Learning Fair Ranking Policies via Differentiable Optimization of Ordered Weighted Averages

My H Dinh*
fqw2tz@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

James Kotary*
jk4pn@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Ferdinando Fioretto
University of Virginia
Charlottesville, Virginia, USA
fioretto@virginia.edu

ABSTRACT

Learning to Rank (LTR) is one of the most widely used machine learning applications. It is a key component in platforms with profound societal impacts, including job search, healthcare information retrieval, and social media content feeds. Conventional LTR models have been shown to produce biases results, stimulating a discourse on how to address the disparities introduced by ranking systems that solely prioritize user relevance. However, while several models of fair learning to rank have been proposed, they suffer from deficiencies either in accuracy or efficiency, thus limiting their applicability to real-world ranking platforms. This paper shows how efficiently-solvable fair ranking models, based on the optimization of Ordered Weighted Average (OWA) functions, can be integrated into the training loop of an LTR model to achieve favorable balances between fairness, user utility, and runtime efficiency. In particular, this paper is the first to show how to backpropagate through constrained optimizations of OWA objectives, enabling their use in integrated prediction and decision models.

CCS CONCEPTS

• **Computing methodologies → Learning to rank**.

KEYWORDS

Predict-then-optimize, Learning To Rank, Algorithmic Fairness

ACM Reference Format:
My H Dinh, James Kotary, and Ferdinando Fioretto. 2024. Learning Fair Ranking Policies via Differentiable Optimization of Ordered Weighted Averages. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24), June 03–06, 2024, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3630106.3661932

1 INTRODUCTION

Ranking models have become a pervasive aspect of everyday life, shaping how we access information online. They serve as the primary tools through which we discover products, consume content, and connect with others. These systems rank a diverse array of items, ranging from job candidates to research papers and beyond. As machine learning-based models, they are primarily trained to provide maximum utility to users, by delivering the most relevant results for their search queries. In today's information-driven economy, an item's position in the ranking significantly impacts its visibility, selection, and ultimately, its economic success.

This influence has drawn attention to the disparate impacts of ranking systems on underrepresented groups, particularly in data-driven systems where item relevance is determined by implicit user feedback like clicks and dwell times. Such systems are prone to disproportionate exposure and self-reinforcing feedback loops that foster winner-take-all dynamics [24, 27]. For instance, in job search systems, male candidates may receive overwhelmingly more exposure, even if female candidates are only marginally less relevant. Research by [9] demonstrates that minor differences in relevance can result in significant exposure disparities for minority group candidates in job candidate ranking systems [31]. Controlling these impacts is essential to avoid reinforcing biases, maintain market health, and implement anti-discrimination measures [8, 23].

Fairness-aware ranking models address these concerns by minimizing disparate exposure between groups while maximizing relevance. However, designing such models is challenging, as they require complex constraints to be integrated into machine learning outputs. For example, the popular listwise learning to rank method, through modifications to its loss function, is only capable of modeling fairness of exposure in the top ranking position [31]. An alternative paradigm, first investigated by [15], prioritizes the precise satisfaction of fairness constraints by integrating fair ranking optimization models end-to-end with predictive models of user relevance. This enables precise control over the trade-off between fairness and utility, but its reliance on hard constraints comes with increased computational costs and limitations in dealing with multi-group fairness criteria. By integrating efficient optimization of OWA functions with relevance score prediction, this paper shows how the efficiency and modeling flexibility of prior end-to-end optimization and learning solutions can be greatly improved upon, while retaining their favorable fairness properties.

Contributions. To address these limitations, this paper makes the following novel contributions: **(1)** It shows how to adopt an alternative approach based on Ordered Weighted Averages (OWA) to design efficient policy optimization modules for the fair learning to rank setting. **(2)** For the first time, it shows how to backpropagate gradients through the highly discontinuous optimization of OWA functions, enabling its use in end-to-end learning. **(3)** The resulting end-to-end optimization and learning scheme, called Smart OWA Optimization for Fair Learning to Rank (**SOFaiR**), is compared with contemporary fair LTR methods, demonstrating not only substantial advantages in fairness over previous fair LTR, but

*Both authors contributed equally to this research.

This work is licensed under a Creative Commons Attribution International 4.0 License.

FAccT '24, June 03–06, 2024, Rio de Janeiro, Brazil
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0450-5/24/06
https://doi.org/10.1145/3630106.3661932

Table 1: Common symbols adopted throughout the paper.

Symbol	Semantic
N	Size of the training dataset
n	Number of items to be ranked
m	Number of protected groups
$\boldsymbol{x}_q = (x_q^i)_{i=1}^n$	List of feature embeddings for items to rank, given query q
$\boldsymbol{a}_q = (a_q^i)_{i=1}^n$	Protected groups associated with items x_q^i
$\boldsymbol{y}_q = (y_q^i)_{i=1}^n$	Relevance scores for each of n items given query q
G	The set of all protected group indicators
$\mathcal{M}_\theta$	End-to-end trainable fair ranking model with weights θ

Symbol	Semantic
σ	A permutation of the list $[n]$ for some n
$\mathbf{P}$	A permutation matrix corresponding to some σ
$\mathcal{P}_n$	The set of all permutations of $[n]$
τ	The sorting operator
Π	A ranking policy, or its representative bistochastic matrix
$\boldsymbol{u}(\Pi, y)$	Expected utility of policy Π under relevance scores y
$\mathcal{B}$	Birkhoff Polytope, the convex set of all ranking policies
$\mathcal{E}(i, \sigma)$	Exposure of item i

also advantages in efficiency and modeling flexibility over the end-to-end fair LTR scheme of [15]. A schematic illustration of the proposed scheme is depicted in Figure 1.

These contributions are significant: They demonstrate that by incorporating modern fair ranking optimization techniques, the integration of post-processing optimization models in end-to-end LTR training can be a viable and scalable paradigm to achieve highly accurate learning to rank system that also provides strong fairness properties.

2 PRELIMINARIES

Throughout the paper, vectors and matrices are denoted in bold font. The inner product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is written $\mathbf{a}^T\mathbf{b}$, while the outer product is $\mathbf{a}\,\mathbf{b}^T$. For a matrix $\mathbf{M}$, the vector $\overrightarrow{\mathbf{M}}$ is formed by concatenation of its rows. A hatted vector $\hat{\mathbf{a}}$ is the prediction of a machine learning model, and a starred vector $\mathbf{a}^\star$ is the optimal solution to some optimization problem. The list of integers $\{1 \dots n\}$ is written $[n]$. When $\mathbf{a} \in \mathbb{R}^n$ and σ is a permutation of $[n]$, $\mathbf{a}_\sigma$ is the corresponding permuted vector. The vectors of all ones and zeros are denoted $\mathbf{1}$ and $\mathbf{0}$, respectively. Commonly used symbols throughout the paper are organized in Table 1 for reference.

2.1 Problem Setting and Goals

Given a user query, the goal is to predict a ranking over n items, in order of most to least relevant, with respect to the query. Relevance of each item to be ranked, with respect to a search query q, is generally measured by a vector of *relevance scores* $\boldsymbol{y}_q \in \mathbb{R}^n$, often modeled on the basis of empirical observations such as historical click rates [26]. This setting considers a ground-truth dataset $(\boldsymbol{x}_q, \boldsymbol{a}_q, \boldsymbol{y}_q)_{q=1}^N$, where $\boldsymbol{x}_q \in \mathcal{X}$ is a list of feature vectors $(x_q^i)_{i=1}^n$, one for each of n items to be ranked in response to query q. $\boldsymbol{a}_q = (a_q^i)_{i=1}^n$ is a vector that indicates which (protected) group g within domain G to which each item belongs. $\boldsymbol{y}_q = (y_q^i)_{i=1}^n \in \mathcal{Y}$ is a vector of relevance scores, for each item with respect to query q. For example, on an image web-search context as depicted in Figure 1, a query denotes the search keywords, e.g., "CEO", the vectors x_q^i in $\boldsymbol{x}_q$ are feature embeddings for the images relative to q, each associated with a gender (attribute a_q^i), and the associated relevance scores y_q^i describe the relevance of item i to query q.

Rankings can be viewed as *permutations* which rearrange the order of a predefined *item list*. Intermediate between the user input and final ranking is often a ranking *policy* which produces discrete rankings (randomly or deterministically).

