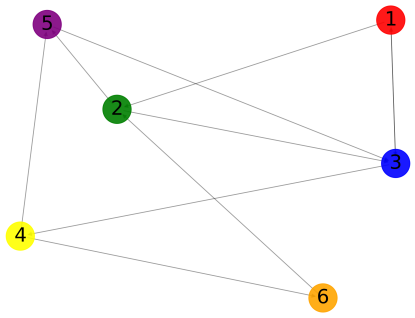
For the uniform restart vector, this becomes

$$c_{PR}(i) = \beta \sum_{j=1}^{n} c_{PR}(j) P_{ji} + \frac{1 - \beta}{n}.$$

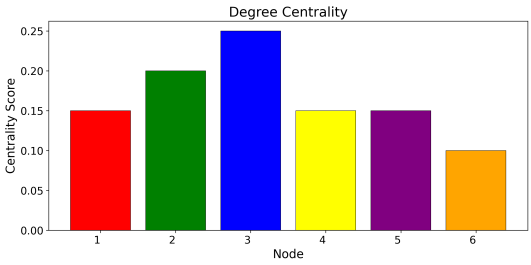
PageRank produces a probability vector [3]: $\sum_{i=1}^{n} c_{PR}(i) = 1$. Therefore, PageRank is normalized by default.

The PageRank centrality measure was developed by Larry Page and Sergey Brin, originally to rank web pages [3]. Later, it has been used in many other applications, including the ranking of citations [7] and protein networks [9].

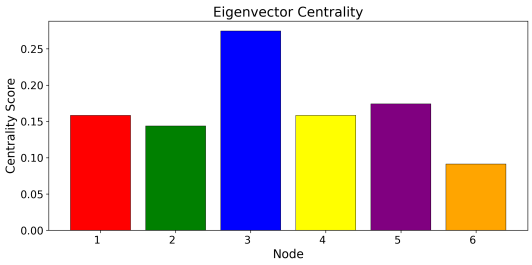
PageRank can be interpreted through the random surfer model. In this interpretation, the PageRank score $c_{PR}(i)$ of a node $i \in V$ represents the probability that a random surfer visits node i [3]. At each step, a surfer located at node $i \in V$ follows an outgoing edge to node j with probability βP_{ij}. If node i has no outgoing edges, the surfer moves to node j with probability v_j. With probability $1 - \beta$, the surfer restarts at node j with probability v_j.



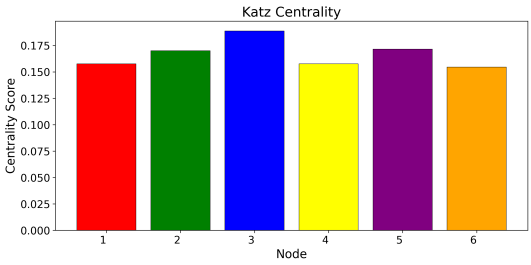
(a) Example graph containing 6 nodes.



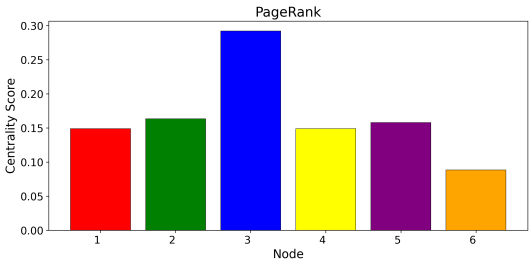
(b) Normalized degree centrality.



(c) Normalized eigenvector centrality.



(d) Normalized Katz centrality.



(e) PageRank centrality.

Figure I.1: Example graph and the corresponding centrality scores for the four centrality measures considered in this thesis.

We now introduce graphons.

Definition I.2.11 (Graphon)**.** A *graphon* is a symmetric measurable function $W\colon [0,1]^2 \to [0,1]$ satisfying $W(x,y) = W(y,x)$ for all $x, y \in [0,1]$.

Graphons can be used as the limit of graphs that increase in size [6]. The idea is that given a graph $G = (V, E)$ the nodes $i \in V$ are placed in the interval $[0,1]$. Given two points $x, y \in [0,1]$, the value $W(x,y)$ can be interpreted as the probability of an edge between the nodes located at the positions x and y.

Chapter II

Fairness Literature Review and Optimization

In this chapter, we first review fairness definitions from the literature. Then, we introduce PageRank fairness optimization algorithms and evaluate these on several networks.

II.1 Fairness Literature Review

With recent technological advances, the fairness of algorithms deployed on social, economic and technological systems has become increasingly important. While algorithmic fairness has attracted more attention [14], research on the fairness of centrality measures remains limited. One exception is PageRank, for which fairness has recently received more attention [10]. The purpose of this chapter is to provide an overview of the existing fairness definitions in current literature and apply them to our context. We start with general *algorithmic fairness* definitions from machine learning and statistics and show how these can be extended to centrality measures. Then, we review the many existing fairness definitions for the notable PageRank centrality measure. We conclude by selecting appropriate fairness definitions for the remainder of the thesis. We also show how we can extend these for the other centrality measures of interest.

In this section, by c we refer to a (nonnegative) centrality measure $c\colon V \to \mathbb{R}_{\geq 0}$, given a graph $G = (V, E)$.

II.1.1 General notions of fairness

Here we aim to provide several important and widely used notions of algorithmic fairness and see how these can be extended to centrality measures. In general, algorithmic fairness is often formulated as a constraint on algorithmic decisions or label predictions, especially in machine learning and classification tasks. However, centrality measures are real valued functions, which inherently produce a ranking. For centrality measures, fairness constraints can be imposed either on the centrality scores, for example on the total centrality mass assigned to a group, or on the induced ranking of the nodes, for example on the group composition of the top-k nodes. This is distinctively covered in this section.
Algorithmic fairness can be structured into that of individual fairness and group fairness [22], which

respectively are concerned with the fair treatment of individuals and the fair treatment of groups of individuals.

Individual fairness. Individual fairness is concerned with the fact that similar individuals should be treated similarly [5]. As in [22] for general algorithmic fairness, we can define the *individual fairness* for a centrality measure c as a Lipschitz condition:

$$|c(u) - c(v)| \leq Ld(u, v).$$

where $|c(u) - c(v)|$ measures the distance between the centrality scores of nodes $u, v \in V$, $L > 0$ is a Lipschitz constant applying the fairness constraint, d is a measure of similarity between nodes $u, v \in V$.
This inequality should be interpreted as: if two individuals (nodes) $u, v \in V$ are similar according to the similarity $d(u, v)$, then their respective centrality scores $c(u)$ and $c(v)$ should also be similar. Hence, the difference between the centrality scores of two nodes is bounded by a multiple of their similarity d.

Demographic parity. Demographic parity is a fairness constraint requiring that the probability of the outcome of an algorithm is similar for different demographic groups [14]. Demographic parity is defined in [17] for centrality measures as follows:
Let $G = (V, E)$ be a graph with communities $C = \{C_1, \ldots, C_l\}$, $k \in \mathbb{N}_{\geq 1}$ and $\tau^k \subseteq V$ being the top-k scored (ranked) individuals according to c. Here, $|S|$ denotes the cardinality of the set S. The centrality measure c should meet the following constraints (inequalities) to achieve demographic parity [17]:

$$|\{u \in \tau^k, u \in C_i\}| \leq \left\lceil \frac{|C_i|}{|V|} k \right\rceil,$$

$$|\{u \in \tau^k, u \in C_i\}| \geq \left\lfloor \frac{|C_i|}{|V|} k \right\rfloor.$$

for all $1 \leq i \leq l$. These inequalities bound the number of top-k scored individuals that are in each community C_i by the proportion of nodes in the community C_i and a multiple k. It is argued by [17] that these inequalities impose a *balance* of the top-k scored users in all communities $C_1, \ldots, C_l$.

Disparate impact. Disparate impact assures that the output of an algorithm is similar across different groups [14].
Let $G = (V, E)$ be a graph consisting of $r \in \mathbb{N}_{\geq 1}$ groups of nodes $V = V_1 \cup \cdots \cup V_r$. Similar to [5], we can define disparate impact for centrality measures as the requirement that the average centrality is similar across different groups. Define the average centrality $\bar{c}_{V_i}$ of V_i as $\bar{c}_{V_i} = \frac{1}{|V_i|} \sum_{v \in V_i} c(v)$, $1 \leq i \leq r$. We assume that for all $1 \leq i \leq r$: $\bar{c}_{V_i} > 0$. We require for a chosen $0 < \epsilon < 1$ that

$$\frac{\bar{c}_{V_i}}{\bar{c}_{V_j}} \geq 1 - \epsilon \text{ for all } 1 \leq i, j \leq r.$$

The concept is that the expected (average) centrality of a group $V_i, 1 \leq i \leq r$ is at least $1 - \epsilon$ of the average centrality of another group $V_j, 1 \leq j \leq r$.

There are more notions of algorithmic fairness [14]. It is important to point out that although these algorithmic fairness constraints can be very useful, all these fairness constraints can be mutually incompatible [15]. Therefore, one is limited to choose and hence justify the choice of a fairness definition.

II.1.2 PageRank fairness

Throughout this subsection, we always assume an underlying directed graph $G = (V, E)$ and we write $c_{PR}\colon V \to \mathbb{R}_{\geq 0}$ for the PageRank centrality measure. We now turn to the existing fairness definitions of the PageRank centrality measure. There is also work on the fairness of personalized PageRank [19, 20]. This will not be covered, since we are concerned with the global centrality distribution, rather than node-specific centrality measures.

ϕ-fairness. In [20], the PageRank centrality measure c_{PR} is defined to be ϕ-fair if for a *protected group* $R \subset V$ and $\phi \in (0, 1)$, $c_{PR}(R) = \phi$.

A related definition given in [20] is the following: given $S \subseteq V$ containing a protected group $S_R \subseteq S$ the PageRank centrality measure c_{PR} is *targeted ϕ-fair* on S if $c_{PR}(S_R) = \phi c_{PR}(S)$. Note that if $S = V$ and $S_R = R$, targeted ϕ-fairness reduces to ϕ-fairness, since $c_{PR}(V) = 1$. For ϕ-fairness, this means that PageRank is ϕ-fair if and only if for a protected group $R \subset V$ the total PageRank mass allocated to this group is exactly ϕ. It is a strict definition of fairness, often used as a target for which the restart vector or transition matrix is modified.

Targeted ϕ-fairness generalizes ϕ-fairness. It requires that, within a subset $S \subseteq V$, the protected subgroup $S_R \subseteq S$ receives a fraction ϕ of the PageRank mass assigned to S. In an experimental setting, one usually sets $\phi = \frac{|R|}{|V|}$; the protected group should get its presence as the proportion of the total allocated PageRank. Next is a more general definition of ϕ-fairness.

ϕ-sum-fairness. Let $V = V_1 \cup \ldots \cup V_k$ for $k \in \mathbb{N}_{\geq 1}$, $\phi_1, \ldots, \phi_k$ be group target values such that $\sum_{i=1}^{k} \phi_i = 1$, $\phi_i \geq 0, 1 \leq i \leq k$. In [10], PageRank c_{PR} is defined to be *ϕ-sum-fair* if $c_{PR}(V_i) = \phi_i, 1 \leq i \leq k$.

It is noted that this however does not take the distribution of PageRank mass within a group V_i into account [10]. Hence, *α-min-fairness* is introduced [10]: given $V = V_1 \cup \cdots \cup V_k$ and specified subsets $A_i \subseteq V_i$, the PageRank centrality measure c_{PR} satisfies α-min-fairness if

$$c_{PR}(v) \geq \alpha_i \quad \text{for all } v \in A_i,\ i = 1, \ldots, k.$$

Then finally the notion of *ϕ-sum + α-min fairness* is introduced as a combination of the above two definitions [10]. As pointed out in [10], ϕ-fairness is a specific case of ϕ-sum-fairness, where $k = 2$. As for the fairness definitions in [20], these definitions are arguably strict. Next are more practically applicable fairness definitions, in the sense of fairness optimization.

PR-unfairness. Given a subset (called a *target group*) $S \subset V$ and a parameter $\phi \in (0, 1)$, the PageRank centrality measure c_{PR} is defined to be PR-unfair [11] for S if $c_{PR}(S) < \phi$. Hence, a subset $S \subset V$ should have at least $0 < \phi < 1$ PageRank mass allocated to it. It is natural to choose $\phi = \frac{|S|}{|V|}$. Unlike ϕ-fairness, PR-unfairness is a one-sided definition: it only detects whether a target group S receives less than the desired PageRank mass ϕ. It is therefore appropriate when under representation is considered unfair, but not when both under representation and over representation are considered unfair. For example, this definition is suitable in the case of resource allocation where a protected group needs at least a prescribed amount of resources. This definition can also be extended to multiple target groups. Given $k \in \mathbb{N}_{\geq 1}$ subsets $S_1, \ldots, S_r \subseteq V$ with parameters $\phi_1, \ldots, \phi_r \in (0, 1)$, $\sum_{i=1}^{r} \phi_i \leq 1$. The PageRank centrality measure c_{PR} is then PR-unfair if there exists a $1 \leq i \leq r$ for which $c_{PR}(S_i) < \phi_i$.

The above fairness definitions are based on equality or inequality constraints. Now, we consider another way of interpreting fairness; in terms of the loss to a certain target distribution.

Fairness loss. Let $G = (V, E)$ be a graph with $r \in \mathbb{N}_{\geq 1}$ disjoint groups $V = V_1 \cup \cdots \cup V_r$. Let $\phi_1, \ldots, \phi_r \geq 0$ be target values with $\sum_{i=1}^{r} \phi_i = 1$. Let $1 - \beta \in (0, 1)$ be the restart probability, P be the transition matrix and v be the restart vector as in Definition I.2.10. The *fairness loss* function $L(P, 1 - \beta, v)$ [21] is defined as

$$L(P, 1 - \beta, v) = \frac{1}{r} \sum_{i=1}^{r} \left(c_{PR}(V_i) - \phi_i \right)^2 .$$

The fairness loss is the mean squared error between the total PageRank assigned to each group and the corresponding target value. It measures how far the observed PageRank of these groups is from a target distribution; the average deviation from a target distribution. From an optimization perspective, this fairness objective is particularly useful because it provides a single quantity that is to be minimized.

It is recognized in [21] that most practical networks are homophilic. Therefore, the *group-adapted fairness loss* function is introduced. Instead of considering only one restart vector, random walks starting uniformly from each group are considered. For each such restart, the fairness loss is considered [21].

Let $\mathbf{1}_{V_i}$ be the indicator vector for group $V_i, 1 \leq i \leq r$. Define $v_i = \frac{1}{|V_i|} \mathbf{1}_{V_i}$; the restart (indicator) vector indicating a (uniform) restart from exclusively group $V_i, 1 \leq i \leq r$. Let $c_{PR;v_i}(V_j)$ be the PageRank allocated to group $V_j, 1 \leq j \leq r$ given this vector v_i. The *group-adapted fairness loss* function [21] L_g is given by

$$L_g(P, 1 - \beta) = \frac{1}{r} \sum_{l=1}^{r} L(P, 1 - \beta, v_l) = \frac{1}{r^2} \sum_{l=1}^{r} \sum_{i=1}^{r} \left(c_{PR;v_l}(V_i) - \phi_i \right)^2$$

II.1.3 Selection of fairness definitions

We now state which PageRank fairness definitions from Section II.1.2 we will use for developing fairness optimization algorithms, and then explain the role of graphons for fairness estimation.

Selection for fairness optimization

As in [11], we turn to a natural optimization setting: given $G = (V, E)$ a budget $b \in \mathbb{N}_{\geq 1}$, how can we improve fairness using at most b edge operations?

Fairness definitions such as ϕ-fairness and ϕ-sum fairness are difficult to use as optimization objectives in this setting. Given $r \in \mathbb{N}_{\geq 1}$ groups $V_1, \ldots, V_r$ and target values $\phi_1, \ldots, \phi_r > 0$, the goal would be to find a set of edge operations within the budget $b \in \mathbb{N}_{\geq 1}$, such that for the modified graph the group PageRank masses match the target values exactly. It is difficult to guarantee that arbitrary target values can be reached within the budget $b \in \mathbb{N}_{\geq 1}$, without exhaustively searching through all possible graphs or using knowledge of the graph structure. A similar issue occurs for α-min fairness. For α-min fairness, it could also be difficult to find appropriate values $\alpha_i, 1 \leq i \leq k$ in a practical application. We therefore do not use these exact fairness constraints as optimization objectives in this thesis. Instead, we use fairness definitions that are more suitable for our optimization setting: PR-unfairness and fairness loss.

Recall that, given a directed graph $G = (V, E)$, a subset $S \subset V$ and $\phi \in (0, 1)$, the PageRank centrality c_{PR} is PR-unfair when $c_{PR}(S) < \phi$.
The optimization objective then simply becomes: apply at most $b \in \mathbb{N}_{\geq 1}$ edge operations to $G = (V, E)$ such that $c_{PR}(S)$ is increased, aiming to achieve $c_{PR}(S) \geq \phi$.

For fairness loss, given groups characterized by an attribute and a target distribution, the optimization objective is to apply at most b edge operations in order to minimize the fairness loss function.

Although the group-adapted fairness loss accounts for homophily, we do not use it in the fairness optimization algorithms. The aim of our optimization methods is to remain applicable to different types of networks.

Selection for graphon-based fairness estimation

The required graphon definitions and the corresponding fairness framework are developed in Section III.1. Graphons are useful in this setting because they provide limiting objects for sequences of graphs and allow fairness to be studied independently of a particular finite graph.

II.2 Fairness Optimization Methods

In this section, we develop greedy algorithms for structurally optimizing PageRank fairness. We consider two notions of fairness: fairness loss and PR-unfairness. In both settings, the graph can only be modified by a bounded number of edge operations. More precisely, given a directed graph $G = (V, E)$ and a budget $b \in \mathbb{N}_{\geq 1}$, an optimization algorithm may perform at most b edge operations. The edge operations include edge addition, edge deletion and edge rewiring. For both fairness notions, we develop one algorithm.

An exhaustive search through all possible graphs is impractical for all but small graphs. Instead, motivated by the size of the search space, a set of promising candidates is generated using heuristics. These candidates are then ranked. For only a selected subset of best-ranked candidates, the PageRank score of the modified graph is then recalculated. Then, the candidates are evaluated according to the fairness objective. The best candidate is applied, continuing this procedure until the fairness objective is met, the budget is reached or no more improvement is found.

The remainder of this section is organized as follows. First, we formulate the optimization problem for both notions of fairness. We then discuss the size of the search space induced by edge operations, motivating the use of heuristic candidate generation. We continue by presenting the greedy optimization algorithms and their candidate scoring. We finalize the section by discussing the limitations to the algorithms.

II.2.1 Optimization Problem

We now turn to defining the optimization problem for both notions of fairness. First, we describe the setting and notation for both optimization problems. Then, we continue with further definitions and conclude by defining the optimization problems.

Setting and notation.

Given a directed graph $G = (V, E)$ without self-loops, we write $G' = (V, E')$ for the resulting modified (directed) graph after applying $l \in \mathbb{N}_{\geq 1}$ edge operations to G. We also write $E_V :=$

$\{(u, v) : u, v \in V, u \neq v\}$ as the set containing all possible directed edges, excluding self-loops. Note that for all $G = (V, E)$, $E \subseteq E_V$. We further write $b \in \mathbb{N}_{\geq 1}$ for the *budget*; the number of allowed edge operations by an optimization algorithm. Finally, we write $c_{PR} \colon V \to \mathbb{R}_{\geq 0}$ for the PageRank centrality measure and $(P, 1-\beta, v)$ for the PageRank parameters.

