LambdaFair for Fair and Effective Ranking

Federico Marcuzzi[⋆†][0000-0002-8141-8294], Claudio Lucchese[0000-0002-2545-0425], and Salvatore Orlando[0000-0002-4155-9797]

Università Ca' Foscari Venezia, Venice, Italy
{federico.marcuzzi,claudio.lucchese,orlando}@unive.it

Abstract. Traditional machine learning algorithms are known to amplify bias in data or introduce new biases during the learning process, often resulting in discriminatory outcomes that impact individuals from marginalized or underrepresented groups. In information retrieval, one application of machine learning is learning-to-rank frameworks, typically employed to reorder items based on their relevance to user interests. This focus on effectiveness can lead to rankings that unevenly distribute exposure among groups, affecting their visibility to the final user. Consequently, ensuring fair treatment of protected groups has become a pivotal challenge in information retrieval to prevent discrimination, alongside the need to maximize ranking effectiveness. This work introduces LambdaFair, a novel in-processing method designed to jointly optimize effectiveness and fairness ranking metrics. LambdaFair builds upon the LambdaMART algorithm, harnessing its ability to train highly effective models through additive ensembles of decision trees while integrating fairness awareness. We evaluate LambdaFair on three publicly available datasets, comparing its performance with state-of-the-art learning algorithms in terms of both fairness and effectiveness. Our experiments demonstrate that, on average, LambdaFair achieves 6.7% higher effectiveness and only 0.4% lower fairness compared to state-of-the-art fairness-oriented learning algorithms. This highlights LambdaFair's ability to improve fairness without sacrificing the model's effectiveness.

Keywords: Learning to Rank · Fairness · Effectiveness

1 Introduction

Nowadays, machine learning (ML) is widely used in information retrieval (IR), mainly to learn effective ranking systems that provide users with relevant and possibly personalized content. In particular, learning to rank (LTR) involves applying ML techniques to build the ranking models used in IR systems. However, traditional ML algorithms, including those based on LTR, can introduce

⋆ Corresponding author.
† Currently employed at INSAIT, Sofia University "St. Kliment Ohridski", Bulgaria, and part of the Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt, Germany.

bias during the model's learning phase or perpetuate bias in the training data, leading to unequal treatment across individuals or groups.

This unwanted facet of ML has led researchers and practitioners to develop solutions to tackle this issue. In IR, various pre/in/post-processing methods have been designed to minimize this phenomenon and guarantee accurate and unbiased results. Fairness in IR aims to provide equally relevant results to different users, regardless of their group membership, ensuring a fair distribution of individuals from different groups and ranking similar individuals equitably [35, 36].

In this work, we introduce LambdaFair, a novel LTR algorithm based on LambdaMART [8], designed to train fairness-aware ranking models by jointly optimizing fairness and effectiveness metrics. LambdaFair extends LambdaMART's objective function to include fairness constraints; for this reason, LambdaFair can be categorized as an in-processing method. We designed LambdaFair along with three different strategies for training ranking models that optimize the effectiveness metric NDCG [15] and the fairness metric rND [31]. While NDCG optimization aims to rank the most relevant document at the top of a ranking, rND optimization seeks to maximize the statistical parity of items belonging to different groups. By integrating rND-based fairness constraints into the NDCG optimization process, LambdaFair balances ranking effectiveness and fairness, thus addressing the dual objectives of accuracy and equity in ranking tasks.

The contribution of this work is threefold: *i)* we developed LambdaFair, an in-processing method to train fairness-aware ranking models. Such models are a fundamental component to enhance the development of IR systems that promote inclusivity and equity; *ii)* we empirically demonstrated on three publicly available real-world datasets that LambdaFair improves ranking fairness while maintaining competitive effectiveness compared to state-of-the-art methods; *iii)* we highlighted the potential of tree-based and feature-engineered models, such as LambdaMART, for fairness-aware LTR. Such algorithms are also well known for their efficiency in the training and inference stages.

2 Related Work

Providing users with fair and accurate results is a critical challenge in IR. Fairness in ranking systems can encompass different facets. For instance, it is essential to present users with similar interests with comparable content without subjecting them to discrimination based on group membership biases. Furthermore, items belonging to so-called protected groups should not be disproportionately disadvantaged in visibility due to underlying biases in the ranking process.

Substantial efforts have been made in IR to minimize biases in training data or those introduced by machine learning algorithms. The proposed solutions are generally classified into three primary categories [35, 36]: pre-processing [11, 17, 37], in-processing [2, 13, 20, 21, 28, 33], and post-processing [3, 27, 32, 34] methods.

Pre-processing methods aim to mitigate bias in data before the model is trained. These methods mainly address individual fairness and are agnostic to group membership. Lahoti *et al.* [17], designed iFair, an approach to generalize

the items' feature vectors into a fairer representation, following the fairness definition that similar individuals should be treated similarly [11]. Zemel *et al.* [37], proposed LFR, a strategy to learn a fair representation that balances two goals: accurately representing the data to provide effective outcomes and hiding information about protected group membership to optimize fairness.

In-processing methods incorporate notions of fairness directly into the learning process by adding fairness constraints or optimizing fairness metrics. The DELTR [33] algorithm is an LTR framework designed to address group discrimination in rankings. It treats unfairness as disparities in group items' exposure along the ranked list. DELTR learns a re-ranking model that adjusts the output of existing rankers to preserve effectiveness while reducing inequality and enhancing fairness. Similarly, Fair-PG-Rank [28] optimizes fairness by modeling exposure as expected attention. It operates under a merit-based constraint, learning rankers that ensure items receive exposure proportional to their relevance, thereby balancing fairness and effectiveness in ranking. Recent studies identified stochastic Plackett-Luce (PL) ranking models [18, 22] as a robust in-processing solution for optimizing effectiveness and fairness ranking metrics. In contrast to deterministic algorithms, which rely on heuristic optimization methods, PL models are entirely differentiable and adaptable to ranking metric optimization via stochastic gradient descent. Yet, in real-world scenarios, gradient estimation becomes impractical because it requires considering all possible item permutations. To overcome this limitation, Oosterhuis [20], proposed PL-Rank to efficiently estimate the gradient of PL models with respect to both effectiveness and fairness metrics by exploiting specific properties inherent to ranking metrics and PL models. Moreover, the PL-Rank-3 algorithm, introduced by Oosterhuis [21], achieves unbiased gradient estimates while maintaining computational efficiency on par with the most advanced sorting algorithms. Following these works, Gorantla *et al.* [13], introduced Group-Fair-PL to optimize group fairness. This approach features a novel objective where expected ranking utility is computed over only those rankings that adhere to specified group representation constraints. Group-Fair-PL is constructed to ensure equal or proportional representation of protected items within the top-k ranks. Equal representation requires an equal number of items across groups at the top-k, whereas proportional representation aligns item distribution with group proportions within the dataset. In this work, we assess the performance of LambdaFair by comparing it against the PL-Rank-3 and Group-Fair-PL baselines.