Learning to Rank. In learning to rank (LTR), a ML model $\mathcal{M}_\theta$ is often adopted to estimate relevance scores $\hat{\mathbf{y}}_q$ of items given their features $\mathbf{x}_q$ relative to user query q (see figure 1). From this a ranking *policy* $\boldsymbol{\Pi}$ is constructed. Its *expected utility* u is

$$u(\boldsymbol{\Pi}, \mathbf{y_q}) = \mathbb{E}_{\sigma\sim\boldsymbol{\Pi}}[\boldsymbol{\Delta}(\sigma, \mathbf{y_q})], \tag{1}$$

where $\boldsymbol{\Pi}$ is viewed as a distribution from which rankings σ are sampled randomly, and their utility Δ is a measure of the overall relevance of a given ranking σ, with respect to given relevance scores $\mathbf{y}_q$. Although its framework is applicable to any linear utility metric Δ for rankings, this paper uses the widely adopted Discounted Cumulative Gain (DCG):

$$\Delta(\sigma, \mathbf{y_q}) = DCG(\sigma, \mathbf{y_q}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{y_q^i b}_{\sigma_\mathbf{i}} = \mathbf{y_q^T P}^{(\sigma)}\mathbf{b}, \tag{2}$$

where $\mathbf{P}^{(\sigma)}$ is the corresponding permutation matrix, $\boldsymbol{y}_q$ are the true relevance scores, and $\boldsymbol{b}$ is a *position bias* vector which models the probability that each position is viewed by a user, defined with elements $b_j = 1/\log_2(1+j)$, for $j \in [n]$.

Ranking policy representation. The methods of this paper adopt a particular representation of the ranking policy, as bistochastic matrix $\boldsymbol{\Pi} \in \mathbb{R}^{n\times n}$, where $\boldsymbol{\Pi}_{jk}$ indicates the probability that item j takes position k in the ranking. The set of feasible ranking policies is expressed as $\boldsymbol{\Pi} \in \mathcal{B}$ where $\mathcal{B}$ is the *Birkhoff Polytope*:

$$\mathcal{B} = \{\boldsymbol{\Pi} \;\; s.t. \;\; \mathbf{1}^T\boldsymbol{\Pi} = \mathbf{1}, \;\; \boldsymbol{\Pi}\,\mathbf{1} = \mathbf{1}, \;\; \mathbf{0} \le \boldsymbol{\Pi} \le \mathbf{1}\}. \tag{3}$$

Its conditions on a matrix $\boldsymbol{\Pi}$ require, in the order of (3), that each column of $\boldsymbol{\Pi}$ sums to one, each row of $\boldsymbol{\Pi}$ sum to one, and each element of $\boldsymbol{\Pi}$ lie between 0 and 1. Each of these conditions is a linear constraint on the variables $\boldsymbol{\Pi}$.

Linearity of the DCG function (1) w.r.t. $\mathbf{P}$ allows it to commute with the expectation, leading to the practical closed form $u(\boldsymbol{\Pi}, \mathbf{y}) = \mathbf{y}^\top\boldsymbol{\Pi}\,\mathbf{b}$ for u as a linear function of $\boldsymbol{\Pi}$:

$$u(\boldsymbol{\Pi}, \mathbf{y}) = \mathbb{E}_{\sigma\sim\boldsymbol{\Pi}}\boldsymbol{\Delta}(\sigma, \mathbf{y}) = \mathbb{E}_{\sigma\sim\boldsymbol{\Pi}}\left[\mathbf{y}^\top\mathbf{P}^{(\sigma)}\,\mathbf{b}\right] = \mathbf{y}^\top\left(\mathbb{E}_{\sigma\sim\boldsymbol{\Pi}}\,\mathbf{P}^{(\sigma)}\right)\mathbf{b} = \mathbf{y}^\top\boldsymbol{\Pi}\,\mathbf{b}. \tag{4}$$

This is an important observation that enables the constrained optimization of utility functions on the policy $\boldsymbol{\Pi}$ in end-to-end differentiable pipelines, as discussed later in the paper.

2.2 Fairness of Exposure

Item exposure is commonly adopted in ranking systems, where items in higher ranking positions receive more exposure, and it is with respect to this metric that fairness is concerned. This paper aims at learning ranking policies that satisfy **group fairness of exposure**, while maintaining high relevance to user queries. The exposure $\mathcal{E}(i, \sigma)$ of item i within some ranking σ is a function of

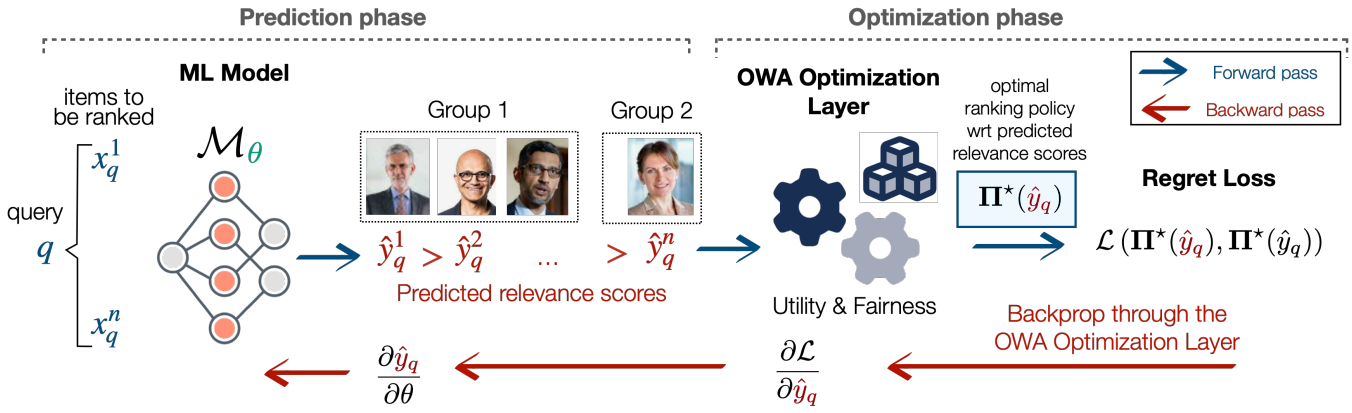

Figure 1: The differentiable optimization module proposed in SOFaiR. Its forward pass is calculated by an efficient Frank-Wolfe method, and its backward pass computes the SPO+ subgradient of the OWA problem's regret due to prediction error.

only its position, with higher positions receiving more exposure than lower ones. Throughout the paper, the common modeling choice $\mathcal{E}(i,\sigma) = b_{\sigma_i}$.

Notions of item exposure in rankings can also be extended to group exposure in ranking policies. The exposure of group g in ranking σ is measured by the mean exposure in σ of items belonging to g. The exposure of group g in ranking policy $\mathbf{\Pi}$ is the mean value of its exposure over all rankings sampled from the policy:

$$\mathcal{E}_g(\mathbf{\Pi}) = \mathbb{E}_{\substack{\sigma\sim\mathbf{\Pi}\\ i\sim[n]}}\left[\mathcal{E}(i,\sigma)\,|a_q^i = g\right], \tag{5}$$

and we let $\mathcal{E}_G(\mathbf{\Pi})$ be the vector of values (5) for each g in G. Derived similarly to (4), linearity of $\mathcal{E}$ leads to a closed form for (5) when $\mathbf{\Pi}$ is represented by a bistochastic matrix, where $\mathbf{1}_g$ indicates 1 for items in g and 0 elsewhere [22]:

$$\mathcal{E}_g(\mathbf{\Pi}) = \frac{1}{|g|}\mathbf{1}_g^T\mathbf{\Pi}\boldsymbol{b}. \tag{6}$$

Imposing fairness in LTR. It is well-known that individual rankings σ, as discrete structures, cannot exactly satisfy most notions of individual or group fairness [30]. Therefore a common strategy in fair ranking optimization is to view ranking policies as random distributions of rankings, upon which a feasible notion of fairness can be imposed *in expectation* [7, 22, 30]. For ranking policy $\mathbf{\Pi}$ and query q, fairness of exposure requires that every group indicated by $g \in G$ receives equal exposure on average over rankings produced by the policy. This condition can be expressed by requiring that the average exposure among items of each group is equal to the average exposure among all items:

$$\mathcal{E}_g(\mathbf{\Pi}) = \mathcal{E}_\alpha(\mathbf{\Pi}), \quad \forall g \in G, \tag{7}$$

where α is the group containing all items. Enforcing the condition (7) on each predicted policy $\mathbf{\Pi}$ is the mechanism by which protected groups are ensured equal exposure in SOFaiR. In the image search example, it corresponds to male and female candidates receiving equal exposure on average over rankings sampled from $\mathbf{\Pi}$. The violation of fairness with respect to group g is measured by the absolute gap in this condition:

$$v_g(\mathbf{\Pi}) = \left|\,\mathcal{E}_g(\mathbf{\Pi}) - \mathcal{E}_\alpha(\mathbf{\Pi})\,\right|. \tag{8}$$

Note that group fairness encompasses individual item fairness is a special case, where each item belongs to a distinct group. While the fairness and utility metrics described above are the ones used throughout the paper, the methodology of the paper is compatible with any alternative metrics u and $\mathcal{E}$ which are *linear* functions of the policy $\mathbf{\Pi}$. This is because the methodology of Sections 5 and 6 depend on linearity of (4) and (6).