Optimization Definitions.

We define $d \colon \mathcal{P}(E_V) \times \mathcal{P}(E_V) \to \mathbb{N}_{\geq 0}$, where $d(E, E')$ is the (minimum) number of edge operations needed to transform $G = (V, E)$ into $G' = (V, E')$.

Since G is finite and edge deletion, addition and rewiring can transform any edge set into any other edge set, d is well-defined. For a graph $G = (V, E)$, budget $b \in \mathbb{N}_{\geq 1}$, we define $\mathcal{B}_b(G)$ as the set of all possible modified graphs obtained after $d(E, E') \leq b$ edge operations:

$$\mathcal{B}_b(G) := \{G' = (V, E') \colon d(E, E') \leq b\}.$$

Given a graph $G = (V, E)$, let $L(P, 1-\beta, v)$ be the fairness loss function as in Section II.1.2 and the nodes V be split into $r \in \mathbb{N}_{\geq 1}$ disjoint groups $V = V_1 \cup \cdots \cup V_{r-1} \cup V_r$, $\phi_1, \ldots, \phi_r \geq 0, \sum_{i=1}^{r} \phi_i = 1$. Since in the optimization setting, the restart probability $1 - \beta$ and the restart vector v are fixed, we write the fairness loss function as

$$L(G) := \frac{1}{r} \sum_{i=1}^{r} \left(c_{PR;G}(V_i) - \phi_i\right)^2.$$

For fairness loss optimization, the *optimization objective* is to find a graph $G' \in \mathcal{B}_b(G)$ which minimizes the fairness loss $L(G')$:

Definition II.2.1 (Fairness Loss Optimization Problem)**.** Given (input graph) $G = (V, E)$, the optimization goal for fairness loss is

$$\text{find a graph } G^* \in \arg\min\{L(G') \colon G' \in \mathcal{B}_b(G)\}.$$

Given $G = (V, E)$, write $c_{PR;G} \colon V \to \mathbb{R}_{\geq 0}$ for the PageRank centrality measure given G, emphasizing that PageRank is evaluated on G. Let $R \subset V$ be a protected group, $\phi \in (0, 1)$.

For PR-unfairness optimization, the *optimization objective* is to find a graph $G' \in \mathcal{B}_b(G)$ such that $c_{PR;G'}(R) \geq \phi$:

Definition II.2.2 (PR-unfairness Optimization Problem)**.** Given (input graph) $G = (V, E)$, the optimization goal for PR-unfairness is

$$\text{find } G' \in \mathcal{B}_b(G) \colon c_{PR;G'}(R) \geq \phi.$$

If no feasible graph is found within the budget b, the goal is to find a graph $G' \in \mathcal{B}_b(G)$ that maximizes $c_{PR;G'}(R)$.

II.2.2 Search Space

Consider a (simple) directed graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges. We derive the number of edge additions $N_{\mathrm{a}}(G)$, edge deletions $N_{\mathrm{d}}(G)$ and edge rewirings $N_{\mathrm{r}}(G)$.

Definition II.2.3 (Number of edge additions)**.** It is clear that the number of possible *edge additions* $N_{\mathrm{a}}(G)$ is

$$N_{\mathrm{a}}(G) = n(n-1) - m.$$

Definition II.2.4 (Number of edge deletions)**.** Equivalently, the number of possible *edge deletions* $N_{\mathrm{d}}(G)$ is

$$N_{\mathrm{d}}(G) = m.$$

Note that the total number of edge additions and deletions is $N_{\mathrm{a}}(G)+N_{\mathrm{d}}(G) = n(n-1) \in O(n^2)$.

Definition II.2.5 (Number of edge rewirings)**.** The number of *edge rewirings* $N_{\mathrm{r}}(G)$ can be derived as follows: for $i \in V$, consider the out-degree $d^{\mathrm{out}}(i)$. For $j \in V, j \neq i$ and an existing edge $(i,j) \in E$, the possible rewirings $(i,j) \to (i,k)$ are the number of nodes $k \in V$ for which $i \neq k, (i,k) \notin E$; hence $n-1-d^{\mathrm{out}}(i)$. Thus, for a node $i \in V$ the number of edge rewirings is $d^{\mathrm{out}}(i)\left(n-1-d^{\mathrm{out}}(i)\right)$ and the total number of *edge rewirings* $N_{\mathrm{r}}(G)$ is

$$N_{\mathrm{r}}(G) = \sum_{i=1}^{n} d^{\mathrm{out}}(i)\left(n-1-d^{\mathrm{out}}(i)\right).$$

It is clear that even for moderately sized graphs the total number of possible edge operations

$$N(G) := N_{\mathrm{a}}(G) + N_{\mathrm{d}}(G) + N_{\mathrm{r}}(G)$$

can already be very large.

A naive approach for developing a fairness optimization algorithm that makes use of edge operations would be to exhaustively search over all graphs that can be obtained within the given budget. In each step, one would consider all possible edge operations on the current graph, apply each operation, recompute PageRank on the resulting graph, and evaluate the corresponding fairness objective.

However, this quickly becomes impractical. If C denotes the set of possible candidate edge operations and the fairness objective is only met after $1 \leq l \leq b$ operations, then up to $O\left(|C|^l\right)$ sequences of edge operations may have to be considered. Since PageRank has to be recomputed for each resulting graph, the number of PageRank computations grows exponentially in the number of allowed operations.

This motivates the development of greedy algorithms. The developed algorithms in this thesis follow a two-stage greedy approach. At each iteration $1 \leq i \leq b$, they heuristically select a number of promising candidates for each edge operation that is used. Then, they apply each candidate to the graph and evaluate the fairness objective on the modified graph. The candidate that gives the largest fairness improvement is then applied to the graph. They then continue with the updated graph, until either the fairness objective is met or the budget b is reached.

II.2.3 Algorithm 1: Greedy Pairwise Fairness Loss Optimization

This subsection presents the optimization algorithm for fairness loss. Before giving the algorithm, we first conceptually explain how the algorithm works.

The algorithm input consists of an initial graph $G = (V, E)$, $r \in \mathbb{N}_{\geq 1}$ disjoint groups $V = V_1 \cup \cdots \cup V_{r-1} \cup V_r$ with target values $\phi_1, \ldots, \phi_r \geq 0, \sum_{i=1}^{r} \phi_i = 1$, a budget of edge operations

$b \in \mathbb{N}_{\geq 1}$, a set of edge operations and heuristic parameters $k \in \mathbb{N}_{\geq 0}, T \in \mathbb{N}_{\geq 1}$. The heuristic parameters will be explained later.

The set of *active groups* $\mathcal{G}_{\text{act}} \subseteq \{V_1, \ldots, V_{r-1}, V_r\}$ is defined to be the set of groups that are actively being optimized. Initially, $\mathcal{G}_{\text{act}} = \{V_1, \ldots, V_{r-1}, V_r\}$. For group $V_i, 1 \leq i \leq r$, consider the *target fairness difference*

$$\Delta_i(G) := c_{PR;G}(V_i) - \phi_i.$$

Note that we can now write the fairness loss function $L(G)$ as $L(G) = \frac{1}{r}\sum_{i=1}^{r} \Delta_i(G)^2$.
The algorithm makes use of the concept of *advantaged* and *disadvantaged* groups. We call $V_i \in \mathcal{G}_{\text{act}}$ advantaged or disadvantaged when respectively $\Delta_i(G) > 0$ or $\Delta_i(G) < 0$. When $\Delta_i(G) = 0$, the group does not contribute to the fairness loss. We therefore call such a group *neutral*, and do not consider it for optimization.

At each iteration, the algorithm selects the most advantaged active group, written $V_{\max} \in \mathcal{G}_{\text{act}}$, and most disadvantaged group, written $V_{\min} \in \mathcal{G}_{\text{act}}$. They are defined to be the active groups with the largest and smallest target fairness difference respectively: $V_{\max} := V_{i_{\max}}, V_{\min} := V_{i_{\min}}$, where

$$i_{\max} \in \arg\max\{\Delta_i(G), V_i \in \mathcal{G}_{\text{act}}\},$$

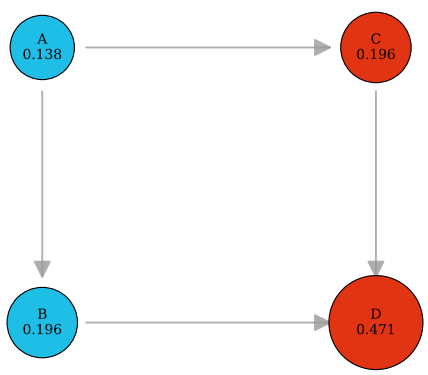
$$i_{\min} \in \arg\min\{\Delta_i(G), V_i \in \mathcal{G}_{\text{act}}\}.$$

If multiple groups attain the same maximum or minimum target fairness difference, the group with the smallest index is selected.

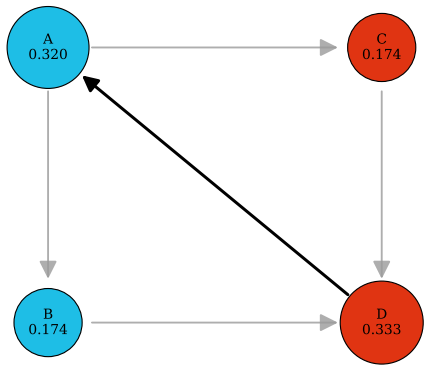
For each included edge operation, candidate edge operations are chosen heuristically, where the aim is to (pairwise) move PageRank mass from $V_{\max}$ to $V_{\min}$. Specifically, edge additions are considered from $V_{\max}$ to $V_{\min}$, edge deletions are considered within $V_{\max}$ and edge rewirings are considered from $V_{\max}$ to $V_{\min}$: replacing an edge from $V_{\max}$ to $V_{\max}$ by an edge from $V_{\max}$ to $V_{\min}$.

If none of the candidates decreases the fairness loss, the algorithm removes the group among $V_{\max}$ and $V_{\min}$ with the smaller absolute target fairness difference.

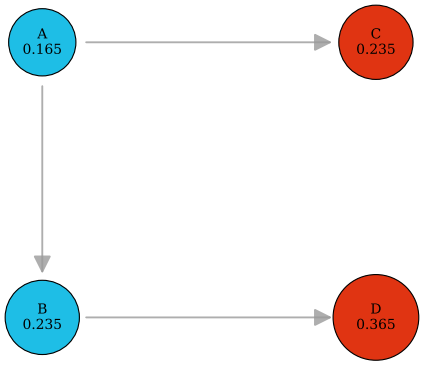
The algorithm terminates when either $L(G) = 0$, the budget $b \in \mathbb{N}_{\geq 1}$ is reached or the algorithm is stuck in a local optimum and no more group can be deactivated. Then, the final modified graph and the corresponding fairness loss are returned.



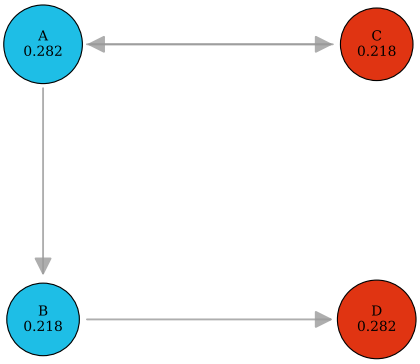
(a) Visualization of graph G and its (rounded) PageRank scores.



(b) Visualization of graph G_{add} and its (rounded) PageRank scores, where G_{add} is obtained from G by adding the edge (D, A).



(c) Visualization of graph G_{del} and its (rounded) PageRank scores, where G_{del} is obtained from G by deleting the edge (C, D).



(d) Visualization of graph G_{rew} and its (rounded) PageRank scores, where G_{rew} is obtained from G by rewiring the edge $(C, D) \to (C, A)$.

Figure II.1: Example illustrating how the PageRank scores change relative to the original graph under the three candidate edge operations considered by the algorithm: adding (D, A), deleting (C, D), and rewiring $(C, D) \to (C, A)$. For $V_{\min} = V_1 = \{A, B\}$, $V_{\max} = V_2 = \{C, D\}$, and $\phi_1 = \phi_2 = 1/2$, the corresponding fairness losses are $L(G) \approx 0.0275$, $L(G_{\text{add}}) \approx 0.00004$, $L(G_{\text{del}}) \approx 0.01$, and $L(G_{\text{rew}}) = 0$.

Next is a complete description of the algorithm, followed by an explanation of its functions and heuristics.

Algorithm 1 Greedy Pairwise Fairness Loss Optimization

Input: $G = (V, E)$, $V = V_1 \cup \cdots \cup V_r$, $\phi_1, \ldots, \phi_r \geq 0, \sum_{i=1}^{r} \phi_i = 1$,
$k \in \mathbb{N}_{\geq 0}, b, T \in \mathbb{N}_{\geq 1}, \mathrm{OP} \subseteq \{\mathrm{ADD}, \mathrm{DELETE}, \mathrm{REWIRE}\}, \mathrm{OP} \neq \emptyset$.
Output: $(G', L(G')) : G' \in \mathcal{B}_b(G), L(G') \leq L(G)$

$G_{\text{current}} \leftarrow G$
$\mathcal{G}_{\text{act}} = \{V_i \colon 1 \leq i \leq r\}$
$t \leftarrow 0$
while $t < b$ **do**
  if $L(G_{\text{current}}) = 0$ **then**
    return $(G_{\text{current}}, L(G_{\text{current}}))$
  end if
  if $|\mathcal{G}_{\text{act}}| = 1$ **then**
    return $(G_{\text{current}}, L(G_{\text{current}}))$
  end if
  Cand $\leftarrow \emptyset$
  Compute c_{PR} on G_{current}
  $i_{\max} \leftarrow \arg\max\{c_{PR}(V_i) - \phi_i, V_i \in \mathcal{G}_{\text{act}}\}$
  $i_{\min} \leftarrow \arg\min\{c_{PR}(V_i) - \phi_i, V_i \in \mathcal{G}_{\text{act}}\}$
  $V_{\max} \leftarrow V_{i_{\max}}$
  $V_{\min} \leftarrow V_{i_{\min}}$
  if ADD $\in$ OP **then**
    $C_{\text{add}} \leftarrow$ CandidatesAddition $(G_{\text{current}}, V_{\max}, V_{\min}, k, T)$
    Cand $\leftarrow$ Cand $\cup\, C_{\text{add}}$
  end if
  if DELETE $\in$ OP **then**
    $C_{\text{del}} \leftarrow$ CandidatesDeletion $(G_{\text{current}}, V_{\max}, V_{\min}, k, T)$
    Cand $\leftarrow$ Cand $\cup\, C_{\text{del}}$
  end if
  if REWIRE $\in$ OP **then**
    $C_{\text{rew}} \leftarrow$ CandidatesRewiring $(G_{\text{current}}, V_{\max}, V_{\min}, k, T)$
    Cand $\leftarrow$ Cand $\cup\, C_{\text{rew}}$
  end if
  for each candidate $c \in$ Cand **do**
    Apply c to G_{current}, obtain modified graph G_c
    Compute c_{PR} on G_c
    Compute fairness loss $L(G_c)$
  end for
  Choose $c_{\text{best}} = \arg\min_{c \in \text{Cand}} L(G_c)$
  $G_{\text{best}} \leftarrow G_{c_{\text{best}}}$
  if $L(G_{\text{best}}) < L(G_{\text{current}})$ **then**
    $G_{\text{current}} \leftarrow G_{\text{best}}$
    $t \leftarrow t + 1$
  else
    DeactivateBestGroup$(V_{\min}, V_{\max})$
    if No Group Can Be Deactivated **then**
      return $(G_{\text{current}}, L(G_{\text{current}}))$
    end if
  end if
end while
return $(G_{\text{current}}, L(G_{\text{current}}))$

The DeactivateBestGroup function works as follows. For each active group $V_i \in \mathcal{G}_{\text{act}}$, consider the absolute fairness loss difference $|\Delta_i(G)|$. The set of advantaged and disadvantaged groups, respectively $\mathcal{G}^+_{\text{act}}$ and $\mathcal{G}^-_{\text{act}}$ are generated:

$$\mathcal{G}^+_{\text{act}} = \{V_i \in \mathcal{G}_{\text{act}} \colon \Delta_i(G) > 0\}, \mathcal{G}^-_{\text{act}} = \{V_i \in \mathcal{G}_{\text{act}} \colon \Delta_i(G) < 0\}.$$

Now, the group to be removed $R \subset V$ is decided according to the following rules:

$$R = \begin{cases} \emptyset, & \text{if } |\mathcal{G}^+_{\text{act}}| = |\mathcal{G}^-_{\text{act}}| = 1, \\ V_{\min} & \text{if } |\mathcal{G}^+_{\text{act}}| = 1, |\mathcal{G}^-_{\text{act}}| > 1, \\ V_{\max} & \text{if } |\mathcal{G}^+_{\text{act}}| > 1, |\mathcal{G}^-_{\text{act}}| = 1, \\ V_{\min} & \text{if } |\Delta_{i_{\max}}(G)| > |\Delta_{i_{\min}}(G)|, |\mathcal{G}^+_{\text{act}}| > 1, |\mathcal{G}^-_{\text{act}}| > 1, \\ V_{\max} & \text{if } |\Delta_{i_{\min}}(G)| \geq |\Delta_{i_{\max}}(G)|, |\mathcal{G}^+_{\text{act}}| > 1, |\mathcal{G}^-_{\text{act}}| > 1. \end{cases}$$

If $R = \emptyset$, the algorithm terminates. When $R \neq \emptyset, \mathcal{G}_{\text{act}} = \mathcal{G}_{\text{act}} \setminus R$.

When the algorithm reaches a local optimum, it should consider all advantaged and all disadvantaged groups in the set of active groups. If after removing the most advantaged group there are no more advantaged groups, then it should try removing the most disadvantaged group from the set of active groups. If there is also 1 disadvantaged group left, this means it is stuck in a local optimum and it cannot consider any other group. If there are multiple advantaged and disadvantaged groups, then remove the group with the lowest fairness difference from the set of active groups, since this contributes the least to the fairness loss.

Heuristic scoring

The CandidatesAddition heuristic works as follows. Given a graph G, groups $V_{\max}$ and $V_{\min}$, and parameters $k \in \mathbb{N}_{\geq 0}, T \in \mathbb{N}_{\geq 1}$, we consider candidate edges $(u, v) \notin E, u \neq v$ with $u \in V_{\max}$ and $v \in V_{\min}$. The source node $u \in V_{\max}$ is scored as

$$s^{\text{src}}_{\text{add}}(u) = \frac{c_{PR}(u)}{d^{\text{out}}(u) + 1}.$$

A destination node $v \in V_{\min}$ is scored as

$$s^{\text{dst}}_{\text{add}}(v) = c_{PR}(v) \frac{\sum_{v' \in V_{\min}, (v,v') \in E} c_{PR}(v') + \epsilon}{\sum_{v' \in V_{\max} \cup V_{\min}, (v,v') \in E} c_{PR}(v') + 2\epsilon}, \epsilon = \frac{1}{n}.$$

The candidate addition (u, v) is defined as

$$s_{\text{add}}(u, v) = s^{\text{src}}_{\text{add}}(u) s^{\text{dst}}_{\text{add}}(v).$$

The source score reflects the approximate PageRank mass that u can pass to v after adding the edge (u, v). Since adding this edge increases the out-degree of u to $d^{\text{out}}(u) + 1$, this contribution is approximated by

$$\frac{c_{PR}(u)}{d^{\text{out}}(u) + 1}.$$

The destination score favors nodes $v \in V_{\min}$ with high PageRank score that mainly link to other disadvantaged nodes. Such nodes can potentially distribute additional PageRank mass within $V_{\min}$. The numerator rewards outgoing links from v to nodes in $V_{\min}$, weighted by their PageRank scores, while the denominator also accounts for outgoing links from v to nodes in $V_{\max}$. Thus, the score is lower when a large part of the outgoing PageRank mass of v is directed toward advantaged nodes.

