Rank Aggregation Algorithms for Fair Consensus

Caitlin Kuhlman
Worcester Polytechnic Institute
cakuhlman@wpi.edu

Elke Rundensteiner
Worcester Polytechnic Institute
rundenst@wpi.edu

ABSTRACT

Aggregating multiple rankings in a database is an important task well studied by the database community. High-stakes application domains include hiring, lending, and education where multiple decision makers rank candidates and their input is then combined into an overall consensus ranking. While state-of-art aggregation methods succeed in giving equal say to each decision maker, to date no methods ensure fair representation for groups of candidates being ranked, who risk being impacted by discriminatory bias. We present the first solution to this open problem of guaranteeing fairness for disadvantaged groups of candidates, while still producing a good consensus of the base rankings. We design a family of exact fair algorithms achieving optimality for fair rank aggregation. We also develop approximate methods achieving fairness with guaranteed minimal approximation error scaling to millions of candidates in the rankings. A comparative study evaluates our proposed methods, revealing trade-offs between aggregation accuracy and different degrees of unfair bias in a rich variety of rank aggregation scenarios. Our real-world case study demonstrates that our solutions mitigate unfair bias using real-world data.

PVLDB Reference Format:
Caitlin Kuhlman and Elke Rundensteiner. Rank Aggregation Algorithms for Fair Consensus. *PVLDB*, 13(11): 2706 - 2719, 2020.
DOI: https://doi.org/10.14778/3407790.3407855

1. INTRODUCTION

Ranking is commonly used to prioritize among candidates in a database for desirable outcomes like jobs, loans, or educational opportunities. For such high-impact applications, proportional representation requirements often exist for groups of people that have been historically disadvantaged. This may be enforced by legal standards (e.g., the 80% rule in discrimination law [28]) or by internal policies of an organization aiming to ensure diversity.

Unfortunately, people performing ranking analysis may suffer from implicit bias in their decision making [30], which has had a demonstrated negative impact for critical tasks such as hiring [11, 12, 48]. Complicating matters further, increasingly the judgments of human analysts are augmented by decision support tools or even fully automated screening procedures which rank candidates [10, 19, 29, 49]. Such systems may encode unfair bias [7], perhaps present in the training data [46], reflected by the design of scoring functions [5], or due to differences in the way of members of different groups represent themselves [3]. This new interplay of human bias and machine learning may further impede equitable decision making in unforeseen ways.

To address this important societal problem, recent research focuses on the design of metrics for measuring unfair bias in individual rankings and strategies for mitigating its effect [5, 17, 29, 38, 51, 53]. For instance, LinkedIn recently incorporated a fairness framework into their Talent Search feature that helps recruiters find job candidates [29]. To date, however, these efforts are limited in that they only consider fairness of a *single* ranking in isolation. The critical yet so far overlooked problem of *ranking by consensus* arises when numerous decision makers produce rankings over candidate items, and then those rankings are *aggregated* to create a final *consensus ranking* [14, 35]. While many (non-fairness aware) procedures for aggregating a set of rankings have been put forth by the database community [2, 13, 27, 34], the problem of *fair rank aggregation* remains open. It is largely unexplored whether aggregation might introduce or exacerbate bias disadvantaging particular groups. In this work we thus set out to investigate these open problems.

Hiring Example. Consider the university hiring scenario in Figure 1. After reviewing a pool of faculty applicants (assisted by an automated screening tool [49]), each committee member ranks the candidates based on their individual impressions. Now the committee must come to some overall *consensus ranking* to recommend to their department. As seen in Figure 1, this poses several challenges. First, the procedure to combine the rankings is not obvious. For instance, candidate A is most frequently ranked in the top spot, but also seems to be divisive, since two committee members ranked A last. Candidate B on the other hand is consistently ranked near the top, but never in the number one spot. Another consideration is the committee's desire to have a diverse faculty body. They would like a gender balance among the candidates in the final ranking to compensate for any unintended bias in the input rankings where female candidates seem to be ranked lower than males. Clearly, a principled strategy is required to fairly

Proceedings of the VLDB Endowment, Vol. 13, No. 11
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3407790.3407855

account for this imbalance while still appropriately representing all committee members' preferences.

1.1 State-of-the-Art

Contemporary (group) fairness research [28] is concerned with predictive outcomes for minority or otherwise disadvantaged groups defined by sensitive legally protected data attributes such as race, gender, or age. Unfair practices include explicit use of protected attributes to create biased rankings, as well as discrimination that indirectly or unintentionally exploits group status ("disparate treatment" verses "disparate impact" [28]). The bulk of recent research for ranking targets a *statistical parity* notion of fairness [5, 17, 29, 38, 51, 53]. That is, they aim to ensure that each group receives a proportionate number of preferred rank positions. Alternative definitions of fairness have also been explored [46, 38, 29]. However, all these methods consider fairness only in the context of *one single* ranking. To the best of our knowledge, *group fairness has not yet been explored when aggregating multiple rankings.*

Traditional rank aggregation produces an aggregate ranking (a consensus) by finding the ranking *closest* to the base set of input rankings [35]. This task has seen much interest in the database [14], machine learning [40], and information retrieval [26] fields. Many algorithms have been proposed for finding optimal or approximate consensus [8, 22, 23, 26, 35, 44, 52, 47]. With each ranking representing the preferences of a *voter* for a given candidate set, Social Choice Theory ensures each voter has an equal say [4]. While properties of aggregation algorithms with regard to the voters are well understood [14], fair and equitable treatment of the *candidates being ranked* in the context of rank aggregation remains unaddressed.

1.2 Challenges

Aggregation problem complexity. Classical rank aggregation – even without considering fairness – is a hard problem. Depending on the criteria used to determine the consensus ranking, finding the exact solution may be NP-hard [8, 26]. For a given set of base rankings over n candidates, there are $n!$ possible consensus rankings to chose from, making an exhaustive search over all options intractable. The complexity of the aggregation problem is not only impacted by the number of candidates being ranked, but also by the number of voters (base rankings), and the extent to which the base rankings agree (or disagree) on the placement of individual candidates [2]. This becomes further complicated for fair aggregation, since bias in the individual base rankings could also impact the complexity of the task.

Notions of fair aggregation. Rankings may be produced according to heterogeneous schemes, including human decision makers' preferences or proprietary ranking algorithms. Therefore, fair aggregation must be performed without access to the underlying data and ranking models – rendering causal notions of fairness [50, 46] impossible, as we cannot investigate the relationship between data attributes and outcomes. Associational bias mitigation methods exist for single rankings [5, 17, 29, 51, 53], which typically trade accuracy of the rank order for fairness. However it is unclear how the similarities and differences among a set of rankings relate to these ways of imposing fairness. Therefore how best to incorporate such measures into the rank aggregation problem is an open question.

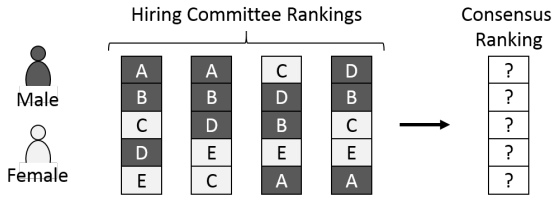

Figure 1: Hiring committee rankings to be aggregated. Four committee members each rank the set of candidates $\{A, B, C, D\}$ from two groups based on gender.

Competing optimization objectives for fair aggregation. If unfair bias against some group is present in base rankings, it is not known how aggregating them into a single ranking will impact this bias. Perhaps aggregation may exaggerate a slight advantage for one group creating a more pronounced bias in the final consensus. This could demand a high toll in aggregation accuracy to ensure fairness. Conversely, a diversity of perspectives among the base rankings might inherently mitigate unfairness present in only a few of the rankings, in which case correction is not necessary. A deep study of these subtle inter-dependencies is needed. Beyond that, a sophisticated strategy is required capable of balancing the competing goals of fairness for the groups of candidates being ranked while concurrently retaining a good representation of the base rankings.

1.3 Proposed Approach

To address these challenges, we formalize fair rank aggregation as a constrained optimization problem balancing the competing objectives above. The solution is defined as the closest consensus ranking to the base set of rankings that satisfies a targeted fairness criterion. We then propose a fair rank aggregation framework which uses pairwise discordance to both compute closeness among consensus and base rankings and measure the advantage given to each group of candidates. This allows group fairness criteria to be seamlessly integrated into Kemeny optimal rank aggregation [35]. We demonstrate the power of our framework to support a *pairwise rank parity* fairness definition [38], proving an equivalence between our approach and popular top-k statistical parity [51] metrics for fair ranking.

Next we leverage the pairwise framing of the problem to tackle the complexity challenge of fair rank aggregation. As a first solution, we propose an integer linear program with parity constraint to produce a Kemeny-optimal fair consensus. Our Fair-ILP approach can aggregate many rankings generated by a large number of *voters*. However, the large number of binary variables is prohibitive when there are many *candidates*. Therefore we extend this by deriving a lower bound on the cost of pairwise fairness criteria. This supports the design of a rank parity-preserving search heuristic integrated into a branch-and-bound fair rank algorithm, which we call Fair-BB. We demonstrate that Fair-BB speeds up computation when ranking many candidates when the fairness requirements are lenient. Finally, we provide a fast approximation post-processing algorithm Fair-Post which guarantees fairness while introducing minimal pairwise error, and scales to millions of candidates.

