



**HAL**  
open science

# Optimizing generalized Gini indices for fairness in rankings

Virginie Do, Nicolas Usunier

► **To cite this version:**

Virginie Do, Nicolas Usunier. Optimizing generalized Gini indices for fairness in rankings. 45th International ACM SIGIR Conference on Research and Development in Information Retrieval ( SIGIR 2022 ), Jul 2022, Madrid, Spain. hal-03636688v2

**HAL Id: hal-03636688**

**<https://hal.science/hal-03636688v2>**

Submitted on 13 Apr 2022 (v2), last revised 23 Apr 2022 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Optimizing generalized Gini indices for fairness in rankings

Virginie Do  
virginiedo@fb.com  
Meta AI Research

LAMSADE, Université Paris Dauphine-PSL

Nicolas Usunier  
usunier@fb.com  
Meta AI Research

## ABSTRACT

There is growing interest in designing recommender systems that aim at being fair towards item producers or their least satisfied users. Inspired by the domain of inequality measurement in economics, this paper explores the use of generalized Gini welfare functions (GGFs) as a means to specify the normative criterion that recommender systems should optimize for. GGFs weight individuals depending on their ranks in the population, giving more weight to worse-off individuals to promote equality. Depending on these weights, GGFs minimize the Gini index of item exposure to promote equality between items, or focus on the performance on specific quantiles of least satisfied users. GGFs for ranking are challenging to optimize because they are non-differentiable. We resolve this challenge by leveraging tools from non-smooth optimization and projection operators used in differentiable sorting. We present experiments using real datasets with up to 15k users and items, which show that our approach obtains better trade-offs than the baselines on a variety of recommendation tasks and fairness criteria.

## CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Theory of computation** → *Theory and algorithms for application domains*.

## KEYWORDS

fairness, ranking, recommender systems, welfare economics

## 1 INTRODUCTION

Recommender systems play an important role in organizing the information available to us, by deciding which content should be exposed to users and how it should be prioritized. These decisions impact both the users and the item producers of the platform. While recommender systems are usually designed to maximize performance metrics of user satisfaction, several audits recently revealed potential performance disparities across users [23, 29, 47, 69]. On the side of item producers, the growing literature on fairness of exposure aims to avoid popularity biases [2] by reducing inequalities in the exposure of different items [66], or aiming for equal exposure weighted by relevance [8, 26, 51]. In most cases, the approaches proposed for user- and item-side fairness aim to reduce inequalities.

In this paper, we propose a new approach to fair ranking based on Generalized Gini welfare Functions (GGFs, [73]) from the economic literature on inequality measurement [20]. GGFs are used to make decisions by maximizing a weighted sum of the utilities of individuals which gives more weight to those with lower utilities. By prioritizing the worse-off, GGFs promote more equality.

The normative appeal of GGFs lies in their ability to address a multiplicity of fairness criteria studied in the fair recommendation

literature. Since GGFs include the well-known Gini inequality index as a special case [33], they can be used to optimize trade-offs between exposure inequality among items and user utility, a goal sought by many authors [51, 84]. GGFs also conveniently specify normative criteria based on utility quantiles [27]: for instance, it is possible to improve the utility of the 10% worse-off users and/or items with GGFs, simply by assigning them more weight in the objective. Moreover, using techniques from convex multi-objective optimization, we show that GGFs cover *all* ranking policies that satisfy *Lorenz efficiency*, a distributive justice criterion which was recently introduced for two-sided fairness in rankings [27].

The difficulty of using GGFs as objective functions for fairness in ranking stems from their non-differentiability, which leads to computational challenges. Indeed, ranking with fairness of exposure requires the solution of a global optimization problem in the space of (randomized) rankings of all users, because the exposure of an item is the sum of its exposure to every users. The Frank-Wolfe algorithm [31] was shown to be a computationally efficient method for maximizing globally fair ranking objectives, requiring only one top- $K$  sort operation per user at each iteration [27]. However, vanilla Frank-Wolfe algorithms only apply to objective functions that are differentiable, which is not the case of GGFs.

We propose a new algorithm for the optimization of GGFs based on extensions of Frank-Wolfe algorithms for non-smooth optimization [44, 71, 83]. These methods usually optimize smoothed surrogate objective functions, while gradually decreasing a smoothing parameter, and a common smoothing technique uses the Moreau envelope [50, 82]. Our main insight is that the gradient of the Moreau envelope of GGFs can be computed in  $O(n \log n)$  operations, where  $n$  is the number of users or items. This result unlocks the use of Frank-Wolfe algorithms with GGFs, allowing us to efficiently find optimal ranking policies while optimizing GGFs.

We showcase the performances of the algorithm on two recommendation tasks of movies and music, and on a reciprocal recommendation problem (akin to dating platforms, where users are recommended to other users), with datasets involving up to 15k users and items. Compared to relevant baselines, we show that our algorithm successfully yields better trade-offs in terms of user utility and inequality in item exposure measured by the Gini index. Our approach also successfully finds better trade-offs in terms of two-sided fairness when maximizing the lower quantiles of user utility while minimizing the Gini index of item exposure.

In the remainder of the paper, we first describe our recommendation framework. We then present the family of generalized Gini welfare functions and its relationship to previously proposed fairness criteria in ranking. In Sec. 3 we provide the details of our algorithm and the convergence guarantees. Our experimental results are reported in Sec. 4, and an extension to reciprocal recommendation problems is discussed in Sec. 5. We position our approach with

respect to the related work in Sec. 6, and Sec. 7 concludes the paper and discusses the limitations of our work.

## 2 FAIR RANKING WITH GENERALIZED GINI

### 2.1 Recommendation framework

We consider a recommendation scenario with  $n$  users, and  $m$  items, and  $K$  recommendation slots.  $\mu_{ij} \in [0, 1]$  denotes the value of item  $j$  for user  $i$  (e.g. a “liking” probability), and we assume the values  $\mu$  are given as input to the system. The goal of the system is to produce a ranked list of items for each of the  $n$  users. Following previous work on fair rankings [e.g. 66], we consider randomized rankings because they enable the use of convex optimization techniques to generate the recommendations, which would otherwise involve an intractable combinatorial optimization problem in the space of all users’ rankings. A *randomized ranking* for user  $i$  is represented by a bistochastic matrix  $P_i \in \mathbb{R}^{m \times m}$ , where  $P_{ijk}$  is the probability that item  $j$  is recommended to user  $i$  at position  $k$ . The recommender system is characterized by a *ranking policy*  $P = (P_i)_{i=1}^n$ . We denote the convex set of ranking policies by  $\mathcal{P}$ .

We use the term *utility* in its broad sense in cardinal welfare economics as a “*measurement of the higher-order characteristic that is relevant to the particular distributive justice problem at hand*” [52]. Similarly to [27, 60, 72], we define the utility of a user as the ranking performance, and the utility of an item as its average exposure to users, which are formalized in (1) below. Utilities are defined according to the position-based model [8, 27, 51] with weights  $\mathbf{b} \in \mathbb{R}_+^m$ . The weight  $b_k$  is the probability that a user examines the item at position  $k$ , and we assume that the weights are non-increasing. Since there are  $K$  recommendation slots, we have  $b_1 \geq \dots \geq b_K$  and  $b_k = 0$  for any  $k > K$ . The user and item utilities are then:

$$\text{User utility: } u_i(P) = \sum_{j=1}^m \mu_{ij} P_{ij}^\top \mathbf{b} \quad \text{Item exposure: } v_j(P) = \sum_{i=1}^n P_{ij}^\top \mathbf{b}. \quad (1)$$

We follow a general framework where the ranking policy  $P$  is found by maximizing a global *welfare function*  $F(P)$ , and the welfare function is a weighted sum of welfare functions for users and items:

$$F(P) = (1 - \lambda)g^{\text{user}}(\mathbf{u}(P)) + \lambda g^{\text{item}}(\mathbf{v}(P)), \quad (2)$$

where  $g^{\text{user}} : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $g^{\text{item}} : \mathbb{R}^m \rightarrow \mathbb{R}$  respectively aggregate the utilities of users and item exposures and  $\lambda \in [0, 1]$  specifies the relative weight of users and items.

### 2.2 Generalized Gini welfare functions

In this work, we focus on the case where  $g^{\text{item}}$  and  $g^{\text{user}}$  are based on Generalized Gini welfare Functions (GGFs) [73]). A GGF  $g_{\mathbf{w}} : \mathbb{R}^n \rightarrow \mathbb{R}$  is a function parameterized by a vector  $\mathbf{w} \in \mathbb{R}^n$  of non-increasing positive weights such that  $w_1 = 1 \geq \dots \geq w_n \geq 0$ , and defined by a weighted sum of its sorted inputs, which is also called an ordered weighted averaging operator (OWA) [80]. Formally, let  $\mathbf{x} \in \mathbb{R}^n$  be a utility vector and denote by  $\mathbf{x}^\uparrow$  the values of  $\mathbf{x}$  sorted in increasing order, i.e.,  $x_1^\uparrow \leq \dots \leq x_n^\uparrow$ . Then:

$$g_{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^n w_i x_i^\uparrow.$$

Let  $\mathcal{V}_n = \{\mathbf{w} \in \mathbb{R}^n : w_1 = 1 \geq \dots \geq w_n \geq 0\}$  be the set of admissible weights of GGFs. Given  $\mathbf{w}^1 \in \mathcal{V}_n$ ,  $\mathbf{w}^2 \in \mathcal{V}_m$  and

$\lambda \in (0, 1)$ , we define the *two-sided GGF* as the welfare function (2) with  $g^{\text{user}} = g_{\mathbf{w}^1}$  and  $g^{\text{item}} = g_{\mathbf{w}^2}$ :

$$F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}(P) = (1 - \lambda)g_{\mathbf{w}^1}(\mathbf{u}(P)) + \lambda g_{\mathbf{w}^2}(\mathbf{v}(P)). \quad (3)$$

With non-increasing, non-negative weights  $\mathbf{w}$ , OWA operators are concave [80]. The maximization of  $F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}(P)$  (3) is thus a convex optimization problem (maximization of a concave function over the convex set of ranking policies). GGFs address fairness from the point of view of distributive justice in welfare economics [52], because they assign more weight to the portions of the population that have the least utility. Compared to a standard average, a GGF thus promotes more equality between individuals.