Post-processing methods aim to mitigate bias by re-ranking the model's output to ensure fairness. These methods do not alter the original ranking algorithm but instead adjust the ranked results after they are generated, ensuring that fairness criteria are met while maintaining the effectiveness of the original model. FA*IR, introduced by Zehlike *et al.* [32], adjusts the model's output to maintain a minimum percentage of protected candidates at each position in the ranking. Zehlike *et al.* [34], presented the Continuous Fairness Algorithm (CFAΘ), which allows for a continuous adjustment between two contrasting fairness definitions: individual and group fairness.

3 Background

Learning-to-rank techniques are widely used in information retrieval to re-rank documents. Given a query q and a set of candidate documents $D = \{d_1, \ldots, d_n\}$, the goal of LTR algorithms is to train a ranker that produces a ranking π over the set of documents. To produce π, the learned ranker assigns a score s_i to each document $d_i \in D$, then sorts the documents in descending score order. Consequently, π is a permutation of D, where $\pi[r] = i$ indicates that document d_i is ranked at position $r \in \mathbb{N}^+$, $1 \leq r \leq n$. Moreover, let $\pi[r_l, r_u]$, $r_l \leq r_u$, be a discrete interval of the ranking π containing the documents ranked between position r_l and r_u. A prefix of ranking π is thus denoted by $\pi[1, r_u]$. Finally, $d_i \prec_\pi d_j$ denotes that d_i precedes d_j in ranking π.

3.1 Evaluation Metrics

To fully understand the main contribution of this work, we need first to introduce the effectiveness and fairness metrics employed in our study.

The *effectiveness* metric used is *Normalized Discounted Cumulative Gain* (NDCG) [15], a well-known and widely used IR metric for assessing the quality of a ranked list. Given a ranking π of a (candidate) document set D, $|D| = n$, produced in response to a query q, NDCG assesses the quality of π by exploiting the *relevance* judgment y_i of each document $d_i \in D$. NDCG is defined as follows:

$$\text{NDCG} = \frac{\text{DCG}}{\text{IDCG}} = \frac{1}{\text{IDCG}} \sum_{r=1}^{n} \frac{\text{G}_r}{\text{D}_r} = \frac{1}{\text{IDCG}} \sum_{r=1}^{n} \frac{2^{y_{\pi[r]}} - 1}{\log_2(r+1)} \ .$$

NDCG is normalized by IDCG, the ideal DCG of the ground-truth ranking to bound it in $[0, 1]$, where higher is better. Through a logarithmic-discount approach, NDCG rewards rankings where the most relevant items appear in the top positions in π. The reason for this is to align with user behavior, as users often only examine the first results and pay little attention to the subsequent ones. Finally, NDCG is usually computed at a fixed cutoff k, denoted by NDCG@k.

The *fairness* metric used in this study is *Normalized Discounted Difference* (rND) [31], which measures group fairness in terms of statistical parity [35]. Specifically, rND considers π fair if the top-r ranked positions have a proportion of protected documents equal to $|\mathcal{G}^+|/n$, for different values of r, where $\mathcal{G}^+ \subseteq D$ denotes the set of documents belonging to the protected group. In more detail, the rND metric splits the ranked list of documents into *overlapping prefixes* of length $r \in \{b, 2b, 3b, \ldots\}$, where b, $b > 1$, is named *bin size*. The difference from the ideal proportion of protected documents is computed for each length prefix r and averaged with a discounting mechanism similar to that of NDCG. The rND metric measures whether each prefix is representative of the entire ranking [24]. More formally, rND is defined as follows:

$$\text{rND} = \frac{\text{rD}}{\text{rDmax}} = \frac{1}{\text{rDmax}} \sum_{r=b,2b,\ldots}^{n} \frac{1}{\log_2(r)} \left| \frac{|\mathcal{G}^+_{\pi[1,r]}|}{r} - \frac{|\mathcal{G}^+|}{n} \right| \ ,$$

where $\mathcal{G}^{+}_{\pi[1,r]}$ is the set of protected documents in the top-r positions. The lower the rND value, the higher the fairness. The rND metric is normalized in the range $[0, 1]$ by dividing rD by rDmax. Given the groups of protected $\mathcal{G}^{+}$ and unprotected $\mathcal{G}^{-}$ individuals, rDmax is rD computed when all items from the group with the smaller cardinality are ranked in the top positions. Finally, like NDCG, rND can be computed up to a fixed cutoff k, denoted by rND@k.

Note that two factors promote fairness in the top positions of the ranking. First, the discounting, and second, the use of overlapping prefixes: for example, the top-b positions contribute to the fairness of all the $\lceil n/b \rceil$ prefixes considered.

Finally, we assess the reasons why we chose rND as the fairness metric in our LambdaMART-based algorithm. *i)* Like NDCG, rND is defined within the $[0, 1]$ interval, facilitating a straightforward trade-off between fairness and effectiveness during optimization. *ii)* It is designed to evaluate fairness in deterministic rankings, making it an ideal choice for evaluation LambdaMART-based algorithms, where we focus on mitigating bias without dynamic changes to the output rankings. *iii)* The formal definition of rND allows us to efficiently compute LambdaMART's gradient. This efficiency directly contributes to the scalability and practical applicability of our approach. *iv)* Like NDCG, rND adopts a logarithmic discount to reward rankings that promote statistical parity at the top positions, where visibility and exposure to users are critical.

Note that, the LambdaFair framework supports various fairness metrics, but their applicability must be assessed carefully. Recent works on bias in document ranking used AWRF [26], NFaiRR [25], and TExFAIR [1]. These metrics rely on term-based group membership, where a document's group is determined by a continuous vector based on its text. This contrasts with our setting, where we use hard-label membership, with each document belonging to exactly one group.

