Budgeted PageRank Fairness

Nikas Themistoklis

Diploma Thesis

Supervisor: Tsaparas Panayiotis

Ioannina, February 2022

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

Περίληψη

Σήμερα, οι αλγόριθμοι χρησιμοποιούνται όλο και περισσότερο για τη λήψη σημαντικών αποφάσεων που έχουν εμφανή αντίκτυπο στη καθημερινή ζωή των ανθρώπων. Ως αποτέλεσμα, το ζήτημα της αλγοριθμικής δικαιοσύνης έχει εγείρει πολλές ανησυχίες, ενώ παράλληλα έχει λάβει μεγάλο ενδιαφέρον. Σε αυτή τη διπλωματική εργασία, εξετάζουμε τη δικαιοσύνη για link analysis αλγορίθμους και συγκεκριμένα για τον περίφημο αλγόριθμο PageRank. Ο αλγόριθμος PageRank εισήχθη από τη μηχανή αναζήτησης Google για την κατάταξη ιστοσελίδων, αλλά χρησιμοποιείται σε μια πληθώρα εφαρμογών και μεταξύ άλλων, στην κατάταξη και κατηγοριοποίηση δεδομένων. Με γνώμονα ότι οι κόμβοι ενός δικτύου ανήκουν σε ομάδες (π.χ., οι ομάδες μπορούν να αποτελούν τα δύο φύλλα, αλλά στη γενικότερη περίπτωση, μη δυαδικά χαρακτηριστικά), ένας δίκαιος PageRank αλγόριθμος επιδιώκει την δίκαιη κατανομή των βαρών του στην ομάδα που αποτελεί τη μειονότητα. Για αυτόν το λόγο, έχουν προταθεί τροποποιημένοι PageRank αλγόριθμοι που επιβάλουν δίκαιη συμπεριφορά σε όλους τους κόμβους που απαρτίζουν το γράφημα. Η τρέχουσα διπλωματική εργασία επιδιώκει την επίτευξη δικαιοσύνης με το λιγότερο πλήθος αλλαγών. Ως εκ τούτου, θέτουμε σταδιακά «δίκαιους» κόμβους μέχρι να φτάσουμε στη δικαιοσύνη. Παρέχουμε αναλυτική φόρμουλα που εκτιμά την επίδραση της επιβολής τέτοιων κόμβων στη συνολική δικαιοσύνη του γράφου και εξετάζουμε την απόκλιση που υπάρχει συγκριτικά με τον πρωτότυπο PageRank αλγόριθμο. Τέλος, αξιολογούμε πειραματικά τόσο τη προτεινόμενη φόρμουλα, όσο και τη δικαιοσύνη σε πραγματικά δίκτυα.

Λέξεις Κλειδιά: PageRank, αλγοριθμική δικαιοσύνη, ανάλυση δικτύου.

Abstract

Algorithms are increasingly used to make important decisions that have significant impact on people's lives. As a result, the issue of algorithmic fairness has raised numerous concerns and received much interest recently. In this diploma thesis, we consider fairness for link analysis algorithms and, in particular, for the celebrated PageRank algorithm. The PageRank algorithm was introduced by the Google search engine, but it is also used for a variety of applications, such as, recommendations. Given a network where the nodes in belong to groups (for example, based on demographic or other characteristics), PageRank fairness asks for a fair allocation of the PageRank weights to the minority (protected) group. For this purpose, modified PageRank algorithms have been proposed that impose a fair behavior to the entire set of nodes from which the network consists of. In this diploma thesis, we purse fairness with the least possible changes and hence, with a lower cost. Therefore, we gradually render "fair" nodes, until we achieve fairness. We provide analytical formulas for computing the effect of such nodes on fairness and consider the utility loss with respect to the original PageRank algorithm. Finally, we present experiments with real networks that examine fairness, as well as experimentally evaluate our formulas.

Keywords: PageRank, algorithmic fairness, link analysis.

Contents

Chapter 1. Introduction

In this day and age, algorithmic systems driven by large amounts of data have penetrated every aspect of our lives. These systems are regularly being used to assist, and often replace human decision making in life affecting scenarios, such as deciding if a person should be let out on bail pending trial or determining whether a resume makes it through the first selection round. Although there are clear benefits to automated decision making (e.g., machines do not get bored), just like people, algorithms are vulnerable to biases that affect their decisions. Therefore, concerns have been raised regarding algorithmic fairness, where the goal is to ensure that algorithms do not treat people with discrimination on the grounds of protected attributes e.g., age, disability, ethnic or racial origin.

In this diploma thesis, we focus on fairness for link analysis algorithms and, more precisely, for the PageRank algorithm, introduced by Google search engine. Given a graph as input, the PageRank algorithm outputs a numerical weight for each node that reflects its relative importance. Apart from producing rankings on nodes, these weights may also be used as input features in a variety of applications and algorithms [19].

There are different approximations for PageRank unfairness. In this work we follow the one introduced in [1] that views fairness as lack of discrimination against a protected group defined by the value of a sensitive attribute and handles unfairness by considering in-processing modifications, that is, modifications to the PageRank algorithm. The proposed families of fair PageRank algorithms were applied to the entire set of nodes from which the network consists of. For this reason, they may be considered a quite intervening policy of allocating weights. In this work, we pursue fairness with the least possible modifications. Therefore, we will gradually attribute the proposed [1] fair characteristics to individual nodes.

We provide analytical formulas for the effect of rendering fair nodes on fairness, as well as propose different selection policies, so as to specify the order in which nodes are selected to behave in a fair manner.

Finally, we present experiments with real networks that examine fairness. We experimentally evaluate our proposed selection and distribution policies and formulas, as well as consider the cost of achieving fairness.

Chapter 2. Related Work

Algorithmic fairness in the area of graph algorithms has received increased attention (see [7, 13] for recent surveys and [14, 4] for recent tutorials), including group-based fairness for centrality measures [18, 1], embeddings [5, 8], influence maximization [3, 15] and clustering [16]. Approaches on handling fairness can be classified as pre-processing, that modify the input data, in-processing, that modify the algorithm and post-processing, that modify the output.

To address PageRank fairness, the work in [1] focused on in-processing techniques and proposed modifications in the inner-working of the PageRank algorithm so that fairness is achieved. They considered the problem of defining families of PageRank algorithms that are fair, as well as defined the *utility loss* of a fair algorithm as the difference between its output and the output of the original PageRank algorithm. Finally, they posed the problem of achieving fairness while minimizing utility loss compared to the original PageRank algorithm. For this purpose, they presented the fairness-sensitive PageRank family of algorithms that modifies the jump vector so as to achieve fairness, and the locally fair PageRank family of algorithms which guarantees that individually each node behaves in a fair manner.

The work in [2] had a similar, group-based approach, and focused on PageRank centrality. More precisely, they did not modify the PageRank or personalized PageRank algorithm. Instead, they modified the network through link recommendations so that the output of these algorithms on the modified network is fairer. Furthermore, they provided analytical formulas for the effect of edge additions and deletions in PageRank fairness, as well as presented efficient link recommendation algorithms.

Our work combines the two aforementioned approaches. Similar to the work in [1] we modify the PageRank algorithm only for a subset (budget) of nodes which we make fair, in order to achieve fairness. The effect of making a node fair can be estimated using similar techniques to those in [2].

Other related research has studied network fairness in the matter of degree centrality. It was shown that the combination of homophily, preferential attachment and imbalances in group sizes may lead to uneven degree distributions between groups, i.e., the underrepresentation of the minority group in top degree positions [9]. There is also evidence of occurring degree inequalities in real social networks [6]. In addition, recent research has found divergences in the PageRank distributions among the groups that consist the underlying network [1, 10]. There are also individual fairness approaches where the goal is to produce a similar output for similar nodes [17].

Chapter 3. Definitions - Preliminaries

In this section we introduce the necessary background for our work.

3.1 The PageRank Algorithm

The PageRank algorithm (PR) algorithm [30] is an algorithm introduced by the Google search engine to measure the authority of a webpage, by assigning a numerical value to each page as a score. Nevertheless, it has also been employed in a broad-spectrum of applications for different purposes, such as, recommendations [10].

The PR algorithm weights correspond to the stationary probability distribution of a random walk on the graph of the Web. Intuitively, the PageRank algorithm can be thought as modeling the behavior of a random surfer (user) on the Web. Our random surfer starts from a web page chosen at random and keeps clicking on links, never hitting “back” but eventually gets bored and restarts from another page chosen at random. The probability that a random surfer visits a page is its PageRank. More generally, we can consider random surfer models on a graph with an arbitrary set of nodes instead of pages, and transition probabilities instead of randomly clicked links.

The algorithm takes as input the (directed) adjacency matrix that corresponds to a (directed) graph $G = (V, E)$ and calculates the stationary distribution of the random walk over it. In other words, the algorithm produces a scoring vector **p***,* the PageRank vector, that assigns a weight to each node $v \in V$ in the graph.

The PageRank random walk is a random walk with restarts. This follows from the existence of the probability that at each page the “random surfer” will get bored and request another random page, that is the probability that the random walk will restart at any step. The aforementioned probability is defined as the restart probability and denoted by γ. The destination of the random jump is chosen accordingly to the probability distribution given in the jump vector **v**. In most cases, the restart probability is set to $\gamma = 0.15$, and although there are plenty different variations for the jump vector for different purposes, usually the jump vector is a uniform distribution over all nodes (pages).

In general, we replace the notion of "clicking on links according to the structure of the web" with "transitioning according to a stochastic matrix **P**", that is defined as the normalized adjacency matrix of the graph G. The aforementioned matrix defines the transition probability $P[i, j]$ between any two nodes i and j. Special treatment is required for the dangling nodes in the graph, that is, nodes with no outgoing links, that correspond to zero-rows at matrix **P**. In this diploma thesis, we adopt the convention that when at a sink node, the random walk performs a jump to a node chosen uniformly at random. Hence, zero-rows in the matrix **P** are replaced by the uniform probability vector **u**.