We thoroughly evaluate these alternate solution strategies in a rich test bed of rank aggregation scenarios. The pre-

viously unknown relationship between fairness and consensus among multiple rankings is explored, using the Mallows' model [41] to generate distributions of rankings and expose the tradeoffs between our competing objectives. Finally, we demonstrate the ability of our framework to produce fair consensus on real-world using a case study of sports rankings. Our methods consistently produce fair aggregations, extending contemporary fairness to ranking by consensus.

Contributions of our work include:

1. We formulate the open problem of fair rank aggregation as finding consensus among a set of input rankings while ensuring the fair treatment of candidates being ranked.
2. We propose a novel pairwise fair rank aggregation framework for parity-preserving Kemeny aggregation.
3. We design a series of algorithms which guarantee to find optimal fair consensus, leveraging integer linear programming and custom branch-and-bound strategies.
4. We also design a fast approximation algorithm which finds a fair solution with minimal aggregation error.
5. We study the interplay between rank consensus and group fairness, evaluating the relative performance of our solutions for a wide spectrum of aggregation scenarios.
6. We investigate unfair group bias for rankings generated by human decision makers using a real-world case study of expert rankings of sports players.

2. PROBLEM FORMULATION

We are given a database X of a fixed set of n *candidates* $x_i \in X$. A *ranking* over X is a permutation $\rho = [x_1 \prec x_2 \prec ... \prec x_n]$ over all candidates. Here $\prec$ is a complete ordering relation on X such that $x_i \prec_\rho x_j$ implies that x_i appears at a more preferred position than x_j in the ranking ρ. The position of a single candidate x_i in the ranking ρ is denoted $\rho(x_i)$, with low number positions favored over higher ones, i.e. $\rho(x_i) = 1$ is the best rank position.

Traditional Rank Aggregation. The set of all possible rankings of X is S_n the symmetric group of permutations. In the traditional rank aggregation problem [35] we are given a subset of base rankings $R \subseteq S_n$ created by some *voters*. To be broadly applicable, we do not make assumptions about how the base rankings were determined, but rather consider R as a fixed input. We are tasked with finding a single *consensus ranking* ρ^* that best represents R as given in Definition 1. The consensus ranking ρ^* is the median ranking in S_n with the minimum average distance to the rankings in R according to some distance function d. A median ranking always exists, however it may not be unique.

DEFINITION 1. *Given a set of base rankings $R \subseteq S_n$ and distance function d* ***the traditional rank aggregation problem*** *is to find a closest ranking $\rho^* \in S_n$ to R such that:*

$$\rho^* = \text{argmin}_{\rho \in S_n} \frac{1}{|R|} \sum_{\sigma \in R} d(\rho, \sigma)$$

Fair Ranking. In our problem setting, each candidate being ranked also has an associated *protected attribute* (e.g., race, gender, or age). This attribute partitions the dataset into two or more disjoint *groups* $G = \{G_1, \ldots, G_m \mid \cup_{i=1}^{m} G_i \in G = X$ Our fairness analysis aims to mitigate unfair bias against these candidate groups.

Unfair bias can be defined in different ways. Since we do not necessarily have access to the underlying data attributes or ranking procedures used by the voters, in this work we aim to meet a group fairness criterion F determined only by the rank order of the candidates and their group membership. In particular, we first focus on achieving the fairness notion *statistical parity* which has been targeted by the majority of recent proposed fair ranking methods [5, 17, 29, 51, 53]. First proposed for classification [25], statistical parity requires that the same proportion of each group receives favorable outcomes. This type of diversity requirement is one approach to achieving distributional justice for historically disadvantaged groups. Definition 2 defines statistical parity for a classification task [20].

DEFINITION 2. ***Statistical Parity for classification:*** *Given a dataset X of candidates belonging to mutually exclusive groups $G_i \in G$ and a binary classifier $f(x) = \hat{y}$ which assigns each item $x \in X$ to a class in $0, 1$, where $\hat{y} = 1$ denotes the preferred outcome for item x, the predictor satisfies statistical parity if the following condition is met for all groups $G_i, G_j \in G, i \neq j$:*

$$P(\hat{y} = 1 \mid x \in G_i) = P(\hat{y} = 1 \mid x \in G_j)$$

The *preferred outcome* in classification is often straightforward, for example getting approved for a loan or being hired for a job. For ranking tasks the preferred outcome is more subtle. Most fair ranking approaches define it as inclusion in a top-k prefix of the ranking [5, 17, 29, 51, 53]. Definition 3 gives a general formulation of this notion of statistical parity for rankings.

DEFINITION 3. ***Top-k Parity:*** *Given a ranking $\rho = [x_1 \prec x_2 \prec ... \prec x_n]$ of candidates belonging to mutually exclusive groups $G_i \in G$, and $0 \leq k \leq n$, the ranking satisfies top-k parity if the following condition is met for all groups $G_i, G_j \in G, i \neq j$:*

$$P(\rho(x) \leq k \mid x \in G_i) = P(\rho(x) \leq k \mid x \in G_j)$$

Since this definition is highly dependent on the choice of k, some methods consider outcomes averaged over fixed intervals of top-k prefixes [5, 17, 51, 53] or combined using a discounting cumulative function over all prefixes [29, 53] to give more emphasis to ensuring fair outcomes toward the top of the ranking. Bias-correction methods then impose parity (group fairness) on an unfair ranking by rearranging the candidates to ensure sufficient representation of the minority group in top-k prefixes of the ranking.

Fair Rank Aggregation Problem Formulation. We now formalize our fair aggregation problem in Definition 4 as the goal of finding the closest consensus ranking to R that satisfies a fairness criterion F. Henceforth in this work we will focus on rank aggregation with statistical parity for binary, disjoint groups of candidates. We discuss extensions of our framework to more general scenarios, including multiple and intersectional group identities, as well as alternative fairness criteria in Section 6.

DEFINITION 4. *Given a set of base rankings $R \subseteq S_n$ and a fairness criterion F,* ***the fair rank aggregation problem*** *is to find a closest ranking $\rho^* \in S_n$ to the base rankings that satisfies F.*

3. PROPOSED FRAMEWORK FOR FAIR RANK AGGREGATION

3.1 Solution Spectrum for Fair Aggregation

We now examine the spectrum of alternate approaches to rank aggregation that concurrently guarantee fairness for the candidates being ranked while assuring maximal representation of the interests expressed in the base rankings. For this, let us suppose we have a fairness correction method $f(\rho) = \rho'$ that can rearrange the items *in a single ranking* to satisfy a fairness criterion F (such as top-k parity) in a way that minimizes the distance to the original ranking $d(\rho, \rho')$. As we show, extreme solutions of applying this correction as a pre- or post-processing step for traditional rank aggregation are not adequate. This necessitates the development of a novel integrated strategy which optimizes for these two competing goals concurrently – the subject of our work.

The **pre-processing** strategy illustrated at the top of Figure 2 first applies this correction method one by one to each of the given rankings in the base set R, to make a new set of fair rankings R'. Thereafter, R' is aggregated using a traditional rank aggregation method. This approach guarantees neither optimal distance to the original base set R, nor fairness of the resulting ranking. One, f may minimize the distance between the fair rankings in R' and their original counterparts in R, however this does not necessarily imply that the median ranking for R' is close to the original R. Two, even if each ranking in R' meets the fairness criterion, the order of individual items may differ greatly across those adjusted rankings in R'. Therefore we do not know whether aggregating R' would also be fair.

On the other hand, the **post-processing** strategy shown in the middle of Figure 2 first aggregates R without considering fairness, producing a consensus ranking ρ^*. Then ρ^* is corrected for fairness to produce a ranking $f(\rho^*) = \rho'$. This time we can be sure ρ' is fair. However, we have no guarantee about the quality of the resulting aggregation. Even if the corrected consensus ranking ρ' is guaranteed to be close to the original ρ^*, it may not be the closest fair solution to the base rankings in R. Therefore neither extreme approach can guarantee both fairness and aggregation accuracy.

3.2 Integrated Pairwise Solution

We propose a conceptual framework for an integrated fair aggregation solution that elegantly balances aggregation accuracy with fairness, achieving both goals concurrently as shown in the bottom of Figure 2. Our key insight here is that we must align the measure of distance between rankings with the measure of candidate group advantage using a *pairwise rank representation*. As we will demonstrate below, this then allows for fairness criteria to be integrated seamlessly into a fair aggregation framework.

Any ranking can be represented as a series of pairwise comparisons between the candidates. Given a base set of rankings R, they will agree on the order of some pairs, and disagree on others. The number of pairs which appear in an inverted order in one ranking compared to the other corresponds to the distance measure known as the **Kendall Tau** [36], defined in Equation 1 for two rankings $\rho, \sigma \in S_n$.

$$K(\rho, \sigma) = \sum_{x_i, x_j \in X} \mathrm{I}\big(\rho(x_i) \prec \rho(x_j) \text{ and } \sigma(x_j) \prec \sigma(x_i)\big) \quad (1)$$

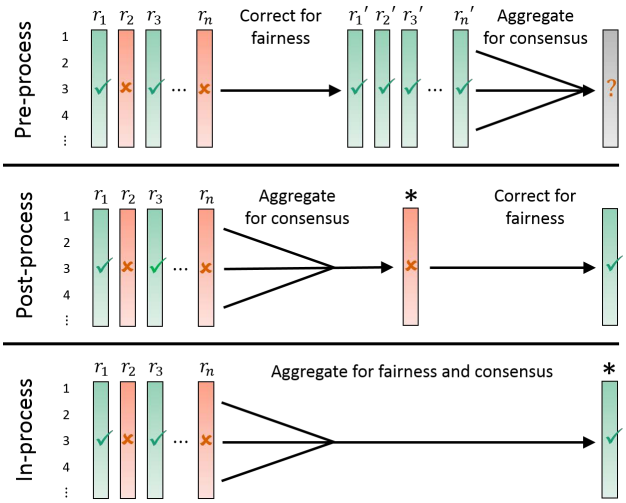

Figure 2: Alternative fair rank aggregation strategies.