3.2 LambdaMART

LambdaMART [8] is a learning algorithm widely used in IR to efficiently learn effective ranking models by optimizing a given ranking metric. LambdaMART overcomes the problem of non-differentiability and flatness of ranking metrics [5, 6, 10, 19] by exploiting a smooth approximation of the gradient. Formally, given a query q, the candidate documents $D = \{d_1, \ldots, d_n\}$, and their relevance labels $Y = \{y_1, \ldots, y_n\}$, a ground-truth pairwise preference set P is created. Specifically, $(i, j) \in P$ *iff* document d_i is to be ranked higher than d_j for the query q in the training set, i.e., $y_i > y_j$. For each query-document pair (q, d_i), LambdaMART learns how to score d_i by computing its gradient $\lambda(P)^{\mathrm{Z}}_i$ as follows:

$$\lambda(P)^{\mathrm{Z}}_i = \sum_{j:(i,j)\in P} \lambda^{\mathrm{Z}}_{ij} - \sum_{k:(k,i)\in P} \lambda^{\mathrm{Z}}_{ki} \ , \tag{1}$$

where Z is the ranking metric being optimized, e.g, NDCG@k. Each partial contribution $\lambda^{\mathrm{Z}}_{ij}$ in Equation 1 is defined as:

$$\lambda^{\mathrm{Z}}_{ij} = \frac{\partial C(s_i - s_j)}{\partial s_i} = \frac{-\sigma}{1 + e^{\sigma(s_i - s_j)}} \left|\Delta \mathrm{Z}_{ij}\right| \ ,$$

where C is the RankNet [7] pairwise cost function, s_i and s_j are the model-predicted scores for d_i and d_j, and $\Delta Z_{ij} = Z_{\pi_{ji}} - Z_{\pi_{ij}}$ is the change in the metric after *swapping* the ranks of d_i and d_j in current predicted ranking π_{ij}. The parameter σ determines the shape of the sigmoid. The $\lambda(P)_i^Z$ gradient can be used in iterative optimization learning algorithms such as artificial neural networks (e.g., LambdaRank [6]) or gradient-boosted decision trees (e.g., LambdaMART).

4 LambdaFair

In this work, we propose LambdaFair, a LambdaMART-based in-processing method to jointly optimize two ranking metrics: NDCG and rND. On the basis of LambdaMART's gradient definition, the joint optimization of the two metrics can be achieved by convex combination as follows:

$$\lambda_i = \alpha\lambda(E)_i^{\text{NDCG}} + (1-\alpha)\lambda(F)_i^{\text{rND}} \quad ,$$

where E and F are the sets of pairwise preferences related to NDCG (effectiveness) and rND (fairness) metrics, respectively. The hyper-parameter α weights the relative importance of the two. Note that E is built based on the ground-truth relevance labels and thus is the same as the set P used in Equation 1.

Despite a simple definition, this joint optimization hides some subtle challenges that must be carefully handled to jointly optimize the two ranking metrics. Regarding NDCG, the set E is naturally generated by the ground-truth relevance labels paired with each document in the query. The labels define a *partial order* over the set D for each query q, and thus it is sufficient to put in E all possible ordered pairs (i, j) that ensure $y_i > y_j$ to achieve the maximum NDCG.

Conversely, since to minimize rND, each ranking prefix has to contain a protected item proportion of $|\mathcal{G}^+|/n$, there is no predefined partial order of documents in D to derive F.

document	d_1	d_2	d_3	d_4	d_5	d_6
rank	1	2	3	4	5	6
relevance	0	0	0	0	1	0
group	-	-	-	-	+	+

Fig. 1. Toy example to discuss joint metric optimization. Six documents $d_1, \ldots, d_6$, with different relevance and group membership, with $\mathcal{G}^+ = \{d_5, d_6\}$. Note that d_5 is relevant and also belongs to the protected group.

Furthermore, considering sets E and F in isolation can provide sub-optimal optimization for both metrics. Let's consider the example in Figure 1. In terms of NDCG metric, document d_5 must be ranked first, and clearly $E = \{(5, i) \mid i \neq 5\}$. From a fairness perspective, given an rND bin size $b = 3$, the ideal ranking must have one of the two protected documents in the first bin (top-3) and the other in the second bin (bottom-3). To achieve this, we may define (at least) two sets of preference pairs: $F' = \{(5, i) \mid i \neq 5 \wedge i \neq 6\}$ and $F'' = \{(6, i) \mid i \neq 6 \wedge i \neq 5\}$. The set F' would push d_5 in the first bin and leave d_6 in the second one, while F'' has the

symmetric behavior. When looking at fairness, the two are equivalent. Indeed, F' perfectly agrees with E, as both push d_5 higher in the ranking, leading to the ranking $[5, 1, 2, 3, 4, 6]$ optimal for both NDCG and rND. Instead, combining E and F'' produces the ranking $[5, 6, 1, 2, 3, 4]$, which is sub-optimal for rND since both protected documents d_5 and d_6 are ranked in the first bin.

From the above example, we can conclude that the definition of pairwise preferences plays a crucial role and that preferences for one metric cannot be defined independently of the other metric.

4.1 Strategies

This section provides three different heuristic strategies for computing the set F for the fair-ranking metric rND based on the fixed set E used to optimize NDCG. For all strategies discussed in the following, a new set F is computed every time the learning algorithm requires the estimation of the lambdas, so it is not pre-computed as for the NDCG set E. We made LambdaFair's implementation, along with its three strategies, publicly available.[1]

Before discussing how to derive F for LambdaFair, let's recall that computing the gradients of LambdaMART-based algorithms requires estimating how swapping documents within the ranking π affects the optimized metrics at each learning iteration. We denote by ΔrND_{ij} the change in terms of rND after swapping the rank positions of d_i and d_j in π_{ij} (i.e., a ranking where $d_i \prec_\pi d_j$), namely $\Delta\text{rND}_{ij} = \text{rND}_{\pi_{ji}} - \text{rND}_{\pi_{ij}}$. Note that for items belonging to the same group or ranked in the same ranking bin, their inter-bin or intra-bin swaps, respectively, do not change the rND metric. Consequently, $\Delta\text{rND}_{ij} = 0$ that implies $\lambda_{ij}^{\text{rND}} = 0$, and for the sake of efficiency, we skip them in estimating $\lambda(F)_i^{\text{rND}}$.