3 LIMITATIONS OF FAIR LTR METHODS

Current Fair LTR models present a combination of the following limitations: **(A)** inability to ensure fairness in each of its generated policies, **(B)** inability or ineffectiveness to handle multiple protected groups, and **(C)** inefficiency at training and inference time. This section reviews current fair LTR methods in light of these limiting factors.

Regardless of how the policy is represented, fair learning to rank methods typically train a model $\mathcal{M}_\theta$ to find parameters θ^* that maximize its empirical utility, along with possibly a weighted penalty term F which promotes fairness:

$$\theta^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \frac{1}{N}\sum_{q=1}^{N} u(\mathcal{M}_{\boldsymbol{\theta}}(\boldsymbol{x}_q), \boldsymbol{y}_q) + \lambda \cdot F(\mathcal{M}_{\boldsymbol{\theta}}(\boldsymbol{x}_q)) \tag{9}$$

For example, the fair LTR method of [31] (called DELTR) is based on listwise learning to rank [4], and thus uses the model $\mathcal{M}_\theta$ to predict activation scores per each individual item, over which a softmax layer defines the probabilities of each item taking the top ranking position. Thus, [31] can only use F to encourage group fairness of exposure in the top position, leading to poor overall satisfaction of the fairness condition (7) (**limitation A**) as illustrated in Figure 2. To impose fairness over all ranking positions, [23] (FULTR) also uses softmax over the activations of $\mathcal{M}_\theta$ to define probabilities, which are sampled without replacement to generate rankings using a policy gradient method. However, this penalty-based method still does not ensure fairness in each predicted policy, as illustrated in figure 2, since the penalty is imposed only *on average* over all predicted policies (**limitation A**). By a similar reasoning, these methods do not translate naturally to the case of multigroup fairness, where $m > 2$ (**limitation B**): Because the penalty F must

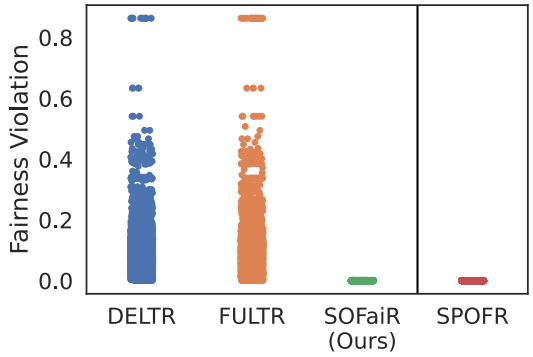

Figure 2: Yahoo-20: Fairness violation at query level.

scalarize the collection of all group fairness violations (8) (by taking their overall sum), it is possible to reduce F while increasing the exposure of a single outlier group [15].

Later work [15] shows how to overcome limitation A, by integrating the fair ranking optimization model of [22] together with prediction of relevance scores $\hat{\mathbf{y}}_q = \mathcal{M}_\theta$. The modeling of predicted policies $\mathbf{\Pi}$ as solutions to an optimization problem under fairness constraints allows for their representation as bistochastic matrices which satisfy the fairness notions (7) exactly. However, this method suffers **limitation C** as it requires to solve a linear programming problem at each iteration of training and at inference, whose number of variables in $\mathbf{\Pi} \in \mathbb{R}^{n \times n}$ scales quadratically as $O(n^2)$ becoming prohibitively large as the item list grows. Additionally, at inference time, the policy must be sampled to produce rankings; this requires a Birhoff-Von Neumann (BVM) decomposition of the matrix $\mathbf{\Pi}$ into a convex combination of permutation matrices, which is also expensive when n is large [22]. Finally, in the case of multiple groups ($m > 2$), the fairness constraints can become infeasible, making this formulation unwieldly (**limitation B**). An extended review of related work is provided in Appendix A.

Figure 2 shows the query-level fairness violations due to each method discussed in this section, where fairness parameters in each case are increased maximally without substantially compromising utility. In addition to higher average violations, penalty-based methods [23, 31] also lead to prevalence of outliers. These three existing fair LTR methods are used as baselines for comparison in Section 7. The SOFaiR framework proposed next most resembles [15], as it combines learning of relevance scores end-to-end with constrained optimization. At the same time, it aims to improve over [15] by addressing the three main limitations stated above. By integrating an alternative optimization component with its predictive model, SOFaiR can achieve faster runtime, and avoid the BVM decomposition at inference time, while naturally accommodating fairness over an arbitrary number of groups.

4 SMART OWA OPTIMIZATION FOR FAIR LEARNING TO RANK (SOFAIR)

This section provides an overview of the proposed SOFaiR framework for learning fair ranking policies that overcomes limitations A, B, and C. Sections 5 and 6 will then detail the core solution approaches required to incorporate its proposed fair ranking optimization module into efficient, end-to-end trainable fair LTR models. As illustrated in figure 1, SOFaiR's core concept is to integrate the learning of relevance scores with a module which optimizes fair ranking policies in-the-loop. By doing so it achieves a favorable balance of fairness and utility relative to other in-processing methods. The key difference in its approach relative to [15] is in the design of its optimization model which leverages Ordered Weighted Average (OWA) objectives (reviewed next) to enforce fairness of exposure. By avoiding the imposition of fairness of exposure (see Equation 7) as a set of hard constraints on the optimization as in [15, 22], it maintains the simple feasible region $\mathbf{\Pi} \in \mathcal{B}$, over which efficient Frank-Wolfe based solution methods can be employed to optimize its OWA objective function as described in Section 5. In turn, the particular form of the OWA optimization model in-the-loop necessitates a novel technique for its backpropagation, detailed in Section 6. The OWA aggregation and its fairness properties in optimization problems are introduced next, followed by its role in the SOFaiR learning framework.

4.1 Ordered Weighted Averaging Operator

The *Ordered Weighted Average* (OWA) operator [28] has found applications in various decision-making fields [29] as a means of fairly aggregating multiple objective criteria. Let $\mathbf{x} \in \mathbb{R}^m$ be a vector of m distinct criteria, and $\tau : \mathbb{R}^m \to \mathbb{R}^m$ be the sorting map for which $\tau(\mathbf{x}) \in \mathbb{R}^m$ holds the elements of $\mathbf{x}$ in increasing order. Then for any $\mathbf{w}$ satisfying $\{\mathbf{w} \in \mathbb{R}^m : \sum_i w_i = 1, \mathbf{w} \geq 0\}$, the OWA aggregation with weight $\mathbf{w}$ is defined as a linear functional on $\tau(\mathbf{x})$:

$$\text{OWA}_\mathbf{w}(\mathbf{x}) = \mathbf{w}^T \tau(\mathbf{x}), \tag{10}$$

which is convex and piecewise-linear in $\mathbf{x}$ [18]. The so-called Generalized Gini Functions, or Fair OWA, are those for which the OWA weights $w_1 > w_2 \ldots > w_n$ are decreasing. Fair OWA functions possess the following three key properties for fairness in optimizing multiple criteria [18]. **(1)** *Impartiality* means that all criteria are treated equally, in the sense that $\text{OWA}_\mathbf{w}(\mathbf{x}) = \text{OWA}_\mathbf{w}(\mathbf{x}_\sigma)$ for any $\sigma \in \mathcal{P}_m$. **(2)** *Equitability* is the property that marginal transfers from a criterion with higher value to one with lower value results in an increase in aggregated OWA value. That is , when $x_i > x_j + \epsilon$ and letting $\mathbf{x}_\epsilon = \mathbf{x}$ except at positions i and j where $(\mathbf{x}_\epsilon)_i = \mathbf{x}_i - \epsilon$ and $(\mathbf{x}_\epsilon)_j = \mathbf{x}_j + \epsilon$, it holds that $\text{OWA}_\mathbf{w}(\mathbf{x}_\epsilon) > \text{OWA}_\mathbf{w}(\mathbf{x})$. **(3)** *Monotonicity* means that $\text{OWA}_\mathbf{w}(\mathbf{x})$ is an increasing function of each element of $\mathbf{x}$. The monotonicity property implies that solutions which optimize (10) are Pareto Efficient solutions of the underlying multiobjective problem, thus that no single criteria can be raised without reducing another [18]. Taken together, it is known that maximization of aggregation functions which satisfy these three properties produces so-called *equitably efficient solutions*, which possess the main intuitive properties needed for a solution to be deemed "fair"; see [13] for a formal definition. As shown next, the SOFaiR framework ensures group fairness by leveraging a fair OWA aggregation of group exposures $\text{OWA}_\mathbf{w}(\mathcal{E}_G(\mathbf{\Pi}))$ in the objective function of its integrated fair ranking optimization module.