The PageRank vector **p** is the stationary distribution of the random walk over the input graph, and it satisfies the equation:

$$p^T = (1 - \gamma)p^T P + \gamma v^T \tag{1}$$

The above equation is recursive, and it may be computed by starting with any initial probability vector and iterating the computation until it converges. Alternatively, it can be computed by simply solving the equation.

One important variation of the PageRank algorithm is the Personalized PageRank algorithm (PPR) where the jump vector is the unit vector e_i that puts all the probability mass on a single node i. We use p_i to denote the PPR vector for node i, and say that node i allocates PageRank $p_i(u)$ to node u. Personalized PageRank provides the importance of nodes in a graph from different points of view. It has found several applications in network analysis, even beyond the web.

The following lemma for PageRank and personalized PageRank appears in [1] and will prove useful for our analysis.

Lemma 3.1. *For the PageRank vector p, it holds that $p^T = v^T Q$ where:*

(1) $Q = \gamma(I - (1 - \gamma)P)^{-1}$.

(2) *The row vector Q_i^T corresponds to the personalized pagerank vector of node i, that is:* $p_i = Q_i$.

Proof (from [1]). We obtain (1) directly from Equation 1. For (2), if we set $v = e_i^T$, then $p = Q_i$, the i-th row of matrix Q.

Given the Lemma 3.1, we will use interchangeably p_i and Q_i to denote the personalized Pagerank vector for node i. The entry Q_{ij} of the matrix Q is the personalized PageRank $p_i(j)$ that node i allocates to node j.

3.2 PageRank Fairness

In this diploma thesis, we will use the group-based notion of fairness introduced in [2]. Specifically, we focus on graphs where nodes belong to groups based on the value of some protected attribute. For example, in the case of social or cooperation networks, where each node is an individual person, the protected attributes may correspond to gender, race or religion. In the following, for simplicity we assume binary such attributes, but both the algorithms to be evaluated and the proposed formulas can be extended for general use.

We thus assume that the given network consists of two types of nodes, red and blue, and the corresponding groups are denoted R and B respectively. Without loss of generality, we assume that group R is the minority (protected) group. We use $\rho = \frac{|R|}{|V|}$ and $\beta = \frac{|B|}{|V|}$ to denote the ratio of group R and B in the overall population respectively.

Abusing the notation, we will refer to the PageRank mass allocated to red group as $p(R)$, that is $p(R) = \sum_{i \in R} p(i)$. The respective weight for blue group equals $p(B) = 1 - p(R)$. Similarly, we will indicate the personalized PageRank mass distributed to the protected group by node v, as $p_v(R)$, that is $p_v(R) = \sum_{i \in R} p_v(i)$.

Given the target group S, and a parameter φ, we say that the network is *PR-unfair* to group S, if $p(S) < \varphi$. Therefore, we measure PageRank fairness by the ratio $p(S)$. With that being said, PageRank fairness asks that $p(S)$ is at least equal to φ. In addition to this, we define a node as *fair* when it distributes its weight according to the specified ratio φ.

Parameter φ is input to our definition, and it may be defined appropriately to achieve different fairness policies. One typical example of this is setting $\varphi = \rho$, in which case we ask that the ratio of the PageRank weights is assigned proportionally to the sizes of the two groups and as a consequence, fairness is analogous to demographic parity [23]. Furthermore, φ may be set in alliance with the 80-percent rule supported by the US Equal Employment Opportunity Commission, or some other formulation of disparate impact [31].

3.3 Locally Fair PageRank

The algorithms we consider in contemplation of achieving fairness derive from a family of fair PageRank algorithms, termed *Locally Fair PageRank (LPFR)*, introduced in [1]. *Locally Fair PageRank* algorithms take a microscopic view of fairness, by asking that *"each individual node acts fairly"*. In random walk terms, local fairness defines a dynamic process that can be viewed as "*a random walk that is fair at each step, and not just at convergence*". More precisely, the LFPR contains all PageRank algorithms, where all rows of the transition matrix **P** are $\varphi -$ fair vectors, i.e., for every node $i \in V$, $\sum_{j \in R} P[i, j] = \varphi$. Also, the jump vector **v** is $\varphi -$ fair: $\sum_{j \in R} v[j] = \varphi$. Therefore, we will modify both the transition matrix and the jump vector. The aforementioned family of algorithms subdivide into specific algorithms, which will now be concisely introduced.

3.3.1 The Neighborhood *LFPR* Algorithm

This specific algorithm considers a node that treats its neighbors fairly by allocating a fraction φ of its PageRank to its red neighbors and the remaining $1 - \varphi$ fraction to its blue neighbors. In the special case where the considered nodes have no red or blue neighbors, we perform a random jump uniformly in the red or blue group with probability φ, or $1 - \varphi$ respectively. In random walk terms, at each node the probability of transitioning to a red neighbor is φ and the respective probability to a blue neighbor is $1 - \varphi$. The algorithm also defines a φ-fair jump vector v_N with $v_N = \frac{\varphi}{|R|}$, if $i \in R$, and $v_N = \frac{1-\varphi}{|B|}$, if $i \in B$.

3.3.2 The Residual-based *LFPR* Algorithms

This family of algorithms considers an alternative fair behavior for individual nodes. Similarly to the Neighborhood LFPR algorithm, each node i acts fairly by respecting the φ ratio when distributing its own pagerank to red and blue nodes. However, in this case, node i treats its neighbors the same, independently of their color (group), and assigns to each one of them the same portion of its PageRank. Moreover, when a node is located in a "biased" neighborhood, i.e., the ratio of its red neighbors is different than φ, to be fair, node i distributes only a fraction of its PageRank to its neighbors, and the remaining portion of its weight to nodes in the locally underrepresented group. The remaining portion is called *residual* and it follows the notation $\delta(i)$. Intuitively, this corresponds to a fair random walker that upon arriving at a node i, with probability $1 - \delta(i)$ follows on of i 's outlinks and with probability $\delta(i)$ jumps to one or more nodes belonging to the group that is locally underrepresented. The way in which $\delta(i)$ is distributed to the underrepresented group is determined by a *residual policy.* For this purpose, two such intuitive policies were introduced in [1], based on which the following two locally fair PageRank algorithms stem from.

The *Uniform Locally Fair PageRank* algorithm distributes the residual uniformly to the corresponding group, that is $\frac{1}{|R|}$ for $i \in R$ and $\frac{1}{|B|}$ for $i \in B$.

The *Proportional Locally Fair PageRank* algorithm distributes the residual proportionally to the original PageRank weights, that is $\frac{p_O[i]}{\sum_{i \in R} p_O[i]}$ for $i \in R$ and $\frac{p_O[i]}{\sum_{i \in B} p_O[i]}$ for $i \in B$.

Chapter 4. Problem Definition

The work in [1] has extensively explained that fairness can be achieved by modifying the parameters of the PageRank algorithm. As already described [1], PageRank algorithm is fully defined by three parameters: the transition matrix **P**, the restart probability γ and the jump vector v. Implementing the proposed fair PageRank algorithms [1] requires the jump probability γ to be fixed, and only consider modifications to the transition matrix **P** and the jump vector **v**. These algorithms were applied to the entire set of nodes from which the network consists of, and therefore may be considered a quite intervening policy of allocating weights. Focused on reducing the cost of achieving fairness, we will gradually attribute the aforesaid characteristics to individual nodes. Simultaneously, inspired by the authors of [2], who modified the network through optimum link recommendations so that the results of PageRank are fairer, we combined the two approaches, addressing the following research questions.

What is the effect of imposing fair behavior to individual nodes on fairness? We will derive analytical formulas that estimate the change in the *PR* ratio for the protected group when rendering a single node as fair.

Given a target φ, which nodes should we render fair, so as to achieve fairness with the minimum budget of nodes? We will propose different approaches for selecting nodes to render fair in order to achieve fairness with the minimum possible changes.

Finally, we will conduct experimental evaluation to study how fairness may be achieved by only imposing fair behavior in some of the nodes (budget) of the network, in practice.

4.1 Fairness comes at a price

To achieve fairness through a modified, fair PageRank algorithm, the output weight vector will obviously differ from that of the original PageRank algorithm. To evaluate the degree of differentiation, we adopt the loss function introduced in [1], and we assume that the weights of the original PageRank algorithm carry some *utility*. We will use these weights to measure the *utility loss* in achieving fairness. More precisely, if **f** is the output of a fair PageRank (FPR) algorithm and p_O is the output of the original PageRank algorithm, we define the utility loss as: $L(FPR) = L(f, p_O) = \|f - p_O\|^2$.We consider the utility loss as a measure of the price we have to pay so as to achieve the desired φ-fairness.

To evaluate the magnitude of the utility loss, the authors of [1] presented an efficient algorithm that computes a lower bound for the utility loss, by constructing the probability vector **w** that is φ-fair, which has the minimum utility loss compared to the original PageRank vector p_O. Nevertheless, it is of importance to identify that vector **w** is not necessarily attainable by any PageRank algorithm.

We will now describe the algorithm proposed in [1] for computing the lower bound for the utility loss. To construct vector **w**, we start with p_O and we redistribute the probability between the two groups to make it fair. Let $p_O(R)$ be the probability assigned to the red group. Without loss of generality, we assume that $p_O(R) < \varphi$, and let $\Delta = \varphi - p_O(R)$. To make the vector fair, we need to remove Δ probability mass from the nodes in B, and redistribute it to the nodes in R. It is easy to show that to minimize the loss, the optimal redistribution would remove uniformly $\Delta/|B|$ probability from all nodes in B, and add uniformly $\Delta/|R|$ to all nodes in R. This follows from the fact that among all distribution vectors, the one with the smallest length is the uniform one. However, this procedure does not guarantee that the resulting vector will not have negative entries, since some blue nodes may have p_O probability less than $\Delta/|B|$. Let β be the smallest non-zero such probability of any blue node. The proposed algorithm transfers β probability from all the non-zero blue nodes to the red nodes, and then recursively applies the same procedure for the residual amount of probability that has not been transferred. This process will produce a fair vector with the minimum utility loss with respect to p_O.