ΔrND: Variation on swap. Given a ranking π for a query q, produced by the model at the current iteration, and let d_i and d_j be two documents in D such that d_i is ranked higher than d_j in π, i.e., $d_i \prec_\pi d_j$. If ranking d_i higher than d_j provides better fairness, then after the swap, the rND metric gets worse, and its value increases, resulting in $\Delta\text{rND}_{ij} > 0$. Conversely, $\Delta\text{rND}_{ij} < 0$ when d_j should be ranked higher than d_i, and thus the swap entails better fairness.

Finally, given any two documents d_i and d_j in D, where $d_i \prec_\pi d_j$, and their ids i and j, we define F as follows:

$$F = \{(i, j) \mid \Delta\text{rND}_{ij} > 0\} \cup \{(j, i) \mid \Delta\text{rND}_{ij} < 0\} \ .$$

The ordered pair (i, j) is included in F if the relative ranking $d_i \prec_\pi d_j$ must be maintained to avoid reducing rND. Conversely, (j, i) is included in F if swapping d_i and d_j in π improves rND. It is worth noting that F is created independently of E. Consequently, no optimal agreements between E and F are guaranteed.

rND+: Fairness driven. Let $\pi_{\text{rND}+}$ be a total ordering of D that maximizes NDCG while ensuring minimum rND. Then, we then define F as follows:

$$F = \{(i, j) \mid ind_b(d_i|\pi_{\text{rND}+}) < ind_b(d_j|\pi_{\text{rND}+})\} \ ,$$

[1] https://github.com/FedericoMarcuzzi/LambdaFair-for-Fair-and-Effective-Ranking

where ind_b is a function that returns the ranking-bin index that contains a given document. More formally, given any ranking π, $h = ind_b(d_i|\pi)$, with $1 \leq h \leq \lceil n/b \rceil$, is the index of the bin that includes document d_i, i.e., the index h such that $d_i \in \pi[(h-1)b+1, hb]$. Hence, this definition of F reflects the bin-wise order of the ideal ranking $\pi_{\text{rND+}}$, which prioritizes fairness over effectiveness.

The ranking $\pi_{\text{rND+}}$ is built in two stages. The first stage, which is precomputed before training to enhance the algorithm's efficiency, generates a static ranking $\tilde{\pi}_{\text{rND+}}$ as follows. We first sort documents in D by decreasing relevance, thus *maximizing* NDCG. Then, we perform the minimal number of rank promotions or demotions of protected documents, without changing their relative order, to put in each ranking bin a number of protected documents that best approximates $|\mathcal{G}^+|/n$. This latter step guarantees the minimum rND without completely undoing the partial order induced by the relevance labels.

The second stage takes place at each training iteration by slightly changing the total ordering $\tilde{\pi}_{\text{rND+}}$. Note that $\tilde{\pi}_{\text{rND+}}$ is one of the possible optimal total ordering due to *ties*. Indeed, two documents of equal relevance belonging to the same group allow us to swap their positions in $\tilde{\pi}_{\text{rND+}}$ without negatively impacting NDCG or rND. Consequently, we update the total ordering $\tilde{\pi}_{\text{rND+}}$ at each training iteration to align with the ranking π of D based on the scores produced by the learned model. More specifically, given a set of ties, i.e., a subset of documents in D with the same relevance and group, we reorder $\tilde{\pi}_{\text{rND+}}$ into $\pi_{\text{rND+}}$ according to π. The rationale for this approach is not to break what has been learned from the model so far by introducing unnecessary order constraints.

NDCG+: Effectiveness driven. This third strategy is similar to rND+, but prioritizing on NDCG. Let $\pi_{\text{NDCG+}}$ be a total ordering of documents that minimizes rND under the constraint of guaranteeing the *maximum* NDCG. The set F is thus defined as follows, where F reflects the bin-wise order of the ideal ranking $\pi_{\text{NDCG+}}$ that prioritizes effectiveness over fairness:

$$F = \{(i, j) \mid ind_b(d_i|\pi_{\text{NDCG+}}) < ind_b(d_j|\pi_{\text{NDCG+}})\} \ .$$

Similarly to $\pi_{\text{rND+}}$, the ranking $\pi_{\text{NDCG+}}$ is derived in two stages. First, documents in D are ordered in decreasing order of relevance to maximize NDCG. Then, a minimal inter-bin swap of equally relevant documents belonging to different groups is performed to distribute protected and unprotected items in each ranking bin to approximate $|\mathcal{G}^+|/n$. Since those documents have the same relevance, they are ties whose swaps do not change NDCG but aim to minimize rND since the swapped documents belong to different groups. The second stage is the same as the one used to generate $\pi_{\text{rND+}}$.

5 Experimental Evaluation

5.1 Datasets

We evaluate LambdaFair on three publicly available datasets: MSLR-30K [23], Statlog (German Credit Data) [14], and Home Mortgage Disclosure Act [9].

Table 1. Datasets properties: query number (#queries), average $|D|$ ($\overline{\text{query len.}}$), number of unique relevance labels (#labels), attribute that identifies the protected group $\mathcal{G}^+$ (pt. attribute), percentage of documents belonging to $\mathcal{G}^+$ (%pt. group), and percentage of documents labeled as relevant ($y_i > 0$) belonging to $\mathcal{G}^+$ (%pt. relevant).

Dataset	#queries	$\overline{\text{query len.}}$	#labels	pt. attribute	%pt. group	%pt. relevant
MSLR-30K	31,531	119.60	5	QS2	42.56	59.97
Statlog (Sex)	100,000	50	2	Sex	61.42	28.73
Statlog (Age)	100,000	50	2	Age	34.85	50.84
HMDA-CT	100,000	50	2	Sex	33.18	31.35

The MSLR-30K dataset consists of feature vectors extracted from query-url pairs along with human-assessed 5-level relevance labels. The dataset does not come with specified groups; for this reason, we followed previous works [4, 16, 29, 30], and used the QualityScore2 (QS2: feature's id 133) as the discriminatory feature. Following the work of Vardasbi *et al.* [29], we used 10 as the threshold to divide the documents into protected (< 10) and unprotected (≥ 10) groups.