Here I is the indicator function, with $I(x) = 1$ when x is true, and $I(x) = 0$ otherwise. The consensus ranking with the minimum average Kendall tau distance to the rankings in R, given in Equation 2, is known as the **Kemeny optimal rank aggregation** [35].

$$\rho^* = \mathrm{argmin}_{\rho \in S_n} \frac{1}{|R|} \sum_{\sigma \in R} K(\rho, \sigma) \quad (2)$$

Although there are different strategies for measuring the distance between rankings, Kemeny aggregation is seen as the gold standard for rank aggregation, which concurrently satisfies multiple axioms from Social Choice Theory [14]. For this reason, it has seen extensive database and machine learning applications [2, 13, 27, 34]. Kemeny aggregation provides our starting place from which to consider group fairness in the ranking utilizing pairs of candidates.

Consider the case where our database X contains candidates from two different groups, G_1 and G_2. The pairs in the Cartesian product X^2 can be divided into three subsets: those pairs containing only candidates from group G_1, those containing only candidates from G_2, and the set of "mixed" pairs containing one item from each group. In recent work, we proposed that the proportion of heterogeneous pairs which favor one group over the other corresponds to a fairness measure of the relative advantage that group enjoys in the ranking [38]. The *Rpar* score in Equation 3 captures this pairwise advantage for G_1, normalized by the total number of mixed pairs in the ranking to account for any imbalance in the size of the groups.

$$Rpar_{G_1}(\rho) = \sum_{x_i \in G_1} \sum_{x_j \in G_2} \frac{\mathrm{I}(x_i \prec_\rho x_j)}{|G_1||G_2|} \quad (3)$$

We now observe that $Rpar_{G_1}$ computes the probability that an item from group G_1 is ranked above an item from group G_2, such that:

$$Rpar_{G_1}(\rho) = P(x_i \prec_\rho x_j \mid x_i \in G_1, x_j \in G_2) \quad (4)$$

Following from this, we propose the following pairwise formulation of statistical parity in Definition 5 for two groups.

DEFINITION 5. ***Pairwise Statistical Parity:*** *Given a ranking $\rho = [x_1 \prec x_2 \prec ... \prec x_n]$ of candidates belonging to mutually exclusive groups G_1, G_2, ρ satisfies pairwise statistical parity if the following condition is met:*

$$Rpar_{G_1}(\rho) = Rpar_{G_2}(\rho)$$

3.3 Equivalence between Top-k and Pairwise Statistical Parity

Defining parity this way gives a compatible fairness criterion for Kemeny aggregation, in that both optimization goals now depend on the pairwise ordering of candidates in the consensus ranking. However, it is not immediately obvious how the pairwise Definition 5 relates to the notions of statistical parity expressed in Definitions 2 and 3. We now show that this proposed formulation indeed corresponds to a variant of top-k parity semantics, namely, a summary formulation that takes all possible prefixes of the list into account. For this, we observe that the probability on the left hand side of the Top-k Parity in Definition 3 corresponds to the cumulative distribution function for the rank $\rho(x)$ of any $x \in G_1$. Let us denote this as $F_{\rho(x)|G_1}(k)$ such that:

$$F_{\rho(x)|G_1}(k) = P(\rho(x) \leq k \mid x \in G_i) \tag{5}$$

To compute the value of $F_{\rho(x)|G_1}$ for a given k, we can simply count the proportion of candidates from group G_1 in the top-k prefix of the ranking as per below.

$$F_{\rho(x)|G_1}(k) = \sum_{x \in G_1} \frac{I(\rho(x) \leq k)}{|G_1|} \tag{6}$$

Since the rank of x is strictly positive, we can use $F_{\rho(x)|G_1}(k)$ to compute the conditional expectation of the rank of x given membership in G_1, where

$$E(P(\rho(x) = k \mid x \in G_1) = \sum_{k=0}^{\infty} \Big(1 - \sum_{x \in G_1} \frac{I(\rho(x) \leq k)}{|G_1|}\Big) \tag{7}$$

Each possible value of k corresponds to a rank position in ρ assigned to some candidate x. Let K_1 and K_2 be the set of all rank positions assigned to candidates in G_1 and in G_2, respectively. We can then re-write Equation 7 as:

$$E(P(\rho(x) = k \mid x \in G_1) = \sum_{k=0}^{n} 1 - \Big(\sum_{x \in G_1} \sum_{k_2 \in K_2} \frac{I(\rho(x) \leq k)}{|G_1|} + \sum_{x \in G_1} \sum_{k_1 \in K_1} \frac{I(\rho(x) \leq h)}{|G_1|}\Big) \tag{8}$$

The first outer term in Equation 8 is simply $n+1$, since for any $k > n$ the probability that $\rho(x) \leq k$ is always 1. The inner summations describe pairwise relationships between the candidates in ρ. The first term is only over those candidates $x \in G_1$ ranked higher than candidates in G_2 (each position $k_2 \in K_2$ corresponding to a candidate in G_2). These are the same pairs used to compute $Rpar_{G_1}$. The second term is only over candidates in G_1. For each consecutive position k_1, this term counts the candidates ranked at that position or higher. This thus corresponds simply to the sum of consecutive integers from 1 to $|G_1|$, given by the constant $\frac{|G_1|(|G_1|+1)}{2}$. Therefore we have:

$$E(P(\rho(x) = k \mid x \in G_1) = n+1-\Big(|G_2|\ Rpar_{G_1} + \frac{|G_1|+1}{2}\Big) \tag{9}$$

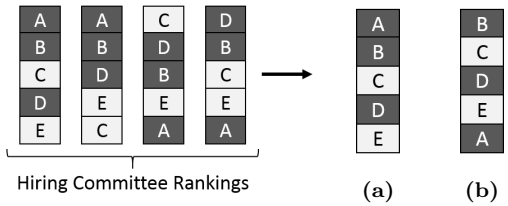

Figure 3: Aggregated hiring committee rankings with and without fairness criteria, namely, Kemeny optimal ranking (a) without considering fairness, and (b) with rank parity.

From Equation 9, we can observe two things. One, the top-k probability taken over all k is a linear function of the *Rpar* measure, where:

$$\sum_{k=0}^{n} P(\rho(x) \leq k \mid x \in G_1) = |G_2|\ Rpar_{G_1} + \frac{|G_1|+1}{2} \tag{10}$$

Two, *Rpar* can be used to measure the difference in the expected rank position of different groups. This is not surprising, as we further observe that the *Rpar* score for a disadvantaged group is equivalent to the Mann–Whitney U-statistic [43], where $Rpar = \frac{U}{|G_2||G_1|}$. The Mann-Whitney U has a well-known relationship to the area under the ROC curve for a probabilistic classifier [32]. Similarly, we put forth here that the *Rpar* score relates to the rates of positive outcomes achieved by each group according to different values of k.

3.4 Rank Parity for Fair Hiring Example.

We return to our example from Section 1 to study how pairwise statistical parity might impact rank aggregation. Figure 3 illustrates the result of aggregation with and without rank parity. Rank 3a is produced using unconstrained Kemeny ranking. We see it reflects the tendency of female candidates to be ranked lower than males, and it places the controversial candidate A at the top. Five mixed pairs favor males over females $\{(a \prec c), (a \prec e), (b \prec c), (b \prec e), (c \prec d)\}$ and only one pair favors females over males $\{(c \prec d)\}$. In contrast, ranking 3b overcomes this group disparity. In this ranking, females enjoy equal pairwise advantage to males, where three pairs favor males $\{(b \prec c), (b \prec e), (d \prec e)\}$ and three favor females $\{(c \prec d), (c \prec a), (e \prec a)\}$.

Any consensus ranking represents a compromise. In a real hiring committee, members might debate or argue for a ranking close to their own point of view. Our proposed framework can be instrumental in resolving differences while adding the benefit of fairness for candidates. In this example, both rankings 3a and 3b are in fact *both optimal*, each having a minimal average Kendall Tau distance of 3.5 to the base rankings. In this case, prioritizing fairness can resolve a tie when more than one optimal consensus ranking exists. Other times, the committee may have flexibility in their choices and may *tune the degree of fairness required* to achieve a solution that is more fair than the base rankings while only introducing minimal additional pairwise error.

4. PROPOSED METHODS FOR KEMENY-OPTIMAL FAIR AGGREGATION

Given our integrative pairwise framework, we now design optimal strategies to solve the proposed constrained optimization problem for fairness-preserving rank aggregation.

To facilitate the design of our methods, we employ a compact representation of the rankings in R in the form of the *precedence matrix* C (Def. 6). Matrix C summarizes the pairwise relationships between candidates in X.

