DeepFairRank: A Multi-Objective Framework for Fair Top-k Node Ranking in Network Data

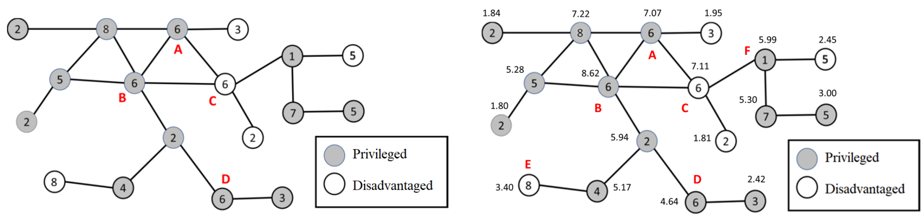
Francisco Santos, Farzan Masrour, Pang-Ning Tan, and Abdol-Hossein Esfahanian

Michigan State University, East Lansing, Michigan, USA 48825
{santosf3, masrours, ptan, esfahanian}@msu.edu

Abstract. The fair top-k node ranking problem aims to find the k most significant nodes in a network without discriminating against particular groups of nodes as defined by their protected attribute. However, unlike fair ranking problems for independent and identically distributed (i.i.d.) data, the rank assigned to a node may influence the perception of fairness among its neighbors with similar acceptability scores due to the interconnectivity among the nodes. Fairness perception, which is an individual-level fairness metric, has thus been proposed to measure the degree to which a node perceives its ranking outcome as fair. While existing fair node ranking algorithms can help maximize its fairness perception, they are susceptible to the oversmoothing effect due to their message passing mechanism. Thus, a key challenge in designing fair node ranking algorithms is to balance the trade-off between maximizing the acceptability of the highly ranked nodes while satisfying both individual-level and group-level fairness criteria. To address this challenge, this paper presents a novel framework called `DeepFairRank` that integrates the potentially diverging criteria in a unified, multi-objective optimization framework using neural networks. Experimental results demonstrate the effectiveness of the framework when applied to real-world data.

1 Introduction

The ranking of nodes in a network [23] plays a central role in a myriad of applications, from viral marketing to recommender systems. Unlike traditional ranking problems, the node ranking algorithm must consider the link structure of the network in addition to its node attribute information. Current node ranking algorithms can be broadly categorized into centrality-based, influence-based and score-based methods. Centrality-based methods [3] identify the top-k most significant nodes based on their relative positions in the network. Such methods would employ node-centric measures such as PageRank and eigenvalue centrality to determine the degree of importance of each node. In contrast, influence-based methods [11] evaluate the prominence of a node in terms of the expected number of nodes influenced given a diffusion model such as the linear threshold or independent cascade models [8]. Score-based ranking methods [24], which are the focus of this study, select the most “qualified" individuals given their acceptability scores.



(a) Toy network. (b) Rank scores of nodes.

Fig. 1: A network of student applicants for college admission. (a) The number associated with each node represents acceptability score while the node color denotes its protected group membership (privileged or disadvantaged). (b) Rank scores after applying a fairness-sensitive PageRank (FSPR) algorithm [20].

To illustrate the score-based node ranking problem, consider the network shown in Figure 1(a). Assume each node represents a high-school student applying for college admission. The acceptability score of a student could be based on a standardized test result or its combination with other criteria such as high school grades. Assume the acceptability score ranges from 1 (unacceptable) to 10 (must accept). Our goal is to select the top-k candidates for admission based on their acceptability and fairness criteria. We further assume that the candidates can be divided into groups based on some sensitive attribute, say, race or gender. For brevity, the groups are denoted as *privileged* and *disadvantaged* in the diagram. To ensure that the admission decision will not discriminate against the disadvantaged group, the ranking algorithm should consider the proportional representation of nodes from different groups in its top-k ranking [24] by incorporating group-level fairness criteria such as statistical parity [4] in its formulation.

Furthermore, due to their social ties, the decision to admit a student can significantly impact the perception of fairness within the student's social circle. For example, suppose node **C** in Figure 1(a) observes both of its neighbors with similar acceptability scores, **A** and **B**, are accepted. This raises **C**'s expectation that it should also be accepted given their similar qualifications. However, if **C** is rejected, then the decision will likely be perceived as unfair since the outcome is below **C**'s expectation. In contrast, if **C** observes both **A** and **B** get rejected, this will lower its expectation, making **C** more amenable to the rejection decision. To ensure such expectation is met, individual-level criteria such as *fairness perception* [12] should be considered when ranking the nodes.

The notion of fairness perception is well-aligned with ideas from social equity theory [1], which suggests that the degree of satisfaction of individuals should be based on what they expect to receive. An individual will more likely perceive an outcome to be fair if the decision meets or exceeds the individual's expectation. However, since it is only an individual-level fairness metric, it does not ensure fairness for all groups of the protected attribute. In addition, the fairness perception measure introduced in [12] was designed for node classification instead of node ranking problems.

Another limitation of existing fairness-aware node ranking algorithms is their primary focus on enhancing fairness of the nodes' PageRank scores in a network [20, 6, 19]. While these approaches can indeed improve individual-level fairness, they are vulnerable to the *oversmoothing* problem due to their inherent message passing mechanism. For example, Figure 1(b) shows the rank scores after applying the FSPR algorithm [20] to rank the nodes shown in Figure 1(a):

$$\mathbf{h} = (1-\gamma)\mathbf{P}^T\mathbf{h} + \gamma\mathbf{s}, \tag{1}$$

where $\mathbf{h}$ is the rank vector of the nodes, $\mathbf{P}$ is the transition probability between nodes, $\mathbf{s}$ is the acceptability score vector of the nodes, and γ is a parameter associated with the jump probability, which is often set to 0.15. Observe that the score for the highly acceptable node **E** has reduced significantly from 8 to 3.4 while the score for node **F** increases from 1 to 5.99 due to the smoothing effect of their neighbors. As a result, node **F** is ranked higher than node **E** despite having a much lower acceptability score.

