Maxmin-Fair Ranking: Individual Fairness under Group-Fairness Constraints

David García-Soriano
ISI Foundation, Turin, Italy
d.garcia.soriano@isi.it

Francesco Bonchi
ISI Foundation, Turin, Italy
Eurecat, Barcelona, Spain
francesco.bonchi@isi.it

ABSTRACT

We study a novel problem of fairness in ranking aimed at minimizing the amount of individual unfairness introduced when enforcing group-fairness constraints. Our proposal is rooted in the distributional maxmin fairness theory, which uses randomization to maximize the expected satisfaction of the worst-off individuals. We devise an exact polynomial-time algorithm to find maxmin-fair distributions of general search problems (including, but not limited to, ranking), and show that our algorithm can produce rankings which, while satisfying the given group-fairness constraints, ensure that the maximum possible value is brought to individuals.

CCS CONCEPTS
• Computing methodologies → Machine learning.

KEYWORDS
fairness, ranking, max-min fairness

ACM Reference Format:
David García-Soriano and Francesco Bonchi. 2021. Maxmin-Fair Ranking: Individual Fairness under Group-Fairness Constraints. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467349

1 INTRODUCTION

As the position in a ranking influences to a great extent the amount of attention that an item receives, biases in ranking can lead to unfair distribution of exposure, thus producing substantial economic impact. If this is important when ranking items (e.g., web pages, movies, hotels, books), it raises even more crucial concerns when ranking people. These concerns have captured the attention of researchers, education (e.g., university admission), or employment (e.g., selection for a job), which can have a direct tangible impact on people’s life. These concerns have captured the attention of researchers, which have thus started devising ranking systems which are fair for the items being ranked [3, 7, 13, 26, 30, 31, 34].

The bulk of the algorithmic fairness literature deals with group fairness along the lines of demographic parity [9] or equal opportunity [16]: this is typically expressed by means of some fairness constraint requiring that the top-k positions (for any k) in the ranking contain enough elements from some groups that are protected from discrimination based on sex, race, age, etc. In fact, [6] shows that in a certain model, group-fairness constraints can eliminate the bias implicit in the ranking scores. Besides, some legal norms enforce these constraints [1, 2]. For these reasons we will consider a ranking valid if it satisfies a given set of group-fairness constraints of this type, as detailed in Section 3.

More formally, consider a set of elements (items or individuals) to be ranked \( \mathcal{U} = \{u_1, \ldots, u_n\} \), a partition of \( \mathcal{U} \) into groups defined by some protected attributes, and a relevance score \( R : \mathcal{U} \to \mathbb{R}^+ \) for each element. For instance, \( \mathcal{U} \) could be the result of a query while \( R \) represents the relevance of each item for the query, or \( \mathcal{U} \) could be the set of applicants for a job while \( R \) their fitness for the job. Let \( \mathcal{S} \) denote all possible rankings of \( \mathcal{U} \) (bijections from \( \mathcal{U} \) to \([n]\)), where \( r(u) \in [n] \) denotes the position of element \( u \) in a ranking \( r \in \mathcal{R} \) and let \( \mathcal{S} \subseteq \mathcal{R} \) denote the subset of valid rankings satisfying the agreed-upon constraints. Let \( W(r, u) \) denote the utility that placing \( u \) at position \( r(u) \) brings to the overall ranking: this is typically a function of the relevance score \( R \), so that having higher relevance elements at top positions is rewarded. In other words, \( W \) is such that, if \( r^* \) denotes the ranking by decreasing \( R \), then \( r^* \) is also the ranking maximizing the total utility (the so-called Probability Ranking Principle [24] in Information Retrieval). As the maximum-utility ranking \( r^* \) might not satisfy the group-fairness constraint, the problem typically addressed in the literature is to find a valid ranking which maximizes the global utility, i.e.,

\[
\tilde{r} \in \arg\max_{r \in \mathcal{S}} \sum_{u \in \mathcal{U}} W(r, u).
\]

| Example 1. Consider the case described in Table 1 and suppose that the group-fairness constraint requires to have at least \( \lfloor k/2 \rfloor \) individuals of each gender in the top-k positions starting from \( k \geq 3 \).

The ranking by decreasing relevance \( r^* = (u_1, u_2, \ldots, u_8) \) is not a valid ranking in this case, as \( \bar{u} \) is underrepresented in the top-k positions for \( k = 4, 5, 6 \). A valid ranking which is as close as possible to \( r^* \) would be \( r = (u_1, u_2, u_3, u_6, u_4, u_7, u_5, u_8) \).

### Table 1: Example instance. Top row: identifiers and protected attribute (gender). Bottom row: relevance score \( R \).

| \( u_1 \), \( \bar{u} \) | \( u_2 \), \( \bar{u} \) | \( u_3 \), \( \bar{u} \) | \( u_4 \), \( \bar{u} \) | \( u_5 \), \( \bar{u} \) | \( u_6 \), \( \bar{u} \) | \( u_7 \), \( \bar{u} \) | \( u_8 \), \( \bar{u} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 0.97            | 0.93            | 0.89            | 0.81            | 0.73            | 0.72            | 0.64            | 0.62            |
This approach stems from an information retrieval standpoint: the set of items to be ranked is the result of a query, and as long as the given group-fairness constraint is satisfied, it suffices for the application at hand to maximize the global utility. While at first sight this setting might seem adequate to rank people, maximizing global utility provides no guarantee to individuals, who care little about global utility. In Example 1, individuals $u_4$ and $u_5$ have been uniquely penalized from a meritocratic fairness point of view: They may accept the group-fairness constraints and agree with the fact that the produced ranking $r'$ is as close as possible to $r^*$, but nevertheless feel discriminated against, for being the only ones in a worse position in $r'$ than in $r^*$ despite other solutions being possible. For example, $(u_4, u_1, u_3, u_6, u_2, u_7, u_5, u_6)$ is valid and more favourable to $u_4$. In other words, while the use of group-fairness constraints is often desirable and may be required by law, certain individuals in a such a valid ranking might feel unfairly penalized, even when comparing only to individuals within the same group. As soon as a group-fairness constraint is enforced in ranking problems, some individual-level unfairness is inevitably introduced.

In this paper we study the problem of minimizing the amount of individual unfairness introduced when enforcing a group-fairness constraint. While much of the literature for ranking attempts to maximize global utility, global quality metrics generally fail to adequately capture the treatment of individuals. Thus, differently from the literature which tries to maximize the global utility, we adopt Rawls’s theory of justice [23], which advocates arranging social and financial inequalities to the benefit of the worst-off. Following this precept, a natural task is to find a ranking that, while satisfying the group-fairness constraint, maximizes the utility of the least-advantaged individual:

$$r' \in \arg \max_{r \in S} \min_{u \in U} V(r, u). \quad (2)$$

Here $V(r, u)$ represents the value (utility) that placing $u$ at position $r(u)$ brings to the individual $u$, relative to $u$’s quality $R(u)$.

In Section 5 we provide an exact optimal solution for (2). This, however, is not the main focus of our paper. In fact, we can improve individual treatment even further through randomization.

### Randomization for individual fairness.

We next show how, by means of randomization, we can improve individual treatment over the best deterministic solution of (2). In particular, we show that there exists a probability distribution over valid rankings, where the minimum expected value that any individual gets is higher than is possible with any single ranking.

**Example 2.** Consider the value function $V(r, u) = r^*(u) - r(u)$, i.e., the difference between the meritocratic ranking by relevance and the ranking produced. This is positive for individuals who are in a better (lower-ranked) position in $r$ w.r.t. $r^*$ and negative for others. It is easy to see that the ranking $r'$ in Example 1 maximizes the minimum value of $V(r, u)$: in fact in order to have $3 \uparrow$ in the first 6 positions, some $r' \in S$ has to give up at least 2 positions w.r.t. $r^*$.

Even when optimizing for (2), individual $u_5$ in Example 2 might have concerns for being the one receiving the largest part of the burden of satisfying the group-fairness constraint. The only way to improve on this situation is to introduce randomization into the process. This means producing a probability distribution over possible valid rankings instead of a single deterministic ranking.