*Relationship to the Gini index.* GGFs are welfare functions so they follow the convention that they should be maximized. Moreover, if  $w_i > 0$  for all  $i$ ,  $g_{\mathbf{w}}$  is increasing with respect to every individual utilities, which ensures that maximizers of GGFs are Pareto-optimal [52]. The Gini index of  $\mathbf{x}$ , denoted  $\text{Gini}(\mathbf{x})$  is associated to the GGF  $g_{\mathbf{w}}(\mathbf{x})$  with  $w_i = (n-i+1)/n$  [for formulas of Gini index, see 81]:

$$\begin{aligned} \text{Gini}(\mathbf{x}) &= 1 - \frac{2}{\|\mathbf{x}\|_1} \sum_{i=1}^n \frac{n-i+1}{n} x_i^\uparrow \\ &= \frac{1}{n^2 \bar{x}} \sum_{i=1}^n \sum_{j=1}^n |x_i - x_j| \quad \text{with } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i. \end{aligned} \quad (4)$$

The second equality gives a more intuitive formula as a normalized average of absolute pairwise differences. The Gini index is an inequality measure, and therefore should be minimized, but, more importantly, it is normalized by the sum of utilities  $\|\mathbf{x}\|_1$ , which means that in general minimizing the Gini index does not yield Pareto-optimal solutions. The importance of this normalization is discussed by e.g., Atkinson [4], and by [27] in the context of fairness in rankings. Yet, when  $\mathbf{x}$  is a vector of item exposures  $\mathbf{x} = \mathbf{v}(P)$ , the normalization is not important because the total exposure is constant. It is then equivalent to minimize the Gini index of item exposures or to maximize its associated GGF.

*Multi-objective optimization of Lorenz curves.* An alternative formula for  $g_{\mathbf{w}}(\mathbf{x})$  is based on the generalized Lorenz curve<sup>1</sup> [64] of  $\mathbf{x}$ , which is denoted  $\mathbf{X}$  and is defined as the vector of cumulative sums of sorted utilities:

$$g_{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^n w'_i X_i \quad \text{where } w'_i = w_i - w_{i+1} \text{ and } X_i = x_1^\uparrow + \dots + x_i^\uparrow. \quad (5)$$

We used the convention  $w_{n+1} = 0$ . Notice that since the weights  $\mathbf{w}$  are non-increasing, we have that  $w'_i \geq 0$ . Thus, family of admissible OWA weights  $\mathbf{w}$  yield weights  $\mathbf{w}'$  that are non-negative and sum to 1. This formula offers the interpretation of GGFs as positively weighted averages of points of the generalized Lorenz curves. Every GGF thus corresponds to a scalarization of the multi-objective problem of maximizing every point of the generalized Lorenz curve [32, 49]. We get back to this interpretation in the next subsections.

<sup>1</sup>Lorenz curves are normalized so that the last value is 1, while generalized Lorenz curves are not normalized.

### 2.3 GGFs for fairness in rankings

To give concrete examples of the relevance of GGFs for fairness in rankings, we provide here two fairness evaluation protocols that have been previously proposed and fall under the scope of maximizing of GGFs as in Eq. (3).

*Trade-offs between user utility and inequality in item exposure.*

The first task consists in mitigating inequalities of exposure between (groups of) items, and appears in many studies [66, 78, 84]. This leads to a trade-off between the total utility of users and inequality among items, and such inequalities are usually measured by the Gini index (as in [8, 51]). Removing the dependency on  $P$  to lighten the notation, a natural formulation of this trade-off uses the two-sided GGF (3) by setting  $\mathbf{w}^1 = (1, \dots, 1)$  and  $\mathbf{w}^2 = \left(\frac{m-j+1}{m}\right)_{j=1}^m$ , which yields:

$$g^{\text{user}}(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n u_i \quad g^{\text{item}}(\mathbf{v}) = \sum_{j=1}^m \frac{m-j+1}{m} v_j^\uparrow. \quad (6)$$

As stated in the previous section, for item exposure, maximizing  $g^{\text{item}}$  is equivalent to minimizing the Gini index. The Gini index for  $g^{\text{item}}$  has been routinely used for *evaluating* inequality in item exposure [8, 51] but there is no algorithm to optimize general trade-offs between user utility and the Gini index of exposure. Morik et al. [51] use the Gini index of exposures in the context of dynamic ranking (with the absolute pairwise differences formula (4)), where their algorithm is shown to asymptotically drive  $g^{\text{item}}(\mathbf{v})$  to 0, equivalent to  $\lambda \rightarrow 1$  in (2). However, their algorithm cannot be used to converge to the optimal rankings for other values of  $\lambda$ . Do et al. [27] use as baseline a variant using the standard deviation of exposures instead of absolute pairwise difference because it is easier to optimize (it is smooth except on 0). In contrast, our approach allows for the direct optimization of the welfare function (2) with this instantiation of  $g^{\text{item}}$  given by eq. (6).

Several authors [8, 51] used *merit-weighted exposure*<sup>2</sup>  $v_j'(P) = v(P)/\bar{\mu}_j$  where  $\bar{\mu}_j = \frac{1}{n} \sum_{i=1}^n \mu_{ij}$  is the average value of item  $j$  across users, rather than the exposure itself. We keep the non-weighted exposure to simplify the exposition, but our method straightforwardly applies to merit-weighted exposure. Note however that the sum of weighted exposures is not constant, so using (6) with merit-weighted exposures is not strictly equivalent to minimizing the Gini index.

*Two-sided fairness.* Do et al. [27] propose to add a user-side fairness criterion to the trade-off above, to ensure that worse-off users do not bear the cost of reducing exposure inequalities among items. Their evaluation involves multi-dimensional trade-offs between specific points of the generalized Lorenz curve. Using the formulation (5) of GGFs, trade-offs between maximizing the cumulative utility at a specific quantile  $q$  of users and total utility can be formulated using a parameter  $\omega \in [0, 1]$  as follows:

$$g^{\text{user}}(\mathbf{u}) = \sum_{i=1}^n w'_i U_i \quad \text{with } w'_{[qn]} = \omega \text{ and } w'_n = 1 - \omega, \quad (7)$$

where all other values of  $w'_i = 0$ . In our experiments, we combine this  $g^{\text{user}}$  with the Gini index for  $g^{\text{item}}$  for two-sided fairness.

<sup>2</sup>also called “equity of attention” [8], “disparate treatment” [66]

### 2.4 Generating all Lorenz efficient solutions

In welfare economics, the fundamental property of concave welfare functions is that they are monotonic with respect to the dominance of generalized Lorenz curves [4, 52, 64], because this guarantees that maximizing a welfare function performs an optimal redistribution from the better-off to the worse-off at every level of average utility. In the context of two-sided fairness in rankings, Do et al. [27] formalize their fairness criterion by stating that a ranking policy is fair as long as the generalized Lorenz curves of users and items are not jointly dominated. In this section, we show that the family of GGFs  $F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}(P)$  (3) allows to generate *every* ranking policy that are fair under this definition, and *only* those. The result follows from standard results of convex multi-objective optimization [32, 49]. We give here the formal statements for exhaustivity.

Let  $\mathbf{x}$  and  $\mathbf{x}'$  two vectors in  $\mathbb{R}_+^n$ . We say that  $\mathbf{x}$  weakly-Lorenz-dominates  $\mathbf{x}'$ , denoted  $\mathbf{x} \succeq_L \mathbf{x}'$ , when the generalized Lorenz curve of  $\mathbf{x}$  is always at least equal to that of  $\mathbf{x}'$ , i.e.,  $\mathbf{x} \succeq_L \mathbf{x}' \iff \forall i, X_i \geq X'_i$ . We say that  $\mathbf{x}$  Lorenz-dominates  $\mathbf{x}'$ , denoted  $\mathbf{x} >_L \mathbf{x}'$  if  $\mathbf{x} \succeq_L \mathbf{x}'$  and  $\mathbf{x} \neq \mathbf{x}'$ , i.e., if the generalized Lorenz curve of  $\mathbf{x}$  is strictly larger than that of  $\mathbf{x}'$  on at least one point. The criterion that generalized Lorenz curves of users and items are not jointly-dominated is captured by the notion of *Lorenz-efficiency*:

*Definition 1* (Do et al. [27]). A ranking policy  $P \in \mathcal{P}$  is *Lorenz-efficient* if there is no  $P' \in \mathcal{P}$  such that either  $[\mathbf{u}(P') \succeq_L \mathbf{u}(P)$  and  $\mathbf{v}(P') >_L \mathbf{v}(P)]$  or  $[\mathbf{v}(P') \succeq_L \mathbf{v}(P)$  and  $\mathbf{u}(P') >_L \mathbf{u}(P)]$ .

We now present the main result of this section:

PROPOSITION 2. Let  $\Theta = (0, 1) \times \mathcal{V}_n \times \mathcal{V}_m$ .

- (1) Let  $(\lambda, \mathbf{w}^1, \mathbf{w}^2) \in \Theta$ , where  $\mathbf{w}^1$  and  $\mathbf{w}^2$  have strictly decreasing weights, and  $P^* \in \operatorname{argmax}_{P \in \mathcal{P}} F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}(P)$ . Then  $P^*$  is Lorenz-efficient.
- (2) If  $P$  is Lorenz efficient, then there exists  $(\lambda, \mathbf{w}^1, \mathbf{w}^2) \in \Theta$  such that  $P \in \operatorname{argmax}_{P \in \mathcal{P}} F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}(P)$ .

PROOF. The proof uses standard results on convex multi-objective optimization from [32, 49]. Written in the form (5), the GGFs corresponds to the scalarization of the multi-objective problem of jointly maximizing the generalized Lorenz curves of users and items, which is a problem with  $n + m$  objectives. Indeed, each objective function is a point of the generalized Lorenz curve  $(\mathbf{U}(P), \mathbf{V}(P))$ . Each objective  $U_i(P)$  is concave because it corresponds to an OWA operator with non-increasing weights  $\rho$  with  $\rho_{i'} = \mathbb{1}_{\{i' \leq i\}}$ , applied to utilities, which are linear functions of the ranking policy. Each objective  $V_i(P)$  is similarly concave. Moreover, we are optimizing over the convex set of stochastic ranking policies  $\mathcal{P}$ . The multi-objective problem is then concave, which means that the maximizers of all weighted sums of the objectives  $(\mathbf{U}(P), \mathbf{V}(P))$  with strictly positive weights are Pareto-efficient. Reciprocally every Pareto-efficient solution is a solution of a non-negative weighted sum of the objectives  $(\mathbf{U}(P), \mathbf{V}(P))$ , where the weights sum to 1 [49].