DEFINITION 6. *Given a set of rankings to be aggregated R over a dataset $X = \{x_1, \ldots, x_n\}$, then the* **precedence matrix** *C is defined as:*

$$C_{ij} = \frac{1}{|R|} \sum_{\sigma \in R} I(x_i \prec_\sigma x_j)$$

Each entry in C indicates the proportion of rankings in R which favor x_i over x_j, compared to the number of times x_j is preferred to x_i. Summing a column j of C captures the overall pairwise advantage given to the item x_j in R.

Fairness Threshold. Recognizing that different parity requirements may arise in various ranking scenarios, we now relax our rank parity definition by introducing a fairness threshold parameter δ_{par} as the maximum allowable difference in the pairwise advantage in ρ^* for each group. This supports us in tuning the fairness criterion according to application domain requirements, trading-off between the accuracy of the consensus ranking and the strictness of parity between groups. The threshold is expressed as a proportion of mixed pairs in ρ^*, such that:

$$|Rpar_{G_1}(\rho^*) - Rpar_{G_2}(\rho^*)| \leq \delta_{par} \tag{11}$$

4.1 Fair-ILP

We now draw on key strategies for tackling the complexity challenge of the aggregation problem, beginning with an Integer Linear Programming (ILP) approach. Exact Kemeny aggregation has been shown to be NP-hard [8, 26], but can be expressed as a variation of the minimum weighted feedback arc set problem [23, 47], and solved using an integer linear program approach [23]. We propose a Fair-ILP by modeling pairwise statistical parity as a constraint in Integer Program 1.

Matrix A specifies the order of all pairs in the desired consensus ranking ρ^*. The first constraint (Equation 13) enforces that A is a boolean matrix representing the final ranking ρ^*, wherein each pair appears exactly once. The second constraint (Equation 14) says that for all pairs, either $x_i \prec x_j$ or $x_j \prec x_i$, preventing loops in a strict ordering over all candidates. The third constraint (Equation 15) enforces transitivity. Conitzer et al. [23] show that together these three constraints are sufficient to produce the Kemeny-optimal aggregation which minimizes the Kendall Tau distance to R.

Equation 16 enforces pairwise statistical parity. This constraint sums the differences between entries in A corresponding to mixed pairs in ρ^*, where $A_{i,j}$ represents the pair $(x_i \prec x_j)$ favoring group G_1 over G_2, and $A_{j,i}$ represents its inverse $(x_j \prec x_i)$ favoring group G_2. Summing over these differences computes exactly the difference between the *Rpar* scores as in Equation 11.

4.1.1 Complexity Analysis for Fair-ILP

While Kemeny aggregation is NP-hard, it is fixed-parameter tractable, depending only on the number of items being ranked n. This intuitively can be understood by considering the size of the n-by-n precedence matrix C. The number of rankings $|R|$ contributes only to a one-time setup cost of $O(|R| * n^2)$ to construct C. The ILP solution for unconstrained Kemeny aggregation requires n^2 binary variables in the matrix A, $\binom{n}{2}$ constraints to enforce strict ordering (Equation 14), and $\binom{n}{3}$ constraints for transitivity (Equation 15). Our fairness requirement in Equation 16 adds additional constraints for each mixed pair ($|G_1||G_2|$ constraints). This, as our experimental study in Section 5 confirms, increases the time to solve the program.

Linear Program 1: Fair-ILP.

$$\text{minimize} \sum_{i,j} C_{ij} A_{ij} \tag{12}$$

$$\text{s.t.} \quad A_{ij} \in 1, 0 \tag{13}$$

$$A_{ij} + A_{ji} = 1 \qquad \textit{strict ordering constraint} \tag{14}$$

$$A_{ij} + A_{jk} + A_{ki} \geq 1 \qquad \textit{transitivity constraint} \tag{15}$$

$$\Big| \sum_{\substack{x_i \in G_1 \\ x_j \in G_2}} (A_{ij} - A_{ji}) \Big| \leq \delta_p \qquad \textit{parity constraint} \tag{16}$$

For rankings over many items, the large number of binary variables in the ILP solution poses practical computational challenges. In benchmarking studies [23, 13], ILP algorithms for unconstrained Kemeny aggregation could not handle more than $n = 60$ items. Similarly, in our empirical evaluation in Section 5 with state-of-the-art highly optimized mathematical GUROBI solver [31] and 500G of dedicated memory, we handle problems on the order of $n = 100$. Further, with commercial solvers for IP being proprietary and not freely available outside of academia, we explore additional solutions next.[1]

4.2 Fair-BB

We now design a Branch-and-Bound based (B+B) approach aiming to handle a larger number of candidates n. Intuitively, any fairness constraints imposed on the rank aggregation task shrink the space of possible outcomes. To capitalize on this problem characteristic, one simple approach would be to incorporate an explicit check into the branching rule for a B+B solution for traditional Kemeny aggregation. This would prune search paths which violate the targeted fairness criterion F. However, we observe that this may require the method to backtrack over many paths in the tree, resulting in an expensive search. Alternatively, to empower more efficient search, we now derive a lower bound on the distance from the base rankings in R to any fair consensus ranking. This can be thought of as the *cost* of each potential solution in terms of pair inversions between rankings. We use this lower bound to design *admissible* fairness-preserving heuristics. which can guide the B+B tree search, and *are guaranteed to underestimate the true cost of the final ranking.* This ensures that whenever a leaf node is reached, an optimal solution has been found [45].

4.2.1 Bounding the Cost of Fairness

The total cost of a potential consensus ranking ρ corresponds to the sum of the costs for all ordered pairs of candidates $(x_i \prec x_j)$ in the ranking as given by the Kendall Tau distance (Equation 1). This cost for each pair is computed

[1] All code, experiments and supplemental material available: `https://arcgit.wpi.edu/cakuhlman/VLDB2020`

by counting the proportion of rankings in R that disagree with its ordering using the precedence matrix C, such that:

$$cost(x_i \prec x_j) = C_{j,i}. \quad (17)$$

Each pair can only be ordered one of two ways, either $(x_i \prec x_j)$ or $(x_j \prec x_i)$. This order will agree with a certain number of rankings in R. Thus, for all $x_i, x_j \in X$, we have:

$$C_{i,j} = 1 - C_{j,i}. \quad (18)$$

Summing the minimum costs associated with each pair in the ranking ($C_{i,j}$ or $C_{j,i}$) provides a lower bound $lb(\rho^k)$ on the true cost of the Kemeny optimal consensus ranking [44], as given in Equation 19. We note here that this order may contain cycles, and therefore may not correspond to the actual cost of a feasible solution.

$$lb(\rho^k) = \sum_{x_i, x_j \in X} min(C_{j,i}, C_{i,j}) \quad (19)$$

Building on this, we now propose Lemma 1 following directly from Definition 4 of the fair rank aggregation problem.

LEMMA 1. *Let a Kemeny optimal consensus ranking over X be any ranking ρ^k with minimum cost distance to R, and a fair consensus ranking over X be any ranking ρ^* with minimum cost out of all rankings that satisfy a given fairness criterion F. Then $cost(\rho^*) \geq cost(\rho^k)$.*

Proof. If a ranking ρ^k exists that satisfies F, then $\rho^k = \rho^*$. Otherwise, ρ^* must have higher cost by definition. □

In a higher cost solution, some number $k \geq 0$ pairs must not conform to the minimum cost ordering in the final ranking ρ^*. Assuming we knew this number k, we could then compute a lower bound $lb'(\rho^*)$ based on Equation 19 by flipping the number of the k pairs that add the *minimum amount of cost overhead* to $lb(\rho^k)$. Lemma 2 identifies the *overhead cost* of flipping a pair.

LEMMA 2. *Given a pair $(x_i \prec x_j)$ with a minimum cost of $C_{j,i}$ contributing to the sum in $lb(\rho^*)$, if the pair is flipped, the **updated bound** $lb'(\rho^*)$ will incur an **overhead cost** of $1 - (2 * C_{j,i})$.*

Proof. By Equation 18, flipping a pair is equivalent to subtracting the minimum cost $C_{j,i}$ for the pair and adding its inverse $(1 - C_{j,i})$. Therefore, when we flip a pair from the order $(x_i \prec x_j)$ to order $(x_j \prec x_i)$, we get:

$$\begin{aligned} lb'(\rho^k) &= lb(\rho^k) - C_{j,i} + (1 - C_{j,i}) \\ &= lb(\rho^k) + 1 - 2 * C_{j,i} \end{aligned}$$ □

COROLLARY 1. *$lb'(\rho^*)$ can be determined from the ordering of pairs used to compute $lb(\rho^k)$ by flipping the k pairs that carry the minimum overhead costs, and that will satisfy the criterion F.*

Proof. Corollary 1 follows directly from Lemmas 1 and 2, given that $C_{i,j} \geq C_{k,l} \implies 1 - 2 * C_{i,j} \leq 1 - 2 * C_{k,l}$ □

4.2.2 Guiding Search for Fair Consensus Ranking

Building on the observations above, we now incrementally construct a search tree such that each node v represents a candidate x being placed in a particular rank position in ρ^*. Each node is expanded by adding children nodes for all items not yet included in the path to v. The full search tree has depth n, with every path through the tree representing one possible ranking, and $n!$ leaf nodes. To bound the search and avoid traversing the entire tree, as each node is expanded, we compute a two-part cost function $f(v) = g(v) + h(v)$.