The parameter $\epsilon = 1/n$ prevents division by zero and represents the average PageRank mass of a node. If v has no outgoing edges to nodes in $V_{\max} \cup V_{\min}$, then v is treated as neutral, and

$$s^{\text{dst}}_{\text{add}}(v) = c_{PR}(v)\frac{\epsilon}{2\epsilon} = \frac{1}{2}c_{PR}(v).$$

The candidate score $s_{\text{add}}(u, v)$ therefore combines a source score and a destination score. The T highest-ranked candidate edges are selected and returned.

The parameter $k \in \mathbb{N}_{\geq 0}$ controls how many of the top-ranked nodes in $V_{\max}$ and $V_{\min}$ are considered when scoring candidate edges. This restriction reduces the cost of candidate generation, since the algorithm does not have to score all possible pairs (u, v) with $u \in V_{\max}$ and $v \in V_{\min}$. If $k = 0$, all nodes are considered.

Next, CandidatesDeletion considers candidate edges $(u, v) \in E$ with $u, v \in V_{\max}$. The source node u is scored as

$$s^{\text{src}}_{\text{del}}(u) = \frac{c_{PR}(u)}{d^{\text{out}}(u)}.$$

The destination node v is scored as

$$s^{\text{dst}}_{\text{del}}(v) = c_{PR}(v)\frac{\sum_{v' \in V_{\max},\, (v,v') \in E} c_{PR}(v') + \epsilon}{\sum_{v' \in V_{\max} \cup V_{\min},\, (v,v') \in E} c_{PR}(v') + 2\epsilon}.$$

The score of the candidate deletion (u, v) is then defined as

$$s_{\text{del}}(u, v) = s^{\text{src}}_{\text{del}}(u)s^{\text{dst}}_{\text{del}}(v).$$

The source score is motivated by the fact that the expected PageRank contribution from u to one of its outgoing neighbors is

$$\frac{c_{PR}(u)}{d^{\text{out}}(u)}.$$

Thus, deleting an outgoing edge from a high-PageRank source node with small out-degree is expected to have a relatively large effect.

The destination score is analogous to the destination score used in CandidatesAddition. It favors destination nodes v whose outgoing PageRank mass is mainly directed toward advantaged nodes, and penalizes nodes whose outgoing mass is directed toward disadvantaged nodes. This way, the deletion heuristic prioritizes edges whose removal is expected to reduce PageRank mass within $V_{\max}$. The parameters $k \in \mathbb{N}_{\geq 0}, T \in \mathbb{N}_{\geq 1}$ have the same interpretation as in CandidatesAddition.

Finally, before discussing CandidatesRewiring, it is important to note that we interpret a rewiring $(u, v) \to (u, w)$ as a deletion of the edge (u, v) followed by the addition of the edge (u, w). We therefore define the rewiring score of the rewiring $(u, v) \to (u, w)$ in CandidatesRewiring as the product of the deletion score $s_{\text{del}}(u, v)$ and the addition score $s_{\text{add}}(u, w)$:

$$s_{\text{rew}}(u, v, w) = s_{\text{del}}(u, v)s_{\text{add}}(u, w).$$

Equivalently,

$$s_{\text{rew}}(u, v, w) = s_{\text{del}}^{\text{src}}(u) s_{\text{del}}^{\text{dst}}(v) s_{\text{add}}^{\text{src}}(u) s_{\text{add}}^{\text{dst}}(w).$$

This score assigns a high value to rewirings that remove an edge within the advantaged group and redirect the source node toward a promising node in the disadvantaged group.

The parameter $k \in \mathbb{N}_{\geq 0}$ is passed to both CandidatesAddition and CandidatesDeletion. To reduce computational cost, the heuristic scores computed by these two procedures can be cached and reused when scoring rewiring candidates. The parameter $T \in \mathbb{N}_{\geq 1}$ has the same interpretation as in CandidatesAddition and CandidatesDeletion.

II.2.4 Algorithm 2: Greedy PR-Unfairness Optimization

We now present the optimization algorithm developed for PR-unfairness. For PR-unfairness, the optimization objective is different: we want to find a graph for which the PageRank for the protected group $R \subset V$ is at least ϕ. Only two groups R and $V \setminus R$ are considered, so there is no need for a concept of active groups. Hence, the algorithm fixes $V_{\text{min}} = R, V_{\text{max}} = V \setminus R$. The candidate generation process is equivalent to that of Algorithm 1. The candidate evaluation process is also similar, the main difference being that the algorithm stops when a candidate graph G_c is found for which $c_{PR;G_c}(R) \geq \phi$. The candidate generation heuristics CandidatesAddition, CandidatesDeletion and
CandidatesRewiring work as in Algorithm 1. We now give the full algorithm.

Algorithm 2 Greedy PR-Unfairness Optimization

Input: $G = (V, E)$, $R \subset V$, $\phi \in (0, 1)$, $k \in \mathbb{N}_{\geq 0}, b, T \in \mathbb{N}_{\geq 1}$, $\mathrm{OP} \subseteq \{\mathrm{ADD}, \mathrm{DELETE}, \mathrm{REWIRE}\}$
Output: $\{G', c_{PR;G'}(R)\} \colon G' \in \mathcal{B}_b(G), c_{PR;G'}(R) \geq c_{PR;G}(R)$

$G_{\text{current}} \leftarrow G$
$V_{\max} \leftarrow V \setminus R$
$V_{\min} \leftarrow R$
for $t = 1, \ldots, b$ **do**
  Cand $\leftarrow \emptyset$
  Compute c_{PR} on G_{current}
  if $c_{PR;G_{\text{current}}}(R) \geq \phi$ **then**
    return $(G_{\text{current}}, c_{PR;G_{\text{current}}}(R))$
  end if
  if ADD $\in$ OP **then**
    $C_{\text{add}} \leftarrow \textsc{CandidatesAddition}\,(G_{\text{current}}, V_{\max}, V_{\min}, k, T)$
    Cand $\leftarrow$ Cand $\cup\, C_{\text{add}}$
  end if
  if DELETE $\in$ OP **then**
    $C_{\text{del}} \leftarrow \textsc{CandidatesDeletion}\,(G_{\text{current}}, V_{\max}, V_{\min}, k, T)$
    Cand $\leftarrow$ Cand $\cup\, C_{\text{del}}$
  end if
  if REWIRE $\in$ OP **then**
    $C_{\text{rew}} \leftarrow \textsc{CandidatesRewiring}\,(G_{\text{current}}, V_{\max}, V_{\min}, k, T)$
    Cand $\leftarrow$ Cand $\cup\, C_{\text{rew}}$
  end if
  for each candidate $c \in$ Cand **do**
    Apply c to G_{current}, obtain modified graph G_c
    Compute c_{PR} on G_c
    Compute $c_{PR;G_c}(R)$
    if $c_{PR;G_c}(R) \geq \phi$ **then**
      return $(G_c, c_{PR;G_c}(R))$
    end if
  end for
  Choose $c_{\text{best}} = \arg\max_{c \in \text{Cand}} c_{PR;G_c}(R)$
  $G_{\text{best}} \leftarrow G_{c_{\text{best}}}$
  if $c_{PR;G_{\text{best}}}(R) > c_{PR;G_{\text{current}}}(R)$ **then**
    $G_{\text{current}} \leftarrow G_{\text{best}}$
  else
    return $(G_{\text{current}}, c_{PR;G_{\text{current}}}(R))$
  end if
end for
return $(G_{\text{current}}, c_{PR;G_{\text{current}}}(R))$

II.2.5 Limitations of the Greedy Algorithms

We conclude this section by discussing the main limitations of Algorithms 1 and 2.

Both algorithms are heuristic greedy algorithms. Hence, they are not guaranteed to find a globally optimal solution for the corresponding optimization problem. Every iteration, only the best edge operation from a set of candidate operations is selected. An edge operation that may later lead to a better solution may therefore be ignored if it does not give the best immediate improvement.

Another limitation is that the candidate generation of both algorithms only considers certain groups for all edge operations, thus not fully exploring the complete search space of edge operations. For example, an edge deletion between the most advantaged and most disadvantaged group could lead to a larger decrease in fairness loss, but this is not considered by the candidate generation process.

A third limitation of the algorithms is that they recompute PageRank for every selected candidate in each iteration. Although this improves the accuracy of candidate selection, it also makes the runtime (and time complexity) dependent on the number of selected candidates T and the cost of PageRank computation on the graph.

Another limitation is that the heuristics only *approximate* the effect of an edge operation on the PageRank group scores, only using local information. Meanwhile, the change in PageRank is propagated through the network, and this is not considered by the heuristics.

Another limitation of the fairness loss algorithm lies in the DeactivateBestGroup function. Once a group is deactivated after a local optimum is reached, it cannot become active again. This prevents the algorithm from repeatedly considering the same pair of groups, but it may also exclude a group that becomes optimizable again after later graph modifications.

Finally, the parameters $k \in \mathbb{N}_{\geq 0}$ and $T \in \mathbb{N}_{\geq 1}$ force a trade-off between runtime and solution quality: for higher values of k and T, the algorithms are expected to perform better, but there are more calculations needed: a higher solution quality means a longer (expected) runtime.

II.3 Fairness Optimization Results

In this section, we show the results of the algorithms developed in Section II.2 applied to several networks. First, we describe the experimental setup for the fairness optimization. Then, we discuss the optimization results for Algorithm 1 and 2. We conclude with a comparative analysis and discussion.

II.3.1 Experimental setup

This subsection describes the experimental setup used for evaluating Algorithms 1 and 2. We first introduce the datasets, then describe parameter settings and target ϕ-values used in the experiments. Finally, we discuss relevant implementation details.

Datasets

We evaluate Algorithms 1, 2 on several types of directed networks. First, we evaluate on subsets of a gender-labeled DBLP collaboration network `DBLP_gender` [8]. These datasets are (filtered)

subsets of a co-authorship network derived from DBLP. Nodes represent authors and edges represent co-authorship relations [8]. Each node has a gender label, either male or female. We have removed nodes from the graph that were originally labeled as "None". Second, on `polblogs`, a network consisting of political blog posts, which are either labeled as "liberal" or "conservative" [1]. Edges represent the existence of hyperlinks between blog posts [1]. Finally, on the `Cora` citation network [12]. This is a citation network consisting of machine learning papers classified into 7 research topics. Nodes represent papers and edges represent citation links between papers [12].

An overview of all networks, their metadata, group attribute and group proportions follows in the table below. For notation purposes, we write for $r \in \mathbb{N}_{\geq 1}$ groups the group proportions $\frac{|V_1|}{|V|}, \ldots, \frac{|V_{r-1}|}{|V|}, \frac{|V_r|}{|V|}$ as a vector $\left(\frac{|V_1|}{|V|}, \ldots, \frac{|V_{r-1}|}{|V|}, \frac{|V_r|}{|V|}\right)$. We also write `datamining`, `database` respectively for the `DBLP_gender: database` and `DBLP_gender: datamining` network.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	Groups	Attribute	Group proportions
`database`	3069	8736	2	gender	$(2411/3069, 658/3069)$
`datamining`	2188	7025	2	gender	$(1773/2188, 415/2188)$
`Cora`	2708	10556	7	Research topic	$(818/2708, 426/2708, 418/2708, 351/2708, 298/2708, 217/2708, 180/2708)$
`polblogs`	1490	19090	2	Political ideology	$(758/1490, 732/1490)$

Table II.1: Overview of datasets included in the experimental setup.

Note that all these datasets are sparse.

Parameter settings

For all experiments, PageRank is computed with restart probability $1 - \beta = 0.15$ and uniform restart vector v. The budget is fixed at $b = 500$ edge operations. Note that we can also evaluate $b = 100, b = 200, b = 300, b = 400$ this way, since the algorithms are deterministic. The candidate generation and evaluation parameters are fixed at $k = 500$ and $T = 10$. This choice is made because calculating PageRank is computationally expensive but the number of nodes in all networks is in the order of 1000 nodes. We set $k = 500$ since naturally $k > T$, and the number of generated candidates should be much larger than the number of evaluated candidates. For each dataset and fairness objective, we evaluate four choices of allowed edge operations OP:

$$\{\text{ADD}\}, \{\text{DELETE}\}, \{\text{REWIRE}\}, \{\text{ADD}, \text{DELETE}, \text{REWIRE}\}.$$

We write $\text{ALL} := \{\text{ADD}, \text{DELETE}, \text{REWIRE}\}$. We refer to these settings as addition-only, deletion-only, rewiring-only and all, respectively.

Target ϕ-values

The choice of target ϕ-values depends on the fairness objective. For fairness loss optimization, a target vector

$$\phi = (\phi_1, \ldots, \phi_{r-1}, \phi_r), \sum_{i=1}^{r} \phi_i = 1.$$

is required. Instead of using only the natural proportional target $\phi_i = \frac{|V_i|}{|V|}$, we evaluate Algorithm 1 on several prescribed target distributions. This is done, because on all datasets, the proportional

target leads to early termination in a local optimum. Therefore, this gives little insight into the behavior of the optimization algorithms. For the two group datasets, we use the target vector $\phi = (\phi_1, 1-\phi_1)$ where $\phi_1 \in \{0.1, \ldots, 0.8, 0.9\}$, as in [21]. This allows us to study how the algorithm behaves when the desired PageRank mass is shifted toward either of the two groups. For multi-group datasets, such as Cora, varying over all possible target vectors is infeasible. Therefore, for the `Cora` dataset, we consider the uniform target

$$\phi = \left(\frac{1}{7}, \ldots, \frac{1}{7}\right).$$

Requiring every group to receive equal PageRank mass.

For PR-unfairness optimization, the target is a threshold ϕ for the protected group $R \subset V$. For the two group datasets, we choose $R \subset V$ to be the initially underrepresented group for which $c_{PR}(R) < \frac{|R|}{|V|}$. For the multi group datasets, we choose R to be the (underrepresented) group $R \subset V$ for which $c_{PR}(R) < \frac{|R|}{|V|}$ and $\frac{|R|}{|V|} - c_{PR}(R)$ is largest. We then evaluate Algorithm 2 for thresholds of the form $\phi = c_{PR}(R) + \delta$, where $\delta \in \{0.1, 0.2\}$. This tests whether Algorithm 2 can increase the PageRank mass of the target group by a prescribed amount.

Implementation details

Both algorithms were implemented in Python, using `networkx` for PageRank calculations and graph representation. Graphs are stored as `.gml` files. The parameters for the `networkx` PageRank calculation are the default parameters.

II.3.2 Fairness Loss Optimization Results

In this section, we present the results for fairness loss optimization for all networks, using the parameter settings as described in the previous section.

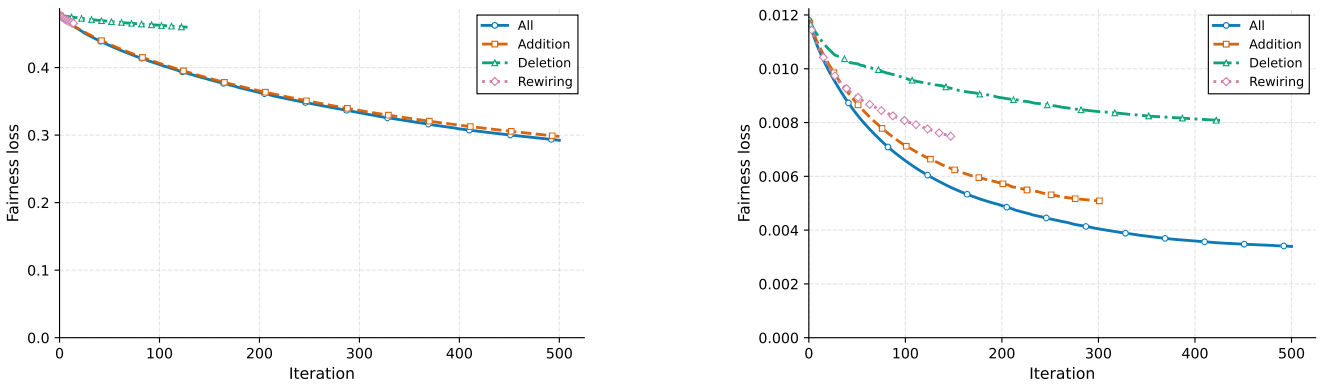
Before presenting the optimization results, Table II.2 gives the original PageRank mass assigned to each group. We write initial PageRank scores $c_{PR}(V_1), \ldots, c_{PR}(V_{r-1}), c_{PR}(V_r)$ as the vector $(c_{PR}(V_1), \ldots, c_{PR}(V_{r-1}), c_{PR}(V_r))$.

Network	Original PageRank scores
`database`	$(0.78560, 0.21440)$
`datamining`	$(0.81183, 0.18817)$
`polblogs`	$(0.50872, 0.49128)$
`Cora`	$(0.29261, 0.15754, 0.15845, 0.13753, 0.10640, 0.08241, 0.065059)$

Table II.2: Original PageRank mass assigned to each group before optimization.

We now present the results for fairness loss optimization, in order of the above table. Although the heuristic scores do not directly depend on the target vector ϕ, the target vector determines the advantaged and disadvantaged groups and is used in the final candidate evaluation. Therefore, when the same advantaged and disadvantaged groups are selected, different target values lead to similar optimization trajectories. For this reason, we only report representative target values:

$\phi = (0.1, 0.9)$ and $\phi = (0.9, 0.1)$. For the optimization results for the remaining target values ϕ and budgets $b \neq 500$, we refer the reader to Appendix A. Figures II.2–II.5 and the corresponding tables report the fairness loss before and after optimization. We start with `DBLP_gender: database`.



(a) Result for $\phi = (0.1, 0.9)$.

(b) Result for $\phi = (0.9, 0.1)$.

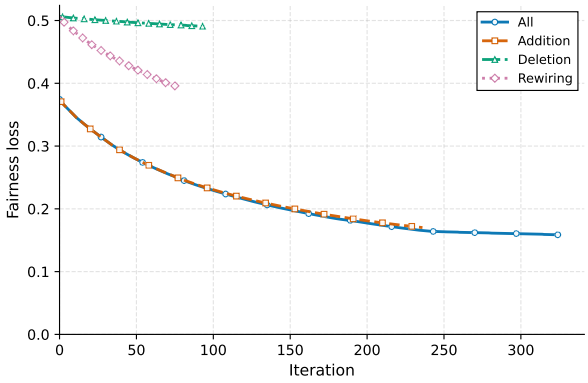
Figure II.2: Fairness loss after $b = 500$ iterations with $\phi = (0.1, 0.9)$ and $\phi = (0.9, 0.1)$ for `DBLP_gender: database` dataset. Addition-only, deletion-only, rewiring only and all edge operations are included.