4.2 Counter Fairness, a drastic change

Working towards minimizing the subset of nodes rendered as fair, in contemplation of achieving fairness, in this section, we introduce a quite extreme and drastic change to the behavior of the selected nodes.

This extreme policy of weights allocation forces the nodes that do not respect the φ ratio, that is, the nodes of which the ratio of red neighbors is smaller than the desired φ ratio, to distribute their *entire* PageRank weights *only* to the protected group. In addition to this, the respective jump vectors will be modified accordingly. This subset of nodes includes all nodes x such that $dR_x \ / \ d_x < \varphi$, and this group will be called L_R. By d_x, we refer to the degree, while with dR_x to the number of red neighbors of node x respectively.

Therefore, we are willing to sacrifice the local fairness achieved by imposing a fair behavior to individual nodes, for the faster enhancement of global, network fairness. In some sense, we will impose an un-fair behavior on the nodes that do not respect the φ ratio and therefore the protected group, so that we counter the already existing un-fairness. Although controversial, this extreme strategy of allocating PageRank weights is promising, since we can have a strong impact on the network by completely altering the behavior of a subset of nodes. It would ideal if at the same time we reduce utility loss and we hope to achieve that.

Chapter 5. Fairness Gain Estimation

In this section, we focus on the role of nodes in fairness. We provide a closed-form formula that estimates the effect of imposing a fair behavior on each node with respect to fairness in line with the Neighborhood Locally Fair PageRank algorithm.

5.1 Fairness gain by imposing fair behavior

We will now compute the gain in fairness by rendering a single node (x) as fair. Let *G = (V, E)* denote the underlying (directed) graph of the network, and let (x) be a node in *G*. Let *P* and *P'* denote the corresponding transitions matrices before and after imposing a fair behavior on node (x) and let p' denote the PR vector of matrix *P'.* Note that we did not modify the structure of the graph, but instead imposed a fair behavior on node (x). We define the *fairness gain* for group S (either R or B) by imposing a fair behavior to node (x), as *fgain(x,S) = p'(S)-p(S)*, that is the change in PR ratio of group S. Note that the value of *fgain* may be negative for some nodes. At this point we ought to emphasize the fact that the proposed formula can be used to calculate the gain for both groups (either R or B). Nevertheless, in this thesis, our interest focuses on the protected group, that is the Red group.

The following theorem that estimates analytically the gain and the derivations follows the work in [2] closely. For a node x we use d_x to denote the out-degree, dR_x and dB_x to denote the number of Red and Blue neighbors, N_x to denote the out-neighbors, and R_x, B_x to denote the Red and Blue out-neighbors of the node respectively.

5.1.1 Theorem 5.1:

Let *G = (V, E)* be a (directed) graph, S the target group, and (x) ∈ *V* a vertex-node in *G.* Let

$\Lambda(x, S)$

$$= \begin{cases} \dfrac{\frac{1-\gamma}{\gamma}\left(\frac{1}{dR_x}\sum_{k\in R_x} p_k(S) \; - \frac{1}{dB_x}\sum_{k\in B_x} p_k(S)\right)}{\frac{1}{(\varphi-\rho_x)} - \frac{1-\gamma}{\gamma}\left(\frac{1}{dR_x}\sum_{k\in R_x} p_k(x) \; - \frac{1}{dB_x}\sum_{k\in B_x} p_k(x)\right)}, & dR_x, dB_x \neq 0 \\ \dfrac{\frac{1-\gamma}{\gamma}\left(\frac{\varphi}{|R|}\sum_{k\in R} p_k(S) \; + \left(\frac{1-\varphi}{dB_x} - \frac{1}{d_x}\right)\sum_{k\in B_x} p_k(S)\right)}{1 - \frac{1-\gamma}{\gamma}\left(\frac{\varphi}{|R|}\sum_{k\in R} p_k(x) \; + \left(\frac{1-\varphi}{dB_x} - \frac{1}{d_x}\right)\sum_{k\in B_x} p_k(x)\right)}, & dR_x = 0, dB_x \neq 0 \\ \dfrac{\frac{1-\gamma}{\gamma}\left(\left(\frac{\varphi}{|R|} - \frac{1}{|V|}\right)\sum_{k\in R} p_k(S) \; + \left(\frac{1-\varphi}{|B|} - \frac{1}{|V|}\right)\sum_{k\in B} p_k(S)\right)}{1 - \frac{1-\gamma}{\gamma}\left(\left(\frac{\varphi}{|R|} - \frac{1}{|V|}\right)\sum_{k\in R} p_k(x) \; + \left(\frac{1-\varphi}{|B|} - \frac{1}{|V|}\right)\sum_{k\in B} p_k(x)\right)}, & dR_x, dB_x = 0 \end{cases}$$

The fairness gain for group S of imposing a fair behavior to node (x), to graph G is:

$$fgain = \Lambda(x, S)p(x)$$

Proof. Let $\boldsymbol{P}$ and $\boldsymbol{P'}$ denote the transition matrices of the PageRank random walk on the graphs G and G' before and after rendering node (x) as fair, respectively. To prove our theorem, we first write the transitions matrix $\boldsymbol{P'}$ as the sum of the transition matrix $\boldsymbol{P}$ and a rank-1, perturbation matrix *D*. For the following we assume that $d_x \neq 0$.

$$P' = P + D, \qquad D_{ij} = \begin{cases} \dfrac{\varphi}{dR_x} - \dfrac{1}{d_x} = \dfrac{\varphi - \rho_x}{dR_x}, & j \in R, i = x \\ \dfrac{1-\varphi}{dB_x} - \dfrac{1}{d_x} = \dfrac{\rho_x - \varphi}{dB_x}, & j \in B, i = x \\ 0, & otherwise \end{cases}$$

Where ρ_x is the red ratio of node (x), that is $\rho_x = \frac{dR_x}{d_x}$ for red nodes and $1 - \rho_x = \frac{dB_x}{d_x}$ for blue nodes respectively.

We want to estimate

$$Q' = \gamma(I - (1-\gamma)P')^{-1} = \gamma\big(I - (1-\gamma)(P + D)\big)^{-1}$$

To do so, we exploit a fundamental lemma [11] that states that for a non-singular matrix *M* and a rank-1 matrix *H*, such that $M + H$ is non-singular, we have:

$$(M+H)^{-1} = M^{-1} - \frac{1}{1+g} M^{-1} H M^{-1}, \qquad g \coloneqq tr(HM^{-1})$$

Applying for $M = (I-(1-\gamma)P) = \frac{Q^{-1}}{\gamma}$ $and\ H = -(1-\gamma)D$:

$$\begin{aligned}
Q' &= \gamma (M+H)^{-1} \\
&= \gamma M^{-1} - \gamma \frac{1}{1+g} M^{-1} H M^{-1}, \qquad g \coloneqq tr(HM^{-1}) \\
&= \gamma \frac{Q}{\gamma} - \gamma \frac{1}{1+h} \frac{Q}{\gamma} (-(1-\gamma)\cdot D) \frac{Q}{\gamma}, \qquad h \coloneqq tr\left(-(1-\gamma) D \frac{1}{\gamma} Q\right) \\
&= Q + \frac{\frac{(1-\gamma)}{\gamma}}{1 - \frac{(1-\gamma)}{\gamma} q} QDQ, \qquad where\ q \coloneqq tr(DQ) \qquad (2)
\end{aligned}$$

With mathematical manipulations, we get:

$$DQ_{ij} = \begin{cases} 0, & i \neq x \\ \sum_{k\in R_x} \frac{\varphi - \rho_x}{dR_x} Q_{kj} + \sum_{k \in B_x} \frac{\varphi - \rho_x}{dB_x} Q_{kj}, & i = x \end{cases}$$

$$QDQ_{ij} = Q_{ix} DQ_{xj} = Q_{ix}\left(\frac{\varphi - \rho_x}{d^2 x} \sum_{k\in R_x} Q_{kj} - \frac{(\varphi - \rho_x)}{dB_x} \sum_{k\in B_x} Q_{kj}\right)$$

Using the fact that $q \coloneqq tr(DQ) = DQ_{xx}$ we have:

$$DQ_{xx} = \frac{(\varphi - \rho_x)}{dR_x} \sum_{k\in R_x} Q_{kx} - \frac{(\varphi - \rho_x)}{dB_x} \sum_{k\in B_x} Q_{kx}$$

Substituting in Equation 1 we have:

$$\begin{aligned}
Q'_{ij} &= Q_{ij} + \frac{\frac{1-\gamma}{\gamma}\left(Q_{ix}\left(\frac{\varphi - \rho_x}{dR_x} \sum_{k\in R_x} Q_{kj} - \frac{(\varphi - \rho_x)}{dB_x} \sum_{k\in B_x} Q_{kj}\right)\right)}{1 - \frac{(1-\gamma)}{\gamma}\left(\frac{\varphi - \rho_x}{dR_x} \sum_{k\in R_x} Q_{kx} - \frac{(\varphi - \rho_x)}{dB_x} \sum_{k\in B_x} Q_{kx}\right)} \\
&= Q_{ij} + Q_{ix} \frac{\frac{1}{dR_x} \sum_{k\in R_x} Q_{kj} - \frac{1}{dB_x} \sum_{k\in B_x} Q_{kj}}{\frac{\gamma}{(1-\gamma)(\varphi - \rho_x)} - \left(\frac{1}{dR_x} \sum_{k\in R_x} Q_{kx} - \frac{1}{dB_x} \sum_{k\in B_x} Q_{kx}\right)}
\end{aligned}$$

Summing over $j \in S$ and using the fact that $p'(S) = \frac{1}{n}\sum_{i=1}^{n} p'_i(S)$ and $p(x) = \frac{1}{n}\sum_{i=1}^{n} p_i(x)$:

$$
\begin{aligned}
p'(S) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in S} Q'_{ij} \\
&= p(S) + \sum_{j\in R} \frac{\frac{1}{dR_x}\sum_{k\in R_x} Q_{kj} - \frac{1}{dB_x}\sum_{k\in B_x} Q_{kj}}{\frac{\gamma}{(1-\gamma)(\varphi-\rho_x)} - \left(\frac{1}{dR_x}\sum_{k\in R_x} Q_{kx} - \frac{1}{dB_x}\sum_{k\in B_x} Q_{kx}\right)} \frac{1}{n}\sum_{i=1}^{n} Q_{ix} \\
&= p(S) + p(x)\frac{\frac{1}{dR_x}\sum_{k\in R_x} p_k(S) - \frac{1}{dB_x}\sum_{k\in B_x} p_k(S)}{\frac{\gamma}{(1-\gamma)(\varphi-\rho_x)} - \left(\frac{1}{dR_x}\sum_{k\in R_x} p_k(x) - \frac{1}{dB_x}\sum_{k\in B_x} p_k(x)\right)} \\
p'(S) &= p(S) + p(x)\,\Lambda(x,S)
\end{aligned}
$$

Subtracting gives us:

$$fgain = \Lambda(x,S)p(x)$$

The formula for the case where $d_x = 0$ follows from the fact that the P_x vector in the definition of matrix D is $\frac{\varphi}{|R|} - \frac{1}{|V|}$ and $\frac{1-\varphi}{|B|} - \frac{1}{|V|}$ for all red and blue nodes in the graph accordingly.