We can think of $g(v)$ as the cost of the *pairs-so-far* set by the order of the candidates in the prefix path from the root to the current node v. Heuristic $h(v)$ models an estimate of the cost for the *pairs-to-go* which are yet to be determined. Given a node v, the *pairs-so-far* set by the path to v can be divided into three subsets: pairs of candidates belonging to the same group, pairs favoring group G_1 over G_2 (denoted $v.p1$), and pairs favoring group G_2 over G_1 (denoted $v.p2$). These pairs all contribute to the cost of the *pairs-so-far* $g(v)$.

Rank Parity Heuristic. To determine $h(v)$ for rank parity, we initially assign all *pairs-to-go* their minimum cost ordering as required to compute $lb(\rho^k)$ (Equation 19). Once all the pairs are ordered, they can similarly be divided into subsets. We denote *pairs-to-go* favoring G_1 over G_2 as $v.p2g1$ and pairs favoring G_2 over G_1 as $v.p2g2$. We can now express the rank parity criterion in Equation 11 for a node v as the following requirement:

$$abs(\ |v.p1 \cup v.p2g1| - |v.p2 \cup v.p2g2|\ ) \leq \delta_{par} \quad (20)$$

where $|\cdot|$ denotes the number of pairs in each set. If this condition is not met, we must update our initial ordering of *pairs-to-go* to be sure it only allows for rankings which satisfy rank parity. This is accomplished by flipping the order of some of the *pairs-to-go*, transferring them from the set favoring one group to the set favoring the other and balancing the pairwise advantage of the groups. We determine the required number of pairs to flip to satisfy rank parity k, according to Lemma 3.

Algorithm 1: getParityOverhead

input : node v
output: overhead cost for v

```
if |v.p1 + v.p2g1| > |v.p0 + v.p2g0| then
    k = ⌈(|v.p1+v.p2g1|−δ)/2⌉;
    pairsList ⟵ v.p2g1;
else if |v.p1 + v.p2g1| < |v.p0 + v.p2g0| then
    k = ⌈(|v.p0+v.p2g0|−δ)/2⌉;
    pairsList ⟵ v.p2g0;
else
    k = 0
if k == 0 then
    return 0
else if k ≤ length(pairsList) then
    sort(pairsList) ; // sort in ascending order
                      // of overhead cost
    overhead ⟵ 0;
    for i to k do
        overhead += 1 − 2 * cost(pairsList[i]);
else
    overhead ⟵ ∞ ;    // constraint cannot be
     met
return overhead
```

LEMMA 3. *Given a node v, let $P_{max} = max(|v.p1 \cup v.p2g1|, |v.p2 \cup v.p2g2|)$ and $P_{min} = min(|v.p1 \cup v.p2g1|, |v.p2 \cup v.p2g2|)$. **The number of pairs to flip** to satisfy Equation 20 is:*

$$k = \left\lceil \frac{P_{max} - P_{min} - \delta_{par}}{2} \right\rceil$$

Proof. When we flip a pair, we subtract 1 from P_{max} and add it to P_{min}. Therefore we simply can derive the number of pairs to flip as:

$$(P_{max} - k) - (P_{min} + k) = \delta_{par}$$
$$P_{max} - P_{min} - 2k = \delta_{par}$$
$$\left\lceil \frac{P_{max} - P_{min} - \delta_{par}}{2} \right\rceil = k \quad \square$$

Given the required number of pairs to flip k, we can then check whether there are a sufficient number of *pairs-to-go* that could be flipped. If not, this path cannot yield a feasible solution and can thus be pruned. If there are more than enough *pairs-to-go*, then following Corollary 1, flipping only those k pairs that add the minimum overhead will give us our adjusted lower bound on the cost of *pairs-to-go* $lb'(\rho^*)$. Algorithm 1 gives the procedure for computing the total minimum overhead cost for k flipped *pairs-to-go*.

4.2.3 Complexity Analysis for B&B Solution

Complexity of Search Tree Exploration. In the worst case, every path in the tree will have to be explored yielding an $O(n!)$ search cost. Even for modest values of n, such a search will likely be intractable. However, in [44], Meila et al. observe that performance is greatly impacted by the amount of agreement among the base rankings in R. If there is little agreement, all rankings in S_n will be far from the set R and many paths in the tree will have to be compared. In contrast, if there is strong agreement in R, then only a small number of rankings in S_n will be likely candidates for ρ^*. Given a good search heuristic, only the small number of likely rankings will need to be explored.

The cost of the B+B algorithm also depends on the search strategy used. We implement A* search, using a priority queue which has constant time cost for adding and removing nodes if a Fibonacci heap isused. Overall performance of the search depends heavily on the heuristic used. In Section 5 we empirically evaluate our rank parity heuristic given varying degrees of agreement among the rankings.

Complexity to Compute Parity Heuristic. As is typical in B+B design, there is a complexity tradeoff between computing a tighter lower bound heuristic and its impact on the resulting search [21]. Each time a node v is expanded, we must compute the cost of the *pairs-so-far* $g(v)$ and cost of the *pairs-to-go* $h(v)$ for every child node. Following the strategy in [44], we use the siblings of each node to compute $g(v)$, and to determine the sets of *pairs-to-go*. This requires constant time complexity and $O(n)$ space complexity to store the child nodes. To compute the parity heuristic $h(v)$, Algorithm 1 sorts the *pairs-to-go* in P_{max}. Sorting all $n(n-1)/2$ candidate pairs has complexity $O(n^2(log(n))$. Therefore, traversing a single path from the root to a leaf, we visit n nodes and expand $O(n)$ child nodes. This results in $O(n^4 log(n))$ time complexity.

4.3 Fair-Post

To handle the case when the B+B method exceeds allowable resources, one could revert to an approximate tree search using standard techniques such as best first search or beam search. Unfortunately, this simple approach guarantees neither optimal distance to the base rankings nor fairness. As an alternative, we now return to the post-processing strategy discussed in Section 3 to design a approximation strategy which not only guarantees pairwise statistical parity, but also assures that a minimum number of pair inversions are introduced. Given this, existing fast approximation methods for Kemeny aggregation can be plugged in to first aggregate R and generate an initial consensus ranking, and then correct for fairness while bounding the additional approximation error added.

Algorithm 2: correctParity

input : $\rho = [x_1 \prec x_2 \prec \cdots \prec x_n]$, maxPairs
output: ρ' corrected ranking, $q = K(\rho, \rho')$

```
1  l1, l2 ⟵ empty queues;
2  for i ← 1 to n do
3      if x_i ∈ G1 then
4          insert x_i into l1;
5      else
6          insert x_i into l2;
7  ρ' ⟵ [ ] ;
8  p1, p2, q ⟵ 0;
9  for i ← 1 to n do
10     if ρ(l1.peek()) > ρ(l2.peek()) then
11         if p1 + l2.length() ≤ maxPairs then
12             ρ'[i] = l1.dequeue();
13             p1 = p1 + l2.length();
14         else
15             flip = ρ(l2.peek()) − i);
16             ρ'[i] = l2.dequeue();
17             p2 = p2 + l1.length() − flip;
18             q = q + flip;
19     else
20         ρ'[i] = l2.dequeue();
21         p2 = p2 + l1.length();
22 return ρ', q
```

Our Fair-Post is given in Algorithm 2. The input is an *unfair* ranking $\rho = [x_1 \prec \cdots \prec x_n]$ which does not satisfy pairwise statistical parity. Let us say without loss of generality that more pairs favoring group G_1 is greater than the number of pairs G_2. A fair version of this ranking should only allow a certain number of pairs to favor G_1 – denoted $maxPairs$. To correct the ranking, the algorithm proceeds by forming two queues of candidates l_1, l_2 for each group, respectively, keeping the candidates in their original rank order. Then, iterating through rank position $i = 1$ to n, at each step the candidate ranked highest in ρ in either queue is placed at the current position i in ρ', provided that the fairness constraint will not be violated. To check this, we keep a tally of the *pairs-so-far* favoring each group denoted p_1 and p_2, respectively. If the selection of a candidate from G_1 would cause the number of pairs favoring G_1 to exceed $maxPairs$ then parity would be violated (line 11). In this case the highest ranked item from queue l_2 is chosen instead.

We observe that Algorithm 2 *preserves within-group order* of candidates, and also guarantees a fair solution *with the*

minimal number of pair inversions compared to the initial input ranking. Due to space reasons, we sketch the intuition behind our proof below (see technical report for details [2]).

Sketch of proof. Let us consider the simple case where strict parity is required ($\delta_{par} = 0$) and the total number of mixed pairs m is even. To be fair, each group must be favored in exactly half of the mixed pairs, i.e. $maxPairs = m/2$. In the process of correction, some number q pairs will be flipped from their original order. We denote the pairs favoring group G_1, G_2 in original ranking as p_1^ρ, p_2^ρ. Let h be the minimum number of pairs that must be flipped from favoring group G_1 in ρ to favoring G_2 in ρ' in order for ρ' to satisfy rank parity, such that $p_1^\rho - h = \frac{m}{2} = p_2^\rho + h$. If $q^i \leq h$ for all rank positions i, then the Kendall Tau distance $K(\rho, \rho')$ is optimal. In our full proof we show that $q^i \leq h \iff p_2^i + q^i \leq \frac{m}{2}$, and prove by induction that for all rank positions i: $p_1^i \leq \frac{m}{2}$ and $p_2^i + q^i \leq \frac{m}{2}$.