And the corresponding table containing the (approximated) data:

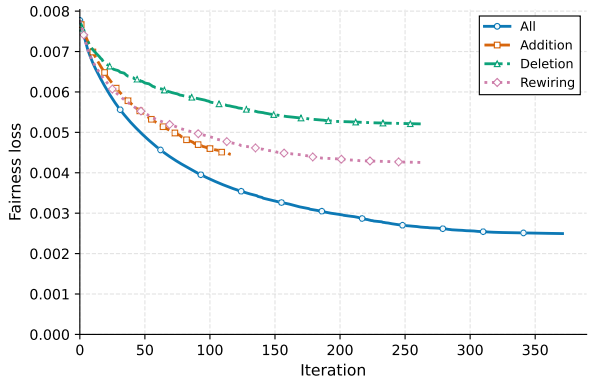
Operation	Original Fairness Loss	Final Fairness Loss	ϕ	Operations used	Decrease
Addition-only	0.47794	0.29806	$(0.1, 0.9)$	$b = 500$	37.64%
Addition-only	0.01181	0.00509	$(0.9, 0.1)$	$b = 301$	56.90%
Deletion-only	0.47794	0.45985	$(0.1, 0.9)$	$b = 128$	3.78%
Deletion-only	0.01181	0.00807	$(0.9, 0.1)$	$b = 430$	31.67%
Rewiring-only	0.47794	0.46558	$(0.1, 0.9)$	$b = 14$	2.59%
Rewiring-only	0.01181	0.00748	$(0.9, 0.1)$	$b = 148$	36.66%
All	0.47794	0.29226	$(0.1, 0.9)$	$b = 500$	38.85%
All	0.01181	0.00339	$(0.9, 0.1)$	$b = 500$	71.30%

Table II.3: Table containing original and resulting fairness loss with a budget $b = 500$, $\phi = (0.1, 0.9)$ and $\phi = (0.9, 0.1)$ for `DBLP_gender: database` dataset.

Next, we present the optimization results for the `DBLP_gender: datamining` dataset.



(a) Result for $\phi = (0.1, 0.9)$.



(b) Result for $\phi = (0.9, 0.1)$.

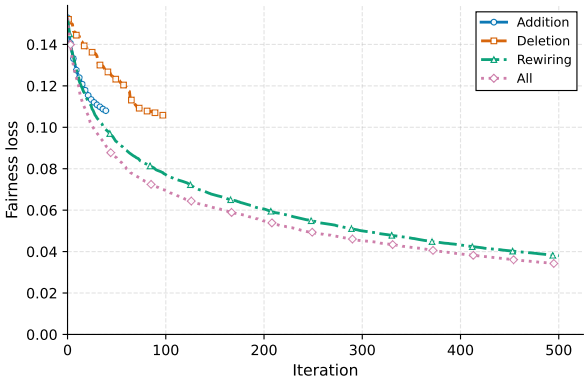
Figure II.3: Fairness loss after $b = 500$ iterations with $\phi = (0.1, 0.9)$ and $\phi = (0.9, 0.1)$ for `DBLP_gender: datamining` dataset. Addition-only, deletion-only, rewiring only and all edge operations are included.

The corresponding approximated values are reported in Table II.4.

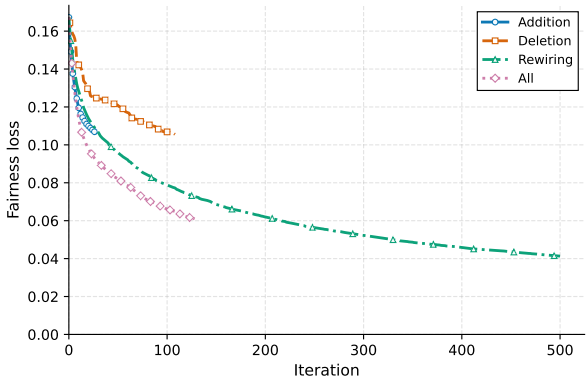
Operation	Original Fairness Loss	Final Fairness Loss	ϕ	Operations used	Decrease
Addition-only	0.50671	0.26267	$(0.1, 0.9)$	$b = 236$	48.16%
Addition-only	0.00777	0.00446	$(0.9, 0.1)$	$b = 116$	42.60%
Deletion-only	0.50671	0.49082	$(0.1, 0.9)$	$b = 128$	3.14%
Deletion-only	0.00777	0.00521	$(0.9, 0.1)$	$b = 262$	32.95%
Rewiring-only	0.50671	0.39433	$(0.1, 0.9)$	$b = 77$	22.18%
Rewiring-only	0.00777	0.00425	$(0.9, 0.1)$	$b = 264$	45.30%
All	0.50671	0.24829	$(0.1, 0.9)$	$b = 325$	51.00%
All	0.00777	0.00250	$(0.9, 0.1)$	$b = 371$	67.82%

Table II.4: Table containing original and resulting fairness loss with a budget $b = 500$, $\phi = (0.1, 0.9)$ and $\phi = (0.9, 0.1)$ for `DBLP_gender: datamining` dataset.

Now the results for the `polblogs` dataset.



(a) Result for $\phi = (0.1, 0.9)$.



(b) Result for $\phi = (0.9, 0.1)$.

Figure II.4: Fairness loss after $b = 500$ iterations with $\phi = (0.1, 0.9)$ and $\phi = (0.9, 0.1)$ for `polblogs` dataset. Addition-only, deletion-only, rewiring only and all edge operations are included.

Table II.5 reports the corresponding approximated values.

Operation	Original Fairness Loss	Final Fairness Loss	ϕ	Operations used	Decrease
Addition-only	0.15288	0.10776	$(0.1, 0.9)$	$b = 40$	29.51%
Addition-only	0.16729	0.10619	$(0.9, 0.1)$	$b = 27$	36.52%
Deletion-only	0.15288	0.10562	$(0.1, 0.9)$	$b = 100$	30.91%
Deletion-only	0.16729	0.10556	$(0.9, 0.1)$	$b = 108$	36.90%
Rewiring-only	0.15288	0.03794	$(0.1, 0.9)$	$b = 500$	75.18%
Rewiring-only	0.16729	0.04133	$(0.9, 0.1)$	$b = 500$	75.30%
All	0.15288	0.03405	$(0.1, 0.9)$	$b = 500$	77.73%
All	0.16729	0.06091	$(0.9, 0.1)$	$b = 128$	63.59%

Table II.5: Table containing original and resulting fairness loss with a budget $b = 500$, $\phi = (0.1, 0.9)$ and $\phi = (0.9, 0.1)$ for `polblogs` dataset.

The results for `Cora` follow below.

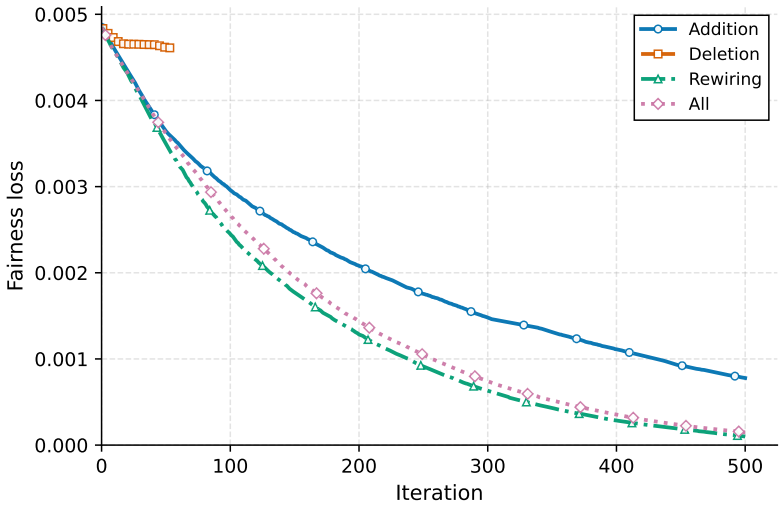


Figure II.5: Result of $b = 500$ on `Cora` dataset, $\phi_i = 1/7, 1 \leq i \leq 7$.

With the corresponding Table II.6.

Operation	Original Fairness Loss	Final Fairness Loss	ϕ	Operations used	Decrease
Addition-only	0.00485	0.00078	$(1/7, \ldots, 1/7)$	$b = 500$	83.92%
Deletion-only	0.00485	0.00461	$(1/7, \ldots, 1/7)$	$b = 53$	4.95%
Rewiring-only	0.00485	0.0001	$(1/7, \ldots, 1/7)$	$b = 500$	97.94%
All	0.00485	0.00015	$(1/7, \ldots, 1/7)$	$b = 500$	96.91%

Table II.6: Table containing original and resulting fairness loss with a budget $b = 500$, $\phi = (1/7, \ldots, 1/7)$ for `Cora` dataset.

II.3.3 PR-Unfairness Optimization Results

In this section, we evaluate the PR-unfairness optimization algorithm on the same datasets. For each dataset, we choose one protected group R and aim to increase its PageRank mass. Table II.7 gives the protected group and its original PageRank mass for each dataset.

Dataset	Protected group attribute	Original PageRank score
`database`	female	0.20867
`datamining`	female	0.18967
`Cora`	Neural Networks	0.13753
`polblogs`	liberal	0.49099

Table II.7: Overview of protected group attribute value and PageRank score for all datasets.

Since the behavior of Algorithm 2 is independent of the chosen ϕ value until reaching that threshold, we will only show figures containing the results for $c_{PR}(R) + 0.2$, and report after which

budget the thresholds $c_{PR}(R) + 0.1$ and $c_{PR}(R) + 0.2$ are reached, where $R \subset V$ is the chosen protected group. The datasets are presented in the same order as in Section II.3.2. Additional PR-unfairness results are provided in Appendix B. Figures II.6–II.9 and the corresponding tables report the PR-unfairness before and after optimization. We start with the `DBLP_gender: database` dataset.

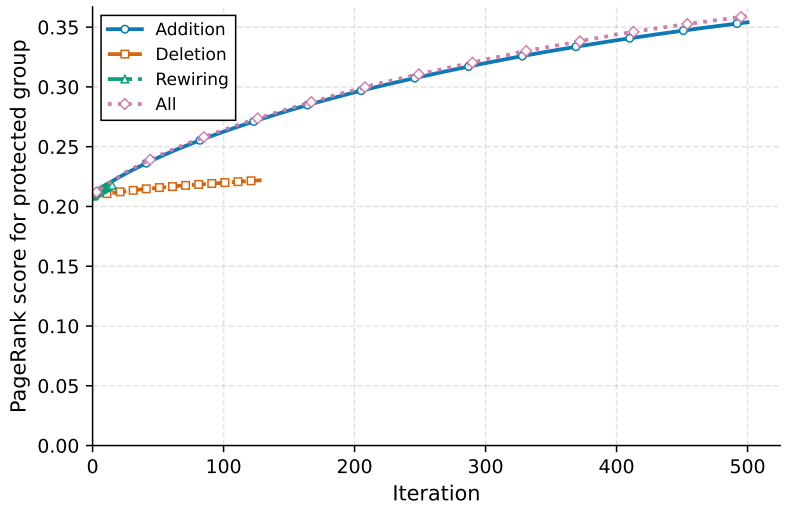


Figure II.6: Result of PR-unfairness algorithm with $b = 500$ for `DBLP_gender: database` dataset. Operations include: addition-only, deletion-only, rewiring-only and all operations.

Below is an overview of the fairness objective results for all operation types.

Operation	Target $c_{PR}(R) + 0.1$ reached	Target $c_{PR}(R) + 0.2$ reached
Addition-only	Yes, $b = 252$	No, final value 0.35405
Deletion-only	No, final value 0.22187	No, final value 0.22187
Rewiring-only	No, final value 0.21767	No, final value 0.21766
All	Yes, $b = 242$	No, final value 0.35939

Table II.8: Overview of PR-unfairness results for `DBLP_gender: database` dataset.

Next, we continue with the `DBLP_gender: datamining` dataset.

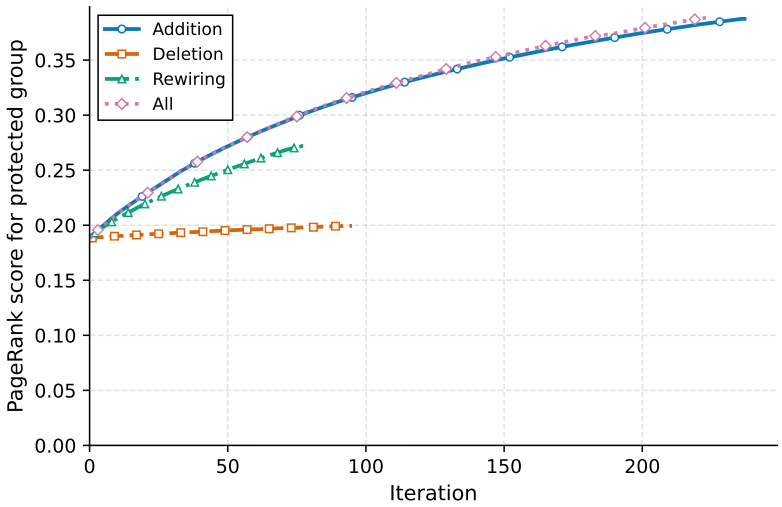


Figure II.7: Result of PR-unfairness algorithm with $b = 500$ for `DBLP_gender: datamining` dataset. Operations include: addition-only, deletion-only, rewiring-only and all operations.

Operation	Target $c_{PR}(R) + 0.1$ reached	Target $c_{PR}(R) + 0.2$ reached
Addition-only	Yes, $b = 65$	Yes, $b = 237$
Deletion-only	No, final value 0.19941	No, final value 0.19941
Rewiring-only	No, final value 0.27204	No, final value 0.27204
All	Yes, $b = 65$	Yes, $b = 223$

Table II.9: Overview of PR-unfairness results for `DBLP_gender: datamining` dataset.

Thirdly, we show results for the `polblogs` dataset.

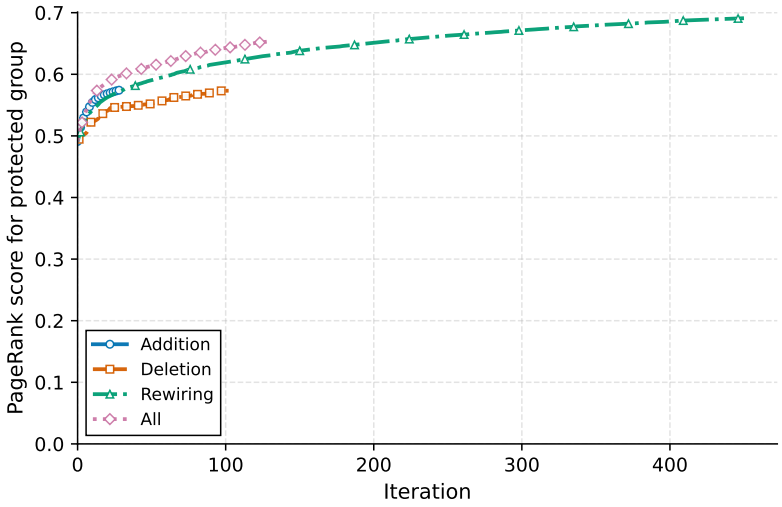


Figure II.8: Result of PR-unfairness algorithm with $b = 500$ for `polblogs` dataset. Operations include: addition-only, deletion-only, rewiring-only and all operations.

Operation	Target $c_{PR}(R) + 0.1$ reached	Target $c_{PR}(R) + 0.2$ reached
Addition-only	No, final value 0.57412	No, final value 0.57412
Deletion-only	No, final value 0.57332	No, final value 0.57332
Rewiring-only	Yes, $b = 51$	Yes, $b = 450$
All	Yes, $b = 23$	No, final value 0.65321

Table II.10: Overview of PR-unfairness results for `polblogs` dataset.

We finish with results for the `Cora` dataset.

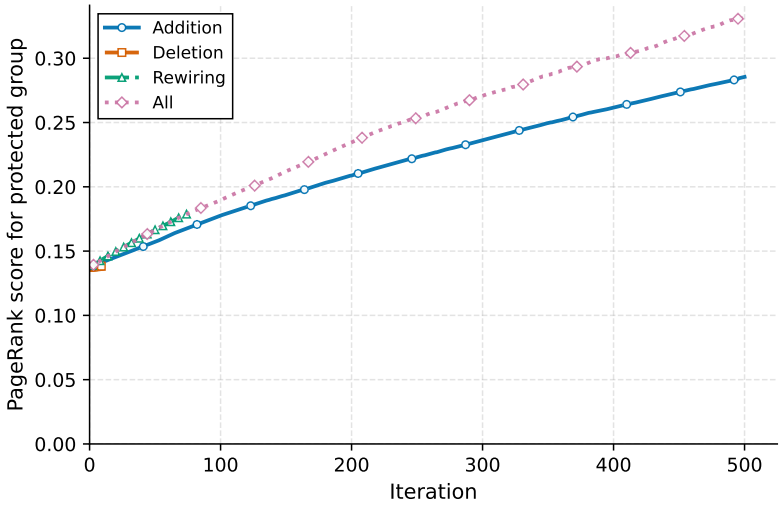


Figure II.9: Result of PR-unfairness algorithm with $b = 500$ for `Cora` dataset. Operations include: addition-only, deletion-only, rewiring-only and all operations.

Operation	Target $c_{PR}(R) + 0.1$ reached	Target $c_{PR}(R) + 0.2$ reached
Addition-only	Yes, $b = 297$	No, final value 0.28548
Deletion-only	No, final value 0.13825	No, final value 0.13825
Rewiring-only	No, final value 0.17879	No, final value 0.17879
All	Yes, $b = 207$	No, final value 0.33148

Table II.11: Overview of PR-unfairness results for `Cora` dataset.

II.3.4 Comparison and Discussion

We now compare the results of the fairness loss optimization and PR-unfairness optimization experiments. Overall, the experiments show that the proposed greedy algorithms are able to improve the corresponding fairness objectives on all the datasets considered. However, the amount of improvement depends on the dataset, the fairness objective, and the allowed edge operation type.

For fairness loss optimization, the results show that addition-only and all-operations are effective on the two DBLP subnetworks. In both the `DBLP_gender: database` and `DBLP_gender: datamining` networks, addition-only already gives a substantial decrease in fairness loss. The all-operations setting gives the largest improvement, although this is only a slightly larger decrease than addition-only. This suggests that in these networks adding edges from the advantaged (male) group to the disadvantaged (group) is often sufficient. Deletion-only and rewiring-only give only small improvements and get stuck in a local optimum quickly.

For the `polblogs` dataset, this behavior is completely different. Here, rewiring-only and all-operations give the strongest decrease in fairness loss, where rewiring-only performs even better than using all-operations when $\phi = (0.9, 0.1)$. Addition and deletion terminate much earlier and give a worse improvement. For this network, it thus seems that adding and deleting edges is much less effective than rewiring edges. Rewiring combines the effect of deleting an edge and adding a new edge. Therefore, it could potentially change the PageRank mass more effectively, which is what is observed for the `polblogs` dataset.

The multi-group experiment on the `Cora` dataset shows that the algorithm can also improve fairness loss in the multi-group setting. Deletion-only again gives a relatively small decrease in fairness loss, supporting the view that deletion-only has limited effect in these experiments. Meanwhile, addition-only and rewiring-only give a significant decrease in fairness loss, where rewiring-only gives the largest improvement. Furthermore, rewiring-only performs better than the all-operations setting, again showing that rewiring is a powerful edge operation. Since `Cora` consists of seven groups, the algorithm has to repeatedly select the most advantaged and most disadvantaged active groups. The strong improvement for this dataset suggest that Algorithm 1 can still be useful in a multi-group setting, although groups are only optimized pairwise.

For PR-unfairness optimization, the results indicate a similar dependence on the edge operation in the results. For both `DBLP` subnetworks, addition-only clearly performs significantly better than deletion-only and rewiring-only. Similarly, deletion-only and rewiring-only get stuck in a local optimum quickly. For the `DBLP_gender: database` network, both the addition-only and all-operations settings reach the threshold $c_{PR}(R)+0.1$, but do not reach the $c_{PR}(R)+0.2$ threshold. In the `data` mining network, both addition-only and all-operations reach the thresholds $c_{PR}(R) + 0.1$ and $c_{PR}(R) + 0.2$, while deletion-only and rewiring do not. This indicates that the ability to add edges to the protected group is important for increasing its PageRank mass.