4.2 End-to-End Learning in SOFaiR

As illustrated in figure 1, the SOFaiR framework uses a prediction model $\mathcal{M}_\theta$ with learnable weights θ, which produces relevance scores $\hat{\boldsymbol{y}}_q$ from a list of item features $\mathbf{x}_q$. Its key component is an optimization module which maps the prediction $\hat{\boldsymbol{y}}_q$ to an associated ranking policy $\boldsymbol{\Pi}^\star(\hat{\boldsymbol{y}}_q)$. The following optimization problem defines $\boldsymbol{\Pi}^\star(\hat{\boldsymbol{y}}_q)$ as the ranking policy which optimizes a trade-off between fair OWA aggregation of group exposures with the expected DCG (as per Equation 4) under relevance scores $\hat{\boldsymbol{y}}_q$. In SOFaiR, it defines, for any chosen weight $0 \leq \lambda \leq 1$, a mapping which can be viewed akin to a neural network layer, representing the last layer of $\mathcal{M}$:

$$\boldsymbol{\Pi}^\star(\hat{\boldsymbol{y}}_q) = \operatorname{argmax}_{\boldsymbol{\Pi}\in\mathcal{B}} \ (1-\lambda)\cdot u(\boldsymbol{\Pi}, \hat{\boldsymbol{y}}_q) + \lambda \cdot \mathrm{OWA}_{\boldsymbol{w}}(\mathcal{E}_G(\boldsymbol{\Pi})), \quad (11)$$

wherein the Birkhoff Polytope $\mathcal{B}$ is the set of all bistochastic matrices, as defined in Equation 3. Let the objective function of Equation 11 be named $f(\boldsymbol{\Pi}, \hat{\boldsymbol{y}}_q)$. It is a convex combination of two terms measuring user utility and fairness, whose trade-off is controlled by a single coefficient $0 \leq \lambda \leq 1$. The former term measures expected user utility $u(\boldsymbol{\Pi}, \hat{\boldsymbol{y}}_q) = \hat{\boldsymbol{y}}_q^\top \boldsymbol{\Pi}\, \boldsymbol{b}$, while the latter term measures OWA aggregation of the group exposures. It is intuitive to see that when $\lambda = 1$, the optimization (11) returns a ranking policy that minimizes disparities in group exposure, without regard for relevance. When $\lambda = 0$, it returns a deterministic policy which ranks the items in order of the estimated scores $\hat{\boldsymbol{y}}_q$. Intermediate values $0 < \lambda < 1$ result in policies which trade off the effects of each term, balancing utility and fairness to various degrees. As λ increases, disparity between the exposure of protected groups must decrease; this leads to a practical mechanism for achieving a desired level of fairness with minimal compromise to utility.

Since Equation 11 defines a direct mapping from $\hat{\boldsymbol{y}}_q$ to $\boldsymbol{\Pi}^\star(\hat{\boldsymbol{y}}_q)$, the problem of learning fair ranking policies reduces to a problem of learning relevance scores. This corresponds to estimating the objective function f via its missing coefficients $\boldsymbol{y}_q$. The SOFaiR training method defines a loss function between predicted and ground-truth relevance scores, as the loss of optimality in $\boldsymbol{\Pi}^\star(\hat{\boldsymbol{y}}_q)$ with respect to objective f under ground-truth $\boldsymbol{y}_q$, caused by prediction error in $\hat{\boldsymbol{y}}_q$. That is, the training objective is to minimize *regret* in f induced by $\hat{\boldsymbol{y}}_q$, defined as:

$$\mathit{regret}(\hat{\boldsymbol{y}}_q, \boldsymbol{y}_q) = f(\ \Pi^\star(\boldsymbol{y}_q), \boldsymbol{y}_q\ ) - f(\ \Pi^\star(\hat{\boldsymbol{y}}_q), \boldsymbol{y}_q\ ). \quad (12)$$

The composition $\boldsymbol{\Pi}^\star \circ \mathcal{M}_\theta$ defines an integrated prediction and optimization model which maps item features to fair ranking policies. Training the integrated model by stochastic gradient descent follows these steps in a single iteration:

(1) For sample query q and item features $\mathbf{x}_q$, a predictive model $\mathcal{M}_\theta$ produces estimated relevance scores $\hat{\mathrm{y}}_q$.
(2) The predicted scores $\hat{\mathrm{y}}_q$ are used to populate the unknown parameters of an optimization problem (11). A solution algorithm is employed to find $\Pi^*(\hat{\mathrm{y}}_q)$, the optimal fair ranking policy relative to $\hat{\mathrm{y}}_q$.
(3) The regret loss (12) is backpropagated through the calculations of steps (1) and (2), in order to update the model weights θ by a gradient descent step.

The following sections detail the main solution schemes for implementing steps (2) and (3). Section 5 shows how recently proposed fair ranking optimization techniques from [7] can be adapted to the setting of this paper, in which fair ranking policies must be learned from empirical data. From this choice of optimization design arises a novel challenge in the backpropagation step (3), since no known work has shown how to backpropagate the regret of a highly discontinuous OWA optimization program. Section 6 shows how to efficiently backpropagate the regret due to Equation 11 for end-to-end learning. Then, Section 7 evaluates the SOFaiR framework against several other methods for learning fair ranking policies, on a set of benchmark tasks from the web search domain.

5 FORWARD PASS OPTIMIZATION

The main motivation for the formulation (11) of SOFaiR's fair ranking optimization layer is to render the optimization problem efficiently solvable. Its main exploitable attribute is its feasible region $\boldsymbol{\Pi}$, over which a *linear* objective function can be quickly optimized by simply sorting a vector in $\mathbb{R}^n$, which has time complexity $n \log n$ [5]. This suggests an efficient solution by Frank-Wolfe methods, which solve a constrained optimization problem by a sequence of subproblems optimizing a linear approximation of the true objective function [1]. This efficient solution pattern is made possible by the absence of additional group fairness constraints on the policy variable $\boldsymbol{\Pi}$.

Frank-Wolfe methods solve a convex constrained optimization problem $\operatorname{argmax}_{\mathbf{x}\in\mathbb{S}} f(\mathbf{x})$ by computing the iterations

$$\mathbf{x}^{(k+1)} = (1-\alpha^{(k)})\mathbf{x}^{(k)} + \alpha^{(k)} \operatorname*{argmax}_{\mathrm{y}\in\mathbb{S}} \langle \mathrm{y}, \nabla f(\mathbf{x}^{(k)})\rangle. \quad (13)$$

Convergence to an optimal solution is guaranteed when f is *differentiable* and with $\alpha^{(k)} = \frac{2}{k+2}$ [1]. However, the main obstruction to solving (11) by the method (13) is that f in our case includes a *non-differentiable* OWA function. A path forward is shown in [16], which shows convergence can be guaranteed by optimizing a smooth surrogate function $f^{(k)}$ in place of the nondifferentiable f at each step of (13), in such a way that the $f^{(k)}$ converge to the true f as $k \to \infty$.

It is proposed in [7] to solve a two-sided fair ranking optimization with OWA objective terms, by the method of [16], where $f^{(k)}$ is chosen to be a Moreau envelope h^{β_k} of f, a $\frac{1}{\beta_k}$-smooth approximation of f defined as [1]:

$$h^\beta(\mathbf{x}) = \min_{\mathrm{y}} f(\mathrm{y}) + \frac{1}{2\beta}\|\mathrm{y} - \mathbf{x}\|^2. \quad (14)$$

When $f = \mathrm{OWA}_{\boldsymbol{w}}$, let its Moreau envelope be denoted $\nabla\mathrm{OWA}^\beta_{\boldsymbol{w}}$; it is shown in [7] that its gradient can be computed as a projection onto the permutahedron induced by modified OWA weights $\tilde{\boldsymbol{w}} = -(w_m, \ldots, w_1)$. By definition, the permutahedron $C(\tilde{\boldsymbol{w}}) = \textsc{conv}(\{\boldsymbol{w}_\sigma : \forall\sigma \in \mathcal{P}_m\})$ induced by a vector $\tilde{\boldsymbol{w}}$ is the convex hull of all its permutations. In turn, it is shown in [3] that the permutahedral projection $\nabla\mathrm{OWA}^\beta_{\boldsymbol{w}}(\mathbf{x}) = \mathrm{proj}_{C(\tilde{\boldsymbol{w}})}(\boldsymbol{x}/\beta)$ can be computed in $m \log m$ time as the solution to an isotonic regression problem using the Pool Adjacent Violators algorithm. To find the overall gradient of $\mathrm{OWA}^\beta_{\boldsymbol{w}}$ with respect to optimization variables $\boldsymbol{\Pi}$, a convenient form can be derived from the chain rule:

$$\nabla_{\boldsymbol{\Pi}}\ \mathrm{OWA}^\beta_{\boldsymbol{w}}(\mathcal{E}(\boldsymbol{\Pi})) = \boldsymbol{\mu}\boldsymbol{b}^T. \quad (15)$$

