Rank Aggregation with Proportionate Fairness

Dong Wei, Md Mouinul Islam, Baruch Schieber, Senjuti Basu Roy
{dw277,mi257,sbar,senjutib}@njit.edu
Department of Computer Science, New Jersey Institute of Technology
New Jersey, USA

ABSTRACT

Given multiple individual rank orders over a set of candidates or items, where the candidates belong to multiple (non-binary) protected groups, we study the classical rank aggregation problem subject to proportionate fairness or p-fairness (**RAPF** in short), considering Kemeny distance. We first study the problem of producing a closest p-fair ranking to an individual ranked order (**IPF** in short) considering Kendall-Tau distance, and present multiple solutions for **IPF**. We then present two computational frameworks (a randomized RandAlgRAPF and a deterministic AlgRAPF) to solve **RAPF** that leverage the solutions of **IPF** as a subroutine.

We make several non-trivial algorithmic contributions: (i) we prove that when the group protected attribute is binary, **IPF** can be solved exactly using a greedy technique; (ii) we present two different solutions for **IPF** when the group protected attribute is multi-valued, ExactMultiValuedIPF is optimal and ApproxMultiValuedIPF admits a 2 approximation factor; (iii) we design a framework for **RAPF** solution with an approximation factor that is 2+ the approximation factor of the **IPF** solution. The resulting RandAlgRAPF and AlgRAPF solutions exhibit 3 and 4 approximation factors when designed using ExactMultiValuedIPF and ApproxMultiValuedIPF respectively.

We run extensive experiments using multiple real world and large scale synthetic datasets and compare our proposed solutions against multiple state-of-the-art related works to demonstrate the effectiveness and efficiency of our studied problem and proposed solution.

CCS CONCEPTS

• **Information systems → Data management systems**.

KEYWORDS

p-fairness, rank aggregation, algorithms

ACM Reference Format:
Dong Wei, Md Mouinul Islam, Baruch Schieber, Senjuti Basu Roy. 2022. Rank Aggregation with Proportionate Fairness. In *Proceedings of Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22)*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3514221.3517865

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

ACM ISBN 978-1-4503-9249-5/22/06…$15.00
https://doi.org/10.1145/3514221.3517865

1 INTRODUCTION

Ranking is a commonly used method to prioritize desirable outcomes among a set of candidates and is an essential step in many high impact applications, such as, hiring candidates for a job, selecting students for school and college admission or scholarship, finding winning candidates in a competition, or approving loans. Traditionally, producing the final ranking involves aggregating potentially conflicting preferences from multiple individuals and is a central problem in the areas of voting and social choice theory, which is traditionally known as the rank aggregation problem [3, 19, 32]. Our goal in this work is to revisit the rank aggregation problem considering a notion of fairness, namely *proportionate fairness or p-fairness* [8, 31] that ensures proportionate representation of every group based on a protected attribute in every position of the aggregated ranked order. P-fairness has been studied in the theory community to enable resource allocation satisfying temporal fairness or proportionate progress. The classical problem in this context is known as the *Chairman Assignment Problem* [6, 31] which studies how to select a chairman of a union every year from a set of r states such that that at any time the accumulated number of chairmen from each state is proportional to its weight. We formalize the *rank aggregation subject to p-fairness* or **RAPF** to that end.

RAPF is defined formally as follows: m conflicting rankings are given over a database of n candidates, where candidates have a protected attribute A with ℓ associated values (defined, e.g., over seniority level, ethnicity, or gender). Let $f(p)$ denote the fraction of candidates with protected attribute value p, that is, $f(p) = \frac{1}{n}\sum_{v \in V} \mathbf{1}_{A(v)=p}$. The goal is to find an aggregated ranking such that the total number of disagreements between the aggregated ranking and each of the individual m rankings is minimized, and for every protected attribute value p and every position k in the aggregated ranking, the representation of the candidates with protected attribute value p in the top k candidates is proportional to $f(p)$. P-fairness is desirable in several compelling rank aggregation applications, such as, French process of admitting students to university (Parcoursup), matching medical students to US hospitals for residency, or faculty hiring in the universities, to name a few. Section 1.1 describes one such application in depth.