4.3.1 Complexity Analysis for Fair-Post

Algorithm 2 has O(n) time complexity, requiring two passes over the input ranking first to populate the queues and then to assign the candidates to their corrected rank positions. It has O(n) space complexity to hold the candidates in the queues. Given we first need to aggregate the rankings which may contain a large number of items, we propose to plug in an existing approximate aggregation method as an initial step. This way we address the complexity concern of the exact aggregation solutions for NP-hard Kemeny aggregation, i.e. (non-fair) ILP [23] and B+B [44] solutions - which have similar limitations on the number of candidates to be ranked as our proposed integrated methods.

5. EVALUATION

5.1 Experimental Methodology

Overall Strategy. We conduct a systematic study analyzing the interplay between critical factors including the number of candidates ranked, number of rankings in R, consensus among rankings in R, and group fairness threshold on the fair rank aggregation algorithms. This is followed by a case study using real-world sports ranking data.

Metrics. To measure the accuracy of the aggregation, we use the average Kendall Tau distance to the rankings in R (Equation 2), denoted $K_{mean}(\rho^*)$. To evaluate the fairness (*pairwise statistical parity*) achieved by the consensus ranking ρ^*, we measure the absolute difference in the $Rpar_{G_i}$ scores (Definition 5) for each group, denoted as $Rpar(\rho^*)$ in Equation 21. Since the $Rpar_{G_i}$ scores for each group are normalized by the total number of mixed pairs, the *Rpar* score is equal to 1 when the ranking is totally biased favoring one group, and 0 when each group is favored in half of the pairs for perfect parity.

$$Rpar(\rho^*) = abs(Rpar_{G_1}(\rho^*) - Req_{G_2}(\rho^*)) \quad (21)$$

Methods. As a baseline, we compare against two existing strategies for (nonfair) exact Kemeny aggregation: an integer linear program (ILP) by Conitzer et al. [23], and a B+B algorithm by Mandhani and Meila [42] which uses the lower bound in Equation 19. We compare these to our Fair-ILP method with the rank parity constraint (Section 4.1) and Fair-BB method with the parity-preserving heuristic (Section 4.2). For approximate aggregation, we employ the classic Borda [24] scoring method as a first step, and correct for fairness using our Fair-Post algorithm (Section 4.3). In a comparative study of algorithms for Kemeny aggregation [2], Borda was suggested as the best approximation approach when optimizing for speed, and to give low approximation error. Borda sorts the candidates by the overall pairwise advantage in R, which can be done efficiently by summing the columns of the precedence matrix.

Experimental Setup. All experiments were performed on a Linux server running Ubuntu 14 with 500G of RAM. Integer programming solutions were implemented using the commercial highly optimized and parallelized mathematical solver GUROBI [31]. Borda and Fair-Post methods we implemented in Python. B+B algorithms were implemented in Java, adapted from implementation by authors of [42]. For Java methods, we fix the heap size of JVM at 50G. Direct comparisons of run-times should consider these differences.

5.2 Controlled Study of Fair Kemeny Aggregation using Mallows' Model

Dataset Generation. We adopt the Mallows' Model probability distribution over rankings [41] which provides a natural means to evaluate Kemeny rank aggregation methods extensively in previous studies [2, 13]. The Kemeny optimal consensus ranking has been shown to be a maximum likelihood estimator for this model [52]. For all rankings $\pi \in S_n$, the Mallows model is the probability distribution:

$$P_{\pi_0,\theta} \frac{exp(-\theta K(\pi, \pi_0))}{Z}$$
$$Z = \prod_{i=1}^{n-1} \frac{1 - exp((-n - i + 1)\ \theta_i)}{1 - exp(-\theta_i)} \quad (22)$$

The distribution is parameterized by θ which controls the degree of consensus among the rankings in R. If $\theta = 0$, Equation 22 yields a uniform distribution, i.e., there is no consensus among the rankings. As θ increases, the distribution becomes steeper around a single mode ranking σ_0.

To understand the fairness of ρ^* compared the fairness of the base set of rankings in R, we introduce a second parameter p. We control parity in the base rankings by assigning the candidates in the central ranking σ_0 to two groups G_1 and G_2, starting from the highest ranked item and progressing sequentially to the lowest ranked. For each candidate, the group is chosen with probability p. For $p = 0.5$, the groups are assigned in a uniform random manner. Given enough items, this central ranking will be fair, i.e., $Rpar(\sigma_0)$ will be close to 0. As p increases, candidates from G_1 appear in more favorable positions in the ranking. When $p = 1$, all candidates in G_1 are ranked above those in G_0, resulting in a completely biased ranking with $Rpar(\sigma_0) = 1$.

5.2.1 Descriptive Study of Consensus and Fairness in Mallow's Data.

Figure 4 shows twelve different sets of base rankings generated using the Mallows model with $n = 50$ candidates and $|R| = 20$ rankings. We create different aggregation scenarios by varying the parameters θ and p. A central ranking σ_0 shown at the top of each column is used to generate four versions of R. When θ is close to 0 at the top of the figure,

[2] https://arcgit.wpi.edu/cakuhlman/VLDB2020

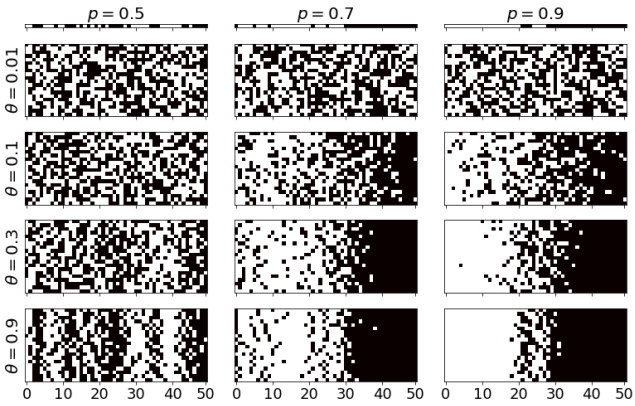

Figure 4: Impact of parameters θ controlling consensus, and p controlling fairness on sets of Mallows generated base rankings with $n = 50$ items and $|R|$=20.

there is little consensus among the rankings. As θ increases, the rankings in R tend to agree more and more with σ_0.

In each column of Figure 4 the candidates are assigned to groups with different degrees of unfairness p. On the left, items are assigned with probability $p = 0.5$, and both groups are randomly distributed throughout all sets of rankings, even when there is strong consensus. Moving right, group G_1 is favored over group G_0 and as θ increases a distinct advantage is introduced for group G_1 in R corresponding to the unfairness controlled by p (Figure 4, bottom right).

This demonstrates that consensus among rankings in R has a large impact on the overall fairness. When there is little consensus (top row of Figure 4), any bias in an individual ranking is "cancelled out" by the diversity in base rankings. In this case, we expect to see little penalty in aggregation accuracy when enforcing fairness in ρ^*. Any ranking chosen will have a large distance to R given rankings are so dissimilar from each other. Conversely, when θ is large (bottom row Figure 4), the distance to the consensus ranking will be small, and accuracy tradeoff for fairness will be higher.

5.2.2 *Experimental Study using Mallows' Data*

Next we demonstrate that our fair aggregation methods learn accurate consensus rankings, while enforcing fairness - even when base rankings are unfairly biased.

Accuracy versus Fairness Tradeoff. First, we verify observations in our descriptive study using unconstrained ILP and parity-preserving Fair-ILP in the same setting. To reveal the impact of strict fairness criteria on aggregation accuracy, we set a tight fairness threshold δ_{par} allowing an advantage of at most 0.02% of the mixed pairs for either group. On the left, Figure 5 shows the impact of enforcing parity in ρ^* on the average Kendall Tau distance to R for base rankings with different degrees of agreement and different amount of bias. When θ is close to zero, R is very noisy, and therefore the overall distance to (ρ^*) is high for both unconstrained and fair aggregation. As θ increases, there is more agreement among rankings in R, and the distance to ρ^k found by ILP gets smaller. Fair-ILP carries a higher distance penalty for requiring parity in ρ^*, which becomes more pronounced with stronger unfair bias in R ($p = 0.9$).

Relationship between Rank Parity and Consensus. On the right, Figure 5 shows $Rpar(\rho^*)$ scores (Equation 21) on the y-axis. The pairwise statistical parity of the ρ^k ILP solution reflects the unfair advantage in base rankings introduced by parameter p. Conversely, Fair-ILP succeeds in enforcing the parity threshold, yielding a flat score across all values of θ and p.

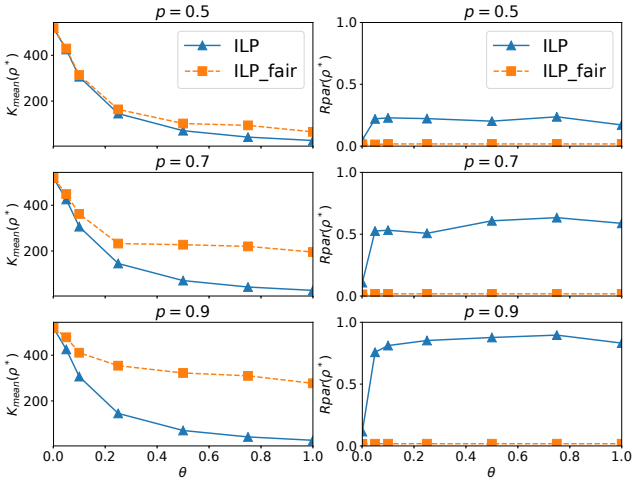

Figure 5: Impact of agreement in R on distance to ρ^* (left) and on the rank parity of ρ^* (right) on sets of Mallows generated base rankings with $n = 50$ candidates and $|R|$=20 for unconstrained and fairness-preserving ILP methods.