**Example 3.** Consider the same instance of Example 1. The following distribution over four rankings $r_1 \ldots r_4$ maximizes the minimum expected value of $V(r, u) = r^*(u) - r(u)$ among all individuals in $U$:

$$
\begin{align*}
Pr(⟨u_1, u_4, u_3, u_7, u_2, u_6, u_5, u_6⟩) &= 1/4 \\
Pr(⟨u_2, u_1, u_3, u_6, u_4, u_5, u_7, u_5⟩) &= 1/2 \\
Pr(⟨u_2, u_1, u_3, u_7, u_5, u_6, u_4, u_6⟩) &= 1/16 \\
Pr(⟨u_5, u_1, u_3, u_7, u_3, u_6, u_4, u_6⟩) &= 3/16
\end{align*}
$$

It is easy to check that, under this distribution, everyone has expected value at least $-0.75$ (which is achieved by the four $r'$), while under the best deterministic solution (Example 2) we had $V(r, u_5) = -2 < -0.75$.

While in Example 2 the burden required for ensuring the group-fairness constraint was all on $u_4$ and $u_5$, in Example 3 it has been equally distributed among the four $r'$. Notice that all four rankings in the distribution above satisfy the group-fairness constraint in Example 1. However, by combining these four rankings probabilistically, we have succeeded in achieving a higher minimum expected value than is possible via any single deterministic ranking. In fact, we have also minimized the disparity in the expected value that each individual receives: whereas requiring all expected values to be the same is not mathematically possible when satisfying group constraints, the solution above comes as close as possible by minimizing the maximum gap. A complete problem definition formalizing these ideas is given in Section 3.

### Implications and practical deployment.

In order to guarantee the maximum possible value is brought to each individual, in this paper we embrace randomization and produce a probability distribution over possible valid rankings. This distributional fairness approach is very well suited for a search context in which the same query can be served many times for different users of a platform (e.g., headhunters searching for a specific type of professional on a career-oriented social networking platform such as LinkedIn or XING). Notice also that amortized fairness in the sense of [4, 26] is an immediate application of this distributional approach: if there are several rankings to be made, we can draw them independently from a fair distribution of rankings, so that the empirical properties of the sample approach those of the fair distribution.

However, the usefulness of randomization extends to settings with a single, non-repeated trial (as in, e.g., university admissions). In this case it is an essential tool to secure “ex-ante” (procedural) individual fairness, i.e., fairness of the procedure by which the outcome is selected, as opposed to “ex-post” fairness, which is based on the final outcome alone (see, e.g., [5]).

Regarding implementation and transparency issues, notice that instead of treating the algorithm as a black box outputting a single ranking, one can make the entire distribution public. For instance, we can publish the distribution described in Example 3 above, letting all the individuals verify the expected value, as well as the fact that this distribution is optimal under the maximin-fair criterion (see Section 3). Then one of the four rankings $r_1 \ldots r_4$ can be picked at random, via any fair and transparent lottery mechanism or coin-tossing protocol. Moreover, our algorithms guarantee that the

---

1This situation resembles some cases in fair classification in which enforcing statistical parity constraints cause a form of unfairness from an individual viewpoint [9].
optimal distribution found is supported on a small (polynomial-size) set of rankings, even if the space of all valid rankings is exponential.

**Paper contributions and roadmap.** In the rest of this paper, following the randomized maxmin-fairness framework [12], we study how to efficiently and accurately compute this type of distributions over the rankings satisfying a given set of group-fairness constraints. We achieve the following contributions:

- We introduce the distributional maxmin-fair ranking framework and provide the formal problem statement (Section 3.1). We show that maxmin-fair ranking distribution maintains within-group meritocracy and, in certain cases, it has the desirable properties of being generalized Lorenz-dominant, and minimizing social inequality (Section 3.2).
- Our main result is an exact polynomial-time algorithm to find maxmin-fair distributions of many problems, including ranking (Section 4). A quicker method to find maxmin-fair distributions approximately is explained in Appendix A.4.
- We also provide an exact optimal solution (Section 5) for the deterministic version of the problem as in (2). This is achieved by a means of a variant of Celis et al. [7]. We use this as a baseline allowing us to quantify the advantage of probabilistic rankings over the optimal deterministic ranking.
- Our experiments on two real-world datasets confirm empirically the advantage of probabilistic rankings over deterministic rankings in terms of minimizing the inequality for the worst-off individuals (Section 6).

To the best of our knowledge, this is the first work studying the problem of minimizing the amount of individual unfairness introduced when enforcing group-fairness constraints in ranking. A major contribution is showing how randomization can be a key tool in reconciling individual and group fairness: we believe that this might hold for other problems, besides ranking.

## 2 RELATED WORK

There are some works on algorithmic fairness focused on individual fairness, but none of them considers them in conjunction with group fairness. Dwork et al [9] introduce a notion of individual fairness in classification problems. Roughly speaking, their definition requires that all pairs of similar individuals should be treated similarly. This is impossible to satisfy with a deterministic classifier so, similarly to ours, their definition of fairness requires randomized algorithms. The individual similarity metric is assumed given, while they base their notion of ‘similar treatment’ on the difference between the probabilities of a favourable classification. Kearns et al. [19] introduce the notion of meritocratic fairness in the context of selecting a group of individuals from incomparable populations (with no group-fairness constraint). Their notion intuitively requires that less qualified candidates do not have a higher chance of getting selected than more qualified ones. Another work focusing on individual fairness is that of Biega et al. [4], which aims at achieving equity-of-attention fairness amortized across many rankings, by requiring that exposure be proportional to relevance.

Our previous work [12] presents a very general framework to deal with individual fairness, based on randomized maxmin-fairness: the idea is to use a distribution of solutions in order to maximize the expected value for the worst-off individual. In particular, [12] analyzes the case of unweighted matching with no group-fairness constraint: presents efficient algorithms and shows that these maxmin-fair matching distributions minimize inequality. While the techniques from [12] are combinatorial and can only deal with unrestricted matchings, we greatly generalize the algorithmic results therein via convex optimization techniques, showing that for a wide class of problems (including weighted matching and ranking with constraints), a maxmin-fair distribution may be found in polynomial time; we only require the existence of a weighted optimization oracle (see Section 4).

The bulk of recent literature on fairness in ranking [3, 7, 13, 26, 30, 31, 34] and learning-to-rank [10, 21, 27] deals with group fairness. Singh and Joachims [26] propose an algorithm computing a fair probabilistic ranking maximizing expected global utility. The fairness constraints expressible in their framework apply to the ranking distribution and not to each single ranking, as required by the group-fairness constraints we use. Celis et al. [7] also investigate fair ranking with group-fairness constraints with an objective function of the form (1), assuming the values \( W(r, u) \) satisfy the Monge condition. They give a polynomial-time algorithm for disjoint protected groups, and a faster greedy algorithm that works when only upper bound constraints are given. When the protected groups are allowed to overlap the problem becomes NP-hard and a polynomial-time approximation algorithm is provided in [7].

## 3 MAXMIN-FAIR RANKING

We are given a set of \( n \) individuals to be ranked \( \mathcal{U} \), a partition of \( \mathcal{U} \) into groups \( C_1, \ldots, C_t \), and a relevance function \( R : \mathcal{U} \rightarrow \mathbb{R} \). For the sake of simplicity we assume that ties are broken so that all \( R(u) \) are distinct. Moreover, we are given group-fairness constraints as in [7], defined by setting, for each \( i \in [n] \) and \( k \in [t] \), a lower bound \( l^k_i \in \mathbb{N} \) and an upper bound \( u^k_i \in \mathbb{N} \) on the number of individuals from class \( k \) in the first \( i \) positions. We denote by \( \mathcal{R} \) the set of all possible rankings of \( \mathcal{U} \) (bijections from \( \mathcal{U} \) to \( [n] \)), and by \( S \subseteq \mathcal{R} \) the set of all valid rankings:

\[
S = \left\{ r \in \mathcal{R} \mid l^k_i \leq \sum_{u \in C_k \mid r(u) \leq i} u^k_i \leq u^k_i \forall i \in [n], k \in [t] \right\}.
\] (3)

Finally, we consider a value function \( V : S \times \mathcal{U} \rightarrow \mathbb{R} \) such that \( V(r, u) \) represents the value (utility) that placing \( u \) at position \( r(u) \) brings to the individual \( u \), relative to \( u \)’s quality \( R(u) \). As we are interested in modeling meritocratic fairness, our value function must take into consideration the input relevance score \( R(u) \) and the produced ranking \( r(u) \). We consider value functions of the form:

\[
V(r, u) = f(r(u)) - g(u),
\] (4)

where \( f : [n] \rightarrow \mathbb{R} \) is a decreasing function and \( g : \mathcal{U} \rightarrow \mathbb{R} \) is increasing in \( R(u) \).