We initiate this investigation by studying the **Individual p-Fairness** or **IPF** problem that finds a closest p-fair ranking to an individual ranking, which we believe is an important problem in its own merit. A similar problem is studied in the past [22] with weaker notion of fairness and the designed solutions are just heuristic. We investigate how a solution designed for **IPF** could solve **RAPF**.

1.1 Motivation

Running Example: p-fairness in faculty hiring. Consider a toy database of n (12) applicants who are interviewed to be hired for a small number of faculty positions in a university. The hiring committee comprises of a set of m (4) members, each of whom ranks these n candidates (refer to Table 1) based on their credentials and interview performance. After that, these individual ranks are to be aggregated to create an overall order based on which the candidates would be made job offers until the positions are filled. Potential protected attributes of the candidates are seniority level, research areas, and gender. As an example, considering seniority level, 3 applicants are junior, 4 are mid-career, and 5 are senior, making the proportion over seniority level to be 3/12, 4/12, and 5/12, respectively.

The goal of **RAPF** is to produce a ranked order over the 12 candidates by aggregating all 4 ranked lists such that the produced order is closest to the individual 4 ranks and *for each of the* 12 *positions and for each of the values of a particular protected attribute the candidates appear proportionate to their representation in the original data.* Indeed, it is *important to ensure fairness in each of the* 12 *positions considering the given protected attribute - otherwise, depending on who accepts/declines the job offer, the proportionate representation of the candidates based on the underlying protected attribute would get disrupted.* Intuitively speaking, assuming seniority level as the protected attribute, a solution designed for **RAPF** must ensure that the representation of junior, mid-career, and senior candidates is (0.75, 1, 1.25) up to integral rounding in the top 3 positions, (1, 1.33, 1.67) up to integral rounding in the top 4 positions, and so on.

Candidate Name	Gender	Seniority level	Area	Mem 1	Mem 2	Mem 3	Mem 4
Molly	Female	Junior	DB	1	3	4	6
Amy	Female	Junior	DB	2	2	1	5
Abigail	Female	Junior	AI	3	5	2	7
Kim	Male	Mid career	HCI	4	7	3	8
Lee	Male	Mid career	Theory	5	9	6	1
Park	Male	Mid career	Vision	6	1	5	2
Kabir	Male	Mid career	NLP	7	4	8	3
Damien	Male	Senior	ML	8	6	7	4
Andres	Male	Senior	Security	9	8	10	9
Aaliyah	Female	Senior	Systems	10	10	9	10
Kiara	Female	Senior	DM	11	11	12	11
Jazmine	Female	Senior	PL	12	12	11	12

Table 1: Original ranks provided by 4 members

We acknowledge that the existing popular group based fairness definition statistical parity *[18] is somewhat similar to p-fairness, however, the best adapted version of top-k statistical parity studied in a recent paper [25] does not account for proportionate representation in every position of the top-k, limiting its applicability. Section 3.4 contains further details.*

1.2 Contributions

Our first contribution is to formalize two optimization problems, **Individual p-Fairness** or **IPF** and the rank aggregation problem subject to proportionate fairness (**RAPF**) **(Section 2)** considering binary ($\ell = 2$) and multi-valued ($\ell > 2$) protected attributes.

Our second contribution is theoretical and algorithmic **(Sections 4, 5)**. For the **IPF** problem, we present an efficient greedy solution GrBinaryIPF for a binary protected attribute that runs in $O(n)$ time. For a multi-valued protected attribute, we prove that the proposed algorithms studied in a recent work [22] for **IPF** are heuristics and do not ensure optimality (refer to Section 3.1 for details). In fact, we claim that solving **IPF** for multi-valued protected attribute is non-trivial. We present two solutions for multi-valued **IPF** - a dynamic programming based exact algorithm ExactMultiValuedIPF that takes linear time when the number of values on the protected attribute is a constant, and ApproxMultiValuedIPF based on a minimum weight matching on convex bipartite graphs [12], that admits a 2 approximation factor.