The result follows from the observation that the Lorenz-efficiency of  $P$ , defined as the Lorenz-efficiency of  $(\mathbf{u}(P), \mathbf{v}(P))$ , is equivalent to the Pareto-efficiency of its joint user-item Lorenz curves  $(\mathbf{U}(P), \mathbf{V}(P))$ . This is because the Lorenz dominance relation between vectors  $\mathbf{x}, \mathbf{x}'$  is defined as Pareto dominance in the space of their generalized Lorenz curves  $\mathbf{X}, \mathbf{X}'$ .

□

*Additive welfare functions vs GGFs.* Do et al. [27] use additive concave welfare functions to generate Lorenz-efficient rankings. Let  $\phi(x, \alpha) = x^\alpha$  if  $\alpha > 0$ ,  $\phi(x, \alpha) = \log(x)$  if  $\alpha = 0$  and  $\phi(x, \alpha) = -x^{-\alpha}$  if  $\alpha < 0$ . Do et al. [27] use concave welfare functions of the form:

$$g^{\text{user}}(\mathbf{u}) = \sum_{i=1}^n \phi(u_i, \alpha_1) \quad g^{\text{item}}(\mathbf{v}) = \sum_{j=1}^m \phi(v_j, \alpha_2) \quad (8)$$

Where  $\alpha_1$  (resp.  $\alpha_2$ ) specifies how much the rankings should redistribute utility to worse-off users (resp. least exposed items).

Additive separability plays an important role in the literature on inequality measures [4, 19, 22], as well as in the study of welfare functions because additive separability follows from a standard axiomatization [52]. However, this leads to a restricted class of functions, so that varying  $\alpha_1, \alpha_2$  and  $\lambda$  in (8) cannot generate all Lorenz-efficient solutions in general. The GGF approach provides a more general device to navigate the set of Lorenz-efficient solutions, with interpretable parameters since they are weights assigned to points of the generalized Lorenz curve.

### 3 OPTIMIZING GENERALIZED GINI WELFARE

In this section, we provide a scalable method for optimizing two-sided GGFs welfare functions (3)  $F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}$ . The challenge of optimizing GGFs is that they are nondifferentiable since they require sorting utilities. We first describe why existing approaches to optimize GGFs are not suited to ranking in Sec. 3.1. We then show how to efficiently compute the gradient of the Moreau envelope of GGFs in Sec. 3.2 and present the full algorithm in Sec. 3.3.

#### 3.1 Challenges

In multi-objective optimization, a standard approach to optimizing OWAs is to solve the equivalent linear program derived by Ogryczak and Śliwiński [56]. Because the utilities depend on 3d-tensors  $P \in \mathcal{P}$  in our case, the linear program has  $O(n \cdot m^2)$  variables and constraints, which is prohibitively large in practice. Another approach consists in using online subgradient descent to optimize GGFs, like [13, 48]. This is not tractable in our case because it requires to project iterates onto the parameter space, which in our case involves costly projections onto the space of ranking policies  $\mathcal{P}$ . On the other hand, the Frank-Wolfe algorithm [31] was shown to provide a computationally efficient and provably convergent method to optimize over  $\mathcal{P}$  [27]. However, it only applies to smooth functions, and Frank-Wolfe with subgradients may not converge to an optimal solution [55].

We turn to Frank-Wolfe variants for nonsmooth objectives, since Frank-Wolfe methods are well-suited to our structured ranking problem [17, 27, 40]. More precisely, following [44, 71, 83], our algorithm uses the Moreau envelope of GGFs for smoothing. The usefulness of this smooth approximation depends on its gradient, which computation is in some cases intractable [16]. Our main technical contribution is to show that the gradient of the Moreau envelope of GGFs can be computed in  $O(n \log n)$  operations.

#### 3.2 The Moreau envelope of GGFs

In the sequel,  $\|z\|$  denotes the  $\ell_2$  norm. Moreover, a function  $L : \mathcal{X} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  is  $C$ -smooth if it is differentiable with  $C$ -Lipschitz continuous gradients, i.e., if  $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$ ,  $\|\nabla L(\mathbf{x}) - \nabla L(\mathbf{x}')\| \leq C \|\mathbf{x} - \mathbf{x}'\|$ .

*3.2.1 Definition and properties.* Let us fix weights  $\mathbf{w} \in \mathcal{V}$  and focus on maximizing the GGF  $g_{\mathbf{w}}$ . Let  $h(\mathbf{z}) := -g_{\mathbf{w}}(\mathbf{z})$  to obtain a convex function (this simplifies the overall discussion). The function  $h$  is  $\|\mathbf{w}\|$ -Lipschitz continuous, but non-smooth. We consider the smooth approximation of  $h$  given by its Moreau envelope [59] defined as:

$$h^\beta(\mathbf{z}) = \min_{z' \in \mathbb{R}^n} h(z') + \frac{1}{2\beta} \|\mathbf{z} - z'\|^2.$$

It is known that  $h^\beta(\mathbf{z}) \leq h(\mathbf{z}) \leq h^\beta(\mathbf{z}) + \frac{\beta}{2} \|\mathbf{w}\|^2$  and that  $h^\beta$  is  $\frac{1}{\beta}$ -smooth [see e.g., 71]. The parameter  $\beta$  thus controls the trade-off between the smoothness and the quality of the approximation of  $h$ .

---

#### Algorithm 1: Computation of $\Pi_{C(\tilde{\mathbf{w}})}$

---

**input** : GGF weights  $\mathbf{w} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^n$   
**output**: Projection of  $\mathbf{z}$  onto the permutahedron  $C(\tilde{\mathbf{w}})$ .  
1  $\tilde{\mathbf{w}} \leftarrow -(\mathbf{w}_n, \dots, \mathbf{w}_1)$  and  $\sigma \leftarrow \text{argsort}(\mathbf{z})$   
2  $\mathbf{x} \leftarrow \text{PAV}(\mathbf{z}_\sigma - \tilde{\mathbf{w}})$   
3  $\mathbf{y} \leftarrow \mathbf{z} + \mathbf{x}_{\sigma^{-1}}$   
4 Return  $\mathbf{y}$ .

---

*3.2.2 Efficient computation of the gradient.* We now present an efficient procedure to compute the gradient of  $f^\beta(P) := h^\beta(\mathbf{u}(P))$ .

Given an integer  $n \in \mathbb{N}$ , let  $[[n]] := \{1, \dots, n\}$  and let  $\mathfrak{S}_n$  denotes the set of permutations of  $[[n]]$ . For  $\mathbf{x} \in \mathbb{R}^n$ , and  $\sigma \in \mathfrak{S}_n$ , let us denote by  $\mathbf{x}_\sigma = (x_{\sigma(1)}, \dots, x_{\sigma(n)})$ . Furthermore, let  $C(\mathbf{x})$  denote the *permutahedron* induced by  $\mathbf{x}$ , defined as the convex hull of all permutations of the vector  $\mathbf{x}$ :  $C(\mathbf{x}) = \text{conv}\{\mathbf{x}_\sigma : \sigma \in \mathfrak{S}_n\}$ . Finally, let  $\Pi_{\mathcal{X}}(\mathbf{z}) := \underset{z' \in \mathcal{X}}{\text{argmin}} \|\mathbf{z} - z'\|^2$ . denote the projection onto

a compact convex  $\mathcal{X}$ . The following proposition formulates  $\nabla f^\beta$  as a projection onto a permutahedron:

**PROPOSITION 3.** *Let  $\tilde{\mathbf{w}} = -(\mathbf{w}_n, \dots, \mathbf{w}_1)$ . Let  $P \in \mathcal{P}$ . Then for all  $(i, j, k) \in [[n]] \times [[m]]^2$ , we have:*

$$\frac{\partial f^\beta}{\partial P_{ijk}}(P) = y_i \mu_{ij} b_k \quad \text{where } \mathbf{y} = \Pi_{C(\tilde{\mathbf{w}})} \left( \frac{\mathbf{u}(P)}{\beta} \right). \quad (9)$$

**PROOF.** Let  $\text{prox}_{\beta h}(\mathbf{z}) = \underset{z' \in \mathbb{R}^n}{\text{argmin}} \beta h(\mathbf{z}) + \frac{1}{2} \|\mathbf{z}' - \mathbf{z}\|^2$  denote the proximal operator of  $\beta h$ . Denoting by  $\mathbf{u}^*$  the adjoint of  $\mathbf{u}$ , it is known that  $\nabla f^\beta(P) = \frac{1}{\beta} \mathbf{u}^*(\mathbf{u}(P)) - \text{prox}_{\beta h}(\mathbf{u}(P))$  [59].