The intuition is the following: suppose that being assigned at position \( i \) carries intrinsic utility \( f(i) \), while \( u \)’s merit for the ranking problem is \( g(u) \) (which may depend on \( u \) and hence also on \( R(u) \)); then \( V(r, u) \) measures the net difference between \( f(r(u)) \) and \( g(u) \), i.e., how much \( u \) has gained in \( r \) w.r.t. \( u \)’s actual merit. In typical applications we can take any decreasing function \( p : [n] \rightarrow \mathbb{R}^{\geq 0} \) encoding position bias or exposure (see [8] for common models) and set \( f = p \) and \( g = p \circ r^* \). As simple examples, by setting...
\[ p(i) = n - i \text{ and } p(i) = \log(n/i), \] can get \( V(r, u) = r'(u) - r(u) \) and \( V(r, u) = \log(r'(u)/r(u)). \) When the ranking is a selection process where \( k \in \mathbb{N} \) individuals are selected and there is no advantage to being ranked first over \( k_{th} \) as long as one is selected, we may use

\[
V(r, u) = \begin{cases} 
1, & \text{if } r'(u) > k \text{ and } r(u) \leq k \\
-1, & \text{if } r'(u) \leq k \text{ and } r(u) > k \\
0, & \text{otherwise.}
\end{cases}
\]

These are but a few examples. Determining which value function \( V \) is best from a psychological or economical standpoint is beyond the scope of this work. Instead we take \( V \) as given and design algorithms which can efficiently deal with any function of the form (4).

### 3.1 Maxmin-fairness framework

Consider an input \( T \) of a general search problem which defines implicitly a set \( S = S(T) \) of feasible solutions, assumed to be finite and non-empty. Let \( U \) denote a finite set of individuals and let us associate with each solution \( S \in S \) and each individual \( u \in U \) a real-valued satisfaction \( A(S, u) \in \mathbb{R} \) (which in [12] takes binary values). Consider a randomized algorithm \( A \) that, for any given problem instance \( T \), always halts and selects a solution \( A(T) \in S \).

Then \( A \) induces a probability distribution \( D \) over \( S : \Pr[D] = \Pr[A(T) = S] \forall S \in S \). Denote the expected satisfaction of each \( u \in U \) under \( D \) by \( D[u] = \mathbb{E}_{D} [A(S, u)] \). A distribution \( F \) over \( S \) is maxmin-fair for \((U, A)\) if it is impossible to improve the expected satisfaction of any individual without decreasing it for some other individual which is no better off, i.e., if for all distributions \( D \) over \( S \) and all \( u \in U \),

\[
D[u] > F[u] \implies \exists v \in U \setminus D[v] < F[v] \leq F[u].
\]

Maxmin-fair distributions always exist [12]. Due to the convexity of the set of feasible probability distributions, an equivalent definition can be given based on the sorted vectors of expected satisfactions. Given a distribution \( D \) over \( S \), let \( D^\uparrow = (\lambda_1, \ldots, \lambda_n) \) be the vector of expected satisfactions \((D[u])_{u \in U}\) sorted in increasing order. Let \( \lambda_i \) denote the lexicographical order of vectors: i.e., \((v_1, \ldots, v_n) \succ (w_1, \ldots, w_n)\) if there is some index \( i \in [n] \) such that \( v_j > w_j \) and \( v_j = w_j \) for all \( j < i \). Write \( v \succeq w \) if \( v = w \) or \( v \succ w \). Then a distribution \( F \) over \( S \) is maxmin-fair if and only if \( F \uparrow \succeq D^\uparrow \) for all distributions \( D \) over \( S \).

### Problem 1 (Maxmin-fairness in combinatorial search)

Given a fixed search problem, a set \( U \) of individuals, and a satisfaction function \( A \), design a randomized algorithm \( A \) which always terminates and such that, for each instance \( T \), the distribution of \( A(T) \) is maxmin-fair for \((U, A)\).

Problem 1 is a general formulation of maxmin-fairness in search problems. Different choices for the set of feasible solutions \( S \) and the satisfaction function \( A \) lead to different algorithmic problems. The problem involves continuous optimization over infinitely many distributions, each defined over the set \( S \) of valid solutions (which is exponential-size). Despite these difficulties, we will show that Problem 1 is tractable under mild conditions (Section 4).

García–Soriano and Bonchi [12] instantiate Problem 1 with the case of matching. The main problem studied in the rest of this paper is obtained by instantiating Problem 1 with the case of ranking under group-fairness constraints with individual-level value function: in our setting \( S \) is the set of rankings \( r \) over \( U \) satisfying the group-fairness constraints, and \( A(S, u) \) is our value function \( V(r, u) \).

### Problem 2 (Maxmin-fair ranking with group-fairness constraints)

Given a set of individuals to be ranked \( U \), a partition of \( U \) into groups, a set \( S \) of rankings satisfying a given set of group-fairness constraints as defined in (3), and a value function \( V \) as defined in (4), design a randomized algorithm which outputs rankings in \( S \), such that its output distribution over \( S \) is maxmin-fair.

### 3.2 Properties of maxmin-fair rankings

We next state some important properties of maxmin-fair rankings. For the sake of readability, the proofs can be found in the Appendix.

**Theorem 3.1 (Intra-group meritocracy).** For any two individuals \( u_1, u_2 \in U \) belonging to the same group and such that \( R(u_1) \geq R(u_2) \), it holds that, if a distribution \( F \) over valid rankings \( S \) is maxmin-fair, then \( E_{F}[(r(u_1))] \geq E_{F}[(r(u_2))] \).

Our second property employs the notion of (generalized) Lorenz dominance from [25], a property indicating a superior distribution of net incomes. Consider two ranking distributions \( A \) and \( B \). Let \( A_k = A \uparrow \{k\} \) denote the \( k_{th} \) element of the expected satisfaction values sorted in increasing order. Then \( A \) dominates \( B \) if \( \sum_{k=1}^{n} A_k \geq \sum_{k=1}^{n} B_k \) \( \forall k \in [n] \), i.e., the expected cumulative satisfaction of the bottom individuals is always higher or equal in \( A \).

A distribution is generalized Lorenz-dominant if it dominates every other distribution. When it exists, such a distribution has a strong claim to being superior to all others, in terms of equity and efficiency [20, 25, 28]. A generalized Lorenz-dominant distribution must also be maxmin-fair. We show that a dominant distribution does exist for rankings, in the important case where only upper bound constraints are given in (3). Notice that in the case of two groups (e.g., a binary protected attribute), lower bound constraints may be replaced with an equivalent set of upper bound constraints.

**Theorem 3.2.** The maxmin-ranking with upper bounds is generalized Lorenz-dominant.

Since \( \sum_{u \in U} V(r, u) \) is a constant independent of \( r \), an easy consequence of Theorem 3.2 is that the maxmin-ranking distribution also minimizes social inequality in the sense of [12]: i.e., the maximum difference between the expected satisfactions of two users.

**Corollary 3.1.** The maxmin-ranking with upper bounds minimizes \( \max_{u \in U} D[u] - \min_{v \in U} D[v] \) over all ranking distributions \( D \), as well as any other quantile range.

Moreover, by the majorization inequality [18], it also maximizes any social welfare function that is additively separable, concave and symmetric w.r.t. \( U \).

**Corollary 3.2.** Suppose \( f : \mathbb{R} \rightarrow \mathbb{R} \) is concave. When only upper bound constraints are present, the maxmin-fair distribution maximizes \( \sum_{u \in U} f(D[u]) \) over all ranking distributions \( D \).

In particular, in this case the maxmin-fair distribution minimizes the variance of \( D \uparrow \), and when the values \( V(r, u) \) are positive, it also
maximizes, for instance, the Nash social welfare [22] (geometric mean) of expected satisfactions. It must also minimize the Gini inequality index when it is well-defined [25].