To overcome these challenges, this paper presents a novel score-based node ranking algorithm called `DeepFairRank`, designed to balance the trade-off between maximizing acceptability scores and fairness perception while minimizing the disparity among different groups of nodes using a neural network. Specifically, we first introduce an approach to compute fairness perception for node ranking problems. `DeepFairRank` also employs a multi-objective optimization framework to integrate the potentially diverging criteria. Experimental results on real-world datasets demonstrate the effectiveness of our proposed algorithm compared to other conventional fairness-aware ranking algorithms.

2 Related Work

There is a vast body of literature on the node ranking problem. Centrality-based measures [3] such as the Katz index [7] and PageRank [15] employ various metrics to determine the relative importance of a node in a network. Influence-based approaches [11] consider models of the diffusion process [8] to assess the degree of influence of the nodes. Both centrality-based and influence-based methods generally utilize the link information only to rank the nodes. In contrast, score-based models [24] consider other node attribute information to perform the ranking.

Algorithmic fairness is an important topic that has attracted considerable interest in recent years. Fairness can be defined at *individual level* as the absence of any prejudice towards an individual in a decision-making task [14]. In other words, similar people should be treated similarly [4]. For *group fairness*, a fair decision should not favor one group over another. Examples of group fairness metrics include *demographic parity* or *statistical parity* [4] and *equalized odds* [5]. Aside from defining fairness metrics, methods for debiasing decision outcomes from machine learning models have also been developed [13].

Recent years have also witnessed growing interest in ensuring fairness in ranking algorithms [24, 16]. For example, the `FA*IR` algorithm [22] was designed to generate rankings that maximize utility while ensuring fairness across different groups. [2] presented an approach for designing fair ranking schemes by assisting users in selecting better weights for combining attributes when ranking items in a database. [18] considered exposure allocation between groups in designing a framework to achieve fairness in ranking. However, these approaches are mostly designed for i.i.d data. This has led to growing research focusing on imparting fairness into node ranking algorithms. Kang et al. [6] presented the `InFoRM` algorithm to mitigate individual biases when computing the PageRank scores of the nodes. However, since their approach did not consider group-level fairness, it may not be able to prevent discrimination against certain underprivileged groups of nodes. Other algorithms for debiasing the output of the PageRank algorithm include [20, 9]. Nevertheless, such algorithms are susceptible to oversmoothing due to their message passing mechanism.

3 Preliminaries

3.1 Problem Statement

Let $G =< V, E, X >$ be an attributed network, where V is the set of nodes, $E \subseteq V \times V$ is the set of links, and X is the set of node attributes. The node attributes $X = (X^{(p)}, X^{(u)})$ are assumed to be a combination of the protected attribute, $X^{(p)}$, and other attributes, $X^{(u)}$. The protected attribute indicates whether the node belongs to a *privileged* ($X^{(p)} = 0$) or *disadvantaged* ($X^{(p)} = 1$) group, as shown by the example given in Figure 1(a). We assume there exists a function $\phi : V \to [0, 1]$ such that $s_v = \phi(X_v)$ is the normalized ***acceptability (utility) score*** of node v. Note that $s_v = 1$ implies the node is highly acceptable whereas $s_v = 0$ implies the node is unacceptable. The choice of ϕ function is domain-dependent and is assumed to be given. We also consider a kernel function, $K : V \times V \to \mathbb{R}_+$, that measures the similarity between nodes in terms of their acceptability scores. We use the following Gaussian radial basis function as our kernel function for nodes u and v:

$$K(u, v) = \exp\left(-\frac{\|s_u - s_v\|^2}{2\sigma^2}\right), \tag{2}$$

where σ is a hyperparameter.

Let $\Pi =< \pi_1, \pi_2, \cdots, \pi_{|V|} >$ be an ordering of the nodes in the network, where each $\pi_i \in V$. We consider node u to be ranked higher than node v, denoted as $u \succ v$, if $u = \pi_i$, $v = \pi_j$, and $i < j$. Furthermore, let $h : V \to [0, 1]$ be a ranking function that maps each node to a value between 0 and 1. We denote $\Pi(G, h)$ as the node ordering of G induced by the function h, where $\forall u, v : h(u) > h(v) \implies u \succ v$. Our goal is to learn the ranking function h that maximizes the acceptability scores $\frac{1}{k} \sum_{i=1}^{k} s_{\pi_i(G,h)}$ subject to the constraints imposed by both individual and group fairness criteria.

3.2 Fairness Measures

Fairness perception was introduced as an individual-level fairness criterion in [12] based on the premise that “the reaction of an individual to the outcome of a decision process is based on the expectation of the individual ... (which) not only depends on one's own outcome but also the outcomes of other individuals that belong to the same reference group." In [12], the reference group that shapes an individual's expectation is defined by the neighborhood of the node in a network.

Definition 1 (Fairness Perception [12]). *Given a prediction function h and a network $G =< V, E, X >$, the fairness perception of a node $v \in V$ is:*

$$f(v,h) = \begin{cases} 1 & \text{if } \mathbb{E}[h(v)] \leq h(v) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbb{E}[h(v)]$ is the neighborhood-based expectation of $h(v)$.