We first notice that since  $\mathbf{w}$  are non-increasing, the rearrangement inequalities [36] gives:  $h(\mathbf{z}) = -\min_{\sigma \in \mathfrak{S}_n} \mathbf{w}^\top \sigma \mathbf{z} = \max_{\sigma \in \mathfrak{S}_n} -\mathbf{w}^\top \sigma \mathbf{z}$ . Thus,  $h$  is the support function of the convex set  $C(\tilde{\mathbf{w}})$ , since:

$$h(\mathbf{z}) = \max_{\sigma \in \mathfrak{S}_n} -\mathbf{w}^\top \sigma \mathbf{z} = \sup_{\mathbf{y} \in C(\tilde{\mathbf{w}})} \mathbf{y}^\top \mathbf{z}.$$

Then the Fenchel conjugate of  $h$  is the indicator function of  $C(\tilde{\mathbf{w}})$ , and its proximal is the projection  $\Pi_{C(\tilde{\mathbf{w}})}$  [59]. By Moreau decomposition, we get  $\text{prox}(\mathbf{z}) = \mathbf{z} - \beta \Pi_{C(\tilde{\mathbf{w}})}(\mathbf{z}/\beta)$ , and thus:

$$\nabla f^\beta(P) = \mathbf{u}^* \left( \Pi_{C(\tilde{\mathbf{w}})}(\mathbf{u}(P)/\beta) \right).$$

The result follows from the definition of  $\mathbf{u}(P) = \left( \sum_{j,k} \mu_{ij} P_{ijk} b_k \right)_{i=1}^n$ .  $\square$

Overall, computing the gradient of the Moreau envelope boils down to a projection onto the permutahedron  $C(\tilde{\mathbf{w}})$ . This projection was shown by several authors to be reducible to isotonic regression:

**PROPOSITION 4 (REDUCTION TO ISOTONIC REGRESSION [10, 46, 53]).** *Let  $\sigma \in \mathfrak{S}_n$  that sorts  $\mathbf{z}$  decreasingly, i.e.  $z_{\sigma(1)} \geq \dots \geq z_{\sigma(n)}$ . Let  $\mathbf{x}$  be a solution to isotonic regression on  $\mathbf{z}_\sigma - \tilde{\mathbf{w}}$ , i.e.*

$$\mathbf{x} = \underset{\mathbf{x}'_1 \leq \dots \leq \mathbf{x}'_n}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x}' - (\mathbf{z}_\sigma - \tilde{\mathbf{w}})\|^2$$

Then we have:  $\Pi_{C(\tilde{\mathbf{w}})}(\mathbf{z}) = \mathbf{z} + \mathbf{x}_{\sigma^{-1}}$ .

Following these works, we use the Pool Adjacent Violators (PAV) algorithm for isotonic regression, which gives a solution in  $O(n)$  iterations given a sorted input [7]. The algorithm for computing the projection is summarized in Alg. 1 where we use the notation  $\operatorname{argsort}(\mathbf{z}) = \{\sigma \in \mathfrak{S}_n : z_{\sigma(1)} \geq \dots \geq z_{\sigma(n)}\}$  for permutations that sort  $\mathbf{z} \in \mathbb{R}^n$  in decreasing order. Including the sorting of  $\frac{\mathbf{u}(P)}{\beta}$ , it costs  $O(n \log n)$  time and  $O(n)$  space.

**REMARK 1.** *Our method is related to the differentiable sorting operator of Blondel et al. [10], which uses a regularization term to smooth the linear formulation of sorting. The regularized form can itself be written as a projection to a permutahedron. The problem they address is different since they differentiate the multi-dimensional sort operation, but eventually the techniques are similar because the smoothing is done in a similar way.*

**REMARK 2.** *We computed the gradient of  $f^\beta(P) = h^\beta(\mathbf{u}(P))$  with user utilities. The gradient of  $f^\beta(P) = h^\beta(\mathbf{v}(P))$  using item exposures is computed similarly:  $\frac{\partial f^\beta}{\partial P_{ijk}}(P) = y_j b_k$  with  $\mathbf{y} = \Pi_{C(\tilde{\mathbf{w}})}\left(\frac{\mathbf{v}(P)}{\beta}\right)$ .*

### 3.3 Frank-Wolfe with smoothing

We return to the optimization of the two-sided GGF objective (3). In this section, we fix the parameters  $(\lambda, \mathbf{w}^1, \mathbf{w}^2)$  and consider the minimization of  $f := -F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}$  over  $\mathcal{P}$ . For  $\beta > 0$  we denote by  $h_1^\beta$  and  $h_2^\beta$  the Moreau envelopes of  $-g_{\mathbf{w}^1}$  and  $-g_{\mathbf{w}^2}$  respectively. The smooth approximation of  $f$  is then:

$$f^\beta(P) := (1 - \lambda)h_1^\beta(\mathbf{u}(P)) + \lambda h_2^\beta(\mathbf{v}(P)).$$

Our algorithm FW-smoothing (Alg. 2) for minimizing  $f$  uses the Frank-Wolfe method for nonsmooth optimization from Lan [44]<sup>3</sup>. Given a sequence  $(\beta_t)_{t \geq 1}$  of positive values decreasing to 0, the algorithm constructs iterates  $P^{(t)}$  by applying Frank-Wolfe updates to  $f^{\beta_t}$  at each iteration  $t$ . More precisely, FW-smoothing finds an update direction with respect to  $\nabla f^{\beta_t}$  by computing:

$$Q^{(t)} = \underset{P \in \mathcal{P}}{\operatorname{argmin}} \langle P | \nabla f^{\beta_t}(P^{(t-1)}) \rangle. \quad (10)$$

The update rule is  $P^{(t)} = P^{(t-1)} + \frac{2}{t+2}(Q^{(t)} - P^{(t-1)})$ .

Before giving the details of the computation of (10), we note that applying the convergence result of Lan [44], and denoting  $D_{\mathcal{P}} = \max_{P, P' \in \mathcal{P}} \|P - P'\|$  the diameter of  $\mathcal{P}$ , we obtain<sup>4</sup>:

<sup>3</sup>Lan [44] uses the smoothing scheme of Nesterov [54] which is in fact equal to the Moreau envelope (see [6, Sec. 4.3]).

<sup>4</sup>In more details, the convergence guarantee of Lan [44] uses the operator norm of  $\mathbf{u}$  and  $\mathbf{v}$ , which we bound as follows:  $\|\mathbf{u}(P)\|^2 \leq \sum_i \sum_{j,k} (\mu_{ij} P_{ijk} b_k)^2 \leq b_1^2 \|P\|^2$ , because  $\mu_{ij} \in [0, 1]$ ,  $b_k \in [0, b_1]$ , and similarly  $\|\mathbf{v}(P)\|^2 \leq b_1^2 \|P\|^2$ .

**PROPOSITION 5 (Th. 4, [44]).** *With  $\beta_0 = \frac{2D_{\mathcal{P}} b_1}{\|\mathbf{w}\|}$  and  $\beta_t = \frac{\beta_0}{\sqrt{t}}$ , FW-smoothing obtains the following convergence rate:*

$$f(P^{(T)}) - f(P^*) \leq \frac{2D_{\mathcal{P}} b_1 \|\mathbf{w}\|}{\sqrt{T}}.$$

**Algorithm 2:** FW-smoothing. Alg. 1 is used for  $\mathbf{y}^1$  and  $\mathbf{y}^2$ .

---

**input** : values  $(\mu_{ij})$ , # of iterations  $T$ , smoothing seq.  $(\beta_t)_t$   
**output** : ranking policy  $P^{(T)}$

- 1 Initialize  $P^{(0)}$  such that  $P_i^{(0)}$  sorts  $\mu_i$  in decreasing order
- 2 **for**  $t=1, \dots, T$  **do**
- 3     Let  $\mathbf{y}^1 = \Pi_{C(\tilde{\mathbf{w}}^1)}\left(\frac{\mathbf{u}(P^{(t-1)})}{\beta_t}\right)$  and  $\mathbf{y}^2 = \Pi_{C(\tilde{\mathbf{w}}^2)}\left(\frac{\mathbf{v}(P^{(t-1)})}{\beta_t}\right)$
- 4     **for**  $i=1, \dots, n$  **do**
- 5          $\tilde{\mu}_{ij} = (1 - \lambda) y_i^1 \mu_{ij} + \lambda y_j^2$
- 6          $\tilde{\sigma}_i \leftarrow \operatorname{top-K}(-\tilde{\mu}_i)$  // Update direction (10)
- 7     **end**
- 8     Let  $Q^{(t)} \in \mathcal{P}$  such that  $Q_i^{(t)}$  represents  $\tilde{\sigma}_i$
- 9      $P^{(t)} \leftarrow (1 - \frac{2}{t+2})P^{(t-1)} + \frac{2}{t+2}Q^{(t)}$ .
- 10 **end**
- 11 Return  $P^{(T)}$ .

---

*Efficient computation of the update direction.* For smooth welfare functions of user utilities and item exposures, the update direction (10) can be computed with only one top-K sorting operation per user [27]. In our case, the update is given by the following result, where  $\operatorname{top-K}(\mathbf{z}) = \{\sigma \in \mathfrak{S}_n : z_{\sigma(1)} \geq \dots \geq z_{\sigma(K)} \text{ and } \forall k \geq K, z_{\sigma(k)} \geq z_k\}$  is the set of permutations that sort the  $k$  largest elements in  $\mathbf{z}$ .

**PROPOSITION 6.** *Let  $\tilde{\mu}$  defined by  $\tilde{\mu}_{ij} = (1 - \lambda) y_i^1 \mu_{ij} + \lambda y_j^2$  where  $\mathbf{y}^1 = \Pi_{C(\tilde{\mathbf{w}}^1)}(\mathbf{u}(P^{(t-1)})/\beta_t)$  and  $\mathbf{y}^2 = \Pi_{C(\tilde{\mathbf{w}}^2)}(\mathbf{v}(P^{(t-1)})/\beta_t)$ .*

*For all  $i \in [n]$ , let  $\tilde{\sigma}_i \in \operatorname{top-K}(-\tilde{\mu}_i)$  and  $Q_i^{(t)}$  a permutation matrix representing  $\tilde{\sigma}_i$ . Then  $Q^{(t)} \in \operatorname{argmin}_{P \in \mathcal{P}} \langle P | \nabla f^{\beta_t}(P^{(t-1)}) \rangle$ .*

**PROOF.** Using the expression of the gradient of the Moreau envelope derived in Proposition 3, eq. (9), we have:

$$\frac{\partial f^{\beta_t}}{\partial P_{ijk}}(P^{(t-1)}) = (1 - \lambda) \frac{\partial}{\partial P_{ijk}}(h_1^{\beta_t}(\mathbf{u}(P^{(t-1)}))) + \lambda \frac{\partial}{\partial P_{ijk}}(h_2^{\beta_t}(\mathbf{v}(P^{(t-1)})))$$

And thus  $\frac{\partial f^{\beta_t}}{\partial P_{ijk}}(P^{(t-1)}) = \tilde{\mu}_{ij} \times b_k$ . The result then follows from [27, Lem. 3] and is a consequence of the rearrangement inequality [36]:  $Q^{(t)}$  is obtained by sorting  $\tilde{\mu}_{ij}$  in increasing order, or equivalently, by sorting  $-\tilde{\mu}_{ij}$  in decreasing order.  $\square$

Since the computation of the gradient of Moreau envelopes costs  $O(n \ln n + n \ln m)$  operations using Alg. 1, then by Prop. 6 at each iteration, the cost of the algorithm is dominated by the top-K sort per user, each of which has amortized complexity of  $O(m + K \ln K)$ :

**PROPOSITION 7.** *Each iteration costs  $O(nm + nK \ln K)$  operations. The total amount of storage required is  $O(nKT)$ .*

In conclusion, FW-smoothing has a cost per iteration similar to the standard Frank-Wolfe algorithm for ranking with smooth objective functions. The cost of the non-smoothness of the objective function is a convergence rate of  $1/\sqrt{T}$ , while the Frank-Wolfe algorithm converges in  $O(1/T)$  when the objective is smooth [17].