For the `polblogs` dataset, rewiring performs the best for PR-unfairness optimization. Rewiring-only reaches both thresholds, while all-operations only reaches the $c_{PR}(R) + 0.1$ threshold quickly but does not reach the $c_{PR}(R) + 0.2$ threshold within the budget. Addition-only and deletion-only do not reach any threshold. This again shows a limitation of the all-operations setting. Although it has access to all operation types, it may choose locally beneficial operations that do not lead to the best long-term improvement.

The experiment on the `Cora` dataset shows that combining all operations in a multi-group setting performs significantly better than exclusively one edge operation for increasing the PageRank score of a selected protected group. Furthermore, addition-only and all-operations reach the $c_{PR}(R) + 0.1$ threshold, while none of the operation settings reach the $c_{PR}(R) + 0.2$ threshold within the budget. This may be due to the fact that increasing the PageRank score of one group can be more difficult when the graph has several groups.

Across all experiments, deletion-only usually performs worst. Deletion-only can remove existing edges within the most advantaged group, but cannot create paths from the most advantaged group

to the most disadvantaged or protected group. Meanwhile, addition can do the latter and rewiring can remove both existing paths and create new paths simultaneously. The all-operations setting usually performs well, but the results show that it is not always superior to the best individual operation type. This is a consequence of the greedy nature of the algorithms.

The experiments also show that several runs terminate before reaching the full budget or target threshold, meaning that no improving candidate operation was found among the generated candidates. This does not imply that no better graph exists within the budget. Rather, the candidate generation and evaluation process did not find an improving candidate. This is in line with the limitations of the algorithms discussed in Section II.2.5. The algorithms do not explore the whole search space and follow a linear trajectory through the search space. Hence, they are not guaranteed to find a globally optimum sequence of edge operations.

In conclusion, the experiments show that the proposed greedy algorithms can significantly improve PageRank fairness objectives on both two-group and multi-group networks. The improvement is the largest when the allowed operation set includes edge addition or rewiring, while deletion-only is generally less effective. The results also show that the effectiveness of an operation type depends on the (structure of the) network. Therefore, the algorithms should be viewed as heuristic methods for improving fairness.

Chapter III

Fairness Estimation Using Graphons

In this chapter, we develop a fairness framework for graphons. We conclude by proving finite graph fairness convergence and showing that for our centrality measures of interest the proportional fairness in the 2-block SBM is equivalent to the two blocks having the same degree.

III.1 Graphon Framework for Fairness Estimation

In this section, we develop a framework for using graphons as estimators for fairness.

Recall from Section I.2 that a *graphon* is a measurable symmetric function

$$W : [0,1]^2 \to [0,1].$$

Here, $W(x,y)$ can be interpreted as the edge probability between two individuals $x, y \in [0,1]$. Generally, graphons are used as the limit of graphs increasing in size [6]. In this section, we aim to study the following:

Given a graphon W, if a large graph is sampled from W, does the finite graph fairness converge to a deterministic graphon fairness score?

The answer is yes under certain conditions. We now develop the framework for these conditions. We start with empirical graphons.

Definition III.1.1 (Empirical graphon)**.** Let G_n be a simple undirected graph on a vertex set $V = \{1, 2, \ldots, n\}, n \in \mathbb{N}_{\geq 1}$ with adjacency matrix $A = (A_{ij})_{1 \leq i,j \leq n}$. We partition $[0,1]$ into intervals

$$I_i^n = \begin{cases} \left[\frac{i-1}{n}, \frac{i}{n}\right), 1 \leq i \leq n-1, \\ \left[\frac{n-1}{n}, 1\right], i = n. \end{cases}$$

The *empirical graphon* associated with G_n is the step function

$$W_{G_n}(x,y) = A_{ij}, (x,y) \in I_i^n \times I_j^n$$

For comparing the empirical graphon with graphon operators it is often convenient to rescale the adjacency matrix A by $1/n$, that is, $(A/n)_{ij} = \frac{A_{ij}}{n}$, for $n \in \mathbb{N}_{\geq 1}$. Rescaling A to A/n will prevent the operator from diverging (in norms).

Definition III.1.2 ((W)-random-graph)**.** Given a graphon W, the $(W$-$)$*random graph* $G_n \sim W$ is given by: Let $U_1, \ldots, U_n$ be independent and identically distributed (i.i.d.), $U_i \sim U(0,1), 1 \leq i \leq n$. Conditionally on $U_1, \ldots, U_n$, put an edge between i and j independently with probability

$$\mathbb{P}\left(A_{ij} = 1 \mid (U_i)_{i=1}^n\right) = W(U_i, U_j), i < j.$$

If $W \equiv p$ then G_n corresponds with an Erdős-Rényi random graph [2]. If W is a piecewise constant function on a finite partition of $[0,1]$, then this random process describes the stochastic block model [2].

Definition III.1.3 (Labeled graphon)**.** For fairness purposes we will work with *labeled graphons*: besides W we specify either a measurable protected set $R \subseteq [0,1]$ or a measurable group label function $\mathcal{G}\colon [0,1] \to \{1, 2, \ldots, k\}, k \in \mathbb{N}_{\geq 1}$.

Note that given $\mathcal{G}$, we can take $R = \{x \in [0,1]\colon g(x) = 1\}$, where R may represent a *protected group*. Thus, a relevant pair would be (W, R) or $(W, \mathcal{G})$.

Definition III.1.4 (Graphon operator)**.** To every graphon W, we can associate an integral operator, as defined in [4]:

$$T_W\colon L^2([0,1]) \to L^2([0,1])$$

given by

$$(T_W f)(x) = \int_0^1 W(x,y) f(y) dy.$$

Since $W \in L^2([0,1]^2)$, T_W is a Hilbert-Schmidt operator and hence a compact operator. Since W is symmetric, T_W is self-adjoint. T_W has real eigenvalues and an orthonormal basis of eigenfunctions.

Definition III.1.5 (Normalized graphon centrality)**.** For a nonnegative centrality measure $c_W \in L^1([0,1])$ with $\int_0^1 c_W(x)dx > 0$ we define its *normalized version* by

$$\overline{c}_W(x) = \frac{c_W(x)}{\int_0^1 c_W(u)du}$$

Then, $\int_0^1 \overline{c}_W(x)dx = 1$.

We will now define centrality measures for graphons. By W we always mean a *graphon* $W\colon [0,1]^2 \to [0,1]$.

Definition III.1.6 (Graphon degree centrality)**.** The graphon *degree centrality* ([2]) d_W is given by

$$d_W(x) = (T_W \mathbb{1})(x) = \int_0^1 W(x,y)dy$$

The *normalized degree centrality* $\overline{d}_W$ is

$$\overline{d}_W(x) = \frac{d_W(x)}{\int_0^1 d_W(u)du},$$

given that $\int_0^1 d_W(u)du > 0$.

Definition III.1.7 (Graphon eigenvector centrality)**.** Assume that the largest eigenvalue $\lambda_1(W)$ of T_W is positive and simple. Let ϕ_1 be the nonnegative eigenfunction, normalized by

$$\|\varphi_1\|_{L^2} = 1.$$

Thus we mean $T_W\varphi_1 = \lambda_1(W)\varphi_1$.
The graphon *eigenvector centrality* e_W is

$$e_W(x) = \varphi_1(x),$$

and the *normalized eigenvector centrality* $\overline{e}_W$ is

$$\overline{e}_W(x) = \frac{e_W(x)}{\int_0^1 e_W(u)du},$$

assuming $\int_0^1 e_W(u)du > 0$.

Definition III.1.8 (Graphon Katz centrality)**.** Let $\alpha > 0$. The graphon *Katz centrality* $K_{W,\alpha}$ is

$$K_{W,\alpha} = (I - \alpha T_W)^{-1} \mathbb{1}$$

whenever the inverse exists, specifically in the case of $0 < \alpha < \dfrac{1}{\|T_W\|_{op}}$,

$$K_{W,\alpha} = \sum_{l=0}^{\infty} \alpha^l T_W^l \mathbb{1}.$$

This is the graphon analogue of counting walks of all lengths, with walks of length l down weighted by α^l.
The *normalized Katz centrality* is

$$\overline{K}_{W,\alpha}(x) = \frac{K_{W,\alpha}(x)}{\int_0^1 K_{W,\alpha}(u)du}.$$

Definition III.1.9 (Graphon PageRank centrality)**.** Assume $d_W(x) = \int_0^1 W(x,y)dy > 0$ almost everywhere (a. e.) for $x \in [0,1]$. The graphon *PageRank centrality* P_W is given by

$$(P_W f)(x) = \int_0^1 W(x,y)\frac{f(y)}{d_W(y)}\,dy.$$

This is the continuum analogue of AD^{-1}. For a damping factor $\beta \in (0,1)$, define PageRank by

$$p_{W,\beta} = (1-\beta)(I - \beta P_W)^{-1}\mathbb{1},$$

$$p_{W,\beta} = (1-\beta)\sum_{l=0}^{\infty} \beta^l P_W^l \mathbb{1}.$$

PageRank is already normalized; $\int_0^1 p_{W,\beta}(x)dx = 1$. This can be seen as follows:

$$\int_0^1 \left(P_W f\right)(x)dx = \int_0^1 \frac{f(y)\int_0^1 W(x,y)dx}{d_W(y)}dy = \int_0^1 f(y)dy.$$

P_W preserves the mass and the series give

$$\int_0^1 p_{W,\beta}(x)dx = (1-\beta)\sum_{l=0}^{\infty}\beta^l = 1.$$

Having defined centrality measures for graphons, we now turn to fairness definitions for graphon centralities.

Definition III.1.10 (Graphon ϕ-fairness)**.** Let $R \subseteq [0,1]$ be a measurable protected group. For the normalized centrality function $\overline{c}_W$, define $M_c(W,R) = \int_R \overline{c}_W(x)dx$. Let $\phi \in [0,1]$ be a target value. The centrality c is called *ϕ-fair* w.r.t. R if

$$M_c(W,R) = \phi.$$

When $\phi = \lambda(R)$, we call c *proportionally fair* w.r.t. R.

Definition III.1.11 (Graphon fairness gap)**.** Given $R \subseteq [0,1]$, $M_c(W,R)$ and $\phi \in [0,1]$ as in the previous definition, the *fairness gap* $\Delta_c(M,R,\phi)$ is

$$\Delta_c(M,R,\phi) = M_c(W,R) - \phi.$$

A one sided under representation condition is $M_c(W,R) \geq \phi$, or equivalently R is underrepresented if $M_c(W,R) < \phi$.
A normal choice is $\phi = \lambda(R)$, where $\lambda(R)$ is the Lebesgue measure of R. This means that the protected group should receive centrality mass proportional to its population share. Next is a two sided definition (lower and upper bound).

Definition III.1.12 (Graphon ϵ-fairness)**.** Given the fairness gap $\Delta_c(M,R,\phi)$, we call the centrality c *ϵ-fair* if

$$|\Delta_c(M,R,\phi)| \leq \epsilon.$$

Definition III.1.13 (Graphon ϕ-sum fairness)**.** For $k \in \mathbb{N}_{\geq 1}$, write $[0,1] = R_1 \cup \cdots \cup R_k, k \in \mathbb{N}_{\geq 1}$ and target vector $\phi_1, \ldots, \phi_k$ with $\phi_a \geq 0, 1 \leq a \leq k$ and $\sum_{a=1}^k \phi_a = 1$. Define

$$M_c\left(W,R_a\right) = \int_{R_a} \overline{c}_W(x)dx.$$

Then c is *ϕ-sum-fair* if

$$M_c\left(W,R_a\right) = \phi_a, 1 \leq a \leq k.$$

We continue to show the continuity of graphon fairness for our desired centrality measures. This is needed to claim graphons fairness is robust under graph approximations. The general idea is: if empirical graphons converge to W, then the correlated normalized centrality functions converge to graphon centrality functions.

Lemma III.1.14 (Graphon degree fairness continuity)**.** *The degree fairness is continuous: let W_n, W be graphons such that*

$$\|W_n - W\|_{L^2([0,1]^2)} \to 0.$$

Let $d_n = T_{W_n}\mathbb{1}, d = T_W\mathbb{1}$. Then $\|d_n - d\|_{L^2([0,1])} \to 0$. If $\int_0^1 d(x)dx > 0$, then $\|\overline{d}_n - \overline{d}\|_{L^1([0,1])} \to 0$. Hence, for every measurable $R \subseteq [0,1]$,

$$M_d(W_n, R) \to M_d(W, R).$$

Proof. For every $x \in [0,1]$,

$$d_n(x) - d(x) = \int_0^1 W_n(x,y) - W(x,y)dy.$$

By Cauchy-Schwarz,

$$|d_n(x) - d(x)|^2 \leq \int_0^1 |W_n(x,y) - W(x,y)|^2 dy.$$

Integrating gives

$$\|d_n - d\|_{L^2}^2 \leq \|W_n - W\|_{L^2}^2$$

So $d_n \xrightarrow{L^2} d$ and hence in L^1 also.
Let $a_n = \int_0^1 d_n(x)dx, a = \int_0^1 d(x)dx$. Since $d_n \xrightarrow{L^1} d$, we have $a_n \to a > 0$. Also,

$$\overline{d}_n - \overline{d} = \frac{d_n}{a_n} - \frac{d}{a} = \frac{d_n - d}{a_n} + d\left(\frac{1}{a_n} - \frac{1}{a}\right).$$

So $\|\overline{d}_n - \overline{d}\|_{L^1} \leq \frac{\|d_n - d\|_{L^1}}{a_n} + \|d\|_{L^1}\frac{|a_n - a|}{a_n a}$. Since $a > 0, a_n > 0$ for large n. Hence, the right hand side (RHS) tends to 0.
Finally, note

$$|M_d(W_n, R) - M_d(W, R)| \leq \|\overline{d}_n - \overline{d}\|_{L^1} \to 0.$$

□

Lemma III.1.15 (Graphon eigenvector fairness continuity)**.** *The eigenvector fairness is continuous: suppose $\|T_{W_n} - T_W\|_{op} \to 0$. Assume $\lambda_1(W)$ of T_W is simple and separated from the rest of the spectrum, i.e. $\lambda_1(W) - \lambda_2(W) \geq \delta$.*
Let φ_1 be the L^2-normalized, positive eigenfunction of T_W, $\varphi_{1,n}$ be the corresponding for T_{W_n}. Then,

$$\|\varphi_{1,n} - \varphi_1\|_{L^2} \to 0.$$

If $\int_0^1 \varphi_1(x)dx > 0$, then $\|\overline{e}_n - \overline{e}\|_{L^1} \to 0$ and hence

$$M_e(W_n, R) \to M_e(W, R).$$

Proof. Since T_W is compact and self-adjoint, the Davis-Kahan type inequality (Lemma 8 of [2]) shows there exists a constant C_r such that

$$\|\varphi_{1,n} - \varphi_1\|_{L^2} \leq C_r\|T_{W_n} - T_W\|_{op}.$$

Note that we need to choose a sign, so

$$\langle \varphi_{1,n}, \varphi_1 \rangle \geq 0.$$

Hence $\varphi_{1,n} \xrightarrow{L^2} \varphi_1$ and also in L^1. The rest of the argument is similar. □

Lemma III.1.16 (Graphon Katz fairness continuity)**.** *The Katz fairness is continuous: Let W_n, W be graphons such that*

$$\|T_{W_n} - T_W\|_{op} \to 0.$$

Fix $\alpha > 0$ such that $\alpha < \frac{1}{\|T_W\|_{op}}$. Also assume that for all n large, $\alpha < \frac{1}{\|T_{W_n}\|_{op}}$.
Let $\kappa_n = (I - \alpha T_{W_n})^{-1} \mathbb{1}, \kappa = (I - \alpha T_W)^{-1} \mathbb{1}$.
Then $\|\kappa_n - \kappa\|_{L^2} \to 0$. If $\int_0^1 \kappa(x)dx > 0$ then $\|\overline{\kappa}_n - \overline{\kappa}\|_{L^1} \to 0$ and hence $M_\kappa(W_n, R) \to M_\kappa(W, R)$

Proof. We have

$$(I - \alpha T_{W_n})^{-1} - (I - \alpha T_W)^{-1} = \alpha (I - \alpha T_{W_n})^{-1} (T_{W_n} - T_W) (I - \alpha T_W)^{-1}.$$

Multiply by $\mathbb{1}$ on both sides:

$$\kappa_n - \kappa = \alpha (I - \alpha T_{W_n})^{-1} (T_{W_n} - T_W) (I - \alpha T_W)^{-1} \mathbb{1}$$

Hence,

$$\|\kappa_n - \kappa\|_{L^2} \leq \alpha \| (1 - \alpha T_{W_n})^{-1} \|_{op} \|T_{W_n} - T_W\|_{op}$$

The inverse operators are uniformly bounded as $\alpha \|T_{W_n}\|_{op} < 1$.
Since $\|T_{W_n} - T_W\|_{op} \to 0$, this shows $\kappa_n \to \kappa$ in L^2. The rest of the proof follows as in the continuity of the degree fairness. □

Lemma III.1.17 (Graphon PageRank fairness continuity)**.** *The PageRank fairness is continuous: let W_n, W be graphons and assume*

$$\|W_n - W\|_{L^\infty[0,1]^2} \to 0 \text{ [stronger assumption].}$$

Assume $d_W(x) \geq \delta$ a. e. for $x \in [0, 1]$. Fix $\beta \in (0, 1)$, define

$$p_n = (1 - \beta) (I - \beta P_{W_n})^{-1} \mathbb{1},$$

And as in Definition III.1.9,

$$p = (1 - \beta) (I - \beta P_W)^{-1} \mathbb{1}.$$

Then $\|p_n - p\|_{L^1} \to 0$.
Consequently, $M_{P_{W_n}}(W_n, R) \to M_{P_W}(W, R)$.