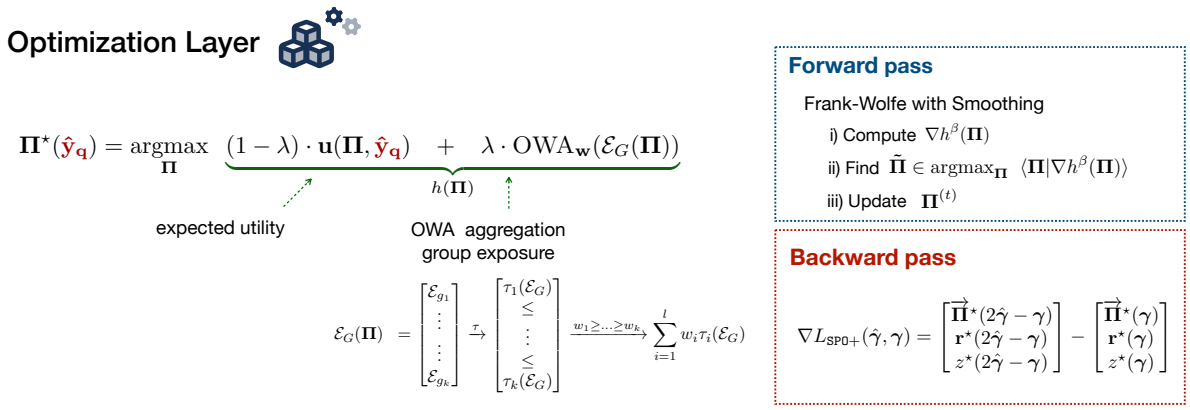

Figure 3: The differentiable optimization module employed in SOFaiR. It forward pass solves the problem (11) by an efficent Frank-Wolfe method. Its backward pass calculates the SPO+ subgradient, relative to its equivalent, but intractably large LP form.

where $\boldsymbol{\mu} = \mathrm{proj}_{C(\tilde{\boldsymbol{w}})}(\mathcal{E}(\mathbf{\Pi})/\beta)$ and $\mathcal{E}(\mathbf{\Pi})$ is the vector of all item exposures [7]. For the case where group exposures $\mathcal{E}_G(\mathbf{\Pi})$ are aggregated by OWA, first note that by Equation 6, $\mathcal{E}_G(\mathbf{\Pi}) = \boldsymbol{A}\mathbf{\Pi}\boldsymbol{b}$, where $\boldsymbol{A}$ is the matrix composed of stacking together all group indicator vectors $\mathbf{1}_g \;\; \forall g \in G$. Since $\mathcal{E}(\mathbf{\Pi}) = \mathbf{\Pi}\boldsymbol{b}$, this implies $\mathcal{E}_G(\mathbf{\Pi}) = \mathcal{E}(\boldsymbol{A}\mathbf{\Pi})$, thus

$$\nabla_{\mathbf{\Pi}} \; \mathrm{OWA}^{\beta}_{\boldsymbol{w}}(\mathcal{E}_G(\mathbf{\Pi})) = (\boldsymbol{A}^T \tilde{\boldsymbol{\mu}}) \; \boldsymbol{b}^T. \tag{16}$$

by the chain rule, and where $\tilde{\boldsymbol{\mu}} = \mathrm{proj}_{C(\tilde{\boldsymbol{w}})}(\mathcal{E}_G(\boldsymbol{A}\mathbf{\Pi})/\beta)$. It remains now to compute the gradient of the user relevance term $u(\mathbf{\Pi}, \hat{\boldsymbol{y}}_q) = \hat{\boldsymbol{y}}_q^T \mathbf{\Pi} \;\; \boldsymbol{b}$ in Equation 11. As a linear function of the matrix variable $\mathbf{\Pi}$, its gradient is $\nabla_{\mathbf{\Pi}} \;\; u(\mathbf{\Pi}, \hat{\boldsymbol{y}}_q) = \hat{\boldsymbol{y}}_q \;\; \boldsymbol{b}^T$, which is evident by comparing to the equivalent vectorized form $\hat{\boldsymbol{y}}_q^T \mathbf{\Pi} \; \boldsymbol{b} = \overrightarrow{\hat{\boldsymbol{y}}_q \, \boldsymbol{b}^T} \cdot \overrightarrow{\mathbf{\Pi}}$. Combining this with Equation 16, the total gradient of the objective function in Equation 11 with smoothed OWA term is $(1-\lambda) \cdot \hat{\boldsymbol{y}}_q \;\; \boldsymbol{b}^T + \lambda \cdot (\boldsymbol{A}^T \tilde{\boldsymbol{\mu}}) \;\; \boldsymbol{b}^T$, which is equal to $\left((1-\lambda) \cdot \hat{\boldsymbol{y}}_q + \lambda \cdot (\boldsymbol{A}^T \tilde{\boldsymbol{\mu}})\right) \;\; \boldsymbol{b}^T$. Therefore the SOFaiR module's Frank-Wolfe linearized subproblem is

$$\underset{\mathbf{\Pi} \in \mathcal{B}}{\operatorname{argmax}} \left\langle \mathbf{\Pi}, \left((1-\lambda) \cdot \hat{\boldsymbol{y}}_q + \lambda \cdot (\boldsymbol{A}^T \tilde{\boldsymbol{\mu}})\right) \;\; \boldsymbol{b}^T \right\rangle \tag{17}$$

To implement the Frank-Wolfe iteration (13), this linearized subproblem should have an efficient solution. To this end, the form of each gradient above as a cross-product of some vector with the position biases $\boldsymbol{b}$ can be exploited. Note that as the expected DCG under relevance scores $\boldsymbol{y}$, the function $\boldsymbol{y}^T \mathbf{\Pi} \; \boldsymbol{b}$ is maximized by the permutation matrix $\boldsymbol{P} \in \mathcal{P}_n$ which sorts the relevance scores $\boldsymbol{y}$ decreasingly. But since $\boldsymbol{y}^T \mathbf{\Pi} \; \boldsymbol{b} = \overrightarrow{\boldsymbol{y} \, \boldsymbol{b}^T} \cdot \overrightarrow{\mathbf{\Pi}}$, we identify $\boldsymbol{y}^T \mathbf{\Pi} \; \boldsymbol{b}$ as the linear function of $\overrightarrow{\mathbf{\Pi}}$ with gradient $\overrightarrow{\boldsymbol{y} \, \boldsymbol{b}^T}$. Therefore equation Equation 17 can be solved in $O(n \log n)$, simply by finding $\boldsymbol{P} \in \mathcal{P}_n$ as the argsort of the vector $((1-\lambda) \cdot \hat{\boldsymbol{y}}_q + \lambda \cdot (\boldsymbol{A}^T \tilde{\boldsymbol{\mu}}))$ in decreasing order. A more formal proof, cited in [6], makes use of [11].

The overall method is presented in Algorithm 1. Decay of the smoothing parameter $\beta_t = \frac{\beta_0}{\sqrt{t}}$ satisfies the conditions for convergence stated in [16] when β_0 is sufficiently large. Sparse matrix additions each require $O(n)$ operations, so that Algorithm 1 maintains $O(n \log n)$ complexity per iteration. An important advantage of Algorithm 1 over the fair ranking LP employed in SPOFR [15], is that the solution iterates $\boldsymbol{P}^{(k)}$ automatically provide a decomposition of the policy matrix $\mathbf{\Pi} = \rho_k \boldsymbol{P}^{(k)}$ as a convex combination of rankings, by which it can be readily sampled as a discrete probability distribution. In contrast, the LP module used in SPOFR [15] provides as its solution only a matrix $\mathbf{\Pi} \in \mathcal{B}$, which must be decomposed using the Birkhoff Von Neumann decomposition, adding substantially to its total runtime.