Moreover, the algorithm produces a *sparse representation* of the stochastic ranking policy as a weighted sum of permutation matrices. In other words, this gives us a Birkhoff-von-Neumann decomposition [9] of the bistochastic matrices *for free*, avoiding the overhead of an additional decomposition algorithm as in existing works on fair ranking [66, 68, 72].

## 4 EXPERIMENTS

We first present our experimental setting for recommendation of music and movies, together with the fairness criteria we explore and the baselines we consider. These fairness criteria have been chosen because they were used in the evaluation of prior work, and they exactly correspond to the optimization of a GGF. We thus expect our two-sided GGF  $F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}$  to fare better than the baselines, because they allow for the optimization of the exact evaluation criterion. We provide experimental results that demonstrate this claim in Sec. 4.2. Note that the GGFs are extremely flexible as we discussed in Sec. 2.1, so our experiments can only show a few illustrative examples of fairness criteria that can be defined with GGFs. In Sec. 4.3, we show the usefulness of FW-smoothing compared to the simpler baseline of Frank-Wolfe with subgradients.

### 4.1 Experimental setup

Our experiments are implemented in Python 3.9 using PyTorch<sup>5</sup>. For the PAV algorithm, we use the implementation of Scikit-Learn.<sup>6</sup>

**4.1.1 Data and evaluation protocol.** We present experiments on two recommendation tasks, following the protocols of [27, 60]. First, we address music recommendation with **Lastfm-2k** from Cantador et al. [14] which contains real listening counts of  $2k$  users for  $19k$  artists on the online music service Last.fm<sup>7</sup>. We filter the 2,500 items having the most listeners. In order to show how the algorithm scales, we also consider the **MovieLens-20m** dataset [37], which contains ratings in  $[0.5, 5]$  of movies by users, and we select the top 15,000 users and items with the most interactions.

We use an evaluation protocol similar to [27, 60, 72]. For each dataset, a full user-item preference matrix  $(\mu_{i,j})_{i,j}$  is obtained by standard matrix factorization algorithms<sup>8</sup> from the incomplete interaction matrix, following the protocol of [27]. Rankings are inferred from these estimated preferences. The exposure weights  $\mathbf{b}$  are the standard weights of the *discounted cumulative gain* (DCG) (also used in e.g., [8, 51, 66]):  $\forall k \in [[K]], b_k = \frac{1}{\log_2(1+k)}$ .

The generated  $\mu_{ij}$  are used as ground truth to evaluate rankings, in order to decouple the fairness evaluation of the ranking algorithms from the evaluation of biases in preference estimates (which are not addressed in the paper). The results are the average of three repetitions of the experiments over different random train/valid/test splits used to generate the  $\mu_{ij}$ .

<sup>5</sup><http://pytorch.org>

<sup>6</sup><https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/isotonic.py>

<sup>7</sup><https://www.last.fm/>

<sup>8</sup>Using the Python library Implicit: <https://github.com/benfred/implicit> (MIT License).

**4.1.2 Fairness criteria.** We remind two fairness tasks studied in the ranking literature and presented in Section 2.3, and describe existing approaches proposed to address them, which we consider as baselines for comparison with our two-sided GGF (3)  $F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}$ .

**Task 1: Trade-offs between user utility and inequality between items.** We use the two-sided GGF  $F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}$  instantiated as in Eq. (6), i.e., with  $\mathbf{w}^1 = (1, \dots, 1)$  and  $\mathbf{w}^2_j = \frac{m-j+1}{m}$ . This corresponds to a trade-off function between the sum of user utilities and a GGF for items with the Gini index weights, where the trade-off is controlled by varying  $\lambda \in (0, 1)$ . We remind though that unlike the standard Gini index, the GGF is un-normalized (see eq. (4), Sec 2.2).

We use three baselines for this task.

First, since the Gini index is non-differentiable, [27] proposed a differentiable surrogate using the standard deviation (std) instead, which we refer to as *eq. exposure*:

$$F^{\text{eq}}(P) = \sum_{i=1}^n u_i(P) - \frac{\lambda}{m} \sqrt{\sum_{j=1}^m \left( v_j(P) - \frac{1}{m} \sum_{j'=1}^m v_{j'}(P) \right)^2}$$

Second, Patro et al. [60] address the trade-off of Task 1, since they compare various recommendation strategies based on the utility of users and the Lorenz curves of items (see [60, Fig. 1]), recalling that the standard Gini index is often defined as  $1 - 2A$  where  $A$  is the area under the Lorenz curve [81]. Their fairness constraints are slightly different though, as their algorithm *FairRec*<sup>9</sup> guarantees envy-freeness for users, and a minimum exposure of  $\frac{\lambda \|\mathbf{b}\|}{m}$  for every item, where  $\lambda$  is the user-item tradeoff parameter.

Finally, we use the additive welfare function (8) (referred to as *welf*) with the recommended values  $\alpha_1 \in \{-2, 0, 1\}$  and  $\alpha_2 = 0$  [27], and varying  $\lambda \in (0, 1)$  as third baseline. We only report the result of  $\alpha_1 = 1$  since it obtained overall better performances on this task.

**Task 2: Two-sided fairness.** We consider trade-offs between the cumulative utility of the  $q$  fraction of worst-off users, where  $q \in \{0.25, 0.5\}$ , and inequality between items measured by the Gini index, as in [27]. For this task, we instantiate the two-sided GGF  $F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}$  as follows: the GGF for users is given by Eq. (7) with parameters  $(q, \omega)$  in  $\{0.25, 0.5\} \times \{0.25, 0.5, 1\}$ , and the GGF for items uses the Gini index weights  $\mathbf{w}_j = \frac{m-j+1}{m}$ . We generate trade-offs between user fairness and item fairness by varying  $\lambda \in (0, 1)$ .

The baseline approach for this task is *welf*, the additive welfare function (8), still with the recommended values  $\alpha_1 \in \{-2, 0, 1\}$  and  $\alpha_2 = 0$  and varying  $\lambda \in (0, 1)$ . We only report the results of  $\alpha_1 = -2$  as they obtained the best performances on this task.

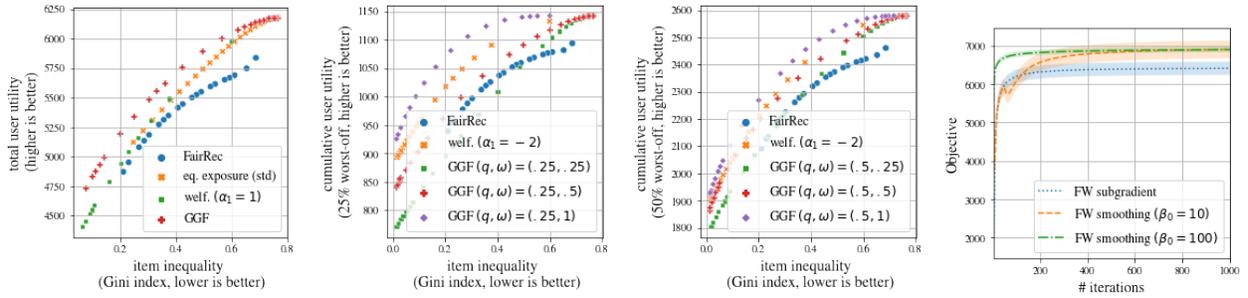
## 4.2 Results

We now present experiments that illustrate the effectiveness of the two-sided GGF approach on Task 1 and 2.

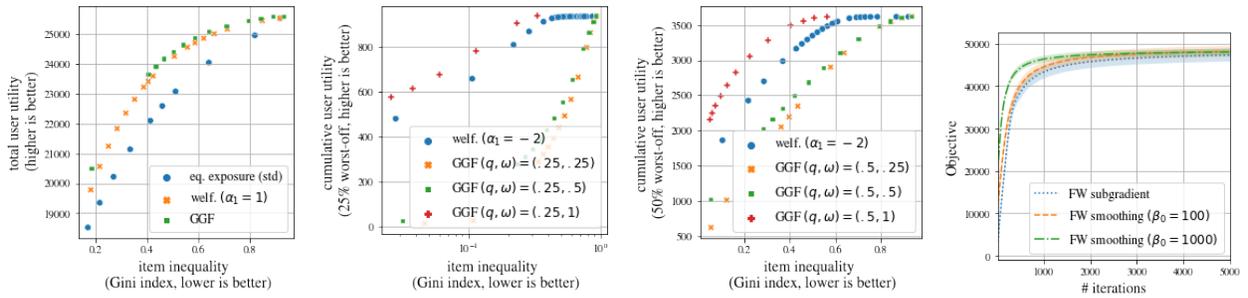
For each fairness method, Pareto frontiers are generated by varying  $\lambda$ . Since Patro et al. [60]'s algorithm *FairRec* does not scale, we compare to *FairRec* only on Lastfm-2k.

We optimize  $F_{\lambda, \mathbf{w}^1, \mathbf{w}^2}$  using FW-smoothing with  $\beta_0 = 100$  and  $T = 5k$  for Lastfm-2k, and  $\beta_0 = 1000$  and  $T = 50k$  for MovieLens.

<sup>9</sup>[60] consider unordered recommendation lists with a uniform attention model. We transform them into ordered lists using the order output by *FairRec*, and adapt the item-side criterion of minimal exposure to the position-based model.