The respective formula that estimates the gain analytically, when imposing the proposed drastic behavior on individual nodes is modified as follows:

$$
\Lambda(x,S) = \begin{cases}
\dfrac{\frac{1-\gamma}{\gamma}\left(\frac{1}{dR_x}\sum_{k\in R_x} p_k(S) \ - \frac{1}{dB_x}\sum_{k\in B_x} p_k(S)\right)}{1 - \frac{1-\gamma}{\gamma}\left(\frac{1}{dR_x}\sum_{k\in R_x} p_k(x) \ - \frac{1}{dB_x}\sum_{k\in B_x} p_k(x)\right)}, & dR_x, dB_x \neq 0 \\
\dfrac{\frac{1-\gamma}{\gamma}\left(\frac{1}{|R|}\sum_{k\in R} p_k(S) \ - \frac{1}{d_x}\sum_{k\in B_x} p_k(S)\right)}{1 - \frac{1-\gamma}{\gamma}\left(\frac{1}{|R|}\sum_{k\in R} p_k(x) \ - \frac{1}{d_x}\sum_{k\in B_x} p_k(x)\right)}, & dR_x = 0, dB_x \neq 0 \\
\dfrac{\frac{1-\gamma}{\gamma}\left(\left(\frac{1}{|R|} - \frac{1}{|V|}\right)\sum_{k\in R} p_k(S) \ - \frac{1}{|V|}\sum_{k\in B} p_k(S)\right)}{1 - \frac{1-\gamma}{\gamma}\left(\left(\frac{1}{|R|} - \frac{1}{|V|}\right)\sum_{k\in R} p_k(x) \ - \frac{1}{|V|}\sum_{k\in B} p_k(x)\right)}, & dR_x, dB_x = 0
\end{cases}
$$

In spite of the fact that Theorem 5.1 estimates analytically the gain when a single node is imposed to behave fair, there are also some particularities to be considered.

First and foremost, Theorem 5.1 and the proposed formula, do not apply modifications to the random jump vector, whereas the fair algorithms implemented for experimental evaluation do, so as to meet the fairness criterion. Taking into consideration that there is no possible way to achieve the desired fairness on a network without properly modifying the jump vector, we are willing to accept a slight deviation from the proposed formula and thus, we follow the convention that whenever a node is imposed to behave in a fair manner, the corresponding jump vector is modified to the fair jump vector accordingly. Although a possible solution would be to adjust parameter φ as $\varphi = \frac{|R|}{|V|}$, so that the jump vector is fair from scratch, the conducted experiments will not be of interest since we will already be close to fairness.

Moreover, the proposed formula estimates the gain when an individual node is forced to behave in a fair manner. With that being said, we ought to elucidate that our formula behaves as if every other node, except the selected one, acts accordingly to the original PageRank. Therefore, it cannot handle and hence estimate, the gain for multiple fair nodes. As a consequence, when comparing the total gain of several individual nodes to the output of one of the proposed fair PageRank algorithms, when the same nodes are imposed to behave fair, there will clearly be a difference.

Chapter 6. Selection Policies and Algorithms

In this section, we describe the selection policies according to which nodes will be chosen to behave fair.

6.1 Selection Policies

We use the following selection policies so as to specify the order in which nodes are selected to behave in a fair manner. We start with no fair nodes and constantly render fair ones. Note that the proposed policies choose nodes among those who do not respect the φ ratio, that is group L_R. The iterating process of rendering fair nodes terminates when all nodes of L_R are imposed to behave in a fair manner.

- *Formula:* This selection policy uses Theorem 5.1 to compute fairness gain ($fgain$) for each node, and then iteratively selects the one with the highest gain. Although accurate, this policy is expensive to compute. The increased complexity derives from the existence of $p_k(x)$ at the denominator of the proposed formula, that is the personalized PageRank that node k allocates to node x. Therefore, we must calculate the personalized PageRank vectors for all nodes of the graph in order to apply it to the equation, which indeed is quite expensive, especially for real, and thus big networks. For this reason, we propose the following policy.
- *FormulaApproximation:* This policy is an approach to Formula selection. We eliminate the term with increased complexity, i.e., denominator, and multiply the outcome by *($\varphi - \rho_x$).* The latter value introduces the concept of "fairness degree", as with $\varphi - \rho_x$ we measure the fairness ratio of a node.
- *Static PR:* This selection policy places the nodes of L_R in descending order, according to their original PageRank weight.

- *Random:* This policy picks nodes at random. To ensure accuracy, we conducted several iterations by constructing different permutations over the L_R group of nodes.

All the aforesaid selection policies calculate the order in which the nodes will be selected in advance. Therefore, they may be considered as static policies, as their outcome only depends on the starting situation of the graph. For this purpose, we introduce a dynamic policy, the output of which changes along with the PageRank vector, as we gradually render fair nodes.

- *Dynamic PR:* At first, this selection policy selects the node with the highest original PageRank value. As expected, imposing a fair behavior to the selected node will result in a different, fairer PageRank vector. Therefore, the next node to be chosen will be the one with the highest PageRank value according to the new, fairer distribution.

6.2 An Efficient Computation

To select the best node to be imposed to behave in a fair manner, we use Theorem 5.1 to compute fairness gain (*fgain*) for each node of the network, and then iteratively select the one with the highest gain, until we reach fairness.

Authors of [2] conducted a similar procedure for link recommendations, proposed a respective formula for fairness gain, as well as presented an efficient algorithm for the selections, that includes *absorbing random walks.* Our work follows closely this approach and therefore, we will adopt the introduced algorithm. This algorithm is of high importance in terms of efficiently computing the personalized PageRank ratio of node i for the red group, that is $p_i(R)$, which appears in several of the proposed selection algorithms (e.g., *FormulaApproximation*), without analytically computing the PPR for each node. We will now briefly describe this efficient algorithm.

The efficient computation relies on the use of *absorbing random walks.* In an absorbing random walk we have two types of nodes: *transient* nodes, from which we transition like in a regular random walk, and *absorbing* nodes, out of which we cannot transition, and thus the walk is *absorbed.* For the purpose of computing the PPR ratio $p_i(R)$, we define the following absorbing random walk. Given the graph G, we add an absorbing node α_r representing the protected (red) group, and with probability γ we add an edge from each red node to the new added state. Furthermore, α_r loops back to itself.

Let $\tilde{B}_{i\alpha_r}$ denote the absorption probabilities for node i to α_r. The following lemma appears in [2].

Lemma 7.1: The PPR ratio of node i for the red group R is equal to the absorption probability of state i to state α_r*:* $p_i(R) = \tilde{B}_{i\alpha_r}$.

Working on the defined absorbing random walk we can efficiently compute the PPR ratio of all nodes for the protected group.

Chapter 7. Experimental Evaluation

Our goal is to rate and measure PageRank fairness in different real networks and evaluate the proposed fair PageRank algorithms when imposed on individual nodes.

Datasets. We use the following real datasets:

- *Zachary's Karate club:* A social network of friendships between members of a karate club. Each node represents a member, and each edge represents a tie between the two members of the club respectively.
- *Books:* A network of books about US politics where links between books represent co-purchasing. [11]
- *Blogs:* A directed network of hyperlinks between web blogs on US politic. [12]

The characteristics of the datasets are summarized in the table below (Table 7.1).

Dataset	#nodes	#edges	ρ	β	PR(R)	PR(B)	Protected attribute
Karate	34	78	0.5	0.5	0.49	0.51	Split after argument
Books	105	441	0.51	0.49	0.48	0.52	Political (left)
Blogs	1,222	16,714	0.48	0.52	0.33	0.67	Political (left)

Table 7.1. Dataset characteristics notation: ρ, β: relative size, PR(R), PR(B): original PageRank assigned to each group respectively.

In every dataset, we define as *red* the protected group, that is the group whose original PR ratio is smaller than its ratio in the overall population. For instance, for the *Blogs* dataset, the red group is the political left. Furthermore, through the documented dataset characteristics, we may observe that PageRank fairness varies among the datasets. Although for some (e.g., *Karate*), it is almost equal to ρ (relative size of red group), for others (e.g., *Blogs*), it is considerably smaller. Last but not least, we will evaluate both undirected and directed graphs.