If $f(v,h) = 1$, then node v will perceive the decision as fair since its decision outcome $h(v)$ is no worse than its expectation. Otherwise, v will perceive the decision as unfair. The overall fairness perception for h on G, also known as fairness visibility [12], is given by the average fairness perception of its nodes:

$$f_G(h) = \frac{1}{|V|} \sum_{u \in V} f(v,h) \tag{4}$$

The formula below was used in [12] to calculate the neighborhood-based expectation of a node:

$$\mathbb{E}[h(v)] = \frac{y_v}{k_1} \Big[\sum_{u \in N(v)} y_u h(u) \Big] + \frac{1-y_v}{k_0} \Big[\sum_{u \in N(v)} (1-y_u) h(u) \Big] \tag{5}$$

where $N(v)$ is the set of nodes in the neighborhood of v, $k_0 = \sum_{u \in N(v)} (1-y_u)$, $k_1 = \sum_{u \in N(v)} y_u$, and y_u is the true class label of node u. Note that the measure requires access to the ground truth labels y for computing the neighborhood-based expectation since it is designed for supervised classification task. It is therefore inapplicable to unsupervised node ranking problems, in which the ground truth ranking is unavailable when computing $\mathbb{E}[h(v)]$.

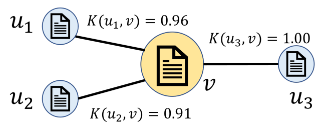
4 Methodology

This section presents our proposed `DeepFairRank` framework. We first introduce our neighborhood expectation function for computing fairness perception.

4.1 Neighborhood Expectation Function

Instead of using Equation (5), we propose the following neighborhood-based conditional expectation $\mathbb{E}[h(v)]$ function:

$$\mathbb{E}[h(v)] = \frac{\sum_{u \in N(v)} h(u) K(v,u)}{\sum_{u \in N(v)} K(v,u)} \tag{6}$$



	v	u_1	u_2	u_3
s	0.6	0.8	0.2	0.6
h	0.6	0.8	0.3	0.7

Fig. 2: An example illustration of fairness perception and calculation of the conditional expectation, $\mathbb{E}[h(v)]$.

Intuitively, the expected value of h for node v is computed based on the weighted average value of its neighbors, where the weights are given by similarity of their acceptability scores s (see Equation (2)). Unlike the approach in [12], h is a continuous-valued ranking function instead of a binary decision function.

Example 1. Consider the example graph shown in Figure 2. The central node v has three neighbors, u_1, u_2, and u_3. Let s_v denotes its normalized acceptability score and $K(v, u_i)$ denotes the similarity of its acceptability score to a neighboring node u_i. The table in the diagram shows the normalized acceptability scores s for all 4 nodes as well as the output of a ranking function h. Using Equation (6), the conditional expectation for node v given its neighbors is:

$$\mathbb{E}[h(v)] = \frac{0.96 \times 0.8 + 0.91 \times 0.3 + 1 \times 0.7}{0.96 + 0.91 + 1} = 0.61.$$

Based on the definition in Equation (3), since $h(v) = 0.6 < \mathbb{E}[h(v)]$, v will perceive the ranking function h as unfair.

4.2 Maximizing Fairness Perception

Let $\mathbb{C} = \{C_1, \ldots, C_k\}$ be the set of connected components in G. The theorem below shows a trivial approach for finding a ranking function h that maximizes fairness perception in G.

Theorem 1. *Given a network $G =< V, E, X >$, the ranking function h maximizes fairness perception, $\forall v \in V: \ f(v, h) = 1$, if and only if $\forall C \in \mathbb{C}: \ h(v) = h(v')$ for all $v, v' \in C$.*

Proof. If h yields the same value for all the nodes in a connected component C, then according to Equation (6), $\forall v \in C: \ \mathbb{E}[h(v)] = h(v)$, and thus, $f(v, h) = 1$. Since this property holds for all $C \in \mathbb{C}$, therefore $\forall v \in V: \ f(v, h) = 1$. Conversely, if $\forall v \in V: \ f(v, h) = 1$, then all the nodes in the same connected component must have the same $h(v)$. This condition is trivially satisfied if the connected component has only one node. Thus, we consider the case when C has at least two nodes. By contradiction, assume that fairness perception is maximized but the nodes in C can have different values of h. Let $C_{min} = \{u \in C | h(u) = m\}$, where $m = min\{h(u) | u \in C\}$. Since the values of h are not uniform in C, there must exist a node $u_m \in C_{min}$ connected to another node $v \in C \setminus C_{min}$, in which $h(u_m) \neq h(v)$. Thus, $\forall v \in N(u_m) : h(u_m) \leq h(v)$ and there exists a neighboring node $v \in N(u_m)$ for which $h(v) > h(u_m)$. Using the definition of neighborhood-based expectation given in Equation (6), we have

$$\mathbb{E}[h(u_m)] = \frac{\sum_{v \in N(u_m)} h(v)K(u_m, v)}{\sum_{v \in N(u_m)} K(u_m, v)} > \frac{\sum_{v \in N(u_m)} h(u_m)K(u_m, v)}{\sum_{v \in N(u_m)} K(u_m, v)} = h(u_m)$$

Since $h(u_m) < \mathbb{E}[h(u_m)]$, $f(u_m, h) = 0$, which contradicts the assumption that the fairness perception of all the nodes in C is maximized. Thus, the original assumption that C has nodes with different values of h must be wrong.

Theorem 1 suggests a trivial way to maximize fairness perception is to assign a constant h to every node in the same connected component. However, as will be shown in the next subsection, maximizing fairness perception alone is insufficient as it does not guarantee high utility of the ranking algorithm since the nodes in the same connected component or the same neighborhood may not have the same acceptability score s. Furthermore, since fairness perception is a local, individual-level fairness criterion instead of a global fairness criterion, it does not guarantee nodes from different groups will be treated equally.