4 ALGORITHM

We show that our fair ranking problem (Problem 2) is efficiently solvable. Notice that the set $S$ of valid solutions can be exponential-size, so enumerating $S$ is out of the question in an efficient algorithm. Instead, we need a method to quickly single out the best solutions to combine for a maxmin-fair distribution. To show how this can be done, we abstract away from the specifics of the problem and show how to find a maxmin-fair distributions of general search problems (Problem 1). The following notion is key:

**Definition 1.** A weighted optimization oracle for $A : S \times U \to \mathbb{R}$ is an algorithm that, given $w : U \to \mathbb{R}^n$, returns $S^*(w)$ and $A(S^*(w), u)$ for all $u \in U$, where

$$S^*(w) \in \arg \max_{S \in S} \sum_{u \in U} w(u) \cdot A(S, u).$$

Roughly speaking, the intuition why these oracles are important is the following. Suppose we have constructed a distribution $D$ which is not maxmin-fair. By putting more weight on the individuals less satisfied by $D$, we can use the weighted optimization oracle to find a new solution $S$ placing more emphasis on them, which can be added to “push $D$ towards maxmin-fairness”.

Designing an efficient weighted optimization oracle is a problem-dependent task. Our first algorithmic result reveals that their existence suffices to solve Problem 1 efficiently.

**Theorem 4.1.** Given a weighted optimization oracle, Problem 1 is solvable in polynomial time.

We emphasize that Theorem 4.1 is very general and its applicability is in no way limited to ranking problems, or to value functions of a certain form. They apply to an arbitrary search problem (e.g., searching for a ranking, a matching, a clustering...) and an arbitrary set of individuals. As long as an efficient weighted optimization algorithm exists for $P$, it yields efficient algorithms for maxmin-fair-$P$ (Problem 1). The wide applicability of this condition implies that maxmin-fair distributions may be efficiently solved in a great many cases of interest: most polynomial-time solvable problems studied in combinatorial optimization (e.g., shortest paths, matchings, polymatroid intersection...) admit a polynomial-time weighted optimization oracle. Thus, Theorem 4.1 extends in a new direction the results of [12]: as efficient weighted matching algorithms exist, the main result of [12] becomes a corollary to Theorem 4.1 (up to the loss of a polynomial factor in runtime).

More importantly for us, the same holds for constrained ranking.

**Theorem 4.2.** Ranking with group-fairness constraints as in (3) and a value function of the form (4) admits a polynomial-time weighted optimization oracle.

**Corollary 4.1.** Maxmin-fair ranking with group-fairness constraints (Problem 2) is solvable in polynomial-time.

In Section 4.1 we prove Theorem 4.1 by solving a sequence of exponentially many constraints to be written down explicitly, but can nonetheless be solved efficiently via the ellipsoid method using a weighted optimization oracle. (As explained in Appendix A.4, if we settle for some approximation error, these LPs can also be solved approximately using techniques to solve zero-sum games and packing/covering LPs [11, 32, 33].) Finally, in Section 4.2 we show the existence of weighted optimization oracles for ranking (Theorem 4.2).

4.1 Proof of Theorem 4.1

We start by showing a weaker result concerning the computation of the optimal expected satisfaction values, rather than the actual distribution of solutions.

**Lemma 4.1.** Given a weighted optimization oracle, the expected satisfactions of a maxmin-fair distribution can be computed in polynomial time.

**Proof.** Let $F$ be a maxmin-fair distribution. We maintain the invariant that we know the expected satisfaction $a_\nu$ of $F$ for all $\nu$ in a subset $K \subseteq U$:

$$F[\nu] = a_\nu \text{ for all } \nu \in K,$$

$$K \neq \emptyset \implies F[\nu] \geq \max_{\nu \not\in K} a_\nu \text{ for all } \nu \not\in K. \tag{8}$$

Initially $K = \emptyset$. We show how to augment $K$ in polynomial time while maintaining (7) and (8), which gives the result since $K = U$ will be reached after at most $|U|$ iterations. We need to find the largest minimum expected satisfaction possible outside $K$ for a distribution $D$ subject to the constraints that the expected satisfaction inside $K$ must be equal to $a_\nu$. By (7), (8) and the lexicographical definition of maxim-fairness, for any distribution $D$ the constraints $D[\nu] = a_\nu$ for all $\nu \in K$ are equivalent to the constraints $D[\nu] \geq a_\nu$ for all $\nu \in K$. We can write our optimization problem as the following (primal) linear program:

$$\max \lambda$$

s.t. $\sum_{S \ni \nu} -p_S \cdot A(S, \nu) \leq -a_\nu \quad \forall \nu \in K$

$$\lambda + \sum_{S \ni \nu} -p_S \cdot A(S, \nu) \leq 0 \quad \forall \nu \not\in K \quad \tag{9}$$

$$\sum_{S \in S} p_S = 1$$

$$p_S \geq 0,$$

whose dual is

$$\min \mu - \sum_{\nu \in K} a_\nu w_\nu$$

s.t. $\sum_{\nu \not\in S} w_\nu \cdot A(S, \nu) \geq 0 \quad \forall S \in S$

$$\sum_{\nu \in U} w_\nu = 1$$

$$w_\nu \geq 0. \tag{10}$$

The dual (10) has $|U|$ variables but a possibly exponential number of constraints (one for each candidate solution $S$). To get around
whose numerators and denominators are specified with weighted optimization oracle. If we run the ellipsoid algorithm (which can be checked separately, or by the constraint \(\sum_{v \in S} w_v \cdot A(S, v) \leq \mu\)) and \(\mathcal{U}\) be a solution with value \(\lambda^*\) and \(\lambda^* \geq \max_{\alpha \in \mathcal{A}} \alpha^0\) since the objective function did not increase since the last iteration. In this case we simply add the constraint \(\mu - \sum_{v \in K} \alpha_v w_v = \lambda^*\) to (10) and change the objective function to minimize \(\sum_{v \in S} w_v\). This also yields an optimal solution to (11). But in this case the new solution \(\mathcal{W}^{**}\) must satisfy \(support(\mathcal{W}^{**}) \setminus K \neq \emptyset\), so we are back to the previous case.

We repeat this process until \(K = \mathcal{U}\). The number of iterations is at most \(|\mathcal{U}|\), and each iteration runs in polynomial time. □

### 4.2 Proof of Theorem 4.2

Proof. Sort \(\mathcal{U} = \{u_1, \ldots, u_n\}\) by decreasing order of \(w\) so that
\[
    w(u_1) \geq w(u_2) \geq \ldots \geq w(u_n).
\]
and let us identify \(\mathcal{U}\) with the set \(\{n\}\) for ease of notation, so that \(u_i = i\). Recall that the positions \([n]\) are sorted by decreasing \(f\):
\[
    f(1) \geq f(2) \geq \ldots \geq f(n).
\]
Because of the orderings defined by (12) and (13), the matrix \(W\) satisfies the following "Monge property":
\[
    W_{ij} = w(i) \cdot B(j, i).
\]
Thus we may apply the algorithm\(^2\) from [7] to find a valid ranking \(r\) maximizing \(\sum_{u \in \mathcal{U}} W^r_{\mathcal{U}, u}\), as required
\[
    W_{ij} + W_{ju} - (W_{iu} + W_{ju}) = (w(u) - w(v)) (f(i) - f(j)) \geq 0.
\]

\(^2\)In [7] an additional monotonicity property is assumed (that \(W_{iu}\) is decreasing with \(u\)), but it is easy to check that it is not actually needed.
by the definition of weighted optimization oracle from Section 4. Then we may compute each \( V(r, u) \) explicitly using \( f \) and \( g \). □

An important case is where only upper bounds are given in the group constraints, i.e., when the set of valid rankings is of the form

\[
S = \left\{ r \in \mathcal{R} \mid \left[ \left[ u \in C_k \mid r(u) \leq i \right] \leq u^k_i \right] \forall i \in [n], k \in [1] \right\}. \quad (14)
\]

Plugging in the algorithm from [7] into Theorem 4.2 we obtain:

Algorithm 2: Weighted optimization oracle for ranking with upper bounds

\[\begin{aligned}
\text{input} : & \text{Set of individuals } \mathcal{U}; \text{ weight function } w : \mathcal{U} \rightarrow \mathbb{R}; \text{ value function } V : S \times U \rightarrow \mathbb{R} \\
\text{output} : & \text{Best response ranking } r \text{ and } V(r, u) \text{ for all } u \in \mathcal{U} \\
1 & \text{Sort individuals in } \mathcal{U} \text{ in order of decreasing weight: } w(u_1) \geq w(u_2) \geq \ldots \geq w(u_n). \\
2 & \text{For each position } i \in [n] \text{ in increasing order (as in (13)), let } u \text{ be the smallest-index unassigned individual whose additional placement at position } i \text{ does not violate the group upper bound} \\
3 & \text{constraints in the first } i \text{ positions, and set } r(u) = i. \\
4 & \text{Return } r \text{ and } V(r, u) \text{ for all } u \in \mathcal{U}.
\end{aligned}\]

Corollary 4.2. Algorithm 2 is a weighted optimization oracle for ranking with upper bounds.