The Statlog dataset is widely used in fairness-aware LTR [4, 16, 28, 30]. It contains 1,000 individuals characterized with binary-relevance labels for classifying an individual's creditworthiness. For each query, we randomly sample 50 individuals with a 4:1 ratio of non-creditworthy to creditworthy individuals for a total of 100,000 queries. Building on previous work, we derived from Statlog dataset two distinct datasets: Statlog (Sex), which divides individuals into protected (*female*) and unprotected (*male*) groups based on their gender [16, 28, 30], and Statlog (Age), which uses 35 as the age threshold to categorize individuals into protected (< 35) and unprotected (≥ 35) groups [4].

The Home Mortgage Disclosure Act dataset includes annual data on home mortgage loans across all 50 states in the US, with records available since 2007. Following Gorantla *et al.* [13], we created HMDA-CT where we focused our analysis on the state of Connecticut, using data from 2013, 2014, and 2015 as the training set, 2016 for validation, and 2017 for testing. Similar to the Statlog datasets, we randomly sample 50 individuals with a 4:1 ratio of not-approved to approved loans for a total of 100,000 queries. Then, we used the sex feature to divide individuals into protected (*female*) and unprotected (*male*) groups.

The four datasets, MSLR-30K, Statlog (Age), Statlog (Sex), and HMDA-CT are partitioned into the train, validation, and test sets following a 60%-20%-20% scheme. Further details can be found in Table 1.

5.2 Learning Algorithms

In this section, we report the experimental results obtained by our LambdaFair and several baseline techniques. While LambdaFair is a LambdaMART-based in-processing method to jointly optimize effectiveness (NDCG) and fairness metrics (rND), we used a plain LambdaMART[2] as the reference baseline aimed to only

[2] https://github.com/microsoft/LightGBM

maximize NDCG. Among the in-processing methods introduced in Section 2 to optimize fairness, we used PL-Rank-3[3] and Group-Fair-PL[4] as fair baselines. In particular, we chose Group-Fair-PL as the reference baseline for maximum fairness since, as LambdaFair, it optimizes group fairness. Furthermore, we optimized Group-Fair-PL under the proportional representation constraint of protected items in the top-k as aligns with LambdaFair optimizing the rND metric.

All algorithms were trained to optimize their respective metrics with cutoff k equal to 15 and 50. For LambdaFair, we kept a common cutoff while jointly optimizing NDCG and rND, and we fixed the rND bin size b to 5 for both cutoff values. Additionally, we fine-tuned the hyperparameter α between 0.1 and 0.9 with a step of 0.1 to minimize rND.

For Group-Fair-PL, we optimize it to ensure a proportional representation of protected items, as this optimization better aligns with our fairness metric rND. All algorithms were trained until early stopping (no changes in the validation set within 100 iterations or 10 epochs). For the best α, we performed model selection based on NDCG@k on the validation set.

Note that during the learning and prediction phases, we assume the unavailability of the protected features or any information about the group membership. In this regard, we removed the corresponding features from each dataset; however, we used the group-membership labels to evaluate the models' fairness.

5.3 Results

Table 2 reports the results in terms of effectiveness and fairness obtained by each LTR algorithm. In particular, we report results for NDCG and rND evaluated at cutoff k (15 and 50) on the test set of each dataset.

As expected, among the LambdaFair variants, NDCG+ is the one providing higher effectiveness and lower fairness since it prioritizes NDCG over rND. Interestingly, for both Statlog datasets (and similarly for HMDA-CT), a clear pattern appears. rND+ is the fairest and the least effective, NDCG+ is the most effective and the least fair, and ΔrND is placed in the middle, with a trade-off between effectiveness and fairness. Again, this is expected since ΔrND does not prioritize a metric over the order.

However, for the MSLR-30K dataset, ΔrND becomes on average the fairest variant. We attribute this to the larger range of relevance labels of MSLR-30K. Indeed, MSLR-30K contains relevant labels ranging from 0 to 4, while the other three datasets have binary relevance labels. A larger range of labels makes it challenging to create a ranking $\pi_{\text{rND}+}$ that agrees with both rND and NDCG. Consequently, the set E that optimizes NDCG can strongly disagree with F used to optimize rND, leading to a sub-optimal optimization for both metrics.

Overall, the LambdaFair variants always achieve statistically higher fairness than LambdaMART with a slight decrease in effectiveness. This demonstrated

[3] https://github.com/HarrieO/2022-SIGIR-plackett-luce
[4] https://github.com/sruthigorantla/Group-Fair-PL

Table 2. Performance in terms of NDCG and rND (percentage). Statistically significant differences compared to LambdaMART are marked with * (Fisher's randomization test [12] with a two-sided $p-$value = 0.01). Differences in performances compared to Group-Fair-PL are marked with better or worst. In **bold** the best rND values.

Model	NDCG@15 ↑	rND@15 ↓	NDCG@50 ↑	rND@50 ↓
	MSLR-30K (QualityScore2)			
LambdaMART	+6.27 55.02	+2.35 26.48	+8.26 63.19	+2.80 22.52
LambdaFair $_{\text{NDCG+}}$	+6.11 54.86	+1.14 25.27*	+8.02 62.95*	+1.25 20.97*
LambdaFair $_{\Delta\text{rND}}$	+5.66 54.41*	-0.86 **23.27***	+7.70 62.63*	-0.39 **19.33***
LambdaFair $_{\text{rND+}}$	+5.64 54.39*	+0.42 24.55*	+7.78 62.71*	+0.63 20.35*
PL-Rank-3	-1.30 47.45*	+0.14 24.27*	-0.75 54.18*	+1.21 20.93*
Group-Fair-PL	48.75*	24.13*	54.93*	19.72*
	HMDA-CT (Sex)			
LambdaMART	+7.72 96.09	+0.21 19.17	+3.03 98.78	+0.46 18.89
LambdaFair $_{\text{NDCG+}}$	+7.13 95.50*	18.96*	+2.64 98.39*	+0.23 18.66*
LambdaFair $_{\Delta\text{rND}}$	+7.12 95.49*	+0.11 19.07	+2.42 98.17*	+0.45 18.88
LambdaFair $_{\text{rND+}}$	+7.12 95.49*	-0.07 **18.89***	+2.59 98.34*	+0.16 18.59*
PL-Rank-3	+0.07 88.44*	+0.14 19.10	-0.87 94.88*	+0.42 18.85*
Group-Fair-PL	88.37*	18.96*	95.75*	**18.43***
	Statlog (Age)			
LambdaMART	+9.16 100.00	+8.70 29.21	+7.66 100.00	+6.72 24.26
LambdaFair $_{\text{NDCG+}}$	+9.16 100.00	+1.45 21.96*	+7.66 100.00	+2.08 19.62*
LambdaFair $_{\Delta\text{rND}}$	+8.95 99.79*	+0.45 20.96*	+7.54 99.88*	+1.69 19.23*
LambdaFair $_{\text{rND+}}$	+8.58 99.42*	-0.15 **20.36***	+6.73 99.07*	+1.02 18.56*
PL-Rank-3	-1.63 89.21*	+0.45 20.96*	-0.47 91.87*	+0.27 17.81*
Group-Fair-PL	90.84*	20.51*	92.34*	**17.54***
	Statlog (Sex)			
LambdaMART	+7.47 100.00	+1.44 20.51	+6.86 100.00	+1.10 19.34
LambdaFair $_{\text{NDCG+}}$	+7.47 100.00	+0.33 19.40*	+6.86 100.00	+0.12 18.36*
LambdaFair $_{\Delta\text{rND}}$	+7.46 99.99*	+0.28 19.35*	+6.86 100.00	+0.14 18.38*
LambdaFair $_{\text{rND+}}$	+7.26 99.79*	-0.08 18.99*	+6.39 99.53*	-0.21 18.03*
PL-Rank-3	-1.51 91.02*	-0.16 **18.91***	-1.41 91.73*	-0.39 **17.85***
Group-Fair-PL	92.53*	19.07*	93.14*	18.24*