(a) Total user utility vs. inequality between items. GGF is instantiated as (6). (b) Utility of the 25% worse-off users vs. inequality between items. GGF is instantiated as (7) with parameters  $(q, \omega)$ . (c) Utility of the 50% worse-off users vs. inequality between items. GGF is instantiated as (7) with parameters  $(q, \omega)$ . (d) Optimization of GGF (6) with  $\lambda = 0.5$



(e) Total user utility vs. inequality between items. GGF is instantiated as (6). (f) Utility of the 25% worse-off users vs. inequality between items. GGF is instantiated as (7) with parameters  $(q, \omega)$ . (g) Utility of the 50% worse-off users vs. inequality between items. GGF is instantiated as (7) with parameters  $(q, \omega)$ . (h) Optimization of GGF (6) with  $\lambda = 0.5$

**Figure 1: Summary of the results on Lastfm-2k (top row) and MovieLens (bottom row). (Left 3 columns): Trade-offs achieved by competing methods on various fairness criteria, when varying  $\lambda \in [0, 1]$ . (Right column): Convergence of FW-subgradient compared to FW-smoothing, for various values of  $\beta_0$ . FW-subgradient is not guaranteed to converge to an optimum.**

$F^{welf}$  and  $F^{eq}$  are optimized with the Frank-Wolfe method of [27] for  $T = 1k$  and  $T = 5k$  iterations respectively for Lastfm-2k and MovieLens. This is the number of iterations recommended by [27], while we need more interactions for FW-smoothing because its convergence is  $O(\frac{1}{\sqrt{T}})$  rather than  $O(\frac{1}{T})$  because of non-smoothness.

We first focus on Lastfm-2k. On Task 1, Fig. 1(a), the GGF (red + curve) obtains the best trade-off between total utility of users and Gini inequality between items, compared to *FairRec* and *eq. exposure*. It fares better than *eq. exposure* (orange  $\times$ ) on this task because *eq. exposure* reduces inequality between items by minimizing the std of exposures, while GGF with weights  $w_j^2 = \frac{m-j+1}{m}$  minimizes the Gini index. This shows that Task 1 can be addressed by directly optimizing the Gini index, thanks to GGFs with FW-smoothing.

For Task 2, Fig. 1(b) and 1(c) depicts the trade-offs achieved between the utility of the 25% / 50% worst-off users and inequality among items. First, we observe that the more weight  $\omega$  is put on the  $q$ -% worst-off in GGF, the higher the curve, which is why we observe the ordering green  $\square <$  red  $+$  purple  $\diamond$  for GGF, on both 25% and 50% trade-offs plots. Second, as expected, *welf* is outperformed by our two-sided GGF with instantiation (7) and  $(q, \omega) = (q, 1)$  (purple  $\diamond$ ), since it corresponds to the optimal settings for this task.

Figures 1(e), 1(f), 1(g) illustrates the same trade-offs on MovieLens. Results are qualitatively similar: by adequately parameterizing GGFs, we obtain the best guarantees on each fairness task.

Overall, these results show that even though the baseline approaches obtain non-trivial performances on the two fairness tasks above, the direct optimization of the trade-offs involving the Gini index or points of the Lorenz curves, which is possible thanks to our algorithm, yields significant performance gains. Moreover, we reiterate that these two tasks are only examples of fairness criteria that GGFs can formalize, since by varying the weights we can obtain all Lorenz-efficient rankings (Prop. 2).

### 4.3 Convergence diagnostics

We now demonstrate the usefulness of FW-smoothing for optimizing GGF objectives, compared to simply using the Frank-Wolfe method of [27] with a subgradient of the GGF (FW-subgradient). We note that a subgradient of  $g_{\mathbf{w}}(\mathbf{x})$  is given by  $\mathbf{w}_{\sigma-1}$ , where  $\sigma \in \text{argsort}(-\mathbf{x})$ . More precisely, FW-subgradient is also equivalent to using subgradients of  $-g_{\mathbf{w}^1}$  and  $-g_{\mathbf{w}^2}$  in Line 3 of Alg. 2, instead of  $\nabla f^{\beta_t}(P^{(t)})$ , ignoring the smoothing parameters  $\beta_t$ . FW-subgradient is simpler than FW-smoothing, but it is not guaranteed

to converge [55]. The goal of this section is to assess whether the smoothing is necessary in practice.

We focus on the two-sided GGF (6) of Task 1 on Lastfm-2k and MovieLens, using FW-subgradient and FW-smoothing with different values of  $\beta_0$ . Figure 1(d) depicts the objective value as a function of the number of iterations, averaged over three seeds (the colored bands represent the std), on Lastfm-2k. We observe that FW-subgradient (blue dotted curve) plateaus at a suboptimum. In contrast, FW-smoothing converges (orange dotted and green dash-dot curves), and the convergence is faster for larger  $\beta_0$ . On MovieLens (Fig 1(h)), FW-subgradient converges to the optimal solution, but it is still slower than FW-smoothing with  $\beta_0 = 1000$ .

In conclusion, even though FW-subgradient reaches the optimal performance on MovieLens for this set of parameters, it is still possible that FW-subgradient plateaus at significantly suboptimal solutions. The use of smoothing is thus not only necessary for theoretical convergence guarantees, but also in practice. In addition, FW-smoothing has comparable computational complexity to FW-subgradient since the computation cost is dominated by the sort operations in Alg. 2.

## 5 RECIPROCAL RECOMMENDATION

### 5.1 Extension of the framework and algorithm

We show that our whole method for fair ranking readily applies to reciprocal recommendation tasks, such as the recommendation of friends or dating partners, or in job search platforms.

*Reciprocal recommendation framework.* The recommendation framework we discussed thus far depicted “one-sided” recommendation, in the sense that only items are being recommended. In reciprocal recommendation problems [57], users are also items who can be recommended to other users (the item *per se* is the user’s profile or CV), and they have preferences over other users.

In this setting,  $n = m$  and  $\mu_{ij}$  denotes the mutual preference value between  $i$  and  $j$  (e.g., the probability of a “match” between  $i$  and  $j$ ). Following [27], we extend our previous framework to reciprocal recommendation by introducing the *two-sided utility* of a user  $i$ , which sums the utility  $\bar{u}_i(P)$  derived by  $i$  from the recommendations it gets, and the utility  $\bar{v}_i(P)$  from being recommended to other users:

$$u_i(P) = \bar{u}_i(P) + \bar{v}_i(P) = \sum_{i,j} (\mu_{ij} + \mu_{ji}) P_{ij}^T \mathbf{b}$$

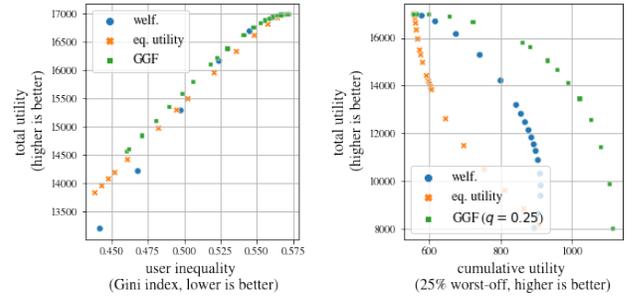
$$\text{where } \bar{u}_i(P) = \sum_{j=1}^n \mu_{ij} P_{ij}^T \mathbf{b} \quad \text{and} \quad \bar{v}_i(P) = \sum_{j=1}^n \mu_{ij} P_{ij}^T \mathbf{b}.$$

*Objective and optimization.* The two-sided GGF objective (3) in reciprocal recommendation simply becomes one GGF of two-sided utilities, and it is specified by a single weighting vector  $\mathbf{w}$ :

$$\max_{P \in \mathcal{P}} \{F_{\mathbf{w}}(P) := g_{\mathbf{w}}(\mathbf{u}(P))\}. \quad (11)$$

The choice of  $\mathbf{w}$  controls the degree of priority to the worse-off in the user population. We show in our experiments in Section 5.2 that in reciprocal recommendation too, the GGF objective can be adequately parameterized to address existing fairness criteria.

$F_{\mathbf{w}}$  can be optimized using our algorithm FW-smoothing. Since there is only one GGF, the subroutine Alg. 1 is simply used once per



(a) Total utility vs. inequality. GGF is instantiated as (12) and we vary  $\lambda$ . (b) Total utility vs. utility of the 25% worst-off. GGF is instantiated as (7) with  $q = 0.25$  and varying  $\omega$ .

**Figure 2: Fairness trade-offs achieved by competing methods on reciprocal recommendation on Twitter. They are generated by varying  $\lambda$  for eq. utility and GGF, and  $\alpha$  for welf.**

iteration to project onto  $C(\tilde{\mathbf{w}})$ , and obtain  $\mathbf{y} = \Pi_{C(\tilde{\mathbf{w}})} \left( \frac{\mathbf{u}(P^{(t-1)})}{\beta_t} \right)$  in Line 3 of Algorithm 2. Line 5 becomes  $\tilde{\mu}_{ij} = (1 - \lambda)y_i\mu_{ij} + \lambda y_j\mu_{ji}$ .

Our method for fair ranking is thus general enough to address both one-sided and reciprocal recommendation, using the notion of two-sided utility from Do et al. [27].

### 5.2 Experiments

Similarly to our experiments in Sec. 4.2, the goal of these experiments is to demonstrate that in reciprocal recommendation, GGF can be parameterized to exactly optimize for existing fairness criteria, outperforming previous approaches designed to address them.

*Data.* We reproduce the experimental setting of [27] who also study fairness in reciprocal recommendation. We simulate a friend recommendation task from the Higgs **Twitter** dataset [24], which contains directed follower links on the social network Twitter. We consider a mutual follow as a “match”, and we keep users having at least 20 matches, resulting in a subset of 13k users. We estimate mutual scores  $\mu_{ij}$  (i.e., match probabilities) by matrix factorization.

*5.2.1 Fairness criteria.* Similarly to Section 4.1.2, we state two fairness tasks that previously appeared in the literature, we instantiate the GGF objective  $F_{\mathbf{w}}$  for each task and describe existing baselines.

*Task 1: Trade-offs between total utility and inequality of utility among users.* Although reciprocal recommendation received less attention in the fairness literature, an existing requirement is to mitigate inequalities in utility between users [5, 41], similarly to the eq. exposure criterion in one-sided recommendation. This leads to a trade-off between the sum of utilities and inequality typically measured by the Gini index. For Task 1, we use the GGF (11) with  $w_i = (1 - \lambda) + \lambda \cdot \frac{n-i+1}{n}$ , which yields a trade-off function between the sum of utilities and inequality of utilities, and we vary  $\lambda$  in (0, 1) to generate such trade-offs:

$$F_{\mathbf{w}}(P) = (1 - \lambda) \sum_{i=1}^n u_i(P) + \lambda \sum_{i=1}^n \frac{n-i+1}{n} u_i^\uparrow(P). \quad (12)$$

We use two baselines for this task.