5 DETERMINISTIC BASELINE

In this section we present an exact optimal solution for the deterministic version of the problem as formulated in (2). This is useful for our experiments (Section 6) as it allows us to quantify the advantage of probabilistic rankings over deterministic rankings in terms of the amount of individual fairness maintained.

Although Celis et al. [7] study the problem of the form (1), we devise a variant of their algorithm to deal with the problem as in (2): this variant can be shown to provide the optimal deterministic ranking solution to the constrained ranking problem (2) when the group-fairness constraints are expressed in terms of upper bounds on the number of elements from each class that appear in the top-\( k \) positions, as in (14). As noted in Section 3 and in [7], in the case of two disjoint groups (e.g., a binary protected attribute), lower bound constraints may be replaced with an equivalent set of upper bound constraints.

Algorithm 3: Deterministic baseline

\[\begin{aligned}
\text{input} : & \text{Set of individuals } \mathcal{U}; \text{ relevance function } R : \mathcal{U} \rightarrow \mathbb{R}_{\geq 0} \\
\text{output} : & \text{Deterministic ranking } r. \\
1 & \text{Sort individuals in } \mathcal{U} \text{ in order of decreasing score: } \text{rk}(u_1) \geq \text{rk}(u_2) \geq \ldots \geq \text{rk}(u_n). \\
2 & \text{For each position } i \in [n] \text{ in increasing order (as in (13)), let } u \text{ be the smallest-index unassigned individual whose additional} \\
3 & \text{placement at position } i \text{ does not violate the group upper bound} \\
4 & \text{constraints in the first } i \text{ positions, and set } r(u) = i. \\
5 & \text{Return } r.
\end{aligned}\]

At the basis of our deterministic baseline (Algorithm 3) lies the idea of using the function \( \text{softmin}(x_1, \ldots, x_n) = -\ln(\sum_{i=1}^n e^{-x_i}) \)

\[
\text{Table 2: Minimum expected value produced by MF(0) and optimal deterministic solution, spread (maximum - minimum) of expected value, Gini inequality index (%), and discounted cumulative gain for IIT-JEE and Law-schools datasets for different values of } \alpha.
\]

| Dataset           | IIT-JEE          | Law-schools       |
|-------------------|------------------|-------------------|
| \( \alpha = 0.1 \) | \( \text{MF} \) | \( \text{MF} \)  |
| \( \text{min}(\text{MF}) \) | -26.82 | -0.87 |
| \( \text{sprd}(\text{det}) \) | -4.4 | -1.03 |
| \( \text{gini}(\text{MF}) \) | 53.76 | -3.58 |
| \( \text{gini}(\text{det}) \) | 433.0 | 899.0 |
| \( \text{DCG}(\text{MF}) \) | 0.6714 | 6.19 |
| \( \text{DCG}(\text{det}) \) | 1.062 | 7.027 |

\[
\text{Figure 1: IIT-JEE: Minimum expected value produced by MF(0) and optimal deterministic solution (left); distribution of expected value } V(r, u) \text{ for } \alpha = 0.3 \text{ (center); number of iterations (calls to the optimization oracle) vs error } \epsilon. \text{ (right).}
\]

to force the algorithm from [7] to approximately maximize a minimum instead of a sum, and observe that the limiting behaviour of the function \( f \rightarrow \text{softmin}(Mx)/M \) must also occur in this case for finite \( M \) because the algorithm from [7] does not depend on the specific values of the matrix \( W_i \), but only on the existence of an ordering of rows/columns of \( W \) where the Monge property holds (see Section 4.2).

Theorem 5.1. When the group-fairness constraints are defined only by upper bounds, Algorithm 3 returns a ranking \( r' \) such that

\[
r' \in \arg \max_{r \in S} \text{rk}(u) \text{ for all } u \in \mathcal{U}.\]

6 EXPERIMENTS

Datasets. We use two real-world datasets containing gender information and one score for each individual. Our first dataset comes from the IIT Joint Entrance Exam (known as IIT-JEE 2009) [6], from which we select the top \( N = 1000 \) scoring males and the top \( N \) scoring females. The score distribution is heavily biased at the top, with just four females making the top-100. Our second dataset is much less skewed: it contains admissions data from all of the public law schools in the United States\(^3\). We use the top \( N = 1000 \) LSAT scorers, of whom 362 are female.

Settings. We impose the following group-fairness constraints, parameterized by \( \alpha \in [0, \frac{1}{2}] \): at least \( \left\lceil \alpha \cdot k - 1 \right\rceil \) females should be ranked in the top \( k \), for \( k = 1, 2, \ldots, 2N \). We employ \( V(r, u) = r^*(u) - r(u) \) as our value function, where \( r^* \) is the ranking by decreasing score.

\(^3\)https://jumpshare.com/v/yRUSJrew3hesGGN0iJ3A
\(^4\)http://www.seaphe.org/databases.php
**Algorithms.** We implement our maxim-fair solver for ranking, using the technique described in Appendix A.4 to solve the LPs approximately with an additive error parameter \( \epsilon \); \( \epsilon = 1 \) corresponds to an additive error in expected ranking position of 1 (out of 2000 for IIT-JEE and out of 1000 for Law school). We denote by \( \text{MF}(\epsilon) \) the ranking distribution produced by our approximate maxim-fair algorithm with parameter \( \epsilon \), and by \( \text{MF}(0) \) the one obtained with the smallest \( \epsilon \) tested (0.5). Our code is available on Dropbox3.

In order to quantify the advantage of probabilistic rankings over the optimal deterministic ranking, we also test the deterministic algorithm we devised (Algorithm 3) to solve the problem in (2). This provides the strongest possible deterministic competitor for our algorithm.

**Measures.** Besides comparing the minimum expected value, which is the main focus of our work, we also report other measures of inequality of the produced solution: spread (maximum - minimum) of expected value and Gini inequality index [14] (after normalizing values to the interval \([0, 1]\) to make the index well-defined). Finally, to examine if there is a loss in global ranking quality, we use the popular discounted cumulative gain metric \([4, 6, 7, 17, 26, 29]\), which can be defined as \( \text{DCCG}(r) = \sum_{u \in U} \text{score}(u)/\log(r(u) + 1) \).

**Results.** The first two rows of Table 2 report the expected value (over a random ranking from the distribution) of the solution for the worst-off individual; we can observe that the maxmin-fair solution improves significantly on the optimal deterministic solution, with the gap between the two increasing with \( \alpha \) (the strength of the group-fairness constraint). The same can be observed in Figure 1 (left) and Figure 2 (left) for the two datasets. We do not report the average value of the solution for all individuals because it is the same for every ranking, as rankings are bijections onto \([n]\).

In Table 2 we can also observe that the inequality measures for the maxim-fair solution are always smaller than the optimal deterministic one. Finally, we report the ranking-quality measure DCG. Since, unlike the three other measures in Table 2 DCG is defined for deterministic rankings, we report average and standard deviation. We see that DCG is nearly the same for \( \text{MF}(0) \) and \( \text{det} \). Thus in this experiment improving individual fairness with respect to a group-only fairness solution incurs a negligible loss in DCG.