5.2.3 *Performance Evaluation*

Next we compare the performance of our proposed exact methods Fair-ILP and Fair-BB for fair rank aggregation. For this, we generate datasets using the Mallows model varying the number of candidates from $n = 40$ to 100. We did not observe significant impact on run-times due to the number of rankings being aggregated. Therefore, we fix $|R| = 1000$.

Impact of number of candidates on performance. From our cost analysis in Section 4.1.1, we know that the number of candidates is the most important determinant of time complexity of the Fair-ILP methods. Indeed, the Fair-ILP method proves to be robust across parameter settings for θ and different fairness thresholds, achieving consistent run-times across all settings. The parity constraint introduced in Section 4.1 adds some overhead impacted by the amount of bias in the base rankings. However, as Figure 6 shows, the ILP run-times increase exponentially in the number of candidates n. This confirms similar analysis in previous studies (see 2015 VLDB survey [13] for $n > 60$).

The B+B approaches tell a different story. In Figure 6, we fixed $\theta = 0.3$ and generated biased data with $p = 0.7$. A loose fairness threshold allowed for an advantage of 25% of the mixed pairs. In this case, Fair-BB gives a > 15x speedup over Fair-ILP when $n = 100$, scaling linearly in the number of candidates. However, Fair-ILP is much more sensitive to the amount of bias in R and agreement among the rankings. This is expected, since the worst case complexity of the method is $O(n!)$ as discussed in Section 4.2.3.

Impact of Unfair Bias on Fair-BB. For their unconstrained B+B method for Kemeny aggregation, [44] claims that the worst case is avoided when the base rankings strongly agree. We observe that for fair aggregation, too much agreement can hinder performance. When strict fairness is

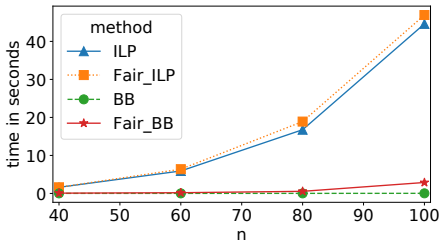

Figure 6: Comparison runtimes for unconstrained and fairness-preserving B+B and ILP methods on sets of $|R| = 1000$ Mallows generated base rankings with $theta = 0.3$, $p = 0.7$ and fairness threshold of 25% mixed pairs.

required and biased base rankings agree strongly, Fair-BB performance suffers greatly. In our experiments, a number of parameter settings (provided in our supplemental report) required more memory than available with heap size of 50GB. For large θ, the base rankings may be tightly clustered far from all potential fair solutions, and thus none can be pruned even using the parity heuristic.

5.2.4 *Approximation methods*

For databases of many candidates, if we accept an approximate solution, our Fair-Post method scales to handle thousands to millions of candidates. Table 1 gives the runtimes when aggregating rankings of $n = 100$ to $n = 1,000,000$ candidates. We fix $|R| = 1000$, $p = 0.7$, and fairness threshold of 1% of the mixed pairs as these settings were not observed to impact runtime in our experimental analysis. We see that the Fair-Post used with the efficient Borda approximation method scales linearly to easily handle large datasets. For rankings with $n > 100,000$ candidates we report only Fair-Post times for the central rankings σ_0 due to the overhead of generating Mallows data at that scale.

5.3 Fantasy Sports Ranking Case Study

Like all human decision making, evaluations of sports players are susceptible to implicit bias by the experts who may unconsciously favor particular groups. For instance, Olympic figure skating judges have been shown to exhibit a bias in favor of their home country [4]. To understand how unfair bias may manifest in the real world, we study fantasy football rankings. Fantasy sports is a billion dollar industry with around 45.9 Million players in the U.S in 2019[3]. Fantasy players choose real athletes for their fantasy teams each week and score points based on the athlete's real performance. Rankings of the athletes are provided by experts

	Time in Seconds	
#Candidates	Borda-Agg.	Fair-Post
100	0.11	0.30
1,000	0.89	2.41
10,000	8.83	23.48
100,000	104.85	235.07
1,000,000	***	2798.57

Table 1: Impact of number of candidates on run time for post-processing approximate aggregation of $|R| = 1000$.

[3]https://thefsga.org/industry-demographics/

to give guidance to players. These public rankings provide an ideal test-bed reflecting a real-world phenomena where human voters judge and rank candidates.

We collected rankings of National Football League (NFL) players for 16 weeks of the 2019 football season from the top 25 experts on the popular website FantasyPros.[4] To examine potential bias, we assigned players to groups based on conferences: American Football Conference (AFC) and National Football Conference (NFC) - each with 16 teams. We hypothesized that experts may favor one conference over the other, perhaps based on favorite teams or reputation.

Using weekly rankings of wide receivers as an example, we observe that week to week, the experts tend to agree strongly in their assessments of the players. On average they disagree on the pairwise order of less than 10% of players. The experts also exhibited a slight bias toward the NFC group, favoring it in on average 18% more pairs than the AFC group. Examining closer, we find that certain experts tend to consistently overestimate the NFC every week, while other are more fair. Figure 7 shows the rank parity scores for each expert's ranking over the season (thin gray lines). We compare this to ground truth rankings based on the actual points scored by the players each week, and consensus rankings generated using the unconstrained ILP and Fair-ILP methods (δ_p set to mean ground truth parity of 12% of pairs). We see that the unconstrained consensus rankings tend to exaggerate the NFC bias compared to the ground truth, while our correction method is able to generate rankings with similar parity. Interestingly, making the rankings fair in this context did not affect the prediction accuracy too much - indicating that the observed bias was not the main reason for the experts' incorrect predictions.

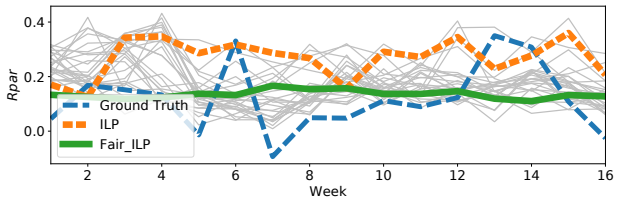

Figure 7: Rank parity scores for weekly rankings of NFL wide receivers for 2019 season.

6. DISCUSSION

Our work on this new fair rank aggregation problem opens many interesting avenues for novel future research, such as further implications for Social Choice Theory. Especially important are questions around supporting different fairness definitions and group identities. Toward this, we now sketch out future extensions of our pairwise framework.

Generalizing Beyond Binary Groups. While it is common to assume binary groups when designing bias mitigation methods, group identity in practice is more complex, and candidates often belong to multiple, possibly overlapping, groups. Much more study is required to fully understand fairness for multiple intersectional groups in the context of fair rank aggregation. However, a first step of supporting categorical sensitive attributes is straightforward to incorporate into our pairwise framework. For multiple

[4]`https://www.fantasypros.com/nfl/rankings`

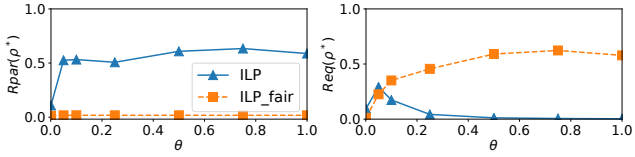

Figure 8: Comparison of rank parity (left) and rank equality (right) error introduced by fair-ILP on base rankings with $n = 50$ candidates,$|R| = 20$ and $p = 0.7$.

groups G_i, *Rpar* scores can be computed by counting the pairs that favor a candidate in G_i over any candidate not in G_i, for all $G_j \neq G_i$. This way, the pairwise preference given to each group is captured. Then, to enforce parity in fair-ILP we can define separate fairness thresholds δ_{G_i} for each group, and simply replace Equation 16 in Linear Program 1 with constraints for each group of the form $|\sum_{x_i \in G_i} \sum_{x_j \notin G_i} (A_{ij} - A_{ji})\ | \leq \delta_{G_i}$.

Alternative Fairness Definitions. Statistical parity is only one of several possible definitions of fairness. For instance, we show in [38] that a pairwise formulation also naturally supports fairness definitions based on *equal error rates*, such as the rank equality metric (Definition 7) which counts only those pairs favoring each group which are inverted in one ranking compared to another. In the context of fair aggregation, rank equality unfairness might occur if the aggregation algorithm erroneously mis-ranks one group more than another in ρ^* *compared to the base rankings.*

DEFINITION 7. ***Rank Equality Error:*** *Given rankings ρ and σ over items from two groups G_1 and G_2, the* ***rank equality error*** *for group G_1 is computed as*

$$Req_{G1}(\rho, \sigma) = \frac{\sum_{x_i \in G_1} \sum_{x_j \in G_2} I(x_i \prec_\rho x_j \text{ and } x_j \prec_\sigma x_i)}{|G_1||G_2|}$$

Figure 8 revisits the same experiment in Figure 5 for $p = 0.7$. This time however we measure the rank equality error. We observe that as the fair-ILP method enforces rank parity, it also introduces rank equality error. To support applications where rank equality fairness is required, our pairwise framework would need to be extended to explicitly target the fairness criterion F where:

$$\frac{1}{|R|} \sum_{\sigma \in R} |Req_{G_1}(\sigma, \rho^*) - Req_{G_2}(\sigma, \rho^*)| \leq \delta_{eq}. \qquad (23)$$

7. RELATED WORK

To our knowledge, contemporary fairness criteria have not been applied in the context of rank aggregation. Recent work on fair ranking has mainly focused on achieving statistical parity between groups in the top-k prefix of a single ranking [51, 17, 53, 5]. Yang and Stoyanovich [51] defined multiple variations of these fairness semantics. Zehlike et al. [53] expand on their work, proposing a greedy post-processing algorithm to correct for unfair bias while optimizing for utility. Celis et al. [17] formulate fair ranking as a constrained bipartite matching problem, optimizing for a number of different rank accuracy metrics alongside fairness. We first propose a pairwise version of statistical parity in [38]. In recent work [5], Asudeh et al. design pre-processing and database indexing strategies to facilitate fair ranking for real-time applications based on linear scoring.