Since rank aggregation problem under Kemeny Optimization is NP-hard for 4 or more lists [3, 19, 32], **RAPF** is also NP-hard. In Section 5, we present two algorithmic frameworks RandAlgRAPF and AlgRAPF for **RAPF**, one is randomized and the other one is deterministic that admit provable approximation factors. Both frameworks are scalable while the randomized one is highly scalable but because of its randomized nature, its approximation factor is expressed in expectation. Both algorithmic frameworks use as subroutine the solutions of **IPF**. They also leverage on variants of the Pick-A-Perm algorithm [3, 19, 32] that is widely used in the classical rank aggregation context. We then prove that the approximation factor of the solution designed for **RAPF** is 2+ the approximation factor of the **IPF** algorithm used as subroutine. This implies that multi-valued **RAPF** with ExactMultiValuedIPF admits a 3 approximation factor; whereas, it admits a 4 approximation factor when ApproxMultiValuedIPF is used instead. Table 2 summarizes our theoretical results.

Our third contribution is experimental (Section 6). We run extensive experiments using 3 real world and a large scale synthetic datasets, and compare an implementation of our solution with the implementation of two state-of-the-art solutions DetConstSort [22] for **IPF** and FairILP [25] for **RAPF**. Our first and foremost observation is that, consistent with our theoretical analysis, p-fairness promotes stronger notion of fairness, by ensuring proportionate representation of each of the protected attribute values for every position in the aggregated ranked order. Our experimental results demonstrate that our proposed model and solutions satisfy the fairness criteria proposed in state-of-the-art solutions [22, 25] - however, existing solutions do not extend to satisfy p-fairness. Our experimental results corroborate our theoretical results in terms of approximation factors and demonstrate that our solutions are highly scalable to large number of items and ranks.

2 PRELIMINARIES & FORMALISM

Database. contains n items or candidates. These two terms will be used interchangeably in the paper. Using the running example, $n = 12$. The set of items will be denoted V, individual items will be denoted by u and v.

Rank. We consider rankings of the items in V. Each such ranking can be viewed as a permutation. We will use the terms ranking and permutation interchangeably.

Multiple Rankings. The input consists m different complete rankings. Using the running example, $m = 4$.

Protected Attribute. Each item/candidate $v \in V$ has a *protected attribute* $A(v)$ that can take any of ℓ different values. As an example, seniority level is a multi-valued protected attribute with three possible values Junior, Mid career, Senior - thus $\ell = 3$. Contrarily, gender is a binary protected attribute with two values male and female, and $\ell = 2$.

Problem	Protected Attribute	Hardness	Algorithm	Approx Factor	Running Time
IPF	binary	p-time	GrBinaryIPF	exact	$O(n)$
	multi-valued	open	ExactMultiValuedIPF	exact	$O(n\ell 2^{\ell})$
			ApproxMultiValuedIPF	2	$O(n^{2.5}\log n)$
RAPF	binary	NP-hard	RandAlgRAPF+GrBinaryIPF	2	$O(n)$
			AlgRAPF+ GrBinaryIPF	2	$O(m^2 n\log n)$
	multi-valued	NP-hard	RandAlgRAPF+ ExactMultiValuedIPF	3	$O(n\ell 2^{\ell})$
			RandAlgRAPF+ ApproxMultiValuedIPF	4	$O(n^{2.5}\log n)$
			AlgRAPF+ ExactMultiValuedIPF	3	$O(m^2 n\log n+ mn\ell 2^{\ell})$
			AlgRAPF+ ApproxMultiValuedIPF	4	$O(m^2 n\log n+ mn^{2.5}\log n)$

Table 2: Summary of technical results

Notation	Meaning
A	Protected attribute
ℓ	Number of different values in A
$f(p)$	proportion of candidates with attribute value p
$\sigma(u)$	position of item u in rank σ

Table 3: Important notations

Rank Aggregation Measures [3, 19]. In this work we consider two popular rank distance measures Kendall-Tau distance and Spearman's footrule distance.