Proof. Since

$$d_{W_n}(x) - d_W(x) = \int_0^1 W_n(x, y) - W(x, y)dy$$

we have

$$\|d_{W_n} - d_W\|_\infty \leq \|W_n - W\|_\infty \to 0.$$

We further have that $d_{W_n} \geq \frac{\delta}{2}$ for n large (since $d_W > \delta$).
For $f \in L^1([0, 1])$ see that

$$\begin{aligned}(P_{W_n} f)(x) - (P_W f) x &= \int_0^1 W_n(x, y) \frac{f(y)}{d_{W_n}(y)} dy - \int_0^1 W(x, y) \frac{f(y)}{d_W(y)} dy \\ &= \int_0^1 \left(\frac{W_n(x, y)}{d_{W_n}(y)} - \frac{W(x, y)}{d_W(y)} \right) f(y)dy\end{aligned}$$

Using the lower bound on degrees and $W_n \to W$ in L^∞ we have that $\|p_{W_n} - p_W\|_{L^1} \to 0$. Moreover,

$$\|P_W f\|_{L^1} \leq \int_0^1 |f(y)| \frac{\int_0^1 W(x,y)dx}{d_W(y)} dy = \|f\|_{L^1}.$$

So, $\| (I - \beta P_W)^{-1} \|_{L^1} \leq \frac{1}{1-\beta}$ and $\| (I - \beta P_{W_n})^{-1} \|_{L^1} \leq \frac{1}{1-\beta}$.
So using resolvent identity as in Katz

$$p_n - p = (1-\beta)\beta (I - \beta P_{W_n})^{-1} (P_{W_n} - P_W) (I - \beta P_W)^{-1} \mathbb{1}$$

and thus

$$\|p_n - p\|_{L^1} \leq c_\beta \|P_{W_n} - P_W\|_{L^1} \to 0$$

So $p_n \xrightarrow{L^1} p$. Since PageRank is normalized,

$$|M_{P_{W_n}}(W_n, R) - M_{P_W}(W, R)| \leq \|p_n - p\|_{L^1} \to 0.$$

□

We are now going to prove the main theorem: convergence of finite graph fairness.
Let G_n be sampled from W as in Definition III.1.2. Let $\hat{W}_n$ be the empirical graphon constructed from the adjacency matrix of G_n as in Definition III.1.1, after ordering the vertices. Let c_n be one of the finite graph centralities computed from A/n. Define a step function

$$C_{\hat{W}_n}(x) = c_n(i), x \in I_i^n.$$

and let $R \subseteq [0,1]$ be a protected group. Define the *empirical protected group* by

$$R_n = \{i : u_i \in R\}.$$

and

$$M_c(G_n, R_n) := \sum_{i \in R_n} \overline{c}_n(i).$$

Then we have the following main theorem:

Theorem III.1.18 (Finite graph fairness convergence)**.** *Assume that* $\|T_{\hat{W}_n} - T_W\|_{op} \xrightarrow{\mathbb{P}} 0$. *Then:*
(i) For the graphon degree centrality d:

$$M_d(G_n, R_n) \xrightarrow{\mathbb{P}} M_d(W, R).$$

(ii) For the graphon Katz centrality K: if $0 < \alpha < \frac{1}{\|T_W\|_{op}}$*:*

$$M_K(G_n, R_n) \xrightarrow{\mathbb{P}} M_K(W, R).$$

(iii) For the graphon PageRank centrality P: if $d_W \geq \delta$ *and* $\|W_n - W\|_{L^\infty[0,1]^2} \to 0$ *[stronger assumption]:*

$$M_P(G_n, R_n) \xrightarrow{\mathbb{P}} M_P(W, R).$$

(iv) For the eigenvector centrality e: if $\lambda_1(T_W) - \lambda_2(T_W) > \delta > 0$*:*

$$M_e(G_n, R_n) \xrightarrow{\mathbb{P}} M_e(W, R)$$

Proof. The results follow from an application of the continuity results (Lemmas III.1.14, III.1.15, III.1.16, III.1.17).
Let f be a step function such that $f(x) = f_j, x \in I_j^n$. Now, for $x \in I_i^n$ we have:

$$\begin{aligned}\left(T_{\hat{W}_n} f\right)(x) &= \int_0^1 \hat{W}_n(x,y) f(y) dy \\ &= \sum_{j=1}^n \int_{I_j^n} A_{ij}^n f_j dy \\ &= \frac{1}{n} \sum_{j=1}^n A_{ij}^n f_j\end{aligned}$$

Thus $T_{\hat{W}_n}$ corresponds to A/n. Moreover,

$$\int_0^1 C_n(x) dx = \sum_{i=1}^n \int_{I_i^n} \frac{c_n(i)}{\frac{1}{n}\sum_{j=1}^n c_n(j)} = 1.$$

Now

$$\begin{aligned}\int_{R_n} C_n(x) &= \sum_{i \in R_n} \int_{I_i^n} \frac{c_n(i)}{\frac{1}{n}\sum_{j=1}^n c_n(j)} dx \\ &= \frac{\sum_{i \in R_n} c_n(i)}{\sum_{j=1}^n c_n(j)} = \overline{c}\end{aligned}$$

Let $R_n^* \subseteq [0,1]$ be the step function representation of R (vertices are ordered):

$$R_n^* = \bigcup_{i:\, U_i \in R} I_i^n.$$

If $\lambda(\partial R) = 0$ then $R_n^* \to R$ in mean. From assuming $\|T_{\hat{W}_n} - T_W\|_{op} \xrightarrow{\mathbb{P}} 0$ it follows that

$$|M_c(G_n, R_n) - M_c(W, R)| \leq \|\overline{C}_{\hat{W}_n} - \overline{C}_W\|_{L^1} + \int_{R_n^*} \overline{C}_W(x) dx - \int_R \overline{C}_W(x) dx$$

From continuity results $\|\overline{C}_{\hat{W}_n} - \overline{C}_W\|_{L^1} \xrightarrow{\mathbb{P}} 0$. When $\lambda(\partial R) = 0$, $R_n^* \to R$ and thus $\int_{R_n^*} \overline{C}_W(x) dx - \int_R \overline{C}_W(x) dx \to 0$. Therefore, $\|\overline{C}_{\hat{W}_n} - \overline{C}_W\|_{L^1} + \int_{R_n^*} \overline{C}_W(x) dx - \int_R \overline{C}_W(x) dx$ goes to 0, and hence $|M_c(G_n, R_n) - M_c(W, R)| \xrightarrow{\mathbb{P}} 0$. □

We have thus shown the convergence (in probability) of finite graph fairness to graphon fairness for the degree, eigenvector, Katz and PageRank centrality. We now apply the developed fairness framework to the *Stochastic Block Model (SBM)*. This follows naturally, since the SBM graphon is a piecewise constant graphon. We first develop the required terminology for the SBM, from [2].

Definition III.1.19 (SBM graphon)**.** Let $m \in \mathbb{N}_{\geq 1}$ and write

$$[0,1] = \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_{m-1} \cup \mathcal{B}_m.$$

where $\mathcal{B}_i \cap \mathcal{B}_j = \emptyset, i \neq j$. Let $P = (P_{ab})_{1 \leq a,b \leq m}$ be the symmetric matrix with entries in $[0,1]$. The *SBM graphon* is

$$W_{\text{SBM}}(x,y) = \sum_{a=1}^{m} \sum_{b=1}^{m} P_{ab} \mathbb{1}_{\mathcal{B}_a}(x) \mathbb{1}_{\mathcal{B}_b}(y).$$

Let $q_a = \lambda(B_a) > 0, 1 \leq a \leq m$, with $\sum_{a=1}^{m} q_a = 1$. Now, q_a is intuitively the size of block $\mathcal{B}_a, 1 \leq a \leq m$. Define

$$Q := \text{diag}(q_1, \ldots, q_m).$$

and

$$E = PQ.$$

where $E_{ab} = P_{ab} q_b$ and write

$$d = E\mathbb{1}.$$

Now,

$$d_a = \sum_{b=1}^{m} P_{ab} q_b, 1 \leq a \leq m.$$

is the graphon degree of every point in block $\mathcal{B}_a$, with $d_a > 0$ for every block.

Suppose $f(x) = f_b$ for $x \in \mathcal{B}_b$. Note that for $x \in \mathcal{B}_a$

$$(T_W f)(x) = \sum_{b=1}^{m} P_{ab} \int_{\mathcal{B}_b} f(y) dy = \sum_{b=1}^{m} P_{ab} q_b f_b = (Ef)_a .$$

Thus, the graphon operator T_W restricted to block constant functions is represented by E. We now derive the graphon block degree formula for the degree, eigenvector, Katz and PageRank centrality.

Definition III.1.20 (Block centrality values)**.** Let W_{SBM} be an SBM graphon from Definition III.1.19. For $1 \leq a \leq m$, consider the block $\mathcal{B}_a$.
(i) For the degree centrality d, taking $f = \mathbb{1}$, the *block degree centrality* is

$$d_a = (E\mathbb{1})_a .$$

(ii) For the eigenvector centrality e, note that eigenfunctions of T_W that are constant on blocks correspond to eigenvectors of E.

$$\|e\|_{L^2}^2 = \sum_{a=1}^{m} q_a e_a^2 = e^T Q e$$

and, when v is the positive right eigenvector of E corresponding to the largest eigenvalue λ_1, it now follows that the *block eigenvector centrality* is

$$e_a = \frac{v_a}{\sqrt{v^T Q v}}.$$

(iii) For the Katz centrality K, if $0 < \alpha < \frac{1}{\rho(E)}$, define

$$k_\alpha = (I - \alpha E)^{-1} \mathbb{1}.$$

The *block Katz centrality* is

$$k_a = (k_\alpha)_a .$$

(iv) Let $D = \operatorname{diag}(E\mathbb{1})$. Given $\beta \in (0,1)$, define

$$p_\beta = (1-\beta)\left(I - \beta E D^{-1}\right)^{-1}\mathbb{1}.$$

The *block PageRank centrality* is

$$p_a = (p_\beta)_a .$$

We conclude this section by showing that, for degree, eigenvector, Katz, and PageRank centrality on a two-block SBM graphon, proportional fairness is equivalent to the two blocks having equal graphon degree.

Theorem III.1.21. *Consider the 2-block SBM graphon W where*

$$[0,1] = \mathcal{B}_1 \cup \mathcal{B}_2.$$

and write $R = \mathcal{B}_1, R^c = \mathcal{B}_2$. Let $d_1 = (E\mathbb{1})_1, d_2 = (E\mathbb{1})_2$ be the respective block degrees of R, R^c and assume that $P_{21} > 0$.

Take $\rho \in (0,1)$ with $\rho = \lambda(R)$ and hence $q_1 = \rho, q_2 = 1-\rho$. Let c be the degree, eigenvector, Katz or PageRank centrality measure. Then:

$$W \text{ is proportionally fair w.r.t. } R \text{ if and only if } d_1 = d_2.$$

Equivalently,

$$\Delta_c(M, R, \rho) = 0 \text{ if and only if } d_1 = d_2.$$

Proof. Write

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} = \begin{pmatrix} P_{11} & P_{21} \\ P_{21} & P_{22} \end{pmatrix}$$

Let

$$E = PQ = \begin{pmatrix} P_{11}\rho & P_{21}(1-\rho) \\ P_{21}\rho & P_{22}(1-\rho) \end{pmatrix}.$$

We get block degrees $d_1 = P_{11}q_1 + P_{12}q_2 = P_{11}\rho + P_{21}(1-\rho), d_2 = P_{21}q_1 + P_{22}q_2 = P_{21}\rho + P_{22}(1-\rho)$. We prove the above theorem separately for the degree (i), eigenvector (ii), Katz (iii) and PageRank centrality (iv):

(i): For the degree centrality d we know that the block centrality values for blocks $\mathcal{B}_1 = R, \mathcal{B}_2 = R^c$ are respectively d_1, d_2. So

$$M_d(W, R) = \frac{\rho d_1}{\rho d_1 + (1-\rho)d_2}.$$

First we assume that W is proportionally fair w.r.t. R, so

$$\phi = \rho = M_d(W, R) = \frac{\rho d_1}{\rho d_1 + (1-\rho)d_2}$$

Hence

$$\frac{d_1}{\rho d_1 + (1-\rho)d_2} = 1 \implies d_1 = \rho d_1 + (1-\rho)d_2.$$

Since $\rho \in (0,1)$, we have $\rho + (1-\rho) = 1$. Now $d_1 = \rho d_1 + (1-\rho)d_2$ and $d_1 = \rho d_1 + (1-\rho)d_1$ so $(1-\rho)d_1 = (1-\rho)d_2$. From $\rho \in (0,1)$ it now follows that $d_1 = d_2$.

Conversely, assume that $d_1 = d_2$. Then

$$M_d(W,R) = \frac{\rho d_1}{\rho d_1 + (1-\rho)d_2} = \frac{\rho d_1}{\rho d_1 + (1-\rho)d_1} = \frac{\rho d_1}{d_1} = \rho.$$

and thus $\Delta_d(W,R,\rho) = 0$.
Hence, $\Delta_d(W,R,\rho) = 0 \iff d_1 = d_2$.
(ii): Since $P_{21} > 0$ and $\rho \in (0,1)$, E is irreducible. Hence, from the Perron-Frobenius theorem the largest eigenvalue λ_1 has a positive right eigenvector v. For the eigenvector centrality e the block centrality values for blocks $\mathcal{B}_1 = R, \mathcal{B}_2 = R^c$ are respectively

$$e_1 = \frac{v_1}{\sqrt{v^T Q v}}, e_2 = \frac{v_2}{\sqrt{v^T Q v}}$$

where v is the positive right eigenvector of E corresponding to the largest eigenvalue λ_1. Now,

$$M_e(W,R) = \frac{\rho e_1}{\rho e_1 + (1-\rho)e_2} = \frac{\rho \frac{v_1}{\sqrt{v^T Q v}}}{\rho \frac{v_1}{\sqrt{v^T Q v}} + (1-\rho)\frac{v_2}{\sqrt{v^T Q v}}}$$

Since $\sqrt{v^T Q v} > 0$, we have

$$M_e(W,R) = \frac{\rho \frac{v_1}{\sqrt{v^T Q v}}}{\rho \frac{v_1}{\sqrt{v^T Q v}} + (1-\rho)\frac{v_2}{\sqrt{v^T Q v}}} = \frac{\sqrt{v^T Q v}\left(\rho \frac{v_1}{\sqrt{v^T Q v}}\right)}{\sqrt{v^T Q v}\left(\rho \frac{v_1}{\sqrt{v^T Q v}} + (1-\rho)\frac{v_2}{\sqrt{v^T Q v}}\right)} = \frac{\rho v_1}{\rho v_1 + (1-\rho)v_2}.$$

Assume W is proportionally fair w.r.t. R:

$$M_e(W,R) = \rho.$$

Then,

$$\rho = M_e(W,R) = \frac{\rho v_1}{\rho v_1 + (1-\rho)v_2}$$

and analogously to (i), now $v_1 = v_2$. We know that $v = (v_1, v_2)$ is an eigenvector of E, so

$$Ev = \lambda_1 v$$

Hence,

$$\begin{aligned}
\begin{pmatrix} \lambda_1 v_1 \\ \lambda_1 v_2 \end{pmatrix} &= \lambda_1 v = Ev \\
&= \begin{pmatrix} P_{11}\rho & P_{21}(1-\rho) \\ P_{21}\rho & P_{22}(1-\rho) \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} P_{11}\rho v_1 + P_{21}(1-\rho)v_2 \\ P_{21}\rho v_1 + P_{22}(1-\rho)v_2 \end{pmatrix} \\
&= \begin{pmatrix} P_{11}\rho v_1 + P_{21}(1-\rho)v_1 \\ P_{21}\rho v_2 + P_{22}(1-\rho)v_2 \end{pmatrix} = \begin{pmatrix} v_1\left(P_{11}\rho + P_{21}(1-\rho)\right) \\ v_2\left(P_{21}\rho + P_{22}(1-\rho)\right) \end{pmatrix} \\
&= \begin{pmatrix} v_1 d_1 \\ v_2 d_2 \end{pmatrix}
\end{aligned}$$

Hence, $d_1 = \lambda_1, d_2 = \lambda_1$ so $d_1 = d_2$.

Now, the converse. Assume that $d_1 = d_2$. Note that $E\mathbb{1} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ and since $d_1 = d_2$ now $\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_1 \end{pmatrix}$, so

$$E\mathbb{1} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_1 \end{pmatrix} = d_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

hence, $(1,1)^T$ is a (positive) eigenvector of E. From the Perron-Frobenius theorem, we have that $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = c \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ for some $c > 0$, so $v_1 = v_2$.

Now, since $\rho \in (0,1)$

$$\frac{\rho v_1}{\rho v_1 + (1-\rho)v_2} = \frac{\rho v_1}{\rho v_1 + (1-\rho)v_1} = \frac{\rho v_1}{v_1} = \rho$$

so W is proportionally fair w.r.t. R.

It now follows that $\Delta_e(W, R, \rho) = 0 \iff d_1 = d_2$.

(iii): For the Katz centrality K, assuming $0 < \alpha < \frac{1}{\rho(E)}$ with

$$k_\alpha = (I - \alpha E)^{-1} \mathbb{1}.$$

The block centrality values for blocks $\mathcal{B}_1 = R, \mathcal{B}_2 = R^c$ are respectively $k_1 = (k_\alpha)_1, k_2 = (k_\alpha)_2$. We get

$$M_K(W, R) = \frac{\rho k_1}{\rho k_1 + (1-\rho)k_2}$$

W is proportionally fair w.r.t. R if and only if $M_K(W, R) = \rho$. Analogously to (i) and (ii)

$$\rho = \frac{\rho k_1}{\rho k_1 + (1-\rho)k_2}$$

if and only if $k_1 = k_2$.

Hence, if we prove that $k_1 = k_2$ if and only if $d_1 = d_2$ then we prove that W is proportionally fair w.r.t. R if and only if $d_1 = d_2$.

First, assume that $k_1 = k_2$. $k_\alpha = (I - \alpha E)^{-1} \mathbb{1}$ we get $(I - \alpha E) k_\alpha = \mathbb{1}$. Now

$$\begin{aligned} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \mathbb{1} = (I - \alpha E) k_\alpha = (I - \alpha E) \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} \\ = \begin{pmatrix} 1 - \alpha P_{11}\rho & -\alpha P_{21}(1-\rho)) \\ -\alpha P_{21}\rho & 1 - \alpha P_{22}(1-\rho)) \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} k_1 - \alpha \left(P_{11}\rho k_1 + P_{21}(1-\rho)k_2\right) \\ k_2 - \alpha \left(P_{21}\rho k_1 + P_{22}(1-\rho)k_2\right) \end{pmatrix} \end{aligned}$$

Substituting $k_2 = k_1$ we get

$$k_1 - \alpha k_1 \left(P_{11}\rho + P_{21}(1-\rho)\right) = 1, k_1 - \alpha k_1 \left(P_{21}\rho + P_{22}(1-\rho)\right) = 1$$

Since $d_1 = P_{11}\rho + P_{21}(1-\rho), d_2 = P_{21}\rho + P_{22}(1-\rho)$ we get

$$1 = (1 - \alpha d_1)k_1, 1 = (1 - \alpha d_2)k_2.$$

Hence, $1-\alpha d_1 = 1-\alpha d_2$. By $\alpha > 0$, we get $d_1 = d_2$.
Now, the converse. Assume $d_1 = d_2$. Analogously as in (ii), we get

$$E\mathbb{1} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_1 \end{pmatrix} = d_1\mathbb{1}$$

Now we deduce k_α:

$$(I-\alpha E)\mathbb{1} = \mathbb{1} - \alpha E\mathbb{1} = \mathbb{1} - \alpha d_1\mathbb{1} = (1-\alpha d_1)\mathbb{1}$$

Now,

$$(I-\alpha E)\mathbb{1} = (1-\alpha d_1)\mathbb{1}.$$

Therefore,

$$k_\alpha = (I-\alpha E)^{-1}\mathbb{1} = \frac{1}{1-\alpha d_1}\mathbb{1}.$$

Hence $k_\alpha = \frac{1}{1-\alpha d_1}\mathbb{1}$. so $k_1 = k_2$ and hence W is proportionally fair w.r.t. R.
Thus, $\Delta_K(W, R, \rho) = 0 \iff d_1 = d_2$. (iv): Taking $\beta \in (0,1)$, for the PageRank centrality PR with

$$p_\beta = (1-\beta)\left(I - \beta E D^{-1}\right)^{-1}\mathbb{1}$$

the block centrality values for blocks $\mathcal{B}_1 = R, \mathcal{B}_2 = R^c$ are respectively $p_1 = (p_\beta)_1, p_2 = (p_\beta)_2$. We first show that W is proportionally fair w.r.t. R if and only if $p_\beta = \mathbb{1}$. Let W be proportionally fair w.r.t. R, so $M_{PR}(W, R) = \rho$. Since PageRank is normalized, $M_{PR}(W, R) = \rho p_1$ so $M_{PR}(W, R) = \rho$ if and only if $p_1 = 1$. We also have that $\rho p_1 + (1-\rho)p_2 = 1$, so also $p_2 = 1$ and hence $p_1 = p_2 = 1$; $p_\beta = \mathbb{1}$. So $M_{PR}(W, R) = \rho$ if and only if $p_\beta = \mathbb{1}$.
Now, we only need to prove that $p_\beta = \mathbb{1}$ if and only if $d_1 = d_2$.
Assume firstly that $p_\beta = \mathbb{1}$. We have

$$\mathbb{1} = p_\beta = (1-\beta)\left(I - \beta E D^{-1}\right)^{-1}\mathbb{1} = (1-\beta)\mathbb{1} + \beta E D^{-1} p_\beta = (1-\beta)\mathbb{1} + \beta E D^{-1}\mathbb{1}$$

hence $ED^{-1}\mathbb{1} = \mathbb{1}$:

$$\begin{aligned} \mathbb{1} &= ED^{-1}\mathbb{1} \\ &= \begin{pmatrix} \frac{P_{11}\rho}{d_1} + \frac{P_{21}(1-\rho)}{d_2} \\ \frac{P_{21}\rho}{d_1} + \frac{P_{22}(1-\rho)}{d_2} \end{pmatrix} \end{aligned}$$

So $\frac{P_{11}\rho}{d_1} + \frac{P_{21}(1-\rho)}{d_2} = 1$, $\frac{P_{21}\rho}{d_1} + \frac{P_{22}(1-\rho)}{d_2} = 1$. Note that $d_1 = P_{11}\rho + P_{21}(1-\rho)$. Hence, $\frac{P_{11}\rho}{d_1} + \frac{P_{21}(1-\rho)}{d_1} = 1$ and hence since $\rho \in (0,1)$, $d_1 = d_2$. For the converse argument, assume $d_1 = d_2$. Then, $D = \mathrm{diag}(d_1, d_1) = d_1 I$ and $ED^{-1} = E\frac{1}{d_1}I = \frac{1}{d_1}IE = \frac{1}{d_1}E$. The PageRank vector p_β is the unique solution of the equation

$$p_\beta = (1-\beta)\mathbb{1} + \beta E D^{-1} p_\beta.$$

We now show that $\mathbb{1}$ is a solution of the equation. Since

$$E\mathbb{1} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix},$$

we get

$$ED^{-1}\mathbb{1} = \frac{1}{d_1}E\mathbb{1} = \mathbb{1}.$$

Therefore,

$$p_\beta = (1-\beta)\mathbb{1} + \beta E D^{-1}\mathbb{1} = (1-\beta)\mathbb{1} + \beta\mathbb{1} = \mathbb{1}.$$

So, by uniqueness of the PageRank solution, $p_\beta = \mathbb{1}$.
Hence $M_{PR}(W, R) = \rho$ and thus $\Delta_{PR}(W, R, \rho) = 0$.
We can hence conclude that $\Delta_{PR}(W, R, \rho) = 0 \iff d_1 = d_2$. □

So we have proven that for the degree, eigenvector, Katz and PageRank centrality proportional fairness in a 2-block SBM is equivalent to the two blocks having the same graphon degree.