Figure 1 (center) and Figure 2 (center) depict the average expected value of the bottom \( k \) individuals in three solutions: our best solution \( \text{MF}(0) \), an approximate solution with \( \epsilon = 10 \), and the optimal deterministic solution. The peculiar behaviour of the curve in Figure 1 (center) (constant up to roughly \( k = n/2 \) for MF) is due to the skew of the input scores, which forces the maxim-fair solution to essentially increase the ranking positions of most men by a certain minimum amount \( X \) and decrease that of most women by \( X \) with the best possible distribution. We notice that the maxim-fair solution yields stronger cumulative value to the worst-off users than the other two do, for any \( k \). In particular, the maxim-fair solution found Lorenz-dominates the approximate one and the deterministic one, in accordance with Theorem 3.2. Because of the error allowed, the approximate solution \( \text{MF}(10) \) stays somewhat below \( \text{MF}(0) \) and its curve crosses that of the deterministic solution sporadically before distancing itself again. Finally, Figure 1 (right) and Figure 2 (right) show the number of calls to the optimization oracle (which is also the size of the support of the ranking distribution) as a function of the additive error parameter \( \epsilon \). Runtime is linear in the number of calls to the optimization oracle. The longest runtime of our Python implementation of MF (which occurred on the IIT-JEE dataset with \( \alpha = 0.3 \) and \( \epsilon = 0.5 \)) was under one hour.

**7 CONCLUSIONS**

We introduced the problem of minimizing the amount of individual unfairness introduced when enforcing group-fairness constraints in ranking. We showed how a randomized approach ensures more individual fairness than the optimal solution to the deterministic formulation of the problem. We proved that our maxim-fair ranking distributions provide strong fairness guarantees such as maintaining within-group meritocracy and, under a mild assumption (i.e., when we have only upper-bound constraints or when the protected attribute is binary), they have the desirable properties of being generalized Lorenz-dominant, and minimizing social inequality. Besides the technical contributions, our work shows how randomization is key in reconciling individual and group fairness. In our future work we plan to extend this intuition beyond ranking.

**ACKNOWLEDGEMENTS**

The authors acknowledge support from Intesa Sanpaolo Innovation Center. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**REFERENCES**

[1] [n.d.]. European Commission Diversity Charters. https://ec.europa.eu/info/policies/justice-and-fundamental-rights/combatting-discrimination/tackling-discrimination/diversity-management/diversity-charters-eu-country_en.

[2] [n.d.]. U.S. Equal Employment Opportunity Commission. https://www.eeoc.gov/laws/index.cfm.

[3] Abolfazl Asudeh, H. V. Jagadish, Julia Stoya novich, and Gautam Das. 2019. Designing Fair Ranking Schemes. In SIGMOD. 1259–1276.

[4] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In SIGIR. 405–414.

[5] Gary E Bolton, Jordi Brandts, and Axel Ockenfels. 2005. Fair Procedures: Evidence from games involving lotteries. The Economic Journal 115, 506 (2005), 1054–1076.

[6] L. Elisa Celis, Anaí Mehrotra, and Nisheeth K. Vishnoi. 2020. Interventions for ranking in the presence of implicit bias. In FAT*’20. 369–380.

[7] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2018. Ranking with Fairness Constraints. In ICALP.

[8] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In WSDM. 87–94.

[9] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In ITCS. 214–226.

[10] Cynthia Dwork, Michael P. Kim, Omer Reingold, Guy N. Rothblum, and Gal Yona. 2019. Learning from Outcomes: Evidence-Based Rankings. In FOCS. 106–125.

[11] Yoav Freund and Robert E Schapire. 1999. Adaptive game playing using multiplicative weights. Games and Economic Behavior 29, 1-2 (1999), 79–103.
Proof. Recall the form of our value function \( V(r, u) = f(r(u)) - g(u) \) and observe that for any distribution \( D \),
\[
D[u] = \mathbb{E}_{r \sim D} [V(r, u)] = \mathbb{E}_{r \sim D} [f(r(u))] - g(u).
\]

Let us write \( r \in F \) to mean that \( r \) occurs with non-zero probability in the max-min fairness distribution \( F \). We show that if \( u_1, u_2 \) belong to the same group and \( R(u_1) \geq R(u_2) \), then the following holds:
\[
f(r(u_2)) > f(r(u_1)) \text{ for some } r \in F \implies F[u_1] \geq F[u_2].
\]

Suppose by contradiction that (16) fails, so \( f(r(u_2)) > f(r(u_1)) \) but \( F[u_1] < F[u_2] \) for some \( r \in F \). Let \( f \) denote a ranking which is identical to \( r \) except that the positions of \( u_1 \) and \( u_2 \) are swapped. As \( u_1 \) and \( u_2 \) belong to the same group, swapping their positions will not affect the group-fairness constraints, so \( f \) is a valid ranking too. Consider a distribution \( D \) over valid rankings \( S \) obtained by drawing \( s \) from \( D \) and returning \( s \) if \( s \neq r \) and \( f \) if \( s = r \). We have \( \mathbb{E}_{s \in D} [f(s(u_1))] = \mathbb{E}_{s \in F} [f(s(u_1))] = \mathbb{E}_{s \in D} [f(s(u_2))] = \mathbb{E}_{s \in F} [f(s(u_2))] \) and therefore, by (15), \( D[u_1] > F[u_2] \). Moreover, \( \forall v \in \mathcal{U} \setminus \{u_1, u_2\} \) it holds that \( F[v] = D[v] \). Therefore \( D \) is a distribution improving the expected satisfaction of \( u_1 \) w.r.t. \( F \) and such that no \( v \in \mathcal{U} \) exists such that \( F[v] \leq F[u_1] \) and \( D[v] < F[v] \), thus contradicting the assumption that \( F \) is maxmin-fair. This proves (16).

To prove (17), consider first the case \( F[u_1] \geq F[u_2] \). Since \( R(u_1) \geq R(u_2) \) implies \( g(u_1) \geq g(u_2) \), in this case substituting \( D = F \) in (15) we trivially obtain (17). If instead \( F[u_1] < F[u_2] \) then, by (16), we conclude that \( f(r(u_1)) \geq f(r(u_2)) \) for all \( r \in F \), which implies (17), as we wished to show.

\( \square \)

### A.2 Proof of Theorem 3.2

In this subsection we consider the case where we only have upper bounds in the group-fairness constraints.

First we need a result characterizing the minimum expected satisfaction of a maxima-fair distribution. It has been inspired by the proof of [12][Theorem 15]. While [12] only considers matroid problems (which do not cover constrained ranking), our key insight is that this type of argument can be generalized whenever there is a weight optimization oracle depending only on the weight order (as opposed to the numerical values of the weights). This is true of the greedy algorithm from Corollary 4.2 (Algorithm 2).

**Lemma A.1.** Let \( \lambda : 2^\mathcal{U} \to \mathbb{R} \). There is a distribution of valid rankings such that \( D[u] \geq \lambda_u \) if and only if
\[
\max_{S \subseteq \mathcal{U}} \sum_{u \in S} A(S, u) \geq \sum_{u \in S} \lambda_u \quad \text{for all } S \subseteq \mathcal{U}. \tag{18}
\]

**Proof of Lemma A.1.** Given a set \( E \), let \( \Delta(E) = \{ x : E \to \mathbb{R}^E \mid \sum_{e \in E} x_e = 1 \} \) denote the set of distributions over \( E \). Consider the following two-player zero-sum game: Player 1 (the maximizer) chooses a distribution of solutions \( p \in \Delta(S) \), Player 2 (the minimizer) chooses a distribution of users \( w \in \Delta(\mathcal{U}) \), and the payoff for Player 1 when she plays \( \delta \in S \) and Player 2 plays \( u \in \mathcal{U} \) is \( A(S, u) - \lambda_u \). The value of this game is
\[
v = \max_{p \in \Delta(S)} \min_{w \in \Delta(\mathcal{U})} \left( \sum_{u \in S} p_S(A(S, u) - \lambda_u) \right);
\]
the required distribution exists when \( v \geq 0 \). By Von Neumann’s minimax theorem we have
\[
v = \min_{w \in \Delta(\mathcal{U})} \max_{S \subseteq \mathcal{U}} \left( \sum_{u \in S} w_u(A(S, u) - \lambda_u) \right). \tag{19}
\]
Thus, \( v \geq 0 \) exactly when for all \( w \in \Delta(\mathcal{U}) \), it holds that
\[
\max_{S \subseteq \mathcal{U}} \sum_{u \in S} w_u(A(S, u) \geq \sum_{u \in S} w_u \lambda_u. \tag{20}
\]