that the joint optimization of NDCG and rND provided by each variant allows LambdaFair to train both fair and effective rankers.

Regarding fairness baselines, in some cases (mostly on Statlog datasets), the Plackett-Luce-based models produce fairer rankings than LambdaFair. However, while the difference in fairness is barely noticeable in most cases, their substantial drop in effectiveness makes them non-preferable solutions when both effectiveness and fairness are required. For example, for the Statlog (Age) dataset and fairness evaluated by rND@50, the best algorithm in terms of fairness is Group-Fair-PL (rND = 17.54), but with a very strong drop in effectiveness, i.e., about 9 points less than LambdaMART. For this dataset, the best LambdaFair's variant is rND+, which achieves a slightly worser fairness (rND = 18.56, i.e., about 1 point more than Group-Fair-PL), but with a very good effectiveness (NDCG = 99.53, i.e., about 0.5 points less than LambdaMART). Therefore,

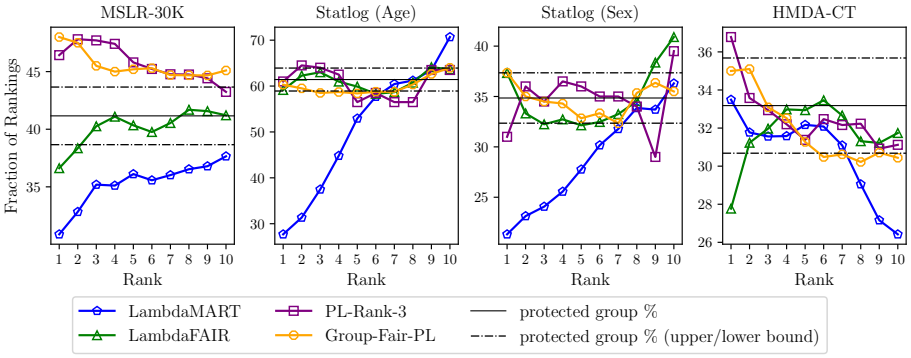

Fig. 2. Average proportion of protected items ranked in the top-10. LambdaFair refers to the rND+ variant. Results for models trained and evaluated with cutoff $k = 15$.

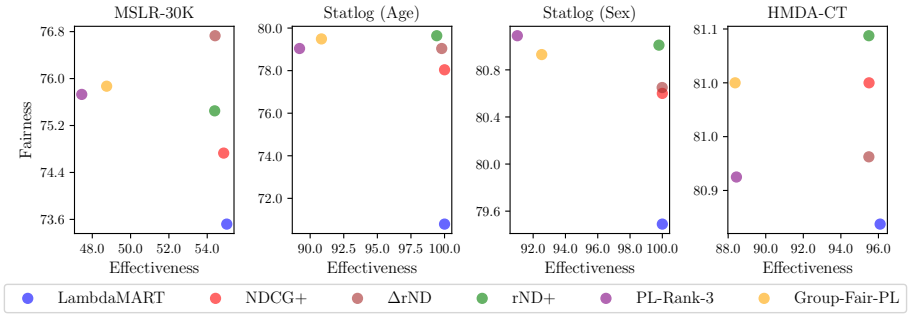

Fig. 3. Effectiveness and fairness trade-off. Fairness = $(1 - \text{rND})\%$. Results for models trained and evaluated with cutoff $k = 15$.

LambdaFair$_{\text{rND+}}$, although not the best in terms of fairness, is the preferable solution if one does not want to lose much in effectiveness. Figure 2 shows the average proportion of protected items in each of the top-10 rank positions. The solid line represents the proportion of the protected group in the test set, while the dashed lines indicate upper and lower bounds at ±2.5% from the average proportion. For the sake of visualization, we report the results only for the rND+ variant and for models trained with cutoff $k = 15$. In most cases, LambdaFair is the algorithm that better proportionally distributes items within the top-10, consistently remaining within the upper and lower bounds. This result is consistent with the findings presented in Table 2 and illustrates that LambdaFair outperforms the baselines in terms of statistical parity, confirming its higher fairness. Interestingly, in some cases, PL-based models appear to offer a more proportional distribution of protected items than LambdaFair in the top-1. Finally, as expected, LambdaMART underexposes items from the protected group in the top positions, as it is trained without any fairness constraints.

Figure 3 plots the effectiveness and fairness trade-off, with fairness expressed as $(1 - \text{rND})\%$ (higher is better). Except for MSLR-30K, where ΔrND is best,

Table 3. Training times per learning algorithm (minutes).

Model	MSLR-30K	Statlog (Age/Sex)	HMDA-CT
LambdaMART	12	5	5
LambdaFair	18	8	8
PL-Rank-3	228	202	201
Group-Fair-PL	1272	1142	1147

rND+ is the overall best variant, providing higher fairness with slightly lower effectiveness than LambdaMART and the other LambdaFair's variants.