Chapter IV

Discussion and Conclusion

In this chapter, we discuss the results and limitations of the thesis, outline directions for future work, and give the final conclusion.

IV.1 Discussion

In this thesis, we studied fairness of centrality measures from both an algorithmic and a theoretical perspective. The algorithmic part focused on improving PageRank fairness through edge operations, while the theoretical part developed a graphon framework for fairness estimation. Together, these two perspectives address different aspects of the same problem: how fairness of centrality measures can be measured, improved, and understood in networks.

The developed optimization algorithms show that PageRank fairness can be improved by local modifications of the graph. Instead of exhaustively searching through all possible edge operations, the algorithms use heuristic candidate generation. This makes the methods computationally less expensive, but also introduces limitations. Since the algorithms are greedy, they only select the best operation according to the current graph and the current fairness objective. Therefore, they may get stuck in local optima.

Another limitation is that the effectiveness of an edge operation depends strongly on the structure of the graph. Adding, deleting, or rewiring an edge can have different effects depending on the PageRank distribution, the out-degrees of nodes, and the connectivity between groups. The proposed heuristics use local structural information to identify promising candidates, but they do not fully capture the global effect of an edge operation on PageRank.

The choice of fairness definition also matters. Fairness loss and PR-unfairness capture different ideas of fairness. Fairness loss is symmetric: it penalizes deviations from a target distribution in both directions. In contrast, PR-unfairness is one-sided and focuses on whether a protected group receives at least a specified amount of PageRank mass. Hence, an operation that is useful for one objective may not be equally useful for the other.

There are also limitations to the graphon framework. Naturally, graphons are suited to dense graphs, while many real-world networks are sparse.

IV.2 Future Work

There are several directions for future work. First, the proposed optimization algorithms are greedy. Therefore, they are not guaranteed to find a globally optimal sequence of edge operations. Future work could compare these algorithms with non-greedy approaches, such as simulated annealing or genetic algorithms.

In addition, the candidate generation heuristics are based on local structural information and only approximate the effect of an edge operation on PageRank. A possible extension is to design heuristics that better estimate the global effect of an edge operation.

The current experiments also use a fixed setting for the parameters $k \in \mathbb{N}_{\geq 0}, T, b \in \mathbb{N}_{\geq 1}$. A more systematic experimental study could investigate how these parameters affect runtime, convergence, and final fairness values. It would also be useful to test the algorithms on larger networks and on networks with different levels of homophily, density, and group imbalance.

Finally, this thesis focuses on PageRank optimization, while degree, eigenvector, and Katz centrality are mainly studied in the graphon framework. Future work could develop and evaluate structural fairness optimization algorithms for these other centrality measures as well. It could also compare whether the same edge operations are effective across different notions of centrality.

IV.3 Conclusion

Networks appear in many real-world domains. Centrality measures quantify the importance of nodes in networks. Nevertheless, when the underlying network is biased, centrality measures can also reflect this bias. This thesis studied this problem from two perspectives: the development of fairness optimization methods for PageRank, and the development of a theoretical framework for fairness estimation of the degree, eigenvector, Katz and PageRank centrality using graphons.

First, we reviewed existing fairness definitions in literature, with particular attention to PageRank fairness. From this, two notions of fairness were selected for PageRank fairness optimization: fairness loss and PR-unfairness. Fairness loss measures the mean squared error between the PageRank assigned to groups and a target distribution, while PR-unfairness considers whether a protected group receives at least a certain target PageRank mass.

We then developed greedy algorithms that aim to improve PageRank fairness by applying a limited number of edge operations. Edge operations are limited to edge addition, edge deletion and edge rewiring. Since the search space of possible modified graphs grows rapidly with the budget, exhaustive search is infeasible. Therefore, the proposed algorithms make use of heuristic candidate generation for each included edge operation, which are then ranked. For each promising candidate, PageRank is then recomputed and evaluated according to the chosen fairness objective.

The experimental results show that the proposed greedy methods are able to improve PageRank fairness on several real-world networks. In particular, they can reduce the fairness loss or increase the PageRank mass assigned to a protected group under a fixed budget. The results also highlight that the effectiveness of the methods depends on the network itself, the budget and the allowed edge operation types.

Finally, we developed a graphon framework for fairness estimation. In this framework, the addressed centrality measures and fairness notions are defined for graphons. We showed that under suitable conditions finite graph fairness converges to graphon fairness. This gives a theoretical

interpretation of graphon fairness as a limiting fairness value for large graphs sampled from a latent graphon. As an application, we consider two-block stochastic block model graphons and showed that for the degree, eigenvector, Katz and PageRank centrality proportional fairness is equivalent to two blocks sharing the same graphon degree.

In conclusion, in this thesis we show that PageRank fairness can be improved through local edge operations, and that graphons provide a useful mathematical framework for studying fairness estimation in large networks. The optimization part gives practical methods for modifying finite graphs, while the graphon part gives a limiting perspective on fairness.

Bibliography

[1] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 U.S. election: divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, page 36–43, New York, NY, USA, 2005. Association for Computing Machinery.

[2] Marco Avella-Medina, Francesca Parise, Michael T. Schaub, and Santiago Segarra. Centrality measures for graphons: Accounting for uncertainty in networks. *IEEE Transactions on Network Science and Engineering*, 7(1):520–537, 2020.

[3] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7, page 107–117, NLD, 1998. Elsevier Science Publishers B. V.

[4] Sourav Chatterjee. *Large Deviations for Random Graphs*, volume 2197 of *Lecture Notes in Mathematics*. Springer Cham, 1 edition, 2017.

[5] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery.

[6] Daniel Glasscock. What is a graphon?, 2016.

[7] David F. Gleich. Pagerank beyond the web. *CoRR*, abs/1407.5107, 2014.

[8] Danylo Honcharov and Sucheta Soundarajan. Fair dense subgraph discovery: algorithms and analysis. *Social Network Analysis and Mining*, 16, 12 2025.

[9] Gábor Iván and Vince Grolmusz. When the web meets the cell: using personalized pagerank for analyzing protein interaction networks. *Bioinformatics*, 27(3):405–407, 02 2011.

[10] Emmanouil Kariotakis and Aritra Konar. Fairrari: A plug and play framework for fairness-aware pagerank, 2026.

[11] Changan Liu, Haoxin Sun, Ahad N. Zehmakan, and Zhongzhi Zhang. Efficient edge rewiring strategies for enhancing pagerank fairness. *Theoretical Computer Science*, 1067:115765, 2026.

[12] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, Jul 2000.

[13] Mark Newman. *Networks*. Oxford University Press, 07 2018.

[14] Dana Pessach and Erez Shmueli. Algorithmic fairness, 2020.

[15] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. On fairness and calibration. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5684–5693, Red Hook, NY, USA, 2017. Curran Associates Inc.

[16] S.M. Rump. Entrywise lower and upper bounds for the perron vector. *Linear Algebra and its Applications*, 653:314–319, 2022.

[17] Akrati Saxena, George Fletcher, and Mykola Pechenizkiy. Fairsna: Algorithmic fairness in social network analysis. *ACM Comput. Surv.*, 56(8), April 2024.

[18] Sergey Shvydun. Zoo of centralities: Encyclopedia of node metrics in complex networks, 2025.

[19] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Konstantinos Semertzidis, and Panayiotis Tsaparas. Link recommendations for pagerank fairness. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 3541–3551, New York, NY, USA, 2022. Association for Computing Machinery.

[20] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. Fairness-aware pagerank. In *Proceedings of the Web Conference 2021*, WWW '21, page 3815–3826, New York, NY, USA, 2021. Association for Computing Machinery.

[21] Honglian Wang, Haoyun Zhou, and Aristides Gionis. Fairness-aware pagerank via edge reweighting. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, page 661–670. ACM, February 2026.

[22] Zichong Wang, Jocelyn Dzuong, Xiaoyong Yuan, Zhong Chen, Yanzhao Wu, Xin Yao, and Wenbin Zhang. Individual fairness with group awareness under uncertainty. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2024, Vilnius, Lithuania, September 9–13, 2024, Proceedings, Part V*, page 89–106, Berlin, Heidelberg, 2024. Springer-Verlag.

Appendix

This appendix contains additional experimental results for the fairness optimization experiments from Section II.3. Appendix A contains the full fairness loss optimization results for the remaining target vectors ϕ. Appendix B contains additional optimization results for PR-unfairness. Here, additional results from Section II.3 are presented for both optimization Algorithms 1 and 2, for the remaining parameter settings of ϕ. Also, the results (if the algorithm is not terminated yet) at $b = 100, b = 200, b = 300, b = 400$ are reported.

A A. Additional Fairness Loss Optimization Results

In this section, we report the remaining optimization results in a table for 1, for each of the datasets. The quantities $L_{100}, L_{200}, L_{300}, L_{400}$ and L_{500} denote the fairness loss after at most $100, 200, 300, 400$ and 500 edge operations, respectively. With b_{term} we refer to the budget at which Algorithm 1 terminates. With L_{init} and L_{final} we refer to the initial and final fairness loss, respectively.

Additional results for DBLP_gender: database

ϕ	Operation	L_{init}	L_{100}	L_{200}	L_{300}	L_{400}	L_{500}	b_{term}	L_{final}	Decrease
$(0.1, 0.9)$	Addition-only	0.477937	0.406321	0.365444	0.336491	0.314693	0.298058	500	0.298058	37.64%
$(0.1, 0.9)$	Deletion-only	0.477937	0.462537	0.459854	0.459854	0.459854	0.459854	128	0.459854	3.78%
$(0.1, 0.9)$	Rewiring-only	0.477937	0.465581	0.465581	0.465581	0.465581	0.465581	14	0.465581	2.59%
$(0.1, 0.9)$	All	0.477937	0.40445	0.36294	0.332896	0.309206	0.292256	500	0.292256	38.85%
$(0.2, 0.8)$	Addition-only	0.349671	0.288835	0.25454	0.230476	0.212498	0.198868	500	0.198868	43.13%
$(0.2, 0.8)$	Deletion-only	0.349671	0.336517	0.334229	0.334229	0.334229	0.334229	128	0.334229	4.42%
$(0.2, 0.8)$	Rewiring-only	0.349671	0.339114	0.339114	0.339114	0.339114	0.339114	14	0.339114	3.02%
$(0.2, 0.8)$	All	0.349671	0.287257	0.252451	0.227501	0.207993	0.194135	500	0.194135	44.48%
$(0.3, 0.7)$	Addition-only	0.241405	0.191348	0.163636	0.14446	0.130303	0.119679	500	0.119679	50.42%
$(0.3, 0.7)$	Deletion-only	0.241405	0.230497	0.228604	0.228604	0.228604	0.228604	128	0.228604	5.30%
$(0.3, 0.7)$	Rewiring-only	0.241405	0.232647	0.232647	0.232647	0.232647	0.232647	14	0.232647	3.63%
$(0.3, 0.7)$	All	0.241405	0.190064	0.161962	0.142107	0.126781	0.116013	500	0.116013	51.94%
$(0.4, 0.6)$	Addition-only	0.153139	0.113861	0.092732	0.078444	0.068108	0.06049	500	0.06049	60.50%
$(0.4, 0.6)$	Deletion-only	0.153139	0.144477	0.142979	0.142979	0.142979	0.142979	128	0.142979	6.63%
$(0.4, 0.6)$	Rewiring-only	0.153139	0.14618	0.14618	0.14618	0.14618	0.14618	14	0.14618	4.54%
$(0.4, 0.6)$	All	0.153139	0.112872	0.091473	0.076713	0.065568	0.057892	500	0.057892	62.20%
$(0.5, 0.5)$	Addition-only	0.084873	0.056375	0.041828	0.032428	0.025913	0.0213	500	0.0213	74.90%
$(0.5, 0.5)$	Deletion-only	0.084873	0.078457	0.077354	0.077354	0.077354	0.077354	128	0.077354	8.86%
$(0.5, 0.5)$	Rewiring-only	0.084873	0.079713	0.079713	0.079713	0.079713	0.079713	14	0.079713	6.08%
$(0.5, 0.5)$	All	0.084873	0.055679	0.040984	0.031319	0.024356	0.01977	500	0.01977	76.71%
$(0.6, 0.4)$	Addition-only	0.036607	0.018888	0.010924	0.006413	0.003718	0.002111	500	0.002111	94.23%
$(0.6, 0.4)$	Deletion-only	0.036607	0.032436	0.031729	0.031729	0.031729	0.031729	128	0.031729	13.33%
$(0.6, 0.4)$	Rewiring-only	0.036607	0.033246	0.033246	0.033246	0.033246	0.033246	14	0.033246	9.18%
$(0.6, 0.4)$	All	0.036607	0.018486	0.010495	0.005925	0.003143	0.001649	500	0.001649	95.50%
$(0.7, 0.3)$	Addition-only	0.008341	0.001401	0.00002	$3.7643 \cdot 10^{-10}$	$3.7643 \cdot 10^{-10}$	$3.7643 \cdot 10^{-10}$	217	$3.7643 \cdot 10^{-10}$	100.00%
$(0.7, 0.3)$	Deletion-only	0.008341	0.006416	0.006104	0.006104	0.006104	0.006104	128	0.006104	26.83%
$(0.7, 0.3)$	Rewiring-only	0.008341	0.006779	0.006779	0.006779	0.006779	0.006779	14	0.006779	18.73%
$(0.7, 0.3)$	All	0.008341	0.001293	0.000006	$7.378276 \cdot 10^{-10}$	$7.378276 \cdot 10^{-10}$	$7.378276 \cdot 10^{-10}$	209	$7.378276 \cdot 10^{-10}$	100.00%
$(0.8, 0.2)$	Addition-only	0.000075	$1.796004 \cdot 10^{-8}$	$1.796004 \cdot 10^{-8}$	$1.796004 \cdot 10^{-8}$	$1.796004 \cdot 10^{-8}$	$1.796004 \cdot 10^{-8}$	23	$1.796004 \cdot 10^{-8}$	99.98%
$(0.8, 0.2)$	Deletion-only	0.000075	$6.024031 \cdot 10^{-14}$	$6.024031 \cdot 10^{-14}$	$6.024031 \cdot 10^{-14}$	$6.024031 \cdot 10^{-14}$	$6.024031 \cdot 10^{-14}$	70	$6.024031 \cdot 10^{-14}$	100.00%
$(0.8, 0.2)$	Rewiring-only	0.000075	$6.430758 \cdot 10^{-12}$	$6.430758 \cdot 10^{-12}$	$6.430758 \cdot 10^{-12}$	$6.430758 \cdot 10^{-12}$	$6.430758 \cdot 10^{-12}$	22	$6.430758 \cdot 10^{-12}$	100.00%
$(0.8, 0.2)$	All	0.000075	$7.421222 \cdot 10^{-15}$	$7.421222 \cdot 10^{-15}$	$7.421222 \cdot 10^{-15}$	$7.421222 \cdot 10^{-15}$	$7.421222 \cdot 10^{-15}$	20	$7.421222 \cdot 10^{-15}$	100.00%
$(0.9, 0.1)$	Addition-only	0.011809	0.007139	0.00574	0.005091	0.00509	0.00509	301	0.00509	56.89%
$(0.9, 0.1)$	Deletion-only	0.011809	0.009651	0.008917	0.008403	0.008136	0.00807	430	0.00807	31.67%
$(0.9, 0.1)$	Rewiring-only	0.011809	0.008059	0.007482	0.007482	0.007482	0.007482	148	0.007482	36.64%
$(0.9, 0.1)$	All	0.011809	0.00658	0.004909	0.004047	0.003592	0.003394	500	0.003394	71.26%

Table 1: All fairness loss optimization results for all operation settings for the DBLP_gender: database dataset.