The result will follow if we can show that the minimization problem (19) has an optimal solution of the form
\[
w_u = \begin{cases}
\frac{1}{|X|} & \text{if } u \in X \\
0 & \text{otherwise}
\end{cases}
\]
for some non-empty \( X \subseteq \mathcal{U} \), because for \( w_u \) of the form (21), (20) simplifies to (18) on multiplication by \(|X|\). We have seen in Corollary 4.2 that for each \( w \), \( \max_{S \subseteq \mathcal{U}} \sum_{u \in \mathcal{U}} w_u(A(S, u) - \lambda_u) \) can
be optimized by an oracle (Algorithm 2) that only depends on the order determined by \( w \) (observe that subtracting \( \lambda u \) from \( A(S, u) \) amounts to adding \( \lambda \) to the function \( g \) in the definition of \( A(S, u) \)). In other words, for any bijection \( \pi : [n] \to \mathcal{U} \) and any weight \( w \geq 0 \) compatible with \( \pi \) (i.e., satisfying \( w_{\pi(1)} \geq w_{\pi(2)} \geq \ldots w_{\pi(n)} \)), we have

\[
\max_{S \in S} \sum_{u \in \mathcal{U}} w_u (A(S, u) - \lambda u) = \sum_{u \in \mathcal{U}} w_u (A(G(\pi), u) - \lambda u),
\]

where \( G(\pi) \) is the solution returned by the greedy weighted optimization oracle.

Fix an order \( \pi : [n] \to \mathcal{U} \) and let \( B_u = A(G(\pi), u) - \lambda u \). Consider the minimization problem

\[
\min \left\{ \sum_{u \in \mathcal{U}} w_u B_u \mid w \in \Delta(\mathcal{U}), w \text{ compatible with } \pi \right\}.
\]

(22)

Let \( d_n = w_{\pi(n)} \) and \( d_i = w_{\pi(i)} - w_{\pi(i+1)} \). The compatibility conditions for \( w \) may be rewritten as \( d_i \geq 0 \) for all \( i \), and the distributional constraint \( \sum_i w_i = 1 \) becomes \( \sum_i i \cdot d_i = 1 \). If we write \( z_i = \sum_{j \leq i} B_j \), then (22) becomes

\[
\min \left\{ \sum_{i \in [n]} d_i \cdot z_i \mid d_i \geq 0, i \sum_{i \in [n]} d_i = 1 \right\} = \min \left\{ \frac{z_t}{t} \mid t \in [n] \right\};
\]

the last equality is easily seen to hold because \( z_i \leq \lambda \cdot i \) for all \( i \) implies \( \sum_i i \cdot z_i \leq \lambda \sum_i i \cdot d_i = \lambda \). Therefore for each \( \pi \), an optimal solution to (22) is of the form (21), where \( X = \{ \pi(1), \ldots, \pi(t) \} \); hence the same is true of an optimal solution to (19). \( \square \)

Corollary A.1. The minimum expected satisfaction in any maxmin-fair distribution of valid rankings is

\[
\min_{S \subseteq S} \frac{\sum_{u \in \mathcal{U}} A(S, u)}{|X|},
\]

where \( H \) is given by (24) and \( S = \bigcup_{j \in I} B_j \).

(25)

We also need the following technical lemma concerning the behaviour of the expression in Corollary (A.1).

Lemma A.2. The following function \( H : 2^\mathcal{U} \to \mathbb{R} \) is submodular:

\[
H(E) = \max_{S \subseteq S} \sum_{u \in E} A(S, u).
\]

(24)

Proof. Let \( J(E) = \max_{S \subseteq S} \sum_{u \in E} f(S(u)) \). Then \( H(E) = J(E) - \sum_{u \in E} f(u) \), i.e., \( H \) is the difference between \( J \) and a modular function. So it suffices to show that \( J \) is submodular; let us fix \( X \subseteq Y \) and \( Z \subseteq \mathcal{U} \setminus Y \). Recall from corollary 4.2 that \( \sum_u w_u f(S(u)) \) is maximized by a greedy algorithm. By setting \( w_0 = 1 \) for \( u \in E \) and \( w_0 = 0 \) elsewhere, it can be used to compute \( J(E) \) for any \( E \); let us denote by \( r_x \) the ranking returned. Whenever we have two equal weights \( w_0 = w_0 \), we can break ties in Algorithm 2 in favor of X, followed by \( Y \setminus X \), and \( \mathcal{U} \setminus (Y \cup Z) \). Then the greedy algorithm to maximize \( f(Y \cup Z) \) attempts to place the elements of \( X \) in top positions whenever possible, then elements of \( Y \), and then elements of \( Z \). This ensures that in \( r_X \) and \( r_X \cup Z \) the position of the elements of \( X \) is the same, allowing us to simplify the marginal gains:

\[
J(X \cup Z) - J(X) = \sum_{u \in \mathcal{U} \setminus Z} f(r_{X \cup Z}(u)) - \sum_{u \in X} f(r_X(u)) = \sum_{u \in Z} f(r_{X \cup Z}(u)).
\]

Similarly,

\[
J(Y \cup Z) - J(Y) = \sum_{u \in \mathcal{U} \setminus Z} f(r_{Y \cup Z}(u)) - \sum_{u \in Y} f(r_Y(u)) = \sum_{u \in Z} f(r_{Y \cup Z}(u)).
\]

Moreover, for any \( x \in Z \), \( r_{X \cup Z}(x) \leq r_{Y \cup Z}(x) \) by the greedy rule in Corollary 4.2 and our tie-breaking rule. Therefore \( f(r_{X \cup Z}(x)) \geq f(r_{Y \cup Z}(x)) \), which implies \( J(X \cup Z) - J(X) \geq J(Y \cup Z) - J(Y) \), as desired.

The following is an analog of the “fair decompositions” of [12]:

Lemma A.3. Define a sequence of sets \( B_1, B_2, \ldots, B_k \) iteratively by \( B_1 \) is a maximal non-empty set \( X \subseteq \mathcal{U} \setminus S_{i-1} \minimizing \)

\[
\frac{|X|}{H(X \cup S_{i-1}) - H(S_{i-1})}.
\]

(28)

We stop when \( S_k = L \) (i.e., \( k \) is the first such \( i \)); this will eventually occur as the sequence \( (S_i) \) is strictly increasing. Then for every \( i \in [k] \), the following hold:

(a) The expected satisfaction of any \( u \in B_i \) in any maxmin-fair distribution \( F \) is \( F[u] = \frac{H(B_i)}{|B_i|} \).

(b) For all \( u \in S_i \setminus B_i \), we have \( F[u] < F[v] \).

(c) For any \( D \subseteq \Delta(S) \) and \( m \leq |\mathcal{U}| \), it holds that \( \sum_{j=1}^m D_{(j)} \leq \sum_{j=1}^m F_{(j)} \).