Finally, Table 3 showcases the results of the efficiency analysis, providing the time required by each algorithm to complete the training process with early stopping criteria. LambdaMART-based algorithms were trained on two Intel(R) Xeon(R) Silver 4110 CPUs using 28 threads. For Plackett-Luce-based algorithms, we used a 12 GB Nvidia Titan Xp with the GP102 architecture and 3840 CUDA cores. The results clearly show that the training cost of Plackett-Luce-based models, despite the use of a GPU, is significantly higher than that of LambdaMART-based models. In fact, training PL-Rank-3 and Group-Fair-PL took one and two orders of magnitude longer, respectively, than training LambdaFair. This ultimately highlights that LambdaFair is not only capable of training fair and effective rankers, but it also excels in doing so with remarkable efficiency. This combination of fairness, effectiveness, and efficiency positions LambdaFair as a highly competitive solution in the realm of ranking algorithms.

6 Conclusion

In this work, we presented LambdaFair, a LambdaMART-based in-processing method to jointly optimize fairness and effectiveness ranking metrics. The main goal of LambdaFair is to provide unbiased rankings without compromising effectiveness. In this work, we designed three variants of LambdaFair tailored to optimize different levels of fairness and effectiveness. Empirical evaluation on publicly available datasets demonstrated that LambdaFair produces fairer rankings than LambdaMART, with only a minor decrease in effectiveness. Furthermore, LambdaFair exhibits similar fairness capability to state-of-the-art fairness baselines while achieving significantly higher effectiveness. Finally, we demonstrated that LambdaFair not only achieves fairer rankings but also accelerates the training process, achieving up to two orders of magnitude faster training time compared to the fairness baseline. This improvement underscores LambdaFair's efficiency, making it a highly practical solution for large-scale applications where fairness, effectiveness, and efficiency are critical.

Acknowledgments. This work was partially supported by the Next Generation EU (EU-NGEU) projects SERICS (Grant NRRP M4C2 Inv.1.3 PE00000014) and iNEST (Grant NRRP M4C2 Inv.1.5 ECS00000043), and the National project PRIN22 WHAM! (Grant n. 2022ZZX57L, CUP: H53D23003750006).

References

1. Abolghasemi, A., Azzopardi, L., Askari, A., de Rijke, M., Verberne, S.: Measuring bias in a ranked list using term-based representations. In: Goharian, N., Tonellotto, N., He, Y., Lipani, A., McDonald, G., Macdonald, C., Ounis, I. (eds.) Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14612, pp. 3–19. Springer (2024). https://doi.org/10.1007/978-3-031-56069-9_1, https://doi.org/10.1007/978-3-031-56069-9_1
2. Beutel, A., Chen, J., Doshi, T., Qian, H., Wei, L., Wu, Y., Heldt, L., Zhao, Z., Hong, L., Chi, E.H., Goodrow, C.: Fairness in recommendation ranking through pairwise comparisons. In: Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., Karypis, G. (eds.) Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019. pp. 2212–2220. ACM (2019). https://doi.org/10.1145/3292500.3330745, https://doi.org/10.1145/3292500.3330745
3. Biega, A.J., Gummadi, K.P., Weikum, G.: Equity of attention: Amortizing individual fairness in rankings. In: Collins-Thompson, K., Mei, Q., Davison, B.D., Liu, Y., Yilmaz, E. (eds.) The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018. pp. 405–414. ACM (2018). https://doi.org/10.1145/3209978.3210063, https://doi.org/10.1145/3209978.3210063
4. Bower, A., Eftekhari, H., Yurochkin, M., Sun, Y.: Individually fair rankings. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=71zCSP_HuBN
5. Bruch, S.: An alternative cross entropy loss for learning-to-rank. In: Leskovec, J., Grobelnik, M., Najork, M., Tang, J., Zia, L. (eds.) WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 118–126. ACM / IW3C2 (2021). https://doi.org/10.1145/3442381.3449794, https://doi.org/10.1145/3442381.3449794
6. Burges, C.J.C., Ragno, R., Le, Q.V.: Learning to rank with nonsmooth cost functions. In: Schölkopf, B., Platt, J.C., Hofmann, T. (eds.) Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006. pp. 193–200. MIT Press (2006), https://proceedings.neurips.cc/paper/2006/hash/af44c4c56f385c43f2529f9b1b018f6a-Abstract.html
7. Burges, C.J.C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.N.: Learning to rank using gradient descent. In: Raedt, L.D., Wrobel, S. (eds.) Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005. ACM International Conference Proceeding Series, vol. 119, pp. 89–96. ACM (2005). https://doi.org/10.1145/1102351.1102363, https://doi.org/10.1145/1102351.1102363
8. Burges, C.J.: From ranknet to lambdarank to lambdamart: An overview. Learning **11**(23-581), 81 (2010)
9. Council, F.F.I.E.: HMDA Data Publication (2017), https://www.consumerfinance.gov/data-research/hmda/historic-data/, released due to the Home Mortgage Disclosure Act

10. Donmez, P., Svore, K.M., Burges, C.J.C.: On the local optimality of lambdarank. In: Allan, J., Aslam, J.A., Sanderson, M., Zhai, C., Zobel, J. (eds.) Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009. pp. 460–467. ACM (2009). https://doi.org/10.1145/1571941.1572021, https://doi.org/10.1145/1571941.1572021
11. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.S.: Fairness through awareness. In: Goldwasser, S. (ed.) Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012. pp. 214–226. ACM (2012). https://doi.org/10.1145/2090236.2090255, https://doi.org/10.1145/2090236.2090255
12. Fisher, R.: The design of experiments. 1935. Oliver and Boyd, Edinburgh (1935)
13. Gorantla, S., Bhansali, E., Deshpande, A., Louis, A.: Optimizing learning-to-rank models for ex-post fair relevance. In: Yang, G.H., Wang, H., Han, S., Hauff, C., Zuccon, G., Zhang, Y. (eds.) Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024. pp. 1525–1534. ACM (2024). https://doi.org/10.1145/3626772.3657751, https://doi.org/10.1145/3626772.3657751
14. Hofmann, H.: Statlog (German Credit Data). UCI Machine Learning Repository (1994), DOI: https://doi.org/10.24432/C5NC77
15. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. **20**(4), 422–446 (2002). https://doi.org/10.1145/582415.582418, http://doi.acm.org/10.1145/582415.582418
16. Kotary, J., Fioretto, F., Hentenryck, P.V., Zhu, Z.: End-to-end learning for fair ranking systems. In: Laforest, F., Troncy, R., Simperl, E., Agarwal, D., Gionis, A., Herman, I., Médini, L. (eds.) WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022. pp. 3520–3530. ACM (2022). https://doi.org/10.1145/3485447.3512247, https://doi.org/10.1145/3485447.3512247
17. Lahoti, P., Gummadi, K.P., Weikum, G.: ifair: Learning individually fair data representations for algorithmic decision making. In: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019. pp. 1334–1345. IEEE (2019). https://doi.org/10.1109/ICDE.2019.00121, https://doi.org/10.1109/ICDE.2019.00121
18. Luce, R.D.: Individual Choice Behavior: A Theoretical analysis. Wiley, New York, NY, USA (1959)
19. Marcuzzi, F., Lucchese, C., Orlando, S.: Lambdarank gradients are incoherent. In: Frommholz, I., Hopfgartner, F., Lee, M., Oakes, M., Lalmas, M., Zhang, M., Santos, R.L.T. (eds.) Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023. pp. 1777–1786. ACM (2023). https://doi.org/10.1145/3583780.3614948, https://doi.org/10.1145/3583780.3614948
20. Oosterhuis, H.: Computationally efficient optimization of plackett-luce ranking models for relevance and fairness. In: Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., Sakai, T. (eds.) SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021. pp. 1023–1032. ACM (2021). https://doi.org/10.1145/3404835.3462830, https://doi.org/10.1145/3404835.3462830
21. Oosterhuis, H.: Learning-to-rank at the speed of sampling: Plackett-luce gradient estimation with minimal computational complexity. In: Amigó, E., Castells,