Algorithm 1: Frank-Wolfe with Moreau Envelope Smoothing to solve (11)

Input: predicted relevance scores $\hat{\boldsymbol{y}} \in \mathbb{R}^n$, group mask $\boldsymbol{A}$, max iteration T, smooth seq. (β_k)
Output: ranking policy $\mathbf{\Pi}^{(T)} \in \mathbb{R}^{n \times n}$

Initialize $\mathbf{\Pi}^{(0)}$ as $\boldsymbol{P} \in \mathcal{P}$ which sorts $\hat{\boldsymbol{y}}$ in decreasing order;
for $k = 1, \ldots, T$ **do**
  $\tilde{\boldsymbol{\mu}} \leftarrow \mathrm{proj}_{C(\tilde{\boldsymbol{w}})}(\mathcal{E}_G(\boldsymbol{A}\mathbf{\Pi})/\beta_k)$;
  $\hat{\boldsymbol{\mu}} \leftarrow (1-\lambda) \cdot \hat{\boldsymbol{y}}_q + \lambda \cdot (\boldsymbol{A}^T \tilde{\boldsymbol{\mu}})$;
  $\hat{\sigma} \leftarrow argsort(-\hat{\boldsymbol{\mu}})$;
  Let $\boldsymbol{P}^{(k)} \in \mathcal{P}$ such that $\boldsymbol{P}^{(k)}$ represents $\hat{\sigma}$;
  $\mathbf{\Pi}^{(k)} \leftarrow \frac{k}{k+2} \mathbf{\Pi}^{(k-1)} + \frac{2}{k+2} \boldsymbol{P}^{(k)}$;
Return $\mathbf{\Pi}^{(T)}$;

6 BACKPROPAGATION

The formulation of the optimization module (11) allows for efficient solution via Algorithm 1, but gives rise to a novel challenge in backpropagating the regret loss function through $\Pi^\star(\hat{\boldsymbol{y}}_q)$. By including an OWA aggregation of group exposure, its objective function is nonlinear and nondifferentiable. This section shows how to train the integrated prediction and OWA optimization model $\mathbf{\Pi}^\star \circ \mathcal{M}_\theta$ to minimize the regret loss (12), despite this challenge.

As a starting point, we recognize the existing literature on "Predict-Then-Optimize" frameworks [14, 17] for minimizing the regret due to prediction error in the objective coefficients of a linear program, denoted $\boldsymbol{c}$ below:

$$\mathbf{x}^\star(\mathbf{c}) = \underset{\mathbf{Ax}\leq\mathbf{b}}{\operatorname{argmin}}\, \mathbf{c}^T\mathbf{x}. \tag{18}$$

Several known methods have been proposed [2, 10, 21, 25] and well-established in the literature [17] for end-to-end training of combined prediction and optimization models employing (18). Due to its OWA objective term, the fair ranking module (11) does not satisfy the LP form (18) for which the aforementioned methods are taylored. The implementation of SOFaiR described here uses the "Smart Predict-Then-Optimize" (SPO) approach [10], since its simple backpropagation rule requires only a solution to (18) using a blackbox solution oracle. This allows its adaptation to the OWA optimization setting by constructing (but not solving) an equivalent but intractable linear programming form to (11), as shown next.

End-to-End learning with SPO+ Loss. Viewed as a loss function, the regret (12) in solutions to problem (18) is nondifferentiable and discontinuous with respect to predicted coefficients $\hat{\mathbf{c}}$, since solutions $\boldsymbol{x}^\star(\boldsymbol{c})$ must occur at one of finitely many vertices in $\mathbf{Ax} \leq \mathbf{b}$. The SPO+ loss function proposed in [10] is by construction a Fischer-consistent, *subdifferentiable* upper bound on regret. In particular, it is shown in [10] that

$$L_{\mathsf{SPO+}}(\hat{\boldsymbol{c}}, \boldsymbol{c}) = \max_{\mathbf{x}}(\mathbf{c}^T\mathbf{x} - 2\hat{\mathbf{c}}^T\mathbf{x}) + 2\hat{\mathbf{c}}^T\mathbf{x}^\star(\mathbf{c}) - \mathbf{c}^T\mathbf{x}^\star(\mathbf{c}), \tag{19}$$

possesses these properties, and a subgradient at $\hat{\mathbf{c}}$ is

$$\nabla L_{\mathsf{SPO+}}(\hat{\boldsymbol{c}}, \boldsymbol{c}) = \boldsymbol{x}^\star(2\hat{\boldsymbol{c}} - \boldsymbol{c}) - \boldsymbol{x}^\star(\boldsymbol{c}). \tag{20}$$

Minimizing the surrogate loss (19) by gradient descent using (20) is key to minimizing the solution regret in problem (18) due to error in a predictive model which predicts the parameter $\mathbf{c}$.

SPO+ loss in SOFaiR. We now show how the SPO training framework described above for the problem type (18) can be used to efficiently learn optimal fair policies in conjunction with problem (11). The main idea is to derive an SPO+ subgradient for regret in (11), through an equivalent linear program (18), but without solving it as such. This is made possible by the fact that the subgradient (20) can be expressed as a difference of two optimal solutions, which can be furnished by any optimization oracle which solves the mapping (11), which includes Algorithm (1).

First note, as it is shown in [18], that the OWA function (10) can be expressed as

$$OWA_{\mathbf{w}}(\mathbf{r}) = \min_{\sigma\in\mathcal{P}} \ \mathbf{w}_\sigma \cdot \mathbf{r}, \tag{21}$$

and as an equivalent linear programming problem which views the minimum inner product above as the maximum lower bound among all possible inner products with the permuted OWA weights:

$$OWA_{\mathbf{w}}(\mathbf{r}) = \max_z \quad z \tag{22a}$$

$$s.t. \quad z \leq \mathbf{w}_\sigma \cdot \mathbf{r}, \qquad \forall\sigma \in \mathcal{P}, \tag{22b}$$

where $\mathcal{P}$ contains all possible permutations of $[n]$ when $\mathbf{w} \in \mathbb{R}^n$. This allows SOFaiR's OWA optimization model (11) to be recast in a linear programming form using auxiliary optimization variables $\mathbf{r}$ and z:

$$(\boldsymbol{\Pi}^\star, \boldsymbol{r}^\star, z^\star)(\hat{\boldsymbol{y}}_q) = \operatorname{argmax}_{\boldsymbol{\Pi}\in\mathcal{B},\, \mathbf{r},\, z} \ (1-\lambda)\cdot\hat{\boldsymbol{y}}_q^\top\,\boldsymbol{\Pi}\,\boldsymbol{b} + \lambda\cdot z \tag{23a}$$

$$\text{subject to:} \quad z \leq \boldsymbol{w}_\sigma \cdot r, \qquad \forall\sigma\in\mathcal{P} \tag{23b}$$

$$r = \mathcal{E}_G(\boldsymbol{\Pi}). \tag{23c}$$

According to [18], this alternative LP form of OWA optimization is mostly of theoretical significance, since the set of constraints (23b) grows factorially in the size of $\mathbf{r}$, one for each possible permutation thereof. This makes (23) impractical for computing a solution to the original OWA problem (11), which we instead solve by Algorithm 1. On the other hand, we show that problem (23) *is* practical for deriving a *backpropagation* rule through the OWA problem (11).

Since the unknown parameters $\hat{\boldsymbol{y}}_q$ appear only in its linear objective function, this parametric LP problem (23) fits the form (18) required for training with SPO+ subgradients. To derive the subgradient explicitly, rewrite the linear objective term $\hat{\boldsymbol{y}}_q^\top\,\boldsymbol{\Pi}\,\boldsymbol{b} = \overrightarrow{\hat{\boldsymbol{y}}_q\,\boldsymbol{b}^T}\cdot\overrightarrow{\boldsymbol{\Pi}}$. Then in terms of the augmented variables $(\boldsymbol{\Pi}, \mathbf{r}, z)$, the objective function (23a) is

$$(1-\lambda)_x\cdot\hat{\boldsymbol{y}}_q^\top\,\boldsymbol{\Pi}\,\boldsymbol{b} + \lambda\cdot z = \underbrace{\begin{bmatrix}(1-\lambda)\overrightarrow{\hat{\boldsymbol{y}}_q\,\boldsymbol{b}^T}\\ \mathbf{0}\\ \lambda\end{bmatrix}^T}_{\hat{\boldsymbol{\gamma}}}\begin{bmatrix}\overrightarrow{\boldsymbol{\Pi}}\\ \mathbf{r}\\ z\end{bmatrix}. \tag{24}$$

Now the SPO+ loss subgradient can be readily expressed with respect to the augmented scores $\hat{\boldsymbol{\gamma}}$ defined as above:

$$\nabla L_{\mathsf{SPO+}}(\hat{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) = \begin{bmatrix}\overrightarrow{\boldsymbol{\Pi}^\star(2\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma})}\\ \mathbf{r}^\star(2\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma})\\ z^\star(2\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma})\end{bmatrix} - \begin{bmatrix}\overrightarrow{\boldsymbol{\Pi}^\star(\boldsymbol{\gamma})}\\ \mathbf{r}^\star(\boldsymbol{\gamma})\\ z^\star(\boldsymbol{\gamma})\end{bmatrix}, \tag{25}$$

using (20), and where $\boldsymbol{\gamma}$ is the augmented score based on ground-truth $\mathbf{y}_q$. Finally, backpropagation from $\hat{\boldsymbol{\gamma}}$ to the base prediction $\hat{\mathbf{y}}_q = \mathcal{M}_{\boldsymbol{\theta}}(\mathbf{x}_q)$ is performed by automatic differentiation, and likewise from $\hat{\mathbf{y}}_q$ to the model weights $\boldsymbol{\theta}$.