4.3 Weighted Statistical Disparity

One way to ensure equity across the different groups of the protected attribute is to apply a group-level criterion such as statistical parity [4]. However, since the measure was designed for classification problems, we consider a ***weighted statistical disparity (WSD)*** metric to quantify the disparity in the average rank scores h of similarly qualified nodes in different groups of the protected attribute. Specifically, we discretize the normalized acceptability scores $\{s_v \in [0, 1]\}$ into K bins, denoted as $\hat{s}_1 < \hat{s}_2 < \cdots < \hat{s}_K$. For example, the normalized acceptability scores can be discretized into 5 bins, $[0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8]$, and $(0.8, 1]$. Let $\hat{s}_1 = 0.1, \hat{s}_2 = 0.3, \cdots, \hat{s}_5 = 0.9$ be the centroids of the bins, $B_i^{(j)} = \{v \in V \mid \hat{s}_{i-1} < s_v \leq \hat{s}_i, X_v^{(p)} = j\}$ be the set of nodes from the protected group $X^{(p)} = j$ assigned to bin $\hat{s}_i$ and $\bar{h}_i^{(j)} = \frac{1}{|B_i^{(j)}|} \sum_{v \in B_i^{(j)}} h(v)$ be its average value of h. Assuming there are 2 groups, the weighted statistical disparity measure is given by:

$$\Gamma(h) = \sum_{i=1}^{K} \hat{s}_i \left| \bar{h}_i^{(0)} - \bar{h}_i^{(1)} \right| \tag{7}$$

Intuitively, $\Gamma(h)$ ensures that the disparity in rank scores of nodes with high acceptability scores be given higher emphasis than the disparity among nodes with low acceptability scores.

Example 2. Consider the plot shown in Figure 3, where the horizontal axis denotes the normalized acceptability score (s) and the vertical axis denotes the rank score given by some function h. Each data point in the plot represents a node in the graph shown in Figure 1(a). Assume the acceptability scores were discretized into 5 bins. The average value of h in bin i for each group j of the

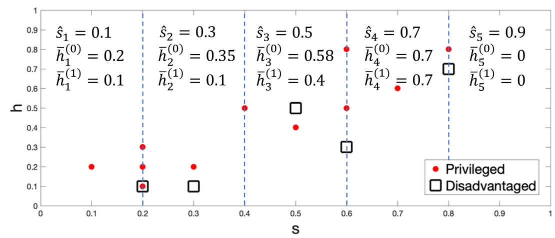


Fig. 3: An illustration of weighted statistical disparity.

protected attribute is denoted as $\bar{h}_i^{(j)}$, respectively. The weighted statistical disparity for the ranking function h can be computed using Equation (7) as follows:

$$\Gamma(h) = 0.1 \times 0.1 + 0.3 \times 0.25 + 0.5 \times 0.18 + \ 0.7 \times 0 + 0.9 \times 0 = 0.175$$

The theorem below shows the relationship between fairness perception and the weighted statistical disparity measure.

Theorem 2. *Given a ranking function h that maximizes fairness perception, i.e., $\forall v \in V: \ f(v,h) = 1$, its weighted statistical disparity satisfies the following inequality:*

$$\kappa \times max\Big\{\delta^*, 0\Big\} \leq \Gamma(h) \leq \kappa \times \Delta^*,$$

where $\kappa := \sum_{i=1}^{K} \hat{s}_i$,

$$\delta^* = sign\Big((h_{\min}^{(0)} - h_{\max}^{(1)})(h_{\max}^{(0)} - h_{\min}^{(1)})\Big) \times \min\left\{ \left|h_{\min}^{(0)} - h_{\max}^{(1)}\right|, \left|h_{\max}^{(0)} - h_{\min}^{(1)}\right| \right\},$$

$$\Delta^* = \max\left\{ \left|h_{\min}^{(0)} - h_{\max}^{(1)}\right|, \left|h_{\max}^{(0)} - h_{\min}^{(1)}\right| \right\}.$$

Here, we denote $h_{\min}^{(j)} = \min\{h(v) \mid X_v^{(p)} = j\}$ *and* $h_{\max}^{(j)} = \max\{h(v) \mid X_v^{(p)} = j\}$.

Proof. First, note that $h_{\min}^{(j)} \leq \bar{h}_i^{(j)} \leq h_{\max}^{(j)}$ since

$$\bar{h}_i^{(j)} = \frac{1}{|B_i^{(j)}|} \sum_{v \in B_i^{(j)}} h(v) \leq \frac{1}{|B_i^{(j)}|} \sum_{v \in B_i^{(j)}} h_{\max}^{(j)} = h_{\max}^{(j)}$$

$$\bar{h}_i^{(j)} = \frac{1}{|B_i^{(j)}|} \sum_{v \in B_i^{(j)}} h(v) \geq \frac{1}{|B_i^{(j)}|} \sum_{v \in B_i^{(j)}} h_{\min}^{(j)} = h_{\min}^{(j)}$$

Using these inequalities, it can be easily shown that

$$h_{\min}^{(0)} - h_{\max}^{(1)} \leq \bar{h}_i^{(0)} - \bar{h}_i^{(1)} \leq h_{\max}^{(0)} - h_{\min}^{(1)}$$

$$\text{and} \left|\bar{h}_i^{(0)} - \bar{h}_i^{(1)}\right| \leq \max\left\{ \left|h_{\min}^{(0)} - h_{\max}^{(1)}\right|, \left|h_{\max}^{(0)} - h_{\min}^{(1)}\right| \right\}$$

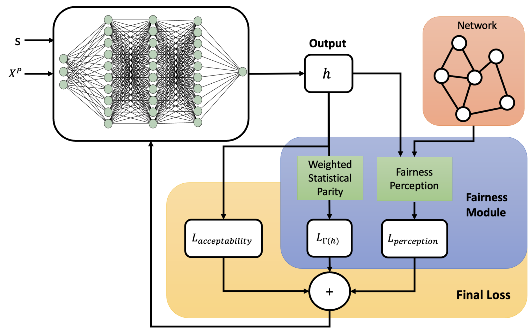


Fig. 4: Overall Framework of `DeepFairRank`.