Additional results for `DBLP_gender: datamining`

ϕ	Operation	L_{init}	L_{100}	L_{200}	L_{300}	L_{400}	L_{500}	b_{term}	L_{final}	Decrease
$(0.1, 0.9)$	Addition-only	0.506708	0.336424	0.275974	0.262668	0.262668	0.262668	236	0.262668	48.16%
$(0.1, 0.9)$	Deletion-only	0.506708	0.490821	0.490821	0.490821	0.490821	0.490821	94	0.490821	3.14%
$(0.1, 0.9)$	Rewiring-only	0.506708	0.394333	0.394333	0.394333	0.394333	0.394333	77	0.394333	22.18%
$(0.1, 0.9)$	All	0.506708	0.335117	0.271537	0.250611	0.248289	0.248289	325	0.248289	51.00%
$(0.2, 0.8)$	Addition-only	0.374342	0.23042	0.180908	0.170166	0.170166	0.170166	236	0.170166	54.54%
$(0.2, 0.8)$	Deletion-only	0.374342	0.360704	0.360704	0.360704	0.360704	0.360704	94	0.360704	3.64%
$(0.2, 0.8)$	Rewiring-only	0.374342	0.278741	0.278741	0.278741	0.278741	0.278741	77	0.278741	25.54%
$(0.2, 0.8)$	All	0.374342	0.229339	0.177319	0.160489	0.158631	0.158631	325	0.158631	57.62%
$(0.3, 0.7)$	Addition-only	0.261975	0.144415	0.105841	0.097663	0.097663	0.097663	236	0.097663	62.72%
$(0.3, 0.7)$	Deletion-only	0.261975	0.250587	0.250587	0.250587	0.250587	0.250587	94	0.250587	4.35%
$(0.3, 0.7)$	Rewiring-only	0.261975	0.183149	0.183149	0.183149	0.183149	0.183149	77	0.183149	30.09%
$(0.3, 0.7)$	All	0.261975	0.14356	0.1031	0.090367	0.088974	0.088974	325	0.088974	66.04%
$(0.4, 0.6)$	Addition-only	0.169608	0.078411	0.050775	0.045161	0.045161	0.045161	236	0.045161	73.37%
$(0.4, 0.6)$	Deletion-only	0.169608	0.160469	0.160469	0.160469	0.160469	0.160469	94	0.160469	5.39%
$(0.4, 0.6)$	Rewiring-only	0.169608	0.107557	0.107557	0.107557	0.107557	0.107557	77	0.107557	36.58%
$(0.4, 0.6)$	All	0.169608	0.077781	0.048882	0.040245	0.039317	0.039317	325	0.039317	76.82%
$(0.5, 0.5)$	Addition-only	0.097241	0.032407	0.015708	0.012659	0.012659	0.012659	236	0.012659	86.98%
$(0.5, 0.5)$	Deletion-only	0.097241	0.090352	0.090352	0.090352	0.090352	0.090352	94	0.090352	7.08%
$(0.5, 0.5)$	Rewiring-only	0.097241	0.051965	0.051965	0.051965	0.051965	0.051965	77	0.051965	46.56%
$(0.5, 0.5)$	All	0.097241	0.032003	0.014663	0.010123	0.00966	0.00966	325	0.00966	90.07%
$(0.6, 0.4)$	Addition-only	0.044874	0.006403	0.000642	0.000157	0.000157	0.000157	236	0.000157	99.65%
$(0.6, 0.4)$	Deletion-only	0.044874	0.040235	0.040235	0.040235	0.040235	0.040235	94	0.040235	10.34%
$(0.6, 0.4)$	Rewiring-only	0.044874	0.016374	0.016374	0.016374	0.016374	0.016374	77	0.016374	63.51%
$(0.6, 0.4)$	All	0.044874	0.006224	0.000445	$3.729078 \cdot 10^{-7}$	$7.575563 \cdot 10^{-11}$	$7.575563 \cdot 10^{-11}$	310	$7.575563 \cdot 10^{-11}$	100.00%
$(0.7, 0.3)$	Addition-only	0.012507	$8.938823 \cdot 10^{-9}$	$8.938823 \cdot 10^{-9}$	$8.938823 \cdot 10^{-9}$	$8.938823 \cdot 10^{-9}$	$8.938823 \cdot 10^{-9}$	77	$8.938823 \cdot 10^{-9}$	100.00%
$(0.7, 0.3)$	Deletion-only	0.012507	0.010118	0.010118	0.010118	0.010118	0.010118	94	0.010118	19.10%
$(0.7, 0.3)$	Rewiring-only	0.012507	0.000782	0.000782	0.000782	0.000782	0.000782	77	0.000782	93.75%
$(0.7, 0.3)$	All	0.012507	$1.723401 \cdot 10^{-10}$	$1.723401 \cdot 10^{-10}$	$1.723401 \cdot 10^{-10}$	$1.723401 \cdot 10^{-10}$	$1.723401 \cdot 10^{-10}$	78	$1.723401 \cdot 10^{-10}$	100.00%
$(0.8, 0.2)$	Addition-only	0.00014	$3.075026 \cdot 10^{-8}$	$3.075026 \cdot 10^{-8}$	$3.075026 \cdot 10^{-8}$	$3.075026 \cdot 10^{-8}$	$3.075026 \cdot 10^{-8}$	5	$3.075026 \cdot 10^{-8}$	99.98%
$(0.8, 0.2)$	Deletion-only	0.00014	$3.437101 \cdot 10^{-7}$	$3.437101 \cdot 10^{-7}$	$3.437101 \cdot 10^{-7}$	$3.437101 \cdot 10^{-7}$	$3.437101 \cdot 10^{-7}$	94	$3.437101 \cdot 10^{-7}$	99.75%
$(0.8, 0.2)$	Rewiring-only	0.00014	$9.095434 \cdot 10^{-11}$	$9.095434 \cdot 10^{-11}$	$9.095434 \cdot 10^{-11}$	$9.095434 \cdot 10^{-11}$	$9.095434 \cdot 10^{-11}$	6	$9.095434 \cdot 10^{-11}$	100.00%
$(0.8, 0.2)$	All	0.00014	$2.021366 \cdot 10^{-10}$	$2.021366 \cdot 10^{-10}$	$2.021366 \cdot 10^{-10}$	$2.021366 \cdot 10^{-10}$	$2.021366 \cdot 10^{-10}$	6	$2.021366 \cdot 10^{-10}$	100.00%
$(0.9, 0.1)$	Addition-only	0.007773	0.004599	0.004457	0.004457	0.004457	0.004457	116	0.004457	42.67%
$(0.9, 0.1)$	Deletion-only	0.007773	0.005754	0.005268	0.005209	0.005209	0.005209	262	0.005209	32.98%
$(0.9, 0.1)$	Rewiring-only	0.007773	0.00489	0.004341	0.004254	0.004254	0.004254	264	0.004254	45.27%
$(0.9, 0.1)$	All	0.007773	0.003844	0.002964	0.002564	0.002495	0.002495	371	0.002495	67.90%

Table 2: All fairness loss optimization results for all operation settings for the `DBLP_gender: datamining` dataset.

Additional results for `polblogs`

ϕ	Operation	L_{init}	L_{100}	L_{200}	L_{300}	L_{400}	L_{500}	b_{term}	L_{final}	Decrease
$(0.1, 0.9)$	Addition-only	0.152876	0.107760	0.107760	0.107760	0.107760	0.107760	40	0.107760	29.51%
$(0.1, 0.9)$	Deletion-only	0.152876	0.105624	0.105624	0.105624	0.105624	0.105624	100	0.105624	30.91%
$(0.1, 0.9)$	Rewiring-only	0.152876	0.077083	0.060614	0.049991	0.043144	0.037944	500	0.037944	75.18%
$(0.1, 0.9)$	All	0.152876	0.069550	0.054818	0.045245	0.038881	0.034046	500	0.034046	77.73%
$(0.2, 0.8)$	Addition-only	0.084677	0.052106	0.052106	0.052106	0.052106	0.052106	40	0.052106	38.46%
$(0.2, 0.8)$	Deletion-only	0.084677	0.050624	0.050624	0.050624	0.050624	0.050624	100	0.050624	40.21%
$(0.2, 0.8)$	Rewiring-only	0.084677	0.031555	0.021374	0.015274	0.011602	0.008985	500	0.008985	89.39%
$(0.2, 0.8)$	All	0.084677	0.026805	0.017992	0.012703	0.009445	0.007143	500	0.007143	91.56%
$(0.3, 0.7)$	Addition-only	0.036479	0.016453	0.016453	0.016453	0.016453	0.016453	40	0.016453	54.90%
$(0.3, 0.7)$	Deletion-only	0.036479	0.015625	0.015625	0.015625	0.015625	0.015625	100	0.015625	57.17%
$(0.3, 0.7)$	Rewiring-only	0.036479	0.006028	0.002134	0.000556	$5.945821 \cdot 10^{-5}$	$1.783096 \cdot 10^{-10}$	458	$1.783096 \cdot 10^{-10}$	100.00%
$(0.3, 0.7)$	All	0.036479	0.004061	0.001165	0.000161	$1.759734 \cdot 10^{-9}$	$1.759734 \cdot 10^{-9}$	381	$1.759734 \cdot 10^{-9}$	100.00%
$(0.4, 0.6)$	Addition-only	0.008280	0.000799	0.000799	0.000799	0.000799	0.000799	40	0.000799	90.35%
$(0.4, 0.6)$	Deletion-only	0.008280	0.000625	0.000625	0.000625	0.000625	0.000625	100	0.000625	92.45%
$(0.4, 0.6)$	Rewiring-only	0.008280	$2.795605 \cdot 10^{-8}$	$2.795605 \cdot 10^{-8}$	$2.795605 \cdot 10^{-8}$	$2.795605 \cdot 10^{-8}$	$2.795605 \cdot 10^{-8}$	57	$2.795605 \cdot 10^{-8}$	100.00%
$(0.4, 0.6)$	All	0.008280	$5.800487 \cdot 10^{-14}$	$5.800487 \cdot 10^{-14}$	$5.800487 \cdot 10^{-14}$	$5.800487 \cdot 10^{-14}$	$5.800487 \cdot 10^{-14}$	42	$5.800487 \cdot 10^{-14}$	100.00%
$(0.5, 0.5)$	Addition-only	$8.111471 \cdot 10^{-5}$	$9.079111 \cdot 10^{-10}$	$9.079111 \cdot 10^{-10}$	$9.079111 \cdot 10^{-10}$	$9.079111 \cdot 10^{-10}$	$9.079111 \cdot 10^{-10}$	1	$9.079111 \cdot 10^{-10}$	100.00%
$(0.5, 0.5)$	Deletion-only	$8.111471 \cdot 10^{-5}$	$8.315225 \cdot 10^{-11}$	$8.315225 \cdot 10^{-11}$	$8.315225 \cdot 10^{-11}$	$8.315225 \cdot 10^{-11}$	$8.315225 \cdot 10^{-11}$	6	$8.315225 \cdot 10^{-11}$	100.00%
$(0.5, 0.5)$	Rewiring-only	$8.111471 \cdot 10^{-5}$	$1.283287 \cdot 10^{-7}$	$1.283287 \cdot 10^{-7}$	$1.283287 \cdot 10^{-7}$	$1.283287 \cdot 10^{-7}$	$1.283287 \cdot 10^{-7}$	1	$1.283287 \cdot 10^{-7}$	99.84%
$(0.5, 0.5)$	All	$8.111471 \cdot 10^{-5}$	$9.079111 \cdot 10^{-10}$	$9.079111 \cdot 10^{-10}$	$9.079111 \cdot 10^{-10}$	$9.079111 \cdot 10^{-10}$	$9.079111 \cdot 10^{-10}$	1	$9.079111 \cdot 10^{-10}$	100.00%
$(0.6, 0.4)$	Addition-only	0.011882	0.000670	0.000670	0.000670	0.000670	0.000670	27	0.000670	94.37%
$(0.6, 0.4)$	Deletion-only	0.011882	0.000723	0.000620	0.000620	0.000620	0.000620	108	0.000620	94.78%
$(0.6, 0.4)$	Rewiring-only	0.011882	$1.73389 \cdot 10^{-9}$	$1.73389 \cdot 10^{-9}$	$1.73389 \cdot 10^{-9}$	$1.73389 \cdot 10^{-9}$	$1.73389 \cdot 10^{-9}$	65	$1.73389 \cdot 10^{-9}$	100.00%
$(0.6, 0.4)$	All	0.011882	$4.339801 \cdot 10^{-12}$	$4.339801 \cdot 10^{-12}$	$4.339801 \cdot 10^{-12}$	$4.339801 \cdot 10^{-12}$	$4.339801 \cdot 10^{-12}$	32	$4.339801 \cdot 10^{-12}$	100.00%
$(0.7, 0.3)$	Addition-only	0.043684	0.015845	0.015845	0.015845	0.015845	0.015845	27	0.015845	63.73%
$(0.7, 0.3)$	Deletion-only	0.043684	0.016099	0.015598	0.015598	0.015598	0.015598	108	0.015598	64.29%
$(0.7, 0.3)$	Rewiring-only	0.043684	0.006516	0.002394	0.000819	0.000203	$1.087029 \cdot 10^{-5}$	500	$1.087029 \cdot 10^{-5}$	99.98%
$(0.7, 0.3)$	All	0.043684	0.003280	0.002189	0.002189	0.002189	0.002189	128	0.002189	94.99%
$(0.8, 0.2)$	Addition-only	0.095485	0.051020	0.051020	0.051020	0.051020	0.051020	27	0.051020	46.57%
$(0.8, 0.2)$	Deletion-only	0.095485	0.051475	0.050577	0.050577	0.050577	0.050577	108	0.050577	47.03%
$(0.8, 0.2)$	Rewiring-only	0.095485	0.032660	0.022179	0.016543	0.013052	0.010670	500	0.010670	88.83%
$(0.8, 0.2)$	All	0.095485	0.024735	0.021547	0.021547	0.021547	0.021547	128	0.021547	77.43%
$(0.9, 0.1)$	Addition-only	0.167286	0.106195	0.106195	0.106195	0.106195	0.106195	27	0.106195	36.52%
$(0.9, 0.1)$	Deletion-only	0.167286	0.106851	0.105556	0.105556	0.105556	0.105556	108	0.105556	36.90%
$(0.9, 0.1)$	Rewiring-only	0.167286	0.078805	0.061964	0.052266	0.045901	0.041330	500	0.041330	75.29%
$(0.9, 0.1)$	All	0.167286	0.066190	0.060905	0.060905	0.060905	0.060905	128	0.060905	63.59%

Table 3: All fairness loss optimization results for all operation settings for the `polblogs` dataset.

Additional results for `Cora`

ϕ	Operation	L_{init}	L_{100}	L_{200}	L_{300}	L_{400}	L_{500}	b_{term}	L_{final}	Decrease
$(1/7, \ldots, 1/7)$	Addition-only	0.00485	0.00296	0.00208	0.001481	0.001112	0.000778	500	0.000778	83.95%
$(1/7, \ldots, 1/7)$	Deletion-only	0.00485	0.004611	0.004611	0.004611	0.004611	0.004611	53	0.004611	4.93%
$(1/7, \ldots, 1/7)$	Rewiring-only	0.00485	0.002449	0.001284	0.00063	0.000285	$9.900624 \cdot 10^{-5}$	500	$9.900624 \cdot 10^{-5}$	97.96%
$(1/7, \ldots, 1/7)$	All	0.00485	0.002671	0.001438	0.000739	0.000355	0.000148	500	0.000148	96.94%

Table 4: All fairness loss optimization results for all operation settings for the `Cora` dataset.

B B. Additional PR-Unfairness Optimization Results

In this section, we report the remaining optimization results in a table for 2, for each of the datasets. With $c_{PR}^{(b)}(R)$ we refer to the PageRank mass for the protected group at $b = 0, b = 100, b = 200, b = 300, b = 400, b = 500$. With $b_{+\delta}$ we refer to the budget needed until the threshold $\phi = c_{PR;G}(R) + \delta$ is reached, with $G = (V, E)$ being the initial graph, for $\delta = 0.1, \delta = 0.2$. With b_{term} we refer to the budget b after which the algorithm terminates. With $c_{PR}^{\text{final}}(R)$ we refer to the final PageRank mass for the protected group R. Finally, $\Delta c_{PR}(R) = \Delta c_{PR}^{\text{final}}(R) - c_{PR}^{0}(R)$.

Additional results for `DBLP_gender: database`

Operation	$c_{PR}^{(0)}(R)$	$c_{PR}^{(100)}(R)$	$c_{PR}^{(200)}(R)$	$c_{PR}^{(300)}(R)$	$c_{PR}^{(400)}(R)$	$c_{PR}^{(500)}(R)$	$b_{+0.1}$	$b_{+0.2}$	b_{term}	$c_{PR}^{\text{final}}(R)$	$\Delta c_{PR}(R)$
Addition-only	0.208670	0.262566	0.295480	0.319921	0.339025	0.354053	252	–	500	0.354053	0.145383
Deletion-only	0.208670	0.219899	0.221875	0.221875	0.221875	0.221875	–	–	129	0.221875	0.013205
Rewiring-only	0.208670	0.217665	0.217665	0.217665	0.217665	0.217665	–	–	15	0.217665	0.008995
All	0.208670	0.264036	0.297555	0.323029	0.343937	0.359393	242	–	500	0.359393	0.150723

Table 5: All PR-unfairness optimization results for all operation settings for the `DBLP_gender: database` dataset. A – indicates that the threshold is not reached.

Additional results for `DBLP_gender: datamining`

Operation	$c_{PR}^{(0)}(R)$	$c_{PR}^{(100)}(R)$	$c_{PR}^{(200)}(R)$	$c_{PR}^{(300)}(R)$	$c_{PR}^{(400)}(R)$	$c_{PR}^{(500)}(R)$	$b_{+0.1}$	$b_{+0.2}$	b_{term}	$c_{PR}^{\text{final}}(R)$	$\Delta c_{PR}(R)$
Addition-only	0.188165	0.319980	0.374667	0.387489	0.387489	0.387489	65	–	237	0.387489	0.199323
Deletion-only	0.188165	0.199414	0.199414	0.199414	0.199414	0.199414	–	–	95	0.199414	0.011248
Rewiring-only	0.188165	0.272041	0.272041	0.272041	0.272041	0.272041	–	–	78	0.272041	0.083875
All	0.188165	0.321107	0.378908	0.388443	0.388443	0.388443	65	223	223	0.388443	0.200277

Table 6: All PR-unfairness optimization results for all operation settings for the `DBLP_gender: datamining` dataset. A – indicates that the threshold is not reached.

Additional results for `polblogs`

Operation	$c_{PR}^{(0)}(R)$	$c_{PR}^{(100)}(R)$	$c_{PR}^{(200)}(R)$	$c_{PR}^{(300)}(R)$	$c_{PR}^{(400)}(R)$	$c_{PR}^{(500)}(R)$	$b_{+0.1}$	$b_{+0.2}$	b_{term}	$c_{PR}^{\text{final}}(R)$	$\Delta c_{PR}(R)$
Addition-only	0.490994	0.574125	0.574125	0.574125	0.574125	0.574125	–	–	28	0.574125	0.083131
Deletion-only	0.490994	0.573120	0.573321	0.573321	0.573321	0.573321	–	–	102	0.573321	0.082327
Rewiring-only	0.490994	0.619278	0.651074	0.671382	0.685754	0.691036	51	450	450	0.691036	0.200043
All	0.490994	0.642726	0.653210	0.653210	0.653210	0.653210	23	–	129	0.653210	0.162216

Table 7: All PR-unfairness optimization results for all operation settings for the `polblogs` dataset. A – indicates that the threshold is not reached.

Additional results for `Cora`

Operation	$c_{PR}^{(0)}(R)$	$c_{PR}^{(100)}(R)$	$c_{PR}^{(200)}(R)$	$c_{PR}^{(300)}(R)$	$c_{PR}^{(400)}(R)$	$c_{PR}^{(500)}(R)$	$b_{+0.1}$	$b_{+0.2}$	b_{term}	$c_{PR}^{\text{final}}(R)$	$\Delta c_{PR}(R)$
Addition-only	0.137533	0.177647	0.208770	0.236163	0.261639	0.285480	305	–	500	0.285480	0.147947
Deletion-only	0.137533	0.138247	0.138247	0.138247	0.138247	0.138247	–	–	9	0.138247	0.000714
Rewiring-only	0.137533	0.178787	0.178787	0.178787	0.178787	0.178787	–	–	74	0.178787	0.041253
All	0.137533	0.189711	0.234687	0.270737	0.301172	0.331478	207	–	500	0.331478	0.193944

Table 8: All PR-unfairness optimization results for all operation settings for the `Cora` dataset. A – indicates that the threshold is not reached.