In addition to statistical parity, other fairness definitions include error-based criteria such as Equalized Odds for fair classification [33] and its analogue for ranking [38]. Recently much attention has focused on causal fairness, including counterfactual [39] and interventional [5] fairness definitions. For instance, Salimi et al. [46] recently put forth a database repair model in which fairness depends on a causal path from the protected attribute to the outcome through any inadmissible attributes. This exciting direction aims to understand subtleties inherent in data attributes and causal mechanisms behind unfair bias, and is therefore distinct from our work.

Kemeny rank aggregation [35] based on the Kendall Tau distance [36] is one of the most popular and important aggregation formulations [14], which we thus target here. Kemeny aggregation has enjoyed much interest by the machine learning community [37], and has been applied for tasks from spam reduction in search results [26, 1] and group recommendation online [6, 9] to biomedical applications [40].

Computing the Kemeny optimal rank aggregation is NP-hard [8, 26]. A full review of methods is beyond the scope of this work (see benchmarking studies [2, 13]), however we note the exact integer programming solution of Conitzer et al. [23] and branch-and-bound method by Meila et al. [44] as inspiring our proposed fairness preserving aggregation methods (Section 4). Ali and Meila [2] characterize the difficulty of Kemeny aggregation based on agreement among the base rankings in R, which informs our evaluation design.

Rank aggregation stems from the study of ranked voting in Social Choice Theory [52, 35, 4]. This discipline provides a rich context for asking questions related to contemporary algorithmic fairness. For instance, Chakraborty et al. [18] apply Social Choice axioms to mitigate the impact of bad actors such as bots on Twitter and ensure fairness for groups of users with underrepresented preferences. Recent work [15, 16] has explored contemporary fairness for another classic problem in social choice: multi-winner voting. This problem differs from rank aggregation in that only a subset of candidates are selected. Methods proposed are being explored for use in real world voting systems [16]. These examples demonstrate the timely nature of these investigations.

8. CONCLUSION

This work offers the first formulation of the fair rank aggregation problem. A rich family of exact and approximate algorithms to solve this new optimization problem are presented. For strict fairness requirements, our exact fairness-aware ILP methods are robust to different amounts of bias agreement among the base rankings for rankings with $n < 100$ candidates. When fairness requirements are lenient, our fairness-aware B+B solution speeds runtime considerably while achieving optimal aggregation results. Lastly, our approximate fairness solution, which guarantees fairness while introducing the minimal amount of approximation error, is shown to scale to rankings of millions of candidates.

9. ACKNOWLEDGMENTS

The authors acknowledge valuable contributions by W. Gerych, and insights from A. Trapp, R. Paffenroth, and anonymous reviewers. We acknowledge funding by NSF IIS-2007932, IIS-1815866, IIS-1852498 and US DoE GAANN Fellowship P200A150306.

10. REFERENCES

[1] L. Akritidis, D. Katsaros, and P. Bozanis. Effective rank aggregation for metasearching. *Journal of Systems and Software*, 84(1):130–143, 2011.

[2] A. Ali and M. Meilă. Experiments with kemeny ranking: What works when? *Mathematical Social Sciences*, 64(1):28–40, 2012.

[3] K. M. Altenburger, R. De, K. Frazier, N. Avteniev, and J. Hamilton. Are there gender differences in professional self-promotion? an empirical case study of linkedin profiles among recent mba graduates. In *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[4] K. J. Arrow. *Social choice and individual values*. Yale University Press, 1963.

[5] A. Asudeh, H. Jagadish, J. Stoyanovich, and G. Das. Designing fair ranking schemes. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1259–1276. ACM, 2019.

[6] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126. ACM, 2010.

[7] S. Barocas and A. D. Selbst. Big data's disparate impact. *Cal. L. Rev.*, 104:671, 2016.

[8] J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and welfare*, 6(2):157–165, 1989.

[9] J. P. Baskin and S. Krishnamurthi. Preference aggregation in group recommender systems for committee decision-making. In *Proceedings of the third ACM conference on Recommender systems*, pages 337–340. ACM, 2009.

[10] T. Berg, V. Burg, A. Gombović, and M. Puri. On the rise of fintechs–credit scoring using digital footprints. Technical report, National Bureau of Economic Research, 2018.

[11] M. Bertrand and S. Mullainathan. Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination. *American economic review*, 94(4):991–1013, 2004.

[12] I. Bohnet, A. Van Geen, and M. Bazerman. When performance trumps gender bias: Joint vs. separate evaluation. *Management Science*, 62(5):1225–1234, 2015.

[13] B. Brancotte, B. Yang, G. Blin, S. Cohen-Boulakia, A. Denise, and S. Hamel. Rank aggregation with ties: Experiments and analysis. *PVLDB*, 8(11):1202–1213, 2015.

[14] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

[15] R. Bredereck, P. Faliszewski, A. Igarashi, M. Lackner, and P. Skowron. Multiwinner elections with diversity constraints. In *AAAI Conference on Artificial Intelligence*, 2018.

[16] L. E. Celis, L. Huang, and N. K. Vishnoi. Multiwinner voting with fairness constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 144–151. AAAI Press, 2018.

[17] L. E. Celis, D. Straszak, and N. K. Vishnoi. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*, 2017.

[18] A. Chakraborty, G. K. Patro, N. Ganguly, K. P. Gummadi, and P. Loiseau. Equality of voice: Towards fair representation in crowdsourced top-k recommendations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 129–138. ACM, 2019.

[19] L. Chen, R. Ma, A. Hannák, and C. Wilson. Investigating the impact of gender on rank in resume search engines. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 651. ACM, 2018.

[20] A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

[21] J. Clausen. Branch and bound algorithms-principles and examples. 1999.

[22] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. In *Advances in Neural Information Processing Systems*, pages 451–457, 1998.

[23] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing kemeny rankings. In *AAAI Conference on Artificial Intelligence*, volume 6, pages 620–626, 2006.

[24] J. C. de Borda. Memoire sur les elections au scrutin, 1781. *Histoire de l'Academie Royale des Sciences, Paris*, 1953.

[25] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012.

[26] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001.

[27] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and aggregating rankings with ties. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 47–58. ACM, 2004.

[28] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

[29] S. C. Geyik, S. Ambler, and K. Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2221–2231. ACM, 2019.

[30] A. G. Greenwald and M. R. Banaji. Implicit social cognition: attitudes, self-esteem, and stereotypes. *Psychological review*, 102(1):4, 1995.

[31] L. Gurobi Optimization. Gurobi optimizer reference manual, 2019.

[32] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[33] M. Hardt, E. Price, N. Srebro, et al. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016.

[34] M. Jacob, B. Kimelfeld, and J. Stoyanovich. A system for management and analysis of preference data. *PVLDB*, 7(12):1255–1258, 2014.

[35] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[36] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[37] A. Korba, S. Clémençon, and E. Sibony. A learning theory of ranking aggregation. In *Artificial Intelligence and Statistics*, pages 1001–1010, 2017.

[38] C. Kuhlman, M. VanValkenburg, and E. Rundensteiner. Fare: Diagnostics for fair ranking using pairwise error metrics. In *The World Wide Web Conference*, pages 2936–2942. ACM, 2019.

[39] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, pages 4066–4076, 2017.

[40] S. Lin. Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570, 2010.

[41] C. L. Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957.

[42] B. Mandhani and M. Meila. Tractable search for learning exponential models of rankings. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 392–399. PMLR, 2009.

[43] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[44] M. Meila, K. Phadnis, A. Patterson, and J. A. Bilmes. Consensus ranking under the exponential model. In *Proceedings of the 23rd Annual Conference on Uncertainty in Artificial Intelligence*, pages 285–294, 2007.

[45] J. Pearl. Heuristics: intelligent search strategies for computer problem solving. 1984.

[46] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*, pages 793–810, 2019.

[47] F. Schalekamp and A. v. Zuylen. Rank aggregation: Together we're strong. In *2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments*, pages 38–51. SIAM, 2009.

[48] E. L. Uhlmann and G. L. Cohen. Constructed criteria: Redefining merit to justify discrimination. *Psychological Science*, 16(6):474–480, 2005.

[49] A. Waters and R. Miikkulainen. Grade: Machine learning support for graduate admissions. *AI Magazine*, 35(1):64–64, 2014.

[50] Y. Wu, L. Zhang, and X. Wu. On discrimination discovery and removal in ranked data using causal graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '18, page 2536–2544, New York, NY, USA, 2018. Association for Computing Machinery.

[51] K. Yang and J. Stoyanovich. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 22:1–22:6. ACM, 2017.

[52] P. Young. Optimal voting rules. *Journal of Economic Perspectives*, 9(1):51–64, 1995.

[53] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578. ACM, 2017.