Both terms in (25) can be produced by using Algorithm 1 to solve (11) for $\boldsymbol{\Pi}^\star$. Then, the remaining variables $\mathbf{r}^\star$ and $z^\star$ are easily completed as groups exposures $\mathbf{r} = \mathcal{E}_G(\Pi^\star)$ and their associated OWA value z, respectively. Importantly, the rightmost term of (25) is independent of any prediction; therefore it is *precomputed* in advance of training. Thus, backpropagation using (25) consists of computing the difference between two solutions, one of which comes from the forward pass and while the other is precomputed before training. The complexity of this backward pass consists of $O(n^2)$ subtractions, which grows only linearly in the size of the matrix variable $\boldsymbol{\Pi} \in \mathbb{R}^{n\times n}$. The differentiable fair ranking optimization module of SOFaiR, with its forward and backward passes, is summarized in figure 3.

7 EXPERIMENTS

Next we evaluate SOFaiR against two prior in-processing methods [23, 30], and the end-to-end framework [14], denoted as FULTR, DELTR, and SPOFR, respectively. We assess the performance on two datasets:

- Microsoft Learn to Rank (MSLR) is a standard benchmark for LTR with queries from Bing and manually-judged relevance labels. It includes 30,000 queries, each with an average of 125 assessed documents and 136 ranking feature. Binary protected groups is defined using the 50th percentile of QualityScore attribute. For multi-group cases, group labels are defined using evenly-spaced quantiles.
- Yahoo! Learning to Rank Challenge (Yahoo LETOR) contains 19,944 queries and 473,134 documents with 519 ranking features. Binary protected groups is defined using feature id 9 following [12] and the 50th percentile as the threshold.

For MSLR, we randomly sample 10,000 queries for training and 1,000 queries each for validation and testing. We create datasets with varying list sizes (20, 40, 60, 80, 100 documents) for MSLR and (20, 40 documents) for Yahoo LETOR.

Models and hyperparameters. A neural network (NN) with three hidden layers is trained using Adam Optimizer with a learning rate of 0.1 and a batch size of 256. The size of each layer is halved, and the output is a scalar item score. Results of each hyperparameter setting is are taken on average over five random seeds.

Fairness parameters, considered as hyperparameters, are treated differently. LTR systems aim to offer a trade-off between utility and group fairness, since the cost of increased fairness results in decreased utility. In DELTR, FULTR, and SOFaiR, this trade-off is indirectly controlled through the fairness weight, denoted as λ in (9) and (11). Larger values of λ indicate more preference towards fairness. In SPOFR, the allowed violation (8) of group fairness is specified directly. Ranking utility and fairness violation are assessed using average DCG (Equation 1) and fairness violation (Equation 8), respectively. The metrics are computed as averages over the entire test dataset.

7.1 Running Time Analysis

We start by comparing the runtime of SOFaiR with other LTR frameworks, emphasizing its efficiency in training and inference (**limitation C**), particularly in comparison to SPOFR. Figure 4 shows the average training and inference time per query for each method, focusing on the binary group MSLR dataset across various list sizes. First notice the drastic runtime reduction of SOFaiR compared to SPOFR, both during training and inference. While SPOFR's training time exponentially increases with the ranking list size, SOFaiR's runtime increases only moderately, reaching over one order of magnitude speedup over SPOFR for large list sizes. Notably, the number of iterations of Algorithm 1 required for sufficient accuracy in training to compute SPO+ subgradients are found to less than those required for solution of (11) at inference. Thus the reported results use 100 iterations in training and 500 at inference. Importantly, reported runtimes under-estimate the efficiency gained by SOFaiR, since its PyTorch [19] implementation in Python is compared against the highly optimized code implementation of Google OR-Tools solver [20]. DELTR and FULTR, as penalty-based methods, are more competitive in runtime. However, this comes at a cost of the achieved fairness level (**limitation A**), as shown in the next section.

7.2 Fairness and Utility Tradeoffs Analysis

Next, we focus on comparing the utility and fairness of the various LTR frameworks analyzed. This section focuses on the two-group case, as none of the methods compared against was able to cope with multi-group case in our experiments (see next section). Figure 5 presents the trade-off between utility and fairness across the test sets for both Yahoo LETOR and MSLR datasets, encompassing their lowest and highest list sizes. For each method, the intensity of colors represents the magnitude of its fairness parameter. A progression from lighter to darker colors indicates an increase in the importance placed on fairness. Consequently, darker colors are expected to correspond with more restrictive models, characterized by lower DCG scores (y-axis) but also fewer fairness violations (x-axis). Each point in the figure represents the largest DCG score obtained from a fairness hyperparameter search, as detailed in Appendix C. Note that points on the grid that are higher on the y-axis and lower on the x-axis represent superior results.

Firstly, notice that most points associated with methods DELTR and FULTR are clustered in a small region with both high DCG and (log-scaled) fairness violations. While these methods reach an order of magnitude reduction in fairness violation on some datasets, the effect is inconsistent, especially as the item list size increases (**limitation A**). In contrast, the end-to-end methods (SPOFR and the proposed SOFaiR) reach much lower fairness violations, underlining their effectiveness of their optimization modules in enforcing the fairness constraint.

Both DELTR and FULTR reach competitive utilities, but they consistently display relatively high fairness violations, underscoring their limitations in providing a fair ranking solution. SOFaiR shows competitive fairness and utility performance compared to SPOFR, with a marked advantage in utility on some datasets. SPOFR ensures fairness but at the expense of efficiency, whereas SOFaiR reaches similar fairness levels at a fraction of the required runtime. Additional results on datasets of various list sizes are included in Appendix B.2.

7.3 Multi-Group Fairness Analysis

Finally, this section analyzes the fairness-utility trade-off in multi-group scenarios using the SOFaiR framework. The SPOFR method returns infeasible solutions for most chosen fairness levels when multiple groups are introduced, preventing its evaluation on these datasets; this is naturally avoided in SOFaiR as the optimization of OWA aggregation without constraints simply increases fairness to the extent feasible. While FULTR provides no code to evaluate multigroup fairness, its penalty function is in principle ill-equipped to handle multiple groups as it must scalarize all group fairness violations into a single loss function as mentioned in Section 3 (**limitation C**). Figure 6 compares the average test DCG against the average fairness violation across various numbers of groups (ranging from 3 to 7) in the MSLR dataset, for list sizes of 40 and 100. Additional results for other list sizes in the MSLR dataset are available in Appendix B.1.

Each data point represents a single model's performance, with fairness parameters λ adjusted between 0 and 1. Models prioritizing fairness show reduced fairness violations and lower utilities, indicated by darker colored points, compared to those with a lower