Replacing the above inequality into Equation (7) and using the definition of Δ^*, we obtain the following upper bound:

$$\Gamma(h) \leq \sum_i \hat{s}_i \max\left\{ \left| h^{(0)}_{\min} - h^{(1)}_{\max} \right|, \left| h^{(0)}_{\max} - h^{(1)}_{\min} \right| \right\} = \kappa \times \Delta^*$$

To obtain the lower bound for $\Gamma(h)$, we use $\left| \bar{h}^{(0)}_i - \bar{h}^{(1)}_i \right| \geq \delta^*$ where

$$\delta^* = \text{sign}\Big((h^{(0)}_{\min} - h^{(1)}_{\max})(h^{(0)}_{\max} - h^{(1)}_{\min}) \Big) \times \min\left\{ \left| h^{(0)}_{\min} - h^{(1)}_{\max} \right|, \left| h^{(0)}_{\max} - h^{(1)}_{\min} \right| \right\}$$

If $(h^{(0)}_{\min} - h^{(1)}_{\max})$ and $(h^{(0)}_{\max} - h^{(1)}_{\min})$ have opposite signs, then δ^* is negative. Since $\left| \bar{h}^{(0)}_i - \bar{h}^{(1)}_i \right|$ is non-negative, we can obtain a tighter bound as follows: $\left| \bar{h}^{(0)}_i - \bar{h}^{(1)}_i \right| \geq \max\left\{ \delta^*, 0 \right\}$. Replacing this into Equation (7) yields

$$\Gamma(h) \geq \frac{1}{K} \sum_i \hat{s}_i \max\left\{ \delta^*, 0 \right\} = \kappa \times \max\left\{ \delta^*, 0 \right\}$$

Theorem 2 suggests that maximizing fairness perception alone may not guarantee group fairness since the upper bound of $\Gamma(h)$ can be small depending on how the nodes from different groups are distributed in the network. Thus, it would be desirable to find an algorithm that considers the trade-off between maximizing fairness perception and minimizing weighted statistical disparity.

4.4 `DeepFairRank`: Proposed Framework

This section presents `DeepFairRank`, our proposed neural network approach to learn a ranking function h that considers (1) the normalized acceptability scores of the nodes, s, and the trade-off between (2) maximizing fairness perception and (3) minimizing weighted statistical disparity measure, $\Gamma(h)$.

Figure 4 depicts the architecture of `DeepFairRank`, which consists of a stack of fully connected neural networks with an input layer, three hidden layers and an output layer. Each hidden layer contains 256 hidden neurons. The model employs a `tanh` activation function in all layers except for its output layer, which uses a `sigmoid` function to restrict the output to be a value between 0 and 1. The network output will be provided as input to a fairness perception module that computes the fairness perception loss associated with the ranking function h.

To illustrate how the fairness perception loss is computed, we first express the neighborhood-based expectation function (see Equation (6)) in matrix notation as follows:

$$\mathbb{E}[h(v_i)] = \frac{\sum_{v_j \in N(v_i)} h(v_j) K(v_i, v_j)}{\sum_{v_j \in N(u_i)} K(v_i, v_j)} = \sum_j W_{ij} h(v_j), \tag{8}$$

where $W = D^{-1}(K \odot A)$, $\odot$ denotes a Hadamard product between the adjacency matrix A and kernel matrix K, while D is a diagonal matrix with $D_{ii} = \sum_j K_{ij} A_{ij}$. Given the output h of the fully connected network, the following fairness perception loss is computed:

$$\ell_{\mathrm{FP}}(h) = \sum_i \big(\mathbb{E}[h(v_i)] - h(v_i)\big) = \sum_i \big[\sum_j W_{ij} h(v_j) - h(v_i)\big]$$

which can be simplified as $\ell_{\mathrm{FP}} = \|(\mathbf{W} - \mathbf{I})h\|_1$, where $\mathbf{I}$ is the identity matrix. The fairness perception module will compute the loss, which will be combined with the weighted statistical disparity to learn the ranking function h. Additionally, h should be close to the acceptability score s to ensure that nodes with high acceptable scores are ranked higher than those with low scores. Putting everything together, the objective function to be optimized by our framework is:

$$\min_h \|s - h\|^2 + \alpha \Gamma(h) + \beta \ell_{\mathrm{FP}} \tag{9}$$

where the hyperparameters α and β control the trade-off between matching h to the node acceptability scores, ensuring group parity, and maximizing fairness perception. The model was trained end-to-end for 1000 epochs using Adam as its optimizer with a learning rate of 0.0001.

During training, the hyperparameter values for α and β are chosen to minimize the following power mean:

$$\alpha^*, \beta^* = \mathrm{argmax}_{\alpha,\beta} \left(\frac{r(h,s)^3 + (1 - \Gamma(h))^3 + f_G(h)^3}{3} \right)^{\frac{1}{3}}, \tag{10}$$

where $r(h, s)$ is the Spearman rank coefficient between the ranked output h and normalized acceptability score s, $\Gamma(h)$ is the weighted statistical disparity, and $f_G(h)$ is the average fairness perception. We randomly choose $\alpha \in [10^{-2}, 10^2]$ and $\beta \in [10^{-2}, 10^2]$ in logarithmic space for our experiments.

5 Experimental Evaluation

This section describes the experiments performed to evaluate the performance of `DeepFairRank`. Our code and data are available at `https://github.com/frsantosp/DeepFairRank`.