Proof. Let \( \lambda_j = \frac{H(B_j)}{|B_j|} \) for all \( j \in [k] \). Notice that, using (25), \( H(S_i) = H(S_{i-1} \cup B_i) = \lambda_i |B_i| + H(S_{i-1}) \) holds for all \( i \), hence

\[
H(S_i) = \sum_{j=1}^i \lambda_j |B_j|.
\]

(26)

Notice also that the definition of \( H \) trivially implies that for any distribution \( D \) we must have

\[
H(X) \geq \sum_{u \in X} D[u].
\]

(27)

We can give an alternative definition of \( S_i \) as

\[
S_i \text{ is a maximal set } X \not\subseteq S_{i-1} \text{ minimizing } \frac{H(X) - H(X \cap S_{i-1})}{|X \setminus S_{i-1}|}.
\]

(28)

Indeed, for any fixed difference \( Y = X \setminus S_{i-1} \), the submodularity of \( H \) (Lemma A.2) implies that the minimum of the numerator in (28) is attained for a set \( X \) which properly contains \( S_{i-1} \). In particular, we have that for any \( X \subseteq \mathcal{U} \) and \( j \in [k] \), \( H(X) - H(X \cap S_{i-1}) \geq \lambda_j |X \setminus S_{i-1}| \), and by replacing \( X \) with \( X \cap S_j \) above we also get \( H(X \cap S_j) - H(X \cap S_{j-1}) \geq \lambda_j |X \cap B_j| \), implying

\[
H(X) \geq \lambda_i |X \setminus S_{i-1}| + \sum_{j \in (i]} \lambda_j |X \cap B_j|, \quad \forall i \in [k].
\]

(29)

We show that properties (a) and (b) hold for all \( i \leq t \leq k \), reasoning by induction on \( t \). There is nothing to show when \( t = 0 \) or \( S_t = \mathcal{U} \) (in the latter case, \( k < 0 \), so assume that \( t \geq 1 \) and the claims hold for all \( i < t \); we show they also hold for \( i = t \).
From property (a) in the induction hypothesis, we know that in the maxmin-fair distribution, \( F[u] \geq \lambda_i \) for \( u \in S_i \), \( i \leq i \). We can use Lemma A.1 to determine the minimum expected satisfaction of \( F \) outside \( S_{i-1} \); we conclude, by (29), that \( \min_{u \notin S_{i-1}} F[u] \geq \lambda_i \). As (26) shows, equality in (29) is attained when \( X = S_i \), thus by (27) we must in fact have \( \min_{u \notin S_{i-1}} F[u] = \lambda_i \) and \( F[u] = \lambda_i \) for all \( u \in B_i \), proving (a).

To prove (b), we need to show the strict inequality \( \lambda_{i-1} < \lambda_i \). By Lemma (A.2), the function \( J(X) = H(U \cup S_{i-1}) - H(S_{i-1}) \) is submodular. A consequence of this is that, if \( X, Y \) are non-empty sets maximizing \( J(X)/|X| \), then \( X \cup Y \) also minimizes \( J(X)/|X| \).

Indeed, suppose \( \frac{J(Y)}{|Y|} > \frac{J(X)}{|X|} \). By the submodularity of \( J \),

\[ J(X \cup Y) + J(X \cap Y) > J(X) + J(Y) = \lambda(|X| + |Y|). \]

Notice that \( J(X \cup Y) \geq \lambda|X \cup Y| \) and \( J(X \cap Y) \geq \lambda|X \cap Y| \) by definition. If any of these two inequalities were strict we would have the contradiction

\[ J(X \cup Y) + J(X \cap Y) > \lambda(|X \cup Y| + |X \cap Y|) = \lambda(|X| + |Y|). \]

Hence these inequalities are not strict, and \( J(X \cup Y) = \lambda|X \cup Y| \).

Now, due to the maximality of \( B_i \) as defined by (25), the set \( B_i \) is the union of all non-empty sets \( X \) minimizing \( J(X)/|X| \). This means that, when \( t > 1 \), the strict inequality \( \lambda_i > \lambda_{i-1} \) holds (otherwise \( B_i \) would not be maximal), which by (a) implies (b).

Finally we show (c). We argue by contradiction. Pick a counterexample with minimum \( m \); then \( m \geq 1 \). Let \( i \) be such that \( |S_{i-1}| < m \leq |S_i| \). Then we have \( \sum_{j=1}^{m-1} D_{ij} \leq \sum_{j=1}^{m-1} F_{ij} \) and \( \sum_{j=1}^{m} D_{ij} > \sum_{j=1}^{m} F_{ij} \), thus \( D_{im} > F_{im} = \lambda_i \) by properties (a) and (b). Now let \( X \) be the individuals with the \( m \) smallest satisfactions in \( D \). It follows that

\[ H(S_i) \geq \sum_{u \in S_i} D[u] \geq \sum_{u \in X} D[u] + (|S_i| - m)D_{im} = \sum_{j \leq m} D_{ij} + (|S_i| - m)D_{im} > \sum_{j=1}^{m} F_{ij} + (|S_i| - m)\lambda_i = \sum_{j \leq m} \lambda_j B_j = H(S_i). \]

This contradiction completes the proof.

**Proof of Theorem 3.2.** Property (c) of Lemma A.3 asserts that the maxmin-fair distribution \( F \) is generalized Lorenz-dominant.

**Proof of Theorem 5.1.**

Proof. Sort \( U = \{u_1, \ldots, u_n\} \) by increasing order of \( g \) so that

\[ g(u_1) \geq g(u_2) \geq \ldots \geq g(u_n). \]

and let us identify \( U \) with the set \([n]\) for ease of notation, so that \( u_i = i \). Recall that the positions \( [n]\) are sorted by decreasing order of \( f \) so that

\[ f(1) \geq f(2) \geq \ldots \geq f(n). \]

Let \( M > 0 \) be a large enough number and define \( W_{u} = -e^{-M}f(u) \). Observe that, because of the orderings defined by (30) and (31), the matrix \( W \) satisfies the Monge property: if \( i < j \) and \( u < v \), then \( W_{iu} + W_{jv} \geq W_{iv} + W_{ju} \). Indeed,

\[ W_{iu} + W_{jv} - (W_{iv} + W_{ju}) = (e^{-Mf(j)} - e^{-Mf(i)}) (e^{Mg(u)} - e^{Mg(v)}) \geq 0 \]

because \( g(u) \geq g(v) \) and \( f(i) \geq f(j) \), so both factors are non-negative. Thus we may apply the algorithm from [7] to maximize \( \sum_{u \in U} W_{r(u),u} \) over valid rankings \( r \). The resulting algorithm is Algorithm 3. For any fixed \( M \), maximizing \( \sum_{u \in U} W_{r(u),u} \) is the same as maximizing \((1/M) \cdot \text{softmin}\{M \cdot A(S, u) \mid u \in U\} \). But since the solution \( S^* = S^*(M) \) returned by this algorithm does not depend on \( M > 0 \) and \( \lim_{M \to \infty} \text{softmin}\{M \cdot z \} = \min(z) \), it follows that \( S^* \) maximizes \( \min(A(S, u) \mid u \in U) \), as we wished to show.

**A.4 Solving maximin-fairness approximately**

Instead of solving the LPs in the proof of Theorem 4.1 exactly, we can use iterative methods designed to approximately solve zero-sum games and packing/covering programs, as sketched next.

Recall that the exact algorithm 1 works by solving the linear program (9) and updating \( K \) and \( \alpha_0 \). Let us apply a positive affine transformation to normalize all \( A(S, e) \) to the range \([0, 1]\) and select an additive approximation parameter \( \epsilon > 0 \), so we want to ensure that in the final solution, the expected satisfaction for every \( e \in E \) is at most an additive \( \epsilon \) below that which would have been computed by solving LP (9) exactly at the point where \( \alpha_0 \) was assigned.

Rather than maximizing \( \lambda \) directly in (9), we can guess an approximation \( \tilde{\lambda} \) to the optimum, and verify if the guess is correct by eliminating the \( \lambda \) variable from this LP and replacing it with our guess \( \tilde{\lambda} \), and then checking if the program is feasible. Denote by \( M \) the resulting LP. \( M \) is a fractional covering program, equivalent to a zero-sum game, hence the techniques from [32] apply. If \( M \) is feasible, the algorithm from [32] returns a non-negative solution with \( \sum_i \epsilon_i = 1 \) and violating the remaining constraints by at most an additive \( \epsilon \) below that which would have been computed by solving LP (9) exactly at the point where \( \alpha_0 \) was assigned.

Several improvements over this basic scheme can be made. First, the above bound for \( m \) is often too pessimistic, and it is more efficient to do a “doubling trick”: start with \( m = 2 \) and keep doubling \( m \) and restarting again with \( K = 0 \) if during the execution of the algorithm sketched above we end up solving more than \( m \) programs. Second, we can use the variable-step increase technique from [33].

Third, in the case of ranking problems with upper bounds, the separation oracle only depends on the order of the weights and not their specific values, so there is no need to call it again if this order does not change; we can simply increase the probability of that solution. Finally, for a given order of weights, (23) allows us to obtain an optimal dual solution that respects that given weight order, which can be used to detect convergence of the iterative algorithm from [33] earlier.