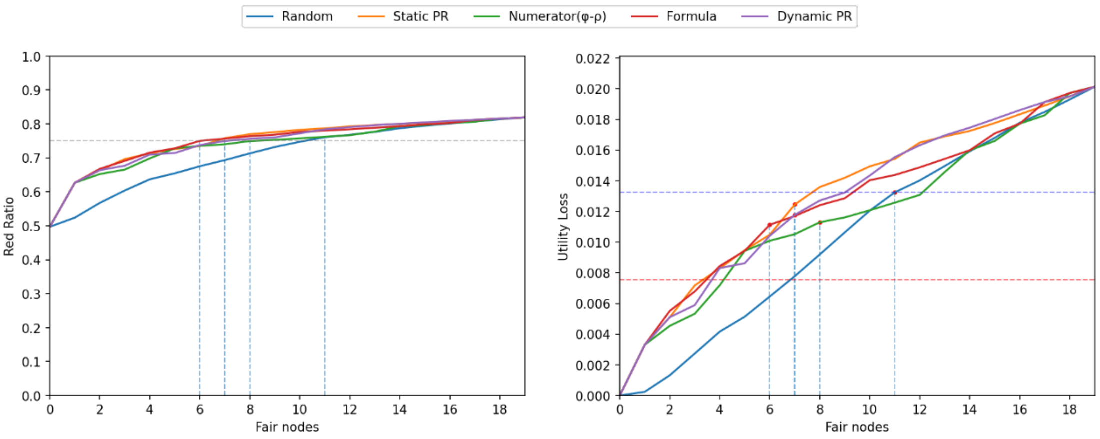
When is Fairness achieved? We study the conditions under which the proposed selection policies achieve fairness and we focus on the required budget of nodes. Note that these policies will be applied on both weight distributions, that is Neighborhood and Residual-based, as well as the proposed extreme behavior of individual nodes. Although it would be reasonable to assume that the algorithms are fair if they respect the demographic parity, that is, if each group obtains PageRank equal to its ratio in the overall population ($\varphi = \rho$), two of the datasets to be examined are already close to fairness and hence, the results will not be of interest. For this purpose, we set $\varphi = 0.75$.

7.1 Neighborhood *LFPR* Algorithm

In this series of experiments, we evaluate the proposed Neighborhood Locally Fair PageRank algorithm.

At first, we plot the *Red ratio*, that is, the PageRank mass allocated by the red group, as a function of the number of fair nodes. We indicate the required fair nodes to achieve fairness for each selection policy with blue, dotted, vertical lines, while the desired φ – fairness with a gray, dotted, horizontal line. We also plot the *utility loss* for achieving fairness and show the respective lower bound with red horizontal, dotted line. Abusing the notation, we will use *upper bound* to denote the value of utility loss when all nodes of the graph are imposed to behave fair.

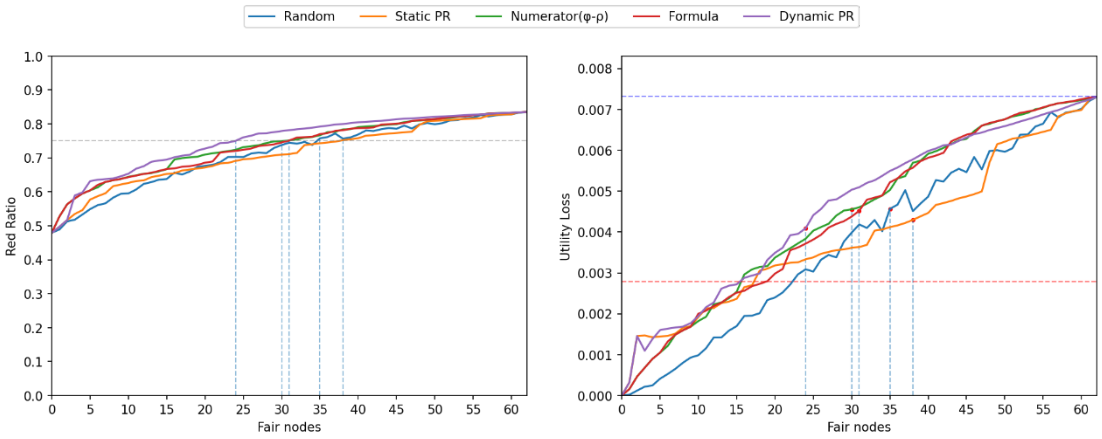
In Figure 7.1.1, we report the red PageRank on the left and utility loss on the right, for the *Karate* dataset. The proposed Formula selection policy seems the best option as it requires the smallest subset of nodes to achieve fairness, while minimizing the utility loss. Although *FormulaApproximation* policy outputs almost the same loss as the previous one, it requires more fair nodes. Last but not least, we might expect the dynamic PR policy to perform better, but for now the difference with the static remains subtle. Note that every selection policy attains lower utility loss from the upper bound, with random being just under the limit.



7.1.1 Neighborhood LFPR for Karate dataset (undirected).

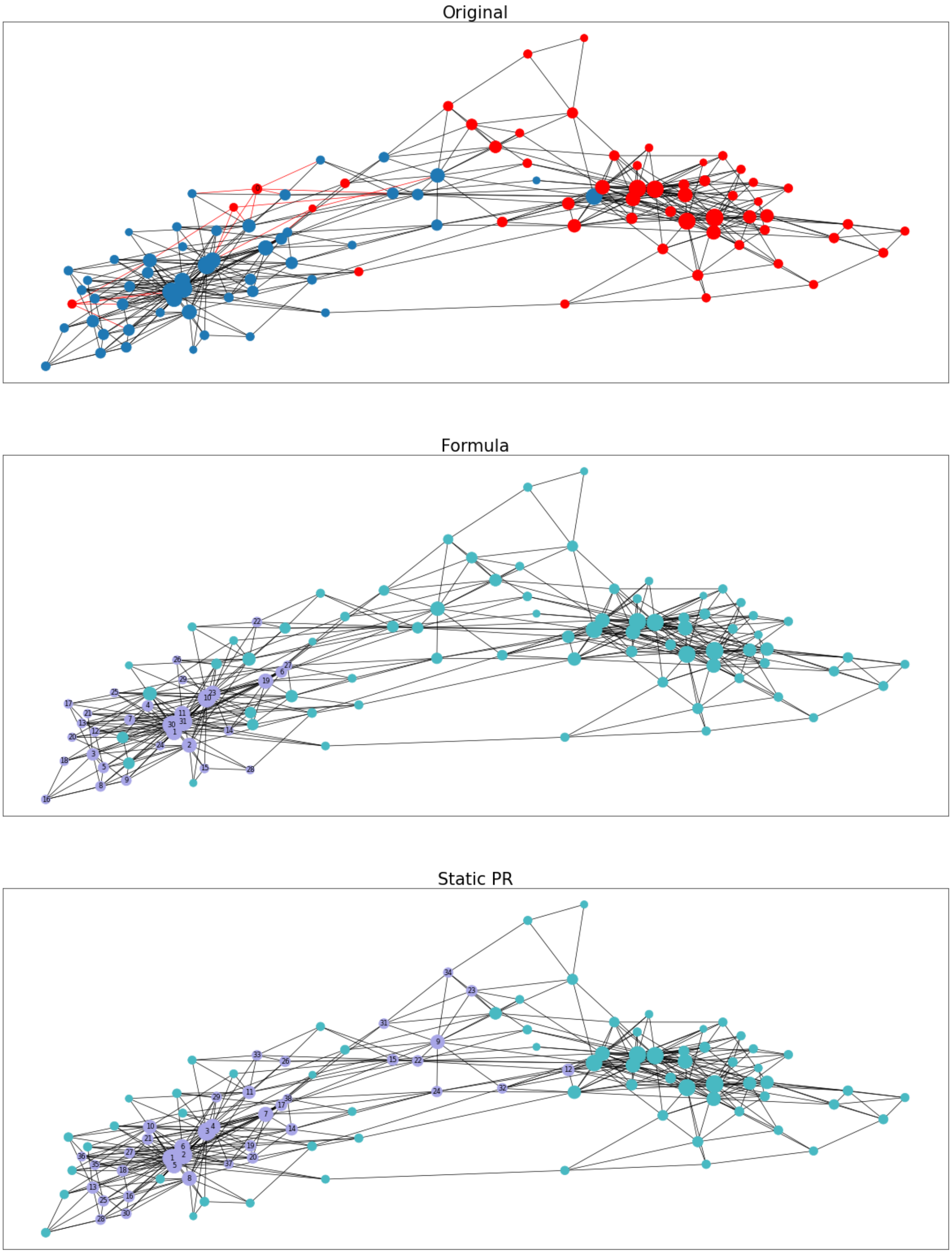
We plot the respective results for the *Books* dataset in Figure 7.1.2. and we observe a completely different situation. Dynamic PR selection policy stands out as not only does it requires a considerable smaller subgroup of nodes to behave fair in order to achieve the desired fairness, but also minimizes the utility loss. Regarding Formula and *FormulaApproximation*, these two policies perform similarly. The former requires one extra fair node, but also reports slightly lower utility loss. Nevertheless, Static PR caught us by surprise, as the requisite subset of nodes surpasses every other policy, including Random. As a matter of fact, Random performs relatively well if we consider the small difference compared to the Formula selection policy.

We addressed the characteristics and the interrelations among the nodes of the graph and observed that this dataset exhibits a very high degree of homophily. Nodes essentially link only to nodes in their own group. In addition to this, there is a small portion of red nodes located in blue neighborhoods, meaning that they only link to blue nodes. Therefore, even though very few cross links from blue to red nodes exist (and vice versa), they do not actually affect fairness. Static PR policy selects such blue nodes to be rendered fair and as a result, they do not channel their given fairness properly to the red group. On the other hand, both Formula and *FormulaApproximation* policies, prefer nodes with no red neighbors, and hence the fair PageRank algorithm distributes their weight uniformly to the red nodes, with respect to the φ – ratio.