Table 1: Summary statistics of submitted and accepted papers for the ICLR conference from 2017 to 2020.

Year	# submitted papers	# accepted papers	% papers by famous authors	% accepted papers by famous authors
2017	488	243	21.7%	28.4%
2018	402	229	17.2%	20.1%
2019	1419	502	14.5%	19.7%
2020	2212	687	12.3%	15.6%

5.1 Data

We consider the following datasets for our experiments.

1. **Co-authorship network** (ICLR): This dataset [12] corresponds to the co-authorship network of papers submitted to the ICLR conference from 2017 to 2020. The nodes of the co-authorship network correspond to submitted papers, while an edge is created if two papers share a co-author. Table 1 shows the summary statistics of the submitted and accepted papers from 2017 to 2020. Similar to the approach used in [12], the submitted papers were categorized into 2 groups using *famous author* as the protected attribute. The acceptance decision of the conference is used as the true label during evaluation.
2. **COMPAS**: The COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) [17], which is a widely used benchmark for evaluating fairness algorithms. Each node corresponds to a jailed offender. An edge is created between two nodes if the time in jail overlaps between two offenders. We use race as the protected attribute. The resulting network contains 8,946 nodes and 2,440,315 edges. We use the COMPAS decile score as acceptability score and the recidivism attribute as the true label during evaluation.
3. **Credit Default** [21]: The dataset contains 2,000 nodes using marital status as the protected attribute. The links in the network are established based on similarity of their spending and payment patterns. Each node (individual) has a binary class indicating whether the individual will default on their credit card payment in the next month.
4. **Facebook** [10]: The dataset is a social network from Facebook. Each node represents a facebook user and the link represents friendship. The protected attribute used is gender. The binary label is based if the user went to college. The dataset contains 1046 nodes with $53,000$ edges.

5.2 Experimental Setup

Baseline Algorithms We compare the performance of `DeepFairRank` (DFR) against the following baselines:

1. **No Calibration**, in which the nodes are ranked according to their normalized acceptability scores, i.e., $\mathbf{h} = \mathbf{s}$.

2. **Fa*ir** [22], a fair ranking algorithm that selects the top-k candidates to recommend based on a ranked group fairness criterion. We use the python implementation[1] provided by the authors for our experiment.
3. **InFoRM** [6], which debiases the normalized acceptability score $\mathbf{s}$ by minimizing the following loss:

$$\min_{\mathbf{h}} \|\mathbf{h} - \mathbf{s}\|^2 + \alpha \text{tr}\left(\mathbf{h}^T \mathbf{L}_s \mathbf{h}\right),$$

   where $\mathbf{L}_s$ is the graph Laplacian matrix. As shown in [6], the optimization problem is equivalent to the PageRank formulation (Equation (1)) if $\mathbf{s}$ corresponds to a uniform probability vector $\mathbf{e}$ (with $\mathbf{L}_s$ equals to the identity matrix). We use the code provided by the authors[2] for our experiment.
4. **FSPR** [20], which is a fairness-aware PageRank algorithm that considers the group fairness criterion. The algorithm is trained to solve the following constraint optimization problem:

$$\begin{aligned} \min_{\mathbf{h}} \quad & \|\mathbf{Q}^T \mathbf{h} - \mathbf{s}\|^2 \\ \text{s.t.} \quad & \mathbf{Q}_p^T \mathbf{h} = \phi, \sum_i h_i = 1 \\ & 0 \leq h_i \leq 1, \forall i \in \{1, 2, \cdots, n\} \end{aligned}$$

   where $\mathbf{Q} = \gamma \left[1 - (1-\gamma)\mathbf{P}\right]^{-1}$, $\mathbf{Q}_p$ is the PageRank mass allocated to the protected group, ϕ is the desired proportion for achieving equity, and $\gamma = 0.15$ is the parameter associated with the jump probability of the PageRank algorithm. The preceding convex optimization problem is solved using the CVXOPT software package.

Evaluation Metrics We compared the performance of the fairness-aware algorithms according to 4 criteria. First, to demonstrate their utility, the *average precision* of each algorithm is computed from their respective ranked output as follows: $\frac{1}{m}\sum_k P@k \times I[r_k]$, where $P@k$ is the precision computed from the top-k highest ranked nodes, $I[r_k]$ is a binary indicator function whose value is 1 only if the i-th ranked node should be accepted according to the ground truth label, and m is the total number of "relevant" nodes. We used the python implementation of average precision from the scikit-learn library to compute the measure. Furthermore, we also compute the *average acceptability scores* of the selected top-k ranked nodes, to ensure that the highly ranked nodes are well-qualified to be selected. Third, to assess their individual-level fairness, we computed their *average fairness perception*, which is given by $\frac{1}{|V|}\sum_{v \in V} f(v, h)$. The metric ranges between 0 and 1, with larger values suggest a higher perception that the ranking is fair among the nodes. Finally, to ensure group fairness, we employed the weighted statistical disparity measure given in Equation (7).