P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022. pp. 2266–2271. ACM (2022). https://doi.org/10.1145/3477495.3531842, https://doi.org/10.1145/3477495.3531842

22. Plackett, R.L.: The analysis of permutations. Journal of the Royal Statistical Society. Series C (Applied Statistics) **24**(2), 193–202 (1975), http://www.jstor.org/stable/2346567
23. Qin, T., Liu, T.: Introducing LETOR 4.0 datasets. CoRR **abs/1306.2597** (2013), http://arxiv.org/abs/1306.2597
24. Raj, A., Ekstrand, M.D.: Measuring fairness in ranked results: An analytical and empirical comparison. In: Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022. pp. 726–736. ACM (2022). https://doi.org/10.1145/3477495.3532018, https://doi.org/10.1145/3477495.3532018
25. Rekabsaz, N., Kopeinik, S., Schedl, M.: Societal biases in retrieved contents: Measurement framework and adversarial mitigation of bert rankers. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 306–316. SIGIR '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3404835.3462949, https://doi.org/10.1145/3404835.3462949
26. Sapiezynski, P., Zeng, W., E Robertson, R., Mislove, A., Wilson, C.: Quantifying the impact of user attentionon fair group representation in ranked lists. In: Companion Proceedings of The 2019 World Wide Web Conference. p. 553–562. WWW '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3308560.3317595, https://doi.org/10.1145/3308560.3317595
27. Singh, A., Joachims, T.: Fairness of exposure in rankings. In: Guo, Y., Farooq, F. (eds.) Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018. pp. 2219–2228. ACM (2018). https://doi.org/10.1145/3219819.3220088, https://doi.org/10.1145/3219819.3220088
28. Singh, A., Joachims, T.: Policy learning for fairness in ranking. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 5427–5437 (2019), https://dl.acm.org/doi/10.5555/3454287.3454774
29. Vardasbi, A., Sarvi, F., de Rijke, M.: Probabilistic permutation graph search: Black-box optimization for fairness in ranking. In: Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022. pp. 715–725. ACM (2022). https://doi.org/10.1145/3477495.3532045, https://doi.org/10.1145/3477495.3532045
30. Yadav, H., Du, Z., Joachims, T.: Policy-gradient training of fair and unbiased ranking functions. In: Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., Sakai, T. (eds.) SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15,

2021. pp. 1044–1053. ACM (2021). https://doi.org/10.1145/3404835.3462953, https://doi.org/10.1145/3404835.3462953

31. Yang, K., Stoyanovich, J.: Measuring fairness in ranked outputs. In: Proceedings of the 29th International Conference on Scientific and Statistical Database Management, Chicago, IL, USA, June 27-29, 2017. pp. 22:1–22:6. ACM (2017). https://doi.org/10.1145/3085504.3085526, https://doi.org/10.1145/3085504.3085526
32. Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., Baeza-Yates, R.: Fa*ir: A fair top-k ranking algorithm. In: Lim, E., Winslett, M., Sanderson, M., Fu, A.W., Sun, J., Culpepper, J.S., Lo, E., Ho, J.C., Donato, D., Agrawal, R., Zheng, Y., Castillo, C., Sun, A., Tseng, V.S., Li, C. (eds.) Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017. pp. 1569–1578. ACM (2017). https://doi.org/10.1145/3132847.3132938, https://doi.org/10.1145/3132847.3132938
33. Zehlike, M., Castillo, C.: Reducing disparate exposure in ranking: A learning to rank approach. In: Huang, Y., King, I., Liu, T., van Steen, M. (eds.) WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020. pp. 2849–2855. ACM / IW3C2 (2020). https://doi.org/10.1145/3366424.3380048, https://doi.org/10.1145/3366424.3380048
34. Zehlike, M., Hacker, P., Wiedemann, E.: Matching code and law: achieving algorithmic fairness with optimal transport. Data Min. Knowl. Discov. **34**(1), 163–200 (2020). https://doi.org/10.1007/S10618-019-00658-8, https://doi.org/10.1007/s10618-019-00658-8
35. Zehlike, M., Yang, K., Stoyanovich, J.: Fairness in ranking, part I: score-based ranking. ACM Comput. Surv. **55**(6), 118:1–118:36 (2023). https://doi.org/10.1145/3533379, https://doi.org/10.1145/3533379
36. Zehlike, M., Yang, K., Stoyanovich, J.: Fairness in ranking, part II: learning-to-rank and recommender systems. ACM Comput. Surv. **55**(6), 117:1–117:41 (2023). https://doi.org/10.1145/3533380, https://doi.org/10.1145/3533380
37. Zemel, R.S., Wu, Y., Swersky, K., Pitassi, T., Dwork, C.: Learning fair representations. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013. JMLR Workshop and Conference Proceedings, vol. 28, pp. 325–333. JMLR.org (2013), http://proceedings.mlr.press/v28/zemel13.html