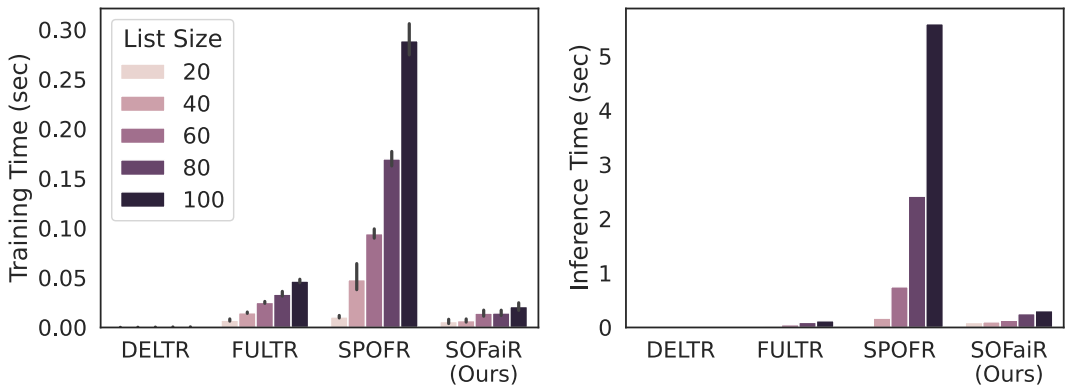

Figure 4: Running time benchmark on MSLR-Web10k dataset

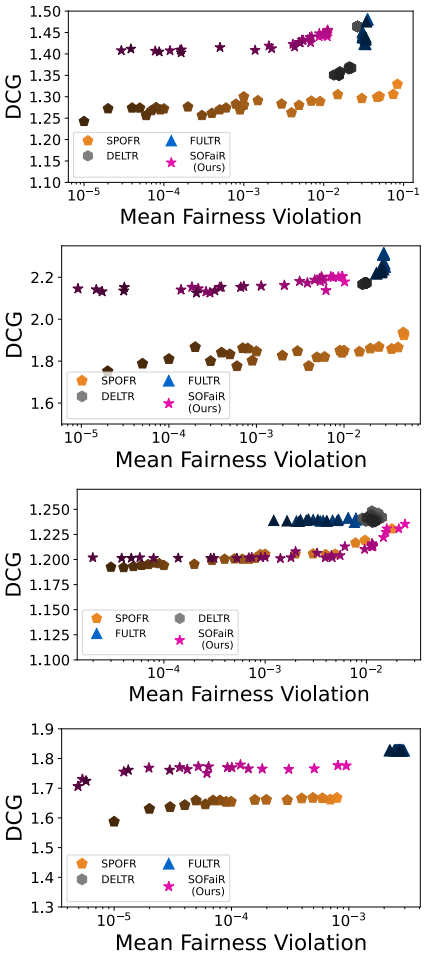

Figure 5: Benchmarking performance in term of fairnesss-utility trade-off on Yahoo-20 (top left), and Yahoo-40(top right). MSLR-20(bottom-left), MSLR-100 (bottom-right)

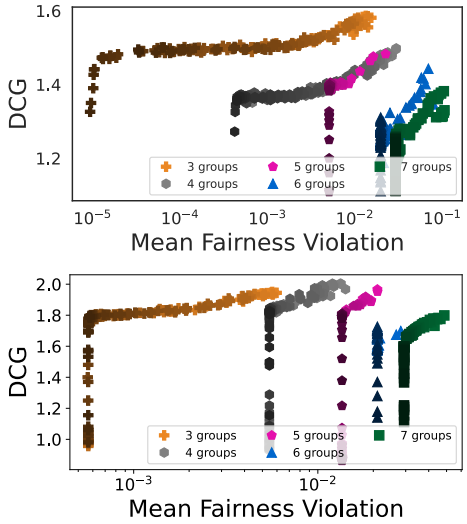

Figure 6: Fairness-utility tradeoff due to SOFaiR with multiple groups on MSLR-40 (left) and MSLR-100 (right) list size

emphasis on fairness, represented by lighter colored points. A distinct trend is observed: as fairness parameters are relaxed, utility increases for all metrics and datasets. It is also evident that multi-group fairness comes at a higher cost to utility. Predictably, saturation occurs in each curve, indicating that beyond a certain point, increasing the fairness weight does not further decrease fairness violations but merely reduces utility.

8 CONCLUSIONS

This paper presented SOFaiR, a method that employs an Ordered Weighted Average optimization model to integrate fairness considerations into ranking processes. Its key contribution is to enable backpropagation through optimization of discontinuous OWA functions, which has makes it possible to precisely enforce flexible group fairness measures directly into the training process of learning to rank, without greatly compromising efficiency. These advantages show that by leveraging modern developments in fair ranking optimization, the integration of constrained optimization and machine

learning techniques can be a promising direction for future research in fair LTR.

9 ETHICAL STATEMENT

This paper was developed on commonly used, open benchmark datasets for learning to rank, and no sensitive data was used in the production of its experiments. As is common in research on fair ranking systems, protected groups were defined on the basis of attributes contained in these datasets, in order to best evaluate the performance of the algorithms. The authors' intended contribution is purely methodological, aimed at enhancing the performance of ranking systems with respect to well-established utility and fairness criteria. When considering possible unintended adverse impacts of the work, it is important to consider that the paper's methodology is generic, and can be used oppositely to its stated goals. The mechanism used to enforce fairness of group exposures in rankings can also be used to enforce arbitrary proportional exposures amongst arbitarily defined groups. Therefore, it is possible to be used in a discriminatory manner. An inherent limitation of the work, with respect to potential fairness impact, is a lack of generalization to two-sided fairness, in which fairness with respect to user utility is enforced in addition to exposure of protected groups. This stems from the fact that the machine learning methodology inherently treats each user query sample independently, thus this fairness goal is usually only pursued by post-processing methods without ML integration.

REFERENCES

[1] Amir Beck. 2017. *First-order methods in optimization.* SIAM.

[2] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. 2020. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems* 33 (2020), 9508–9519.

[3] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. 2020. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*. PMLR, 950–959.

[4] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.

[5] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to algorithms.* MIT press.

[6] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2021. Two-sided fairness in rankings via Lorenz dominance. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 8596–8608. https://proceedings.neurips.cc/paper_files/paper/2021/file/48259990138bc03361556fb3f94c5d45-Paper.pdf

[7] Virginie Do and Nicolas Usunier. 2022. Optimizing generalized Gini indices for fairness in rankings. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 737–747.

[8] Benjamin Edelman, Michael Luca, and Dan Svirsky. 2017. Racial discrimination in the sharing economy: Evidence from a field experiment. *American economic journal: applied economics* 9, 2 (2017), 1–22.

[9] Shady Elbassuoni, Sihem Amer-Yahia, Ahmad Ghizzawi, and Christine El Atie. 2019. Exploring fairness of ranking in online job marketplaces. In *22nd International Conference on Extending Database Technology (EDBT)*.

[10] Adam N Elmachtoub and Paul Grigas. 2021. Smart "predict, then optimize". *Management Science* (2021).

[11] Godfrey Harold Hardy, John Edensor Littlewood, and George Pólya. 1952. *Inequalities.* Cambridge university press.

[12] Yiling Jia and Hongning Wang. 2021. Calibrating Explore-Exploit Trade-off for Fair Online Learning to Rank. arXiv:2111.00735 [cs.IR]

[13] Michael M Kostreva and Włodzimierz Ogryczak. 1999. Linear optimization with multiple equitable criteria. *RAIRO-Operations Research-Recherche Opérationnelle* 33, 3 (1999), 275–297.

[14] James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, and Bryan Wilder. 2021. End-to-End Constrained Optimization Learning: A Survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. 4475–4482. https://doi.org/10.24963/ijcai.2021/610

[15] James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, and Ziwei Zhu. 2022. End-to-end learning for fair ranking systems. In *Proceedings of the ACM Web Conference 2022*. 3520–3530.

[16] Guanghui Lan. 2013. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550* (2013).

[17] Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. 2023. Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities. *arXiv preprint arXiv:2307.13565* (2023).

[18] Włodzimierz Ogryczak and Tomasz Śliwiński. 2003. On solving linear programs with the ordered weighted averaging objective. *European Journal of Operational Research* 148, 1 (2003), 80–91.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

[20] Laurent Perron. 2011. Operations research and constraint programming at google. In *Principles and Practice of Constraint Programming–CP 2011: 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings 17*. Springer, 2–2.

[21] Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. 2020. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations (ICLR)*.

[22] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.

[23] Ashudeep Singh and Thorsten Joachims. 2019. *Policy Learning for Fairness in Ranking*. Curran Associates Inc., Red Hook, NY, USA.

[24] Wenlong Sun et al. 2020. Evolution and impact of bias in human and machine learning algorithm interaction. *PLOS ONE* (2020). https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0235502

[25] Bryan Wilder, Bistra Dilkina, and Milind Tambe. 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *AAAI*, Vol. 33. 1658–1665.

[26] Jingfang Xu, Chuanliang Chen, Gu Xu, Hang Li, and Elbio Renato Torres Abib. 2010. Improving quality of training data for learning to rank using click-through data. In *Proceedings of the third ACM international conference on Web search and data mining*. 171–180.

[27] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. 2019. Fair Learning-to-Rank from Implicit Feedback. *DeepAI* (2019). https://deepai.org/publication/fair-learning-to-rank-from-implicit-feedback

[28] RONALD R. YAGER. 1993. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. In *Readings in Fuzzy Sets for Intelligent Systems*, Didier Dubois, Henri Prade, and Ronald R. Yager (Eds.). Morgan Kaufmann, 80–87. https://doi.org/10.1016/B978-1-4832-1450-4.50011-0

[29] Ronald R. Yager and J. Kacprzyk. 2012. *The Ordered Weighted Averaging Operators: Theory and Applications.* Springer Publishing Company, Incorporated.

[30] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 1569–1578. https://doi.org/10.1145/3132847.3132938

[31] Meike Zehlike and Carlos Castillo. 2020. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of the web conference 2020*. 2849–2855.