7.1.2 Neighborhood LFPR for Books dataset (undirected).

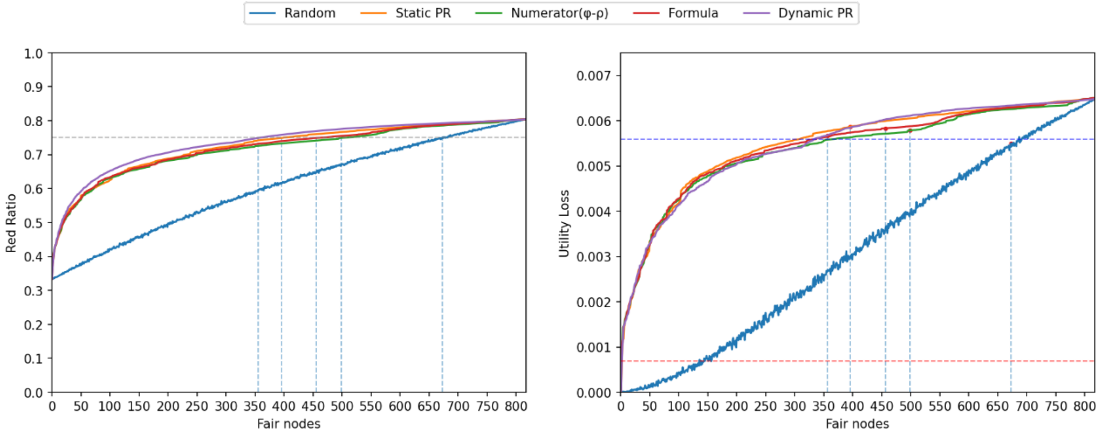
In Figure 7.1.3 we visualize the *Books* dataset. Violet and light blue colored vertices denote fair and original nodes respectively. Labels of fair nodes indicate the order in which they were rendered fair, while red colored links correspond to links from blue nodes that do not respect the φ-ratio, to red nodes that only preserve blue neighbors. In the top left of Figure 7.1.3, we may locate the red node labeled *"0"* which links to five blue nodes. Hence it is in a blue neighborhood. Static PR policy selects three out of five blue neighbors to behave fair and as a result, they do not channel their given fairness properly to the red group. A corresponding situation prevails with several red nodes and therefore, the required subset of fair nodes increases significantly. On the other hand, Dynamic PR prefers nodes with no red neighbors, which will uniformly allocate their weights to underrepresented group. Therefore, this policy does not select any neighbors of red node *"0"*.



7.1.3 Visualization of Books dataset (Selected nodes from Formula and Static PR).

Regarding the *Blogs* dataset, in Figure 7.1.4 we plot the *Red ratio* and the *utility loss* as a function of the number of fair nodes. To begin with, Dynamic PR selection policy requires the smallest subset of nodes in order to achieve fairness. At the same time, utility loss is only above the lowest achieved by any of the proposed policies. As far as nodes are concerned, Static PR performs similarly well, requiring just a few extra nodes.

Nevertheless, the respective utility loss surpasses every other algorithm. In terms of fair nodes, Formula performs hardly better than *FormulaApproximation*. Even so, the latter reports decreased utility loss. A corresponding situation with enhanced subset of nodes but lower utility loss, prevails in the Random selection policy. As a matter of fact, the requisite subset of nodes is the biggest among all. However, the measured utility loss is the lowest attained. This is due to the fact that in order to achieve the desired fairness, this policy requires more than half of the nodes comprising the graph to be rendered as fair. Therefore, the weights circulate smoothly across the entire network, resulting in lower utility loss. On top of that, we should also note that the current graph is directed. Last but not least, all policies except Random selection, exceed the upper bound of utility loss. This phenomenon is linked to the relatively small, required subset of fair nodes. The more nodes we impose to behave in a fair manner, the lower the utility loss will be. Therefore, if we were to render as fair the nodes that already respect the φ - ratio, we would fall into the upper bound of utility loss. In anecdotal conducted experiments, we measured that as we lower the parameter φ, this phenomenon is eliminated.



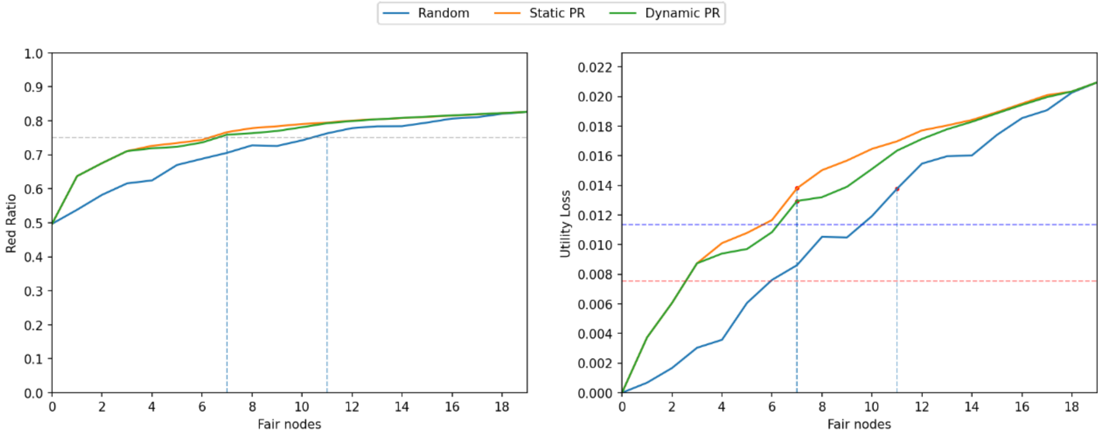
7.1.4 Neighborhood LFPR for Blogs dataset (directed)

7.2 Residual *LFPR* Algorithm

In the occasion of the special interrelations among the nodes of *Books* dataset, we observed that the best nodes to be rendered fair would be the ones with no red neighbors. As a result, the fair PageRank algorithms will distribute the weights uniformly to the protected group. Corresponding behaviors are imposed by the proposed Residual based fair algorithms, with the difference that every node that does not respect the φ – ratio, distributes its residual to all nodes of the protected group, according to the residual distribution policy. Therefore, it is expected that this family of algorithms will perform equally well, if not better. Although the aforesaid characteristics were only observed in the *Books* dataset, for matters of completeness we also implement the Residual fair algorithms to the rest of the datasets. In addition to this, we only present the Proportional distribution policy, as both distribution policies performed equivalently.

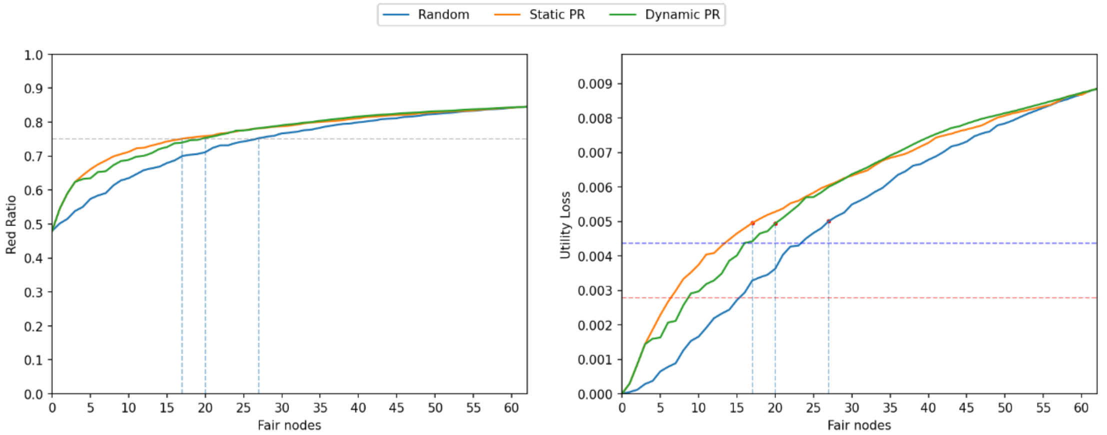
As for the plots, we preserve the same structure. We present red Ratio along with utility loss, with respect to the number of fair nodes. Note that Theorem 5.1 applies only to the Neighborhood LFPR algorithm. Hence, we only present the rest of the proposed selection policies.

In Figure 7.2.1, we report the red PageRank on the left and utility loss on the right, for the *Karate* dataset. Static PR performs similarly well to Dynamic PR, with the latter reporting slightly lower utility loss. As expected, Random selection requires more fair nodes to achieve fairness. Overall, compared to the respective output of the Neighborhood LFPR algorithm, there are no noticeable differences. Nevertheless, we must point out that all three sorting algorithms surpass the upper bound of utility loss.



7.2.1 Residual LFPR for Karate dataset (undirected).

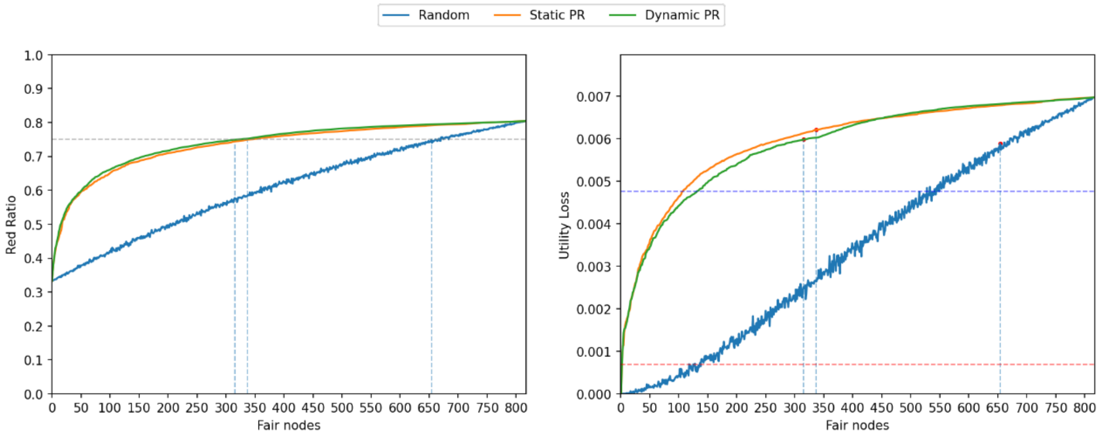
Regarding the *Books* dataset, we plot the respective results in Figure 7.2.2. First and foremost, in terms of fair nodes, Static PR selection outweighs every other policy as it requires the least subset of nodes. Dynamic PR follows with just a few extra fair nodes and slightly lower utility loss. As expected, Random sorting algorithm cannot compete with the rest. As for the upper bound of utility loss, a corresponding situation to the *Karate* dataset prevails. All the proposed selection policies surpass the upper utility loss.