[1] https://github.com/fair-search/fairsearch-fair-python
[2] https://github.com/jiank2/inform

Data	ICLR_{17}	ICLR_{18}	ICLR_{19}	ICLR_{20}	COMPAS	Credit	Facebook	Avg Rank
No Calib	0.9257	0.8480	0.9160	0.8917	0.4911	0.8706	0.8476	1.86
Fa*ir	0.8947	0.8212	0.8630	0.8449	0.3918	0.8581	0.7666	3.29
InFoRM	0.7044	0.6991	0.5175	0.3689	0.4097	0.4152	0.3258	4.85
FSPR	0.7760	0.7745	0.7270	0.7157	0.4648	0.8206	0.7279	3.86
DFR	0.9319	0.8641	0.9269	0.9111	0.5063	0.8668	0.8477	1.14

(a) Average Precision (↗)

No Calib	0.6790	0.6549	0.6674	0.6414	0.7923	0.8275	0.7850	1.00
Fa*ir	0.6746	0.6540	0.6656	0.6413	0.5638	0.8255	0.7849	3.00
InFoRM	0.6044	0.5878	0.5810	0.4727	0.6727	0.7770	0.3687	4.71
FSPR	0.6123	0.5878	0.6244	0.6030	0.7361	0.8100	0.6910	3.85
DFR	0.6773	0.6534	0.6673	0.6414	0.7923	0.8275	0.7849	1.71

(b) Average Acceptability Score (↗)

No Calib	0.7377	0.7537	0.6660	0.5967	0.5872	0.6046	0.4057	2.71
Fa*ir	0.7848	0.7587	0.6596	0.4910	0.4181	0.8140	0.3818	3.00
InFoRM	0.6885	0.6915	0.5441	0.3513	0.4315	0.3379	0.2842	4.71
FSPR	0.6803	0.7040	0.5849	0.6700	0.4735	0.8635	0.3722	3.14
DFR	0.7540	0.7611	0.6821	0.6103	0.5989	0.8170	0.4201	1.43

(c) Average Fairness Perception (↗)

No Calib	1.0397	0.8374	0.0610	0.0108	0.0198	0.00004	0.000093	3.86
Fa*ir	1.4751	1.1510	0.3580	0.1527	0.6084	0.0077	0.0002	4.86
InFoRM	0.0069	0.0055	0.0009	0.0002	0.00001	0.0000	0.0000008	1.14
FSPR	0.3528	0.0941	0.0027	0.0084	0.0023	0.00001	0.00064	3
DFR	0.0043	0.0055	0.0024	0.0028	0.0104	0.0075	0.000002	2.14

(d) Weighted Statistical Disparity (↘)

Table 2: Performance comparison of `DeepFairRank` against baseline algorithms. ↗ or ↘ indicates larger or smaller values are better. The red color represents first place and blue represents second place. The last row in each table denotes average rank of each algorithm over all datasets.

5.3 Experimental Results

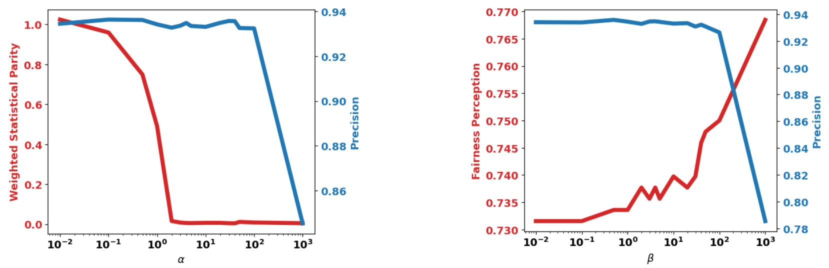
We first compare the average precision of `DeepFairRank` against the ranking results generated by other baseline methods. The results are summarized in Table 2(a). Observe that `DeepFairRank` achieves the highest average precision on 6 out of the 7 datasets. The next best result is obtained by the **No Calibration** method, which is not surprising since there is strong correlation between acceptability score and the true class label of the nodes in the network. The **Fa*ir** algorithm [22] performs relatively well on the ICLR and credit datasets but worse on the COMPAS dataset whereas **FSPR** performs well on COMPAS but struggles with the other datasets. **InFoRM** has the lowest average precision values among all the methods on the ICLR datasets. This is likely due to the over-smoothing effect of the message passing schemes used by the algorithms.

Table 2(b) shows the average acceptability scores of the top-k nodes selected by each algorithm. As expected, the **No Calibration** method is the best performer since it is designed to rank the nodes according to the highest normalized acceptability scores. `DeepFairRank` is the next best performer, achieving acceptability scores that are comparable to **No Calibration** in all the datasets. As its average precision is also the highest, this shows that the nodes chosen by `DeepFairRank` not only have among the highest acceptability scores, their top-k rankings are also consistent with the true class labels despite the unsupervised learning nature of the algorithm.

In terms of their average fairness perception, the results shown in Table 2(c) suggest that `DeepFairRank` has the best average fairness perception in 4 out of the 7 datasets evaluated. This is not surprising as `DeepFairRank` is trained to optimize fairness perception as one of its criteria while accounting for its trade-off with other metrics. The next best performer is **No Calibration** followed by the **Fa*ir** and**FSPR** algorithms. Note that **FSPR** achieves the highest average fairness perception for the ICLR_{20} and the credit dataset. `DeepFairRank` followed in those two datasets.

Finally we compare the performance of the various algorithms in terms of their group fairness criterion. The results in Table 2(d) suggest that **InFoRM** has the best average ranking with respect to the weighted statistical disparity, which is to be expected since the method is designed to explicitly optimize the group fairness criterion only. The next best performer is `DeepFairRank`, which has the best weighted statistical disparity measure in 2 datasets and second place in 3 datasets. While **InFoRM** excels at maintaining group fairness, it performs poorly on all the other metrics including average fairness perception and precision. These results suggest that **InFoRM** has difficulty managing the trade-off between utility and fairness compared to other metrics. Table 2(d) also shows that **Fa*ir** performs worse than **No Calibration** in terms of weighted statistical parity. This is due to the difficulty in tuning the hyperparameters of the algorithm (k, p, and α) using the code provided by the authors. In particular, the code has a tendency of breaking down and returning warning messages indicating the code library has not been tested outside the range of values used.