Definition 2.1. **Kendall-Tau distance.** Given two permutations $\sigma, \eta : V \rightarrow [1..n]$, the Kendall-Tau distance between the two permutations is the sum of pairwise disagreements between σ and η (bubble-sort distance).

$$\mathcal{K}(\sigma,\eta) = \sum_{\{u,v\}\subseteq V} \mathbf{1}_{(\sigma(v)-\sigma(u))(\eta(v)-\eta(u))<0}$$

Note that the Kendall-Tau distance is symmetric, that is, $\mathcal{K}(\sigma,\eta) = \mathcal{K}(\eta,\sigma)$. It also satisfies the triangle inequality, for any three permutations σ, μ, η we have $\mathcal{K}(\sigma,\mu)+\mathcal{K}(\mu,\eta) \geq \mathcal{K}(\sigma,\eta)$.

Definition 2.2. **Spearman's footrule distance.** Given two permutations $\sigma, \eta : V \rightarrow [1..n]$, the Spearman's footrule distance between the two permutations is the sum of the absolute values (ℓ_1 distance) of the differences between two permutations.

$$\mathcal{S}(\sigma,\eta) = \sum_{u\in V} |(\sigma(u) - \eta(u)|$$

Using the running example, the Kendall-Tau distance between the rankings of Member 1 and Member 2 is 12 because there are 12 pairs of items that appear in opposite order in these two rankings. Spearman's footrule distance between them is 22, which is the sum of the absolute values of the difference in the order between these two rankings.

Relationship between the two measures. Diaconis and Graham [17] proved that for any two permutations the Spearman's footrule distance is at least the Kendall-Tau distance between them, and at most twice the Kendall-Tau distance. That is, for any two permutations σ, η, we have $\mathcal{K}(\sigma,\eta) \leq \mathcal{S}(\sigma,\eta) \leq 2\mathcal{K}(\sigma,\eta)$.

In the rest of the paper, we focus on Kendall-Tau distance and when we refer to Spearman's footrule distance we will state it explicitly. The Kemeny distance between a single ranking and multiple rankings is based on Kendall-Tau distance.

Definition 2.3. **Kemeny Distance.** For rankings $\rho_1, \rho_2, \ldots, \rho_m$ the *Kemeny Distance* of the ranking σ to these rankings is

$$\kappa(\sigma, \rho_1, \rho_2, \ldots, \rho_m) = \sum_{i=1}^{m} \mathcal{K}(\sigma, \rho_i)$$

Using the running example, Kemeny Distance between each of the aggregated rankings presented in the three columns of Table 4 and the individual member ranks are 34, 34, and 46, respectively.

We note that Kemeny distance which is based on Kendall-Tau distance is the most popular and accepted measure for quantifying the quality of rank aggregation and has been widely used in the related work on rank aggregation [2, 3, 18]. The Kemeny distance measure has a maximum likelihood interpretation and it is the only known measure that simultaneously satisfies: neutrality, consistency, and the (extended) Condorcet property. Moreover, Kendall-Tau/Kemeny has also been adopted in the only previously known fair rank aggregation FairILP [25] work. Other distance measures are briefly described in Section 3.

Definition 2.4. **Proportionate Fair or p-fair ranking [8, 31].** For any protected attribute value p, let $f(p)$ denote the fraction of items with this value, that is, $f(p) = \frac{1}{n}\sum_{v\in V} \mathbf{1}_{A(v)=p}$. A ranking σ is *proportionate fair* or *p-fair* if for every $k \in [1..n]$, the number of items with protected attribute value p among the k top ranked items in σ is either $\lfloor f(p)\cdot k\rfloor$ or $\lceil f(p)\cdot k\rceil$.

Using the running example, if gender is the protected attribute with 50% representation of male and female, then p-fairness implies 1 male and 1 female in the top-2 items, 2 males and 2 females in the top-4 items , and so on. (Note that for any odd k the difference between the number of males and females in the top-k is exactly 1.) We refer to the 3rd column of Table 4 and note that p-fairness is satisfied.

Definition 2.5. **Relaxed p-fair ranking.** Given an integer input $\delta \geq 0$, a ranking σ is *relaxed proportionate fair* or *relaxed p-fair* if for every $k \in [1..n]$, the number of items with protected attribute value p among the k top ranked items in σ is between $\lfloor f(p)\cdot k\rfloor - \delta$ and $\lceil f(p)\cdot k\rceil + \delta$.