7.2.2 Residual LFPR for Books dataset (undirected).

Nevertheless, it would be interesting to compare the output of the two fair PageRank algorithms, namely Neighborhood and Residual. To begin with, regarding the required subset of nodes, Residual based fair algorithm performs noticeably better in every of the proposed selection policies. As a matter of fact, Residual Random selection almost competes with the best Neighborhood selection policy, that is Dynamic PR. That is not the case, however, in terms of utility loss. In some instances, Neighborhood selection policies attain considerably lower utility loss, but in others, there are no notable differences.

Finally, we present the output of *Blogs* dataset for the Residual based fair PageRank algorithm in Figure 7.2.3. In terms of fair nodes, once again, Dynamic PR requires the least subset. Relatively, well regarding its staticity, performs Static PR selection, while Random comes last requiring double the fair nodes. As for the attained utility loss, Random selection policy reports the lowest value, supported by the fact that the selected nodes, represent half of the graph. Respective values for Dynamic PR and Static PR selections are slightly higher. Corresponding to the previous datasets, so in this one, the proposed selection policies outstrip the upper bound of utility loss. Overall, if we were to compare Residual based with Neighborhood LFPR algorithm for the current dataset, the former shrinks the requisite subset of fair nodes, in the detriment of increased utility loss.

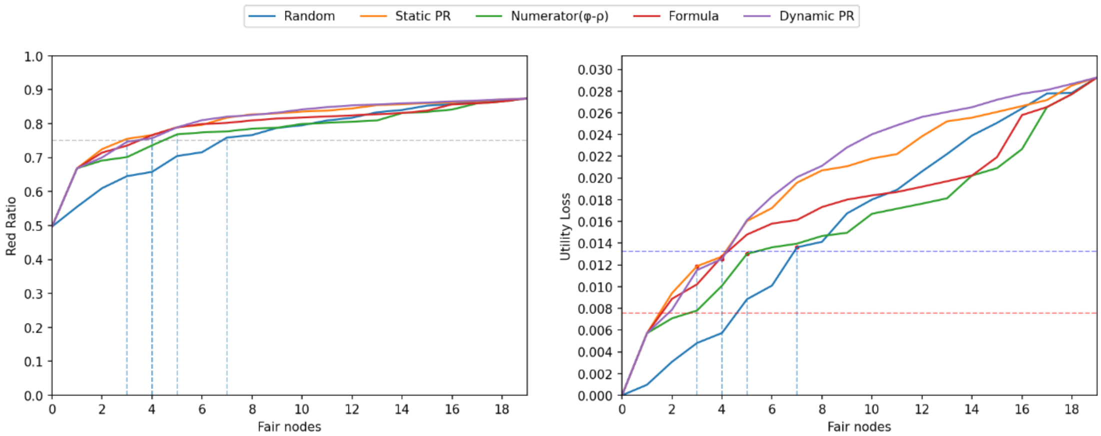


7.2.3 Residual LFPR for Blogs dataset (directed).

7.3 Neighborhood *LFPR*, a drastic change

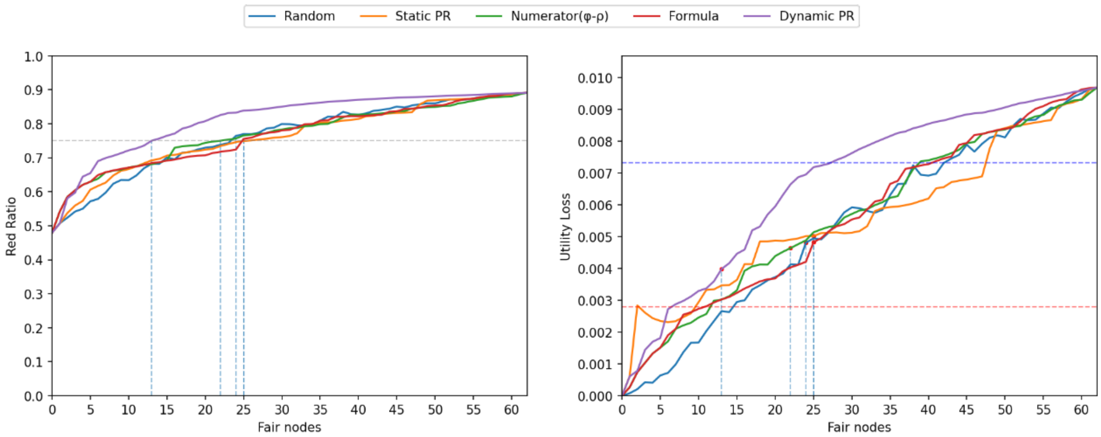
Following the experimental evaluation, we present the respective plots for the Neighborhood LFPR algorithm, when the selected nodes are imposed to behave according to the proposed extreme manner. Although the upper bound of utility loss will obviously escalate, we preserve the previous upper bound, that is of the conventional weights allocation, in order to compare the two distributions.

In Figure 7.3.1, we report the red PageRank on the left and utility loss on the right, for the *Karate* dataset. Regarding the order in which selection policies achieve fairness, there are no remarkable differences. In addition to this, as expected, the subset of nodes required to behave fair considerably decrease for all the proposed selection policies. Utility loss for achieving fairness faced only a subtle increase while at the same time remained under the upper bound of utility loss. It is worth noting that Static PR selection attains lower utility loss while implementing the extreme strategy of allocating weights when compared to the respective one for the conventional. In addition to the remarkable small set of fair nodes, this is also explained by the fact that the extra nodes required by the conventional distribution preserve only blue neighbors. Hence, according to the fair algorithm, they allocate their weight uniformly to the red group which increases utility loss. Taking everything into account, the proposed extreme strategy of allocating weights proved efficient for the current dataset.



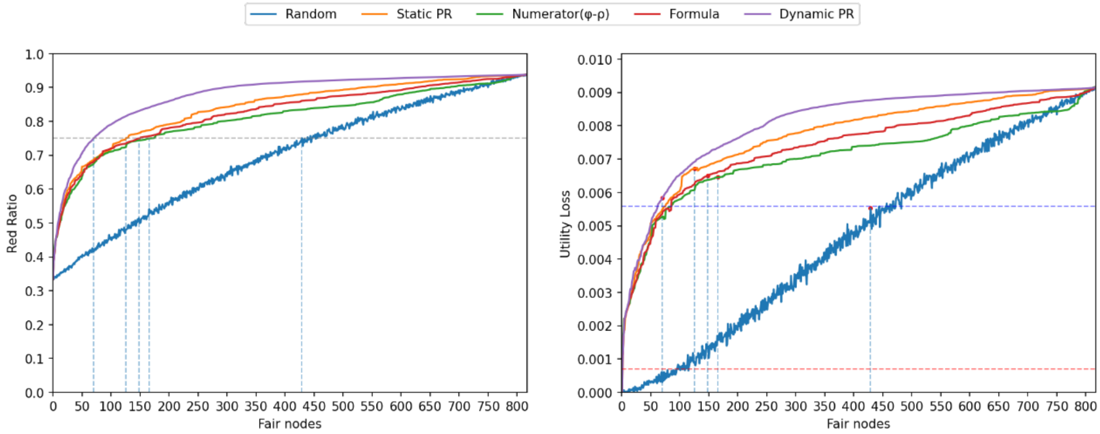
7.3.1 Neighborhood LFPR (extreme strategy) for Karate dataset (undirected)

Continuing with the *Books* dataset, we plot the respective results in Figure 7.3.2. Dynamic PR stands out among every other selection policy and distribution strategy, requiring the least subset of fair nodes while attaining the lowest utility loss. Like before, the drastic strategy of allocating weights attains lower utility loss, since other policies choose nodes which only preserve blue neighbors. Therefore, imposed to behave fair, they distribute their weights uniformly to the red group, escalating utility loss. Apparently, the interrelations across the nodes of the graph also affect the Formula algorithm, as not only the requisite subset of nodes did not noticeably decrease, but also Random selection policy performs equally well. After all, the proposed selection policies, except Dynamic PR, report an increased utility loss, but yet remain below the upper bound.



7.3.2 Neighborhood LFPR (extreme strategy) for Books dataset (undirected)

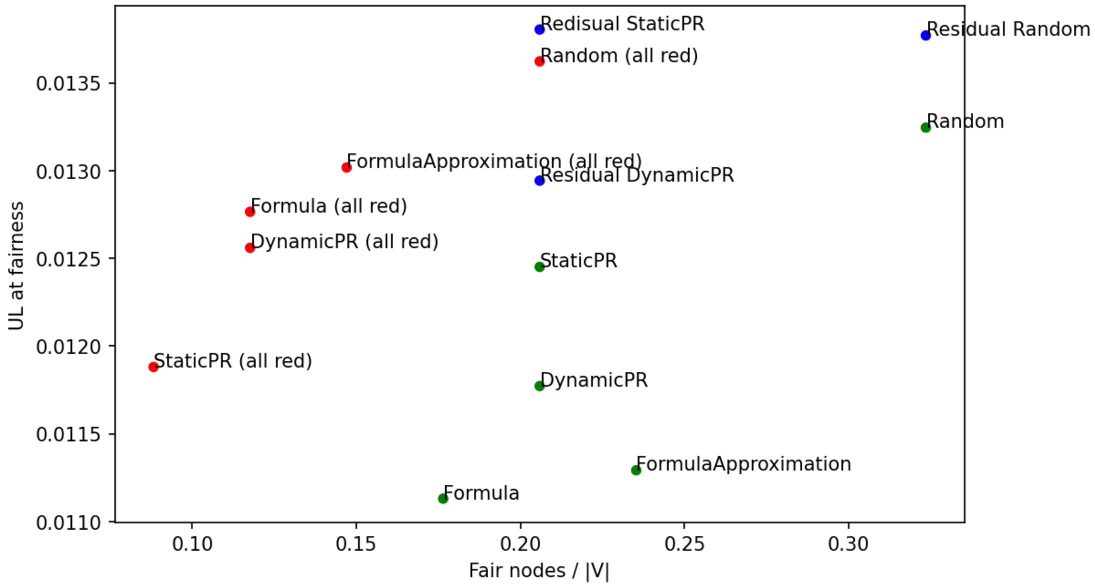
Finally, in Figure 7.3.3 we present the output of implementing the Neighborhood LFPR algorithm for *Blogs* dataset. Additionally, nodes are imposed to behave according to the proposed extreme strategy of weights distribution. As expected, every selection policy achieves fairness with a significantly smaller subset of fair nodes. Nevertheless, that is to the detriment of utility loss. Most of the policies report notably higher values, this is not the case, however, for Dynamic PR. Utility loss of the latter faced an increase, but yet remained relatively close to the respective of conventional distribution policy. Just as importantly, reported utility losses exceed the corresponding one when all nodes are imposed to behave in a fair manner.