In summary, these results demonstrate the effectiveness of using the proposed multi-objective framework in `DeepFairRank` to achieve high average precision in its node rankings while balancing the trade-off between maximizing the individual-level fairness perception measure and minimizing the group-level weighted statistical disparity measure for the different groups. The framework gives flexibility for users to tune the algorithm towards achieving their desired utility and fairness goals for their application domain by choosing the appropriate values of the hyperparameters α and β. For example, increasing α will promote higher group-level fairness, as illustrated in Figure 5a, since the hyperparameter promotes lower weighted statistical disparity in the loss function of `DeepFairRank` when α increases. However, Figure 5b shows that selecting a higher β can hurt both precision and fairness perception as it will unbalance the loss. Nevertheless, it is important to note that the algorithm maintains stable



(a) Varying α with $\beta = 1$.

(b) Varying β with $\alpha = 1$.

Fig. 5: Effects of varying the hyperparameters α and β of the `DeepFairRank` algorithm on average precision and weighted statistical disparity (for varying α) or fairness perception (for varying β) when applied to the $ICLR_{17}$ dataset.

performance on a relatively wide range of hyperparameter values, showing that algorithm is relatively easy to tune.

6 Conclusions

This paper presents a fairness-aware node ranking algorithm called `DeepFairRank` that considers a multi-objective criteria based on acceptability score, fairness perception, and weighted statistical disparity between different groups of the protected attribute. We introduce a novel conditional expectation measure for fairness perception that is designed for node ranking problems. We also provide theoretical analysis to show that maximizing fairness perception alone is insufficient as it may lead to bias in terms of the group fairness criteria. Finally, we empirically demonstrate the effectiveness of `DeepFairRank` in terms of balancing the three competing requirements unlike other fairness-aware algorithms.

Acknowledgment

This material is based upon work supported by NSF under grant #IIS-1939368 and #IIS-2006633. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Adams, J.S.: Inequity in social exchange. Advances in Experimental Social Psychology, vol. 2, pp. 267–299. Academic Press (1965)
2. Asudeh, A., Jagadish, H., Stoyanovich, J., Das, G.: Designing fair ranking schemes. In: Proc of Int'l Conf on Management of Data. pp. 1259–1276 (2019)
3. Das, K., Samanta, S., Pal, M.: Study on centrality measures in social networks: a survey. Social Network Analysis and Mining **8**(1), 1–11 (2018)

4. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proc of 3rd Innovations in Theoretical Comp Sci conf. pp. 214–226 (2012)
5. Hardt, M., Price, E., Srebro, N., et al.: Equality of opportunity in supervised learning. In: NeurIPS. pp. 3315–3323 (2016)
6. Kang, J., He, J., Maciejewski, R., Tong, H.: Inform: Individual fairness on graph mining. In: Proc of ACM SIGKDD Int'l Conf on Knowl Disc & Data Mining. pp. 379–389 (2020)
7. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
8. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proc. of ACM SIGKDD Int'l Conf on Knowl Disc & Data Mining. pp. 137–146 (2003)
9. Krasanakis, E., Papadopoulos, S., Kompatsiaris, I.: Applying fairness constraints on graph node ranks under personalization bias. In: Proc of Int'l Conf on Complex Networks and their Applications. pp. 610–622 (2021)
10. Leskovec, J., Mcauley, J.J.: Learning to discover social circles in ego networks. In: NeurIPS. pp. 539–547 (2012)
11. Li, Y., Fan, J., Wang, Y., Tan, K.L.: Influence maximization on social graphs: A survey. IEEE Trans. on Knowl. and Data Engr. **30**(10), 1852–1872 (2018)
12. Masrour, F., Tan, P.N., Esfahanian, A.: Fairness perception from a network-centric perspective. In: Proc of the IEEE Int'l Conf on Data Mining (2020)
13. Masrour, F., Wilson, T., Yan, H., Tan, P.N., Esfahanian, A.: Bursting the filter bubble: Fairness-aware network link prediction. In: Proc. of the AAAI Conf on Artificial Intelligence. vol. 34, pp. 841–848 (2020)
14. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: Survey on bias and fairness in machine learning. ACM Comp. Surv. **54**(6), 115:1–115:35 (2022)
15. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the Web. Tech. rep., Stanford InfoLab (1999)
16. Patro, G.K., Porcaro, L., Mitchell, L., Zhang, Q., Zehlike, M., Garg, N.: Fair ranking: a critical review, challenges, and future directions pp. 1929–1942 (2022)
17. ProPublica: Compas recidivism risk score data and analysis (2016), data retrieved from ProPublica Data Store, `https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis`
18. Singh, A., Joachims, T.: Fairness of exposure in rankings. In: Proc of ACM SIGKDD Int'l Conf on Knowl Disc & Data Mining. pp. 2219–2228 (2018)
19. Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., Tsaparas, P.: Link Recommendations for PageRank Fairness. In: Proc of ACM Web Conf. pp. 3541–3551 (2022)
20. Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., Mamoulis, N.: Fairness-aware PageRank. In: Proc of the Web Conf 2021. pp. 3815–3826 (2021)
21. Yeh, I.C., Lien, C.h.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Sys. with Appl. **36**(2), 2473–2480 (2009)
22. Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., Baeza-Yates, R.: Fa* ir: A fair top-k ranking algorithm. In: Proc of the 2017 ACM on Conf on Information and Knowledge Management. pp. 1569–1578 (2017)
23. Zehlike, M., Sühr, T., Baeza-Yates, R., Bonchi, F., Castillo, C., Hajian, S.: Fair top-k ranking with multiple protected groups. Information Processing & Management (2022)
24. Zehlike, M., Yang, K., Stoyanovich, J.: Fairness in ranking, part i: Score-based ranking. ACM Comp. Surv. **55**(6), 1–36 (2022)