First, similarly to *eq. exposure*, to bypass the nonsmoothness of the Gini index, [27] optimize a surrogate with std, named *eq. utility*:

$$F^{\text{eq}}(P) = \sum_{i=1}^n u_i(P) - \frac{\lambda}{n} \sqrt{\sum_{i=1}^n \left( u_i(P) - \frac{1}{n} \sum_{i'=1}^n u_{i'}(P) \right)^2}.$$

Second, the welfare function *welf*(8) of [27] is used in reciprocal recommendation as a single sum:  $F^{\text{welf}}(P) = \sum_{i=1}^n \phi(u_i(P), \alpha)$  where  $\phi$  is defined in Sec. 2.3. We study *welf* as baseline by varying  $\alpha$ , which controls the redistribution of utility in the user population.

*Task 2: Trade-offs between total utility and utility of the worse-off.* The main task studied by [27] with *welf* is to trade-off between the total utility and the cumulative utility of the  $q$  fraction of worse-off users. For this task, we instantiate the GGF with (7), with fixed quantile  $q = 0.25$  and we vary  $\omega$  to generate trade-offs between total utility and cumulative utility of the 25% worst-off.

We compare it to the *welf* baseline where  $\alpha$  is varied as in [27].

### 5.2.2 Fairness trade-offs results.

*Results.* We now demonstrate that in reciprocal recommendation too, GGF is the most effective approach in addressing existing fairness criteria. We optimize the GGF  $F_w(P)$  using FW-smoothing with  $\beta_0 = 10$  for  $T = 50k$  iterations, and optimize  $F^{\text{welf}}$  and  $F^{\text{eq}}$  using Frank-Wolfe for  $T = 5k$  iterations.

Figure 2 depicts the trade-offs obtained by the competing approaches on the fairness tasks 1 and 2, on the Twitter dataset. Fig. 2(a) illustrates the superiority of GGF (green  $\square$ ) on Task 1, despite good performance of the baselines *eq. utility* (orange  $\times$ ) and *welf* (blue  $\circ$ ). As in one-sided recommendation with *eq. exposure*, the reason why *eq. utility* achieves slightly worse trade-offs on this fairness task is because it minimizes the std as a surrogate to the Gini index, instead of the Gini index itself as GGF does. For Task 2, on Fig. 2(b), we observe that GGF with parameterization (7) (green  $\square$ ) is the most effective. This is because unlike the *welf* approach (blue  $\circ$ ) of [27] who address this fairness task, this form of GGF is exactly designed to optimize for utility quantiles.

## 6 RELATED WORK

*Algorithmic fairness.* Fairness in ranking and recommendation systems is an active area of research. Since recommender systems involve multiple stakeholders [1, 12], fairness has been considered from the perspective of both users and item producers. On the user side, a common goal is to prevent disparities in recommendation performance across sensitive groups of users [29, 47]. On the item side, authors aim to prevent winner-take-all effects [2] by redistributing exposure across groups of producers, either towards equal exposure, or equal ratios of exposure to relevance [8, 26, 43, 66], sometimes measured by the classical Gini index [51, 74].

Some authors consider fairness for both users and items, often by applying existing user or item criteria simultaneously to both sides, such as [5, 72, 78]. [28, 60] instead discuss two-sided fairness with envy-freeness as user-side criterion, while [25] propose to use generalized cross entropy to measure unfairness among sensitive groups of users and items. [77] recently considered two-sided fairness in recommendation as a multi-objective problem, where each objective corresponds to a different fairness notion, either for

users or items. Similarly, Mehrotra et al. [48] aggregate multiple recommendation objectives using a GGF, in a contextual bandit setting. In their case, the aggregated objectives represent various metrics (e.g., clicks, dwell time) for various stakeholders. Unlike these two works [48, 77], in our case the multiple objectives are the individual utilities of each user and item, and our goal is to be fair towards each entity by redistributing utility. To our knowledge, we are the first to use GGFs as *welfare functions* of users' and items' utilities for two-sided fairness in rankings.

Reciprocal recommender systems received comparatively less attention in the fairness literature, to the exception of [41, 58, 79]. The closest to our work is the additive welfare approach of [27], which addresses fairness in both one-sided and reciprocal recommendation, and is extensively discussed in the paper, see Sec. 2.1.

In the broader fair machine learning community, several authors advocated for economic concepts [30], using inequality indices to quantify and mitigate unfairness [38, 45, 67, 75], taking an axiomatic perspective [18, 34, 76] or applying welfare economics principles [39, 61]. GGFs, in particular, were recently applied to fair multi-agent reinforcement learning, with multiple reward functions [13, 65, 85]. These works consider sequential decision-making problems without ranking, and their GGFs aggregate the objectives of a few agents (typically  $n < 20$ ), while in our ranking problem, there are as many objectives as there are users and items.

*Nonsmooth convex optimization and differentiable ranking.* Our work builds on nonsmooth convex optimization methods [54, 63], and in particular variants of the Frank-Wolfe algorithm [31, 40] for nonsmooth problems [44, 62, 71, 83]. The recent algorithm of [71] is a Frank-Wolfe variant which uses the Moreau envelope like us. Its number of first-order calls is optimal, but this is at the cost of a more complex algorithm with inner loops that make it slow in practice. In our case, since the calculation of the gradient is not a bottleneck, we use the simpler algorithm of Lan [44], which applies Frank-Wolfe to the Moreau envelope of the nonsmooth objective.

Our technical contribution is also related to the literature on differentiable ranking, which includes a large body of work on approximating learning-to-rank metrics [3, 15, 70], and recent growing interest in designing smooth ranking modules [10, 21, 35] for end-to-end differentiation pipelines. The closest method to ours is the differentiable sorting operator of Blondel et al. [10], which also relies on isotonic regression. The differences between our approaches are explained in Remark 1.

## 7 CONCLUSION

We proposed generalized Gini welfare functions as a flexible method to produce fair rankings. We addressed the challenges of optimizing these welfare functions by leveraging Frank-Wolfe methods for nonsmooth objectives, and demonstrated their efficiency in ranking applications. Our framework and algorithm applies to both usual recommendation of movies or music, and to reciprocal recommendation scenarios, such as dating or hiring.

Generalized Gini welfare functions successfully address a large variety of fairness requirements for ranking algorithms. On the one hand, GGFs are effective in reducing inequalities, since they generalize the Gini index in economics. Optimizing them allows to meet the requirements of equal utility criteria, largely advocated by

existing work on fair recommendation [5, 60, 66, 78]. On the other hand, GGFs effectively increase the utility of the worse-off, which is usually measured by quantile ratios in economics, and has been recently considered as a fairness criterion in ranking [27].

Our approach is limited to fairness considerations at the stage of inference. It does not address potential biases arising at other parts of the recommendation pipeline, such as in the estimation of preferences. Moreover, we considered a static model, which does not account for real-world dynamics, such as responsiveness in two-sided markets [68], feedback loops in the learning process [11], and the changing nature of the users' and items' populations [51] and preferences [42]. Addressing these limitations, in combination with our method, are interesting directions for future research.

## ACKNOWLEDGMENTS

We would like to thank Sam Corbett-Davies for his thoughtful feedback on this work.