7.3.3 Neighborhood LFPR (extreme strategy) for Blogs dataset (directed)

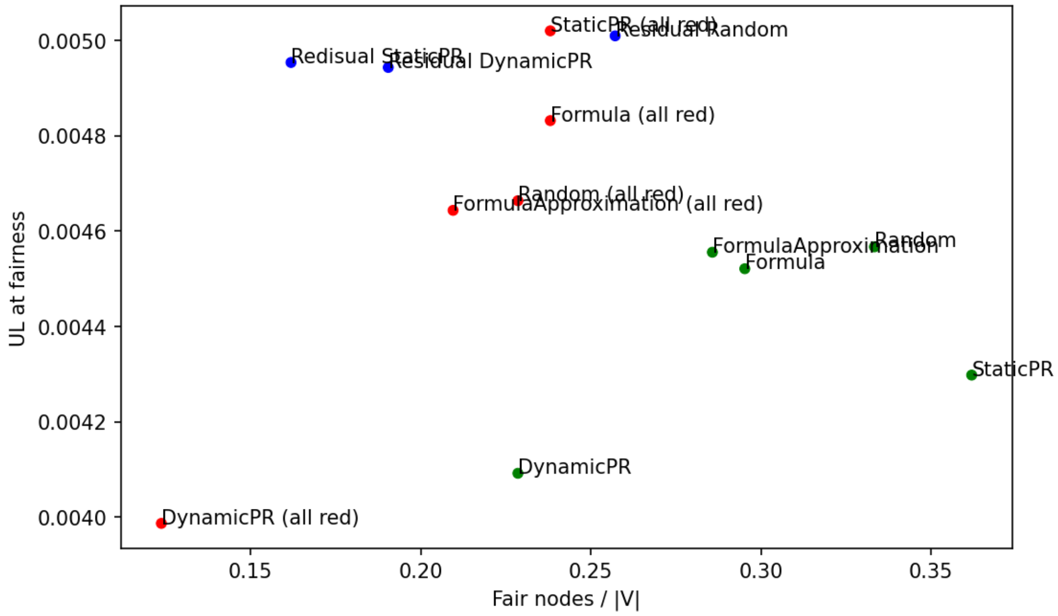
7.4 Overall Performance

Concluding the experimental evaluation, we present the overall performance of the proposed distribution and selection policies. For this purpose, we divide the number of required fair nodes by |V|. Each point represents a different selection policy and each color a different weights allocation policy. In Figure 7.4.1 we plot the results of *Karate* dataset and observe that Formula along with StaticPR when the nodes are imposed to behave according to the extreme distribution policy, are the best alternatives. Of course, we should also consider that Formula policy is quite expensive to compute. Relatively well perform FormulaApproximation and DynamicPR.



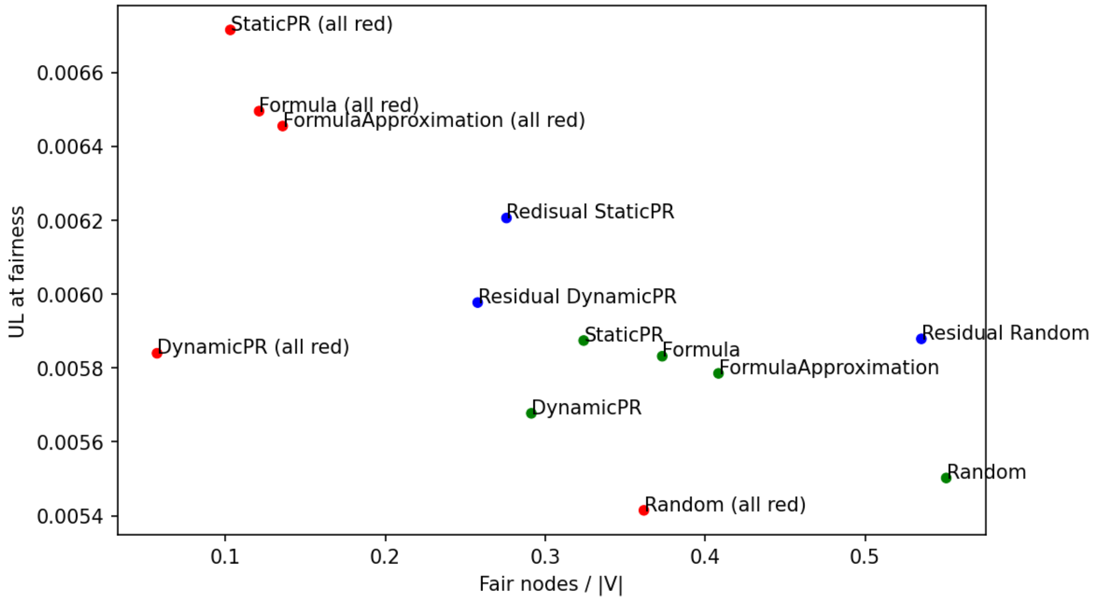
7.4.1 Overall Comparison for Karate (undirected)

Regarding Books dataset, we plot the respective results in Figure 7.4.2 and face a completely different situation. Note that this dataset exhibits a very high degree of homophily. Due to the interrelation across the nodes, DynamicPR policy for both the conventional and the extreme behavior for individual nodes, outweighs every other selection policy. Therefore, in future work we should consider different Dynamic selection policies. In addition to this, Residual policies achieved their goal reducing the subset of fair nodes, but also increased utility loss.



7.4.2 Overall Comparison for Books (undirected)

Finally, in Figure 7.4.3 we plot the output of the directed dataset, e.g., *Blogs.* A corresponding situation prevails as DynamicPR selections consist the best alternative. Although Random selection seems to perform relatively well, we should also consider the significantly enhanced subset of required fair nodes to achieve fairness. The reduced utility loss attained by the aforesaid selection policy, results in the attributed placement on the plot.



7.4.3 Overall Comparison for Blogs (directed)

Chapter 8. Conclusions

In this diploma thesis, we studied fairness for PageRank algorithm and focused on achieving fairness with the least possible modifications. We provided analytical formulas that measure the effect of imposing fair behaviors on individual nodes on fairness, as well as proposed both Static and Dynamic policies for selecting nodes to be rendered fair.

There are many directions for future work. First, we would like to derive corresponding analytical formulas for all the considered policies of allocating weights. In addition to this, it is of our interest to explore different Dynamic selection policies, as they seem to outweigh the respective Static. Last but not least, we are looking forward to study the problem of attaining fairness with the least possible changes, while minimizing the utility loss.

Bibliography

[1] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. 2021. Fairness-Aware PageRank. In WWW '21: *The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23*, 2021. 3815–3826.

[2] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Konstantinos Semertzidis, and Panayiotis Tsaparas. 2022. Link Recommendations for PageRank Fairness. In Proceedings of the *ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.*

[3] Golnoosh Farnadi, Behrouz Babaki, and Michel Gendreau. 2020. A Unifying Framework for Fairness-Aware Influence Maximization. *In Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. 714–722.

[4] Suresh Venkatasubramanian, Carlos Scheidegger, Sorelle A. Friedler, and Aaron Clauset. [n. d.]. Fairness in Networks: Social Capital, Information Access, and Interventions. In KDD '21: *The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 4078–4079.

[5] Avishek Joey Bose and William L. Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 715–724.

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

[7] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in machine learning. *In Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*. 329–338.

[8] Enyan Dai and Suhang Wang. 2021. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. In *WSDM ’21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*. ACM, 680–688.

[9] Chen Avin, Barbara Keller, Zvi Lotker, Claire Mathieu, David Peleg, and Yvonne- Anne Pignolet. 2015. Homophily and the Glass Ceiling Effect in Social Networks. In *ITCS.* 41–50.

[10] Lisette Espín-Noboa, Claudia Wagner, Markus Strohmaier, and Fariba Karimi. 2021. Inequality and Inequity in Network-based Ranking and Recommendation Algorithms. *CoRR* abs/2110.00072 (2021).

[11] http://www-personal.umich.edu/~mejn/netdata/

[12] L. A. Adamic and N. S. Glance. 2005. The political blogosphere and the 2004 U.S. election: divided they blog. In *LinkKDD*.

[13] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.* 54, 6 (2021), 115:1–115:35.

[14] Jian Kang and Hanghang Tong. 2021. Fair Graph Mining. In CIKM ’21: *The 30th ACM International Conference on Information and Knowledge Management.*

[15] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. 2019. Group- Fairness in Influence Maximization. *In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. 5997–6005.

[16] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. 2019. Guarantees for Spectral Clustering with Fairness Constraints. *In Proceedings of the 36th International Conference on Machine Learning,* ICML. 3458–3467.

[17] J. Kang, J. He, R. Maciejewski, and H. Tong. 2020. InFoRM: Individual Fairness on Graph Mining. In *KDD*.

[18] Emmanouil Krasanakis, Symeon Papadopoulos, and Ioannis Kompatsiaris. 2020. Applying Fairness Constraints on Graph Node Ranks Under Personalization Bias. In *COMPLEX NETWORKS*, Vol. 944. Springer, 610–622.

[19] D. F. Gleich. 2015. PageRank Beyond the Web. SIAM Rev. 57, 3 (2015*)*, 321–363.