## REFERENCES

- [1] Himan Abdollahpour, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. 2020. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction* 30, 1 (2020), 127–158.
- [2] Himan Abdollahpour, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286* (2019).
- [3] Ryan Prescott Adams and Richard S Zemel. 2011. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925* (2011).
- [4] Anthony B Atkinson. 1970. On the measurement of inequality. *Journal of economic theory* 2, 3 (1970), 244–263.
- [5] Kinjal Basu, Cyrus DiCiccio, Heloise Logan, and Noureddine El Karoui. 2020. A Framework for Fairness in Two-Sided Marketplaces. *arXiv preprint arXiv:2006.12756* (2020).
- [6] Amir Beck and Marc Teboulle. 2012. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization* 22, 2 (2012), 557–580.
- [7] Michael J Best, Nilotpal Chakravarti, and Vasant A Ubhaya. 2000. Minimizing separable convex functions subject to simple chain constraints. *SIAM Journal on Optimization* 10, 3 (2000), 658–672.
- [8] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*. 405–414.
- [9] Garrett Birkhoff. 1940. *Lattice theory*. Vol. 25. American Mathematical Soc.
- [10] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. 2020. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*. PMLR, 950–959.
- [11] Léon Bottou, Jonas Peters, Joaquin Quiñero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research* 14, 1 (2013), 3207–3260.
- [12] Robin Burke. 2017. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093* (2017).
- [13] Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, and Shie Mannor. 2017. Multi-objective bandits: Optimizing the generalized Gini index. In *International Conference on Machine Learning*. PMLR, 625–634.
- [14] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *Proceedings of the 5th ACM conference on Recommender systems* (Chicago, IL, USA) (RecSys 2011). ACM, New York, NY, USA.
- [15] Olivier Chapelle and Mingrui Wu. 2010. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval* 13, 3 (2010), 216–235.
- [16] Xi Chen, Qihang Lin, Seyoung Kim, Jaime G Carbonell, and Eric P Xing. 2012. Smoothing proximal gradient method for general structured sparse learning. *arXiv preprint arXiv:1202.3708* (2012).
- [17] Kenneth L Clarkson. 2010. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)* 6, 4 (2010), 1–30.
- [18] Cyrus Cousins. 2021. An axiomatic theory of provably-fair welfare-centric machine learning. *Advances in Neural Information Processing Systems* 34 (2021).
- [19] Frank A Cowell. 1988. Inequality decomposition: three bad measures. *Bulletin of Economic Research* 40, 4 (1988), 309–312.
- [20] Frank A Cowell. 2000. Measurement of inequality. *Handbook of income distribution* 1 (2000), 87–166.
- [21] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. 2019. Differentiable Ranking and Sorting using Optimal Transport. In *Advances in Neural Information Processing Systems*. 6858–6868.
- [22] Hugh Dalton. 1920. The measurement of the inequality of incomes. *The Economic Journal* 30, 119 (1920), 348–361.
- [23] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies* 2015, 1 (2015), 92–112.
- [24] Manlio De Domenico, Antonio Lima, Paul Mougél, and Mirco Musolesi. 2013. The anatomy of a scientific rumor. *Scientific reports* 3, 1 (2013), 1–9.
- [25] Yashar Deldjoo, Vito Walter Anelli, Hamed Zamani, Alejandro Bellogin, and Tommaso Di Noia. 2021. A flexible framework for evaluating user and item fairness in recommender systems. *User Modeling and User-Adapted Interaction* (2021), 1–55.
- [26] Fernando Diaz, Bhaskar Mitra, Michael D Ekstrand, Asia J Biega, and Ben Carterette. 2020. Evaluating stochastic rankings with expected exposure. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 275–284.
- [27] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2021. Two-sided fairness in rankings via Lorenz dominance. *Advances in Neural Information Processing Systems* 34 (2021).
- [28] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2022. Online certification of preference-based fairness for personalized recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36.
- [29] Michael D Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D Ekstrand, Oghenamary Anuyah, David McNeill, and Maria Soledad Pera. 2018. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In *Conference on Fairness, Accountability and Transparency*. PMLR, 172–186.
- [30] Jessie Finocchiaro, Roland Maio, Faidra Monachou, Gourab K Patro, Manish Raghavan, Ana-Andreea Stoica, and Stratis Tsirtsis. 2020. Bridging Machine Learning and Mechanism Design towards Algorithmic Fairness. *arXiv preprint arXiv:2010.05434* (2020).
- [31] Marguerite Frank and Philip Wolfe. 1956. An algorithm for quadratic programming. *Naval research logistics quarterly* 3, 1-2 (1956), 95–110.
- [32] Arthur M Geoffrion. 1968. Proper efficiency and the theory of vector maximization. *Journal of mathematical analysis and applications* 22, 3 (1968), 618–630.
- [33] Corrado Gini. 1921. Measurement of inequality of incomes. *The economic journal* 31, 121 (1921), 124–126.
- [34] Paul Gözl, Anson Kahng, and Ariel D Procaccia. 2019. Paradoxes in Fair Machine Learning. *Advances in Neural Information Processing Systems (NeurIPS)*. Forthcoming (2019).
- [35] Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. 2019. Stochastic optimization of sorting networks via continuous relaxations. *arXiv preprint arXiv:1903.08850* (2019).
- [36] G. H. Hardy, J. E. Littlewood, and George Pólya. 1952. *Inequalities*. 2nd ed. Cambridge, Engl.: At the University Press. XII, 324 p. (1952).
- [37] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.
- [38] Hoda Heidari, Claudio Ferrari, Krishna Gummadi, and Andreas Krause. 2018. Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *Advances in Neural Information Processing Systems*. 1265–1276.
- [39] Lily Hu and Yiling Chen. 2020. Fair classification and social welfare. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 535–545.
- [40] Martin Jaggi. 2013. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*. PMLR, 427–435.
- [41] Yongzheng Jia, Xue Liu, and Wei Xu. 2018. When Online Dating Meets Nash Social Welfare: Achieving Efficiency and Fairness. In *Proceedings of the 2018 World Wide Web Conference*. 429–438.
- [42] Dimitris Kalimeris, Smriti Bhagat, Shankar Kalyanaraman, and Udi Weinsberg. 2021. Preference Amplification in Recommender Systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 805–815.
- [43] Till Klettli, Jean-Michel Renders, and Patrick Loiseau. 2022. Introducing the Exposedron for Efficient Pareto-optimal Fairness-Utility Amortizations in Repeated Rankings. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 498–507.
- [44] Guanghui Lan. 2013. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550* (2013).
- [45] Tomo Lazovich, Luca Belli, Aaron Gonzales, Amanda Bower, Uthaiapon Tantipongpipat, Kristian Lum, Ferenc Huszar, and Rummman Chowdhury. 2022. Measuring Disparate Outcomes of Content Recommendation Algorithms with Distributional Inequality Metrics. *arXiv preprint arXiv:2202.01615* (2022).

- [46] Cong Han Lim and Stephen J Wright. 2016. Efficient bregman projections onto the permutahedron and related polytopes. In *Artificial Intelligence and Statistics*. PMLR, 1205–1213.
- [47] Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach, and Emine Yilmaz. 2017. Auditing search engines for differential satisfaction across demographics. In *Proceedings of the 26th international conference on World Wide Web companion*. 626–633.
- [48] Rishabh Mehrotra, Niannan Xue, and Mounia Lalmas. 2020. Bandit based Optimization of Multiple Objectives on a Music Streaming Platform. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3224–3233.
- [49] Kaisa Miettinen. 2012. *Nonlinear multiobjective optimization*. Vol. 12. Springer Science & Business Media.
- [50] Jean Jacques Moreau. 1962. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 255 (1962), 2897–2899.
- [51] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. *Controlling Fairness and Bias in Dynamic Learning-to-Rank*. Association for Computing Machinery, New York, NY, USA, 429–438. <https://doi.org/10.1145/3397271.3401100>
- [52] Hervé Moulin. 2003. *Fair division and collective welfare*. MIT press.
- [53] Renato Negrinho and Andre Martins. 2014. Orbit regularization. *Advances in neural information processing systems* 27 (2014), 3221–3229.
- [54] Yu Nesterov. 2005. Smooth minimization of non-smooth functions. *Mathematical programming* 103, 1 (2005), 127–152.
- [55] Yu Nesterov. 2018. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming* 171, 1 (2018), 311–330.
- [56] 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.
- [57] Iván Palomares, Carlos Porcel, Luiz Pizzato, Ido Guy, and Enrique Herrera-Viedma. 2021. Reciprocal Recommender Systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation. *Information Fusion* 69 (2021), 103–127.
- [58] Dimitris Paraschakis and Bengt J Nilsson. 2020. Matchmaking under fairness constraints: a speed dating case study. In *International Workshop on Algorithmic Bias in Search and Recommendation*. Springer, 43–57.
- [59] Neal Parikh and Stephen Boyd. 2014. Proximal algorithms. *Foundations and Trends in optimization* 1, 3 (2014), 127–239.
- [60] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2020. *FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms*. Association for Computing Machinery, New York, NY, USA, 1194–1204. <https://doi.org/10.1145/3366423.3380196>
- [61] Ashesh Rambachan, Jon Kleinberg, Sendhil Mullainathan, and Jens Ludwig. 2020. *An economic approach to regulating algorithms*. Technical Report. National Bureau of Economic Research.
- [62] Sathya N Ravi, Maxwell D Collins, and Vikas Singh. 2019. A deterministic nonsmooth frank wolfe algorithm with coresets guarantees. *Informs Journal on Optimization* 1, 2 (2019), 120–142.
- [63] Ohad Shamir and Tong Zhang. 2013. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*. PMLR, 71–79.
- [64] Anthony F Shorrocks. 1983. Ranking income distributions. *Economica* 50, 197 (1983), 3–17.
- [65] Umer Siddique, Paul Weng, and Matthieu Zimmer. 2020. Learning Fair Policies in Multi-Objective (Deep) Reinforcement Learning with Average and Discounted Rewards. In *International Conference on Machine Learning*. PMLR, 8905–8915.
- [66] 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*. ACM, 2219–2228.
- [67] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. 2018. A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual & Group Unfairness via Inequality Indices. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2239–2248.
- [68] Yi Su, Magd Bayoumi, and Thorsten Joachims. 2021. Optimizing Rankings for Recommendation in Matching Markets. *arXiv preprint arXiv:2106.01941* (2021).
- [69] Latanya Sweeney. 2013. Discrimination in online ad delivery. *Queue* 11, 3 (2013), 10.
- [70] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. 2008. Sofrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. 77–86.
- [71] Kiran K Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. 2020. Projection Efficient Subgradient Method and Optimal Nonsmooth Frank-Wolfe Method. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 12211–12224. <https://proceedings.neurips.cc/paper/2020/file/8f468c873a32bb0619eab2050ba45d1-Paper.pdf>
- [72] Lequn Wang and Thorsten Joachims. 2021. User Fairness, Item Fairness, and Diversity for Rankings in Two-Sided Markets. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 23–41.
- [73] John A Weymark. 1981. Generalized Gini inequality indices. *Mathematical Social Sciences* 1, 4 (1981), 409–430.
- [74] Colin Wilkie and Leif Azzopardi. 2014. Best and fairest: An empirical analysis of retrieval system bias. In *European Conference on Information Retrieval*. Springer, 13–25.
- [75] Colin Wilkie and Leif Azzopardi. 2015. Retrievalability and retrieval bias: A comparison of inequality measures. In *European Conference on Information Retrieval*. Springer, 209–214.
- [76] Robert Williamson and Aditya Menon. 2019. Fairness risk measures. In *International Conference on Machine Learning*. PMLR, 6786–6797.
- [77] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. 2021. Multi-FR: A Multi-Objective Optimization Method for Achieving Two-sided Fairness in E-commerce Recommendation. *arXiv preprint arXiv:2105.02951* (2021).
- [78] Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. 2021. TFRM: A Two-sided Fairness-Aware Recommendation Model for Both Customers and Providers. *arXiv preprint arXiv:2104.09024* (2021).
- [79] Peng Xia, Benyuan Liu, Yizhou Sun, and Cindy Chen. 2015. Reciprocal recommendation system for online dating. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 234–241.
- [80] Ronald R Yager. 1988. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics* 18, 1 (1988), 183–190.
- [81] Shlomo Yitzhaki and Edna Schechtman. 2013. More than a dozen alternative ways of spelling Gini. In *The Gini Methodology*. Springer, 11–31.
- [82] Kósaku Yosida et al. 1965. Functional analysis. (1965).
- [83] Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. 2018. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In *International Conference on Machine Learning*. PMLR, 5727–5736.
- [84] Meike Zehlke and Carlos Castillo. 2020. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020*. 2849–2855.
- [85] Matthieu Zimmer, Claire Glanois, Umer Siddique, and Paul Weng. 2021. Learning fair policies in decentralized cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 12967–12978.