This alternative fairness definition essentially relaxes p-fair ranking definition, such that for every position, the proportionate representation of items with protected attribute value p is allowed to have at most δ deviation (an input parameter) from its original p-fair ranking. Using the running example, if gender is the protected attribute with 50% representation of male and female, then the relaxed p-fairness with $\delta = 1$ implies at least 1 male and at least

Rank	Rank aggregation (without fairness)	Rank aggregation (with statistical parity) [25]	Rank aggregation (with p-fairness)
1	Amy (Female)	Amy (Female)	Amy (Female)
2	Molly (Female)	Molly (Female)	Park (Male)
3	Abigail (Female)	Abigail (Female)	Molly (Female)
4	Kim (Male)	Kim (Male)	Kabir (Male)
5	Lee (Male)	Lee (Male)	Abigail (Female)
6	Park (Male)	Park (Male)	Kim (Male)
7	Kabir (Male)	Kabir (Male)	Lee (Male)
8	Damien (Male)	Damien (Male)	Aaliyah (Female)
9	Andres (Male)	Andres (Male)	Damien (Male)
10	Aaliyah (Female)	Aaliyah (Female)	Kiara (Female)
11	Kiara (Female)	Kiara (Female)	Andres (Male)
12	Jazmine (Female)	Jazmine (Female)	Jazmine (Female)
Kemeny Distance	34	34	46

Table 4: Rank aggregation results of comparable methods using Section 1.1 example considering gender as the protected attribute.

1 female in the top-4 items, at least 2 males and at least 2 females in the top-6 items, and so on.

2.1 Problem Formulation

P1: Individual p-fair rank (or IPF). Given a ranking ρ find a *p-fair* ranking that is closest to ρ in Kendall-Tau distance.

P2: Rank aggregation under p-fairness (or RAPF). Given m rankings $\rho_1, \rho_2, \ldots, \rho_m$ find a *p-fair* ranking that minimizes the Kemeny distance to these m rankings. We observe that RAPF is NP-Hard which directly follows from the fact that rank aggregation considering unconstrained Kemeny distance minimization is NP-hard when $m \geq 4$ [3].

We study **IPF** and **RAPF** for binary and multi-valued protected attributes considering fairness as a constraint. By that process, it is likely to deteriorate the Kemeny Distance values, i.e., the Kemeny Distance of an unfair rank aggregation is likely to be smaller than that of a fair one (recall Column 1 and Column 3 of Table 4). These choices and other alternative ways of incorporating fairness inside rank aggregation are explored in Section 7.

We also study **IPF** and **RAPF** subject to the relaxed p-fairness. Our proposed solutions trivially adapt for this version and we omit those for brevity. Experimental results based on this relaxed definition are included in Section 6.4.

3 RELATED WORK & COMPARISON

Rank Aggregation. The rank aggregation study was initiated in the early 2000s by Dwork et. al. [19]. Since then, rank aggregation and several of its variants have been well studied, including rank aggregation considering different optimization functions, rank aggregation with partial ranking information, or with ties [2, 3, 7, 11, 20]. Kemeny optimal rank aggregation which minimizes the sum/average Kendall-Tau distances [23, 24] to the individually ranked lists is the most popular variant. In [3, 7], the authors show that computing the Kemeny optimal rank aggregation is NP-hard for 4 or more rankings. There exist both randomized and deterministic approximation algorithms for rank aggregation [3, 32, 33]. In [3], Ailon et al. introduced a randomized approximation algorithm with a $\frac{4}{3}$ approximation factor. In [32, 33], the authors propose deterministic pivoting algorithms with the same approximation factors. In [16] Conitzer et al. propose an exact integer programming solution for the Kemeny optimal rank aggregation.

One of the early yet popular results in this space is the randomized algorithm Pick-a-Perm [3, 19] that is shown to admit a $\frac{1}{2}$ approximation factor for the Kemeny Rank Aggregation Problem in expectation. We adapt Pick-a-Perm in our proposed solution for the **RAPF** *problem.*

Alternative rank aggregation measures. Other than Kemeny, alternative measures of the quality of rank aggregations, such as, those based on Spearman's Footrule and Borda's Method [19]. We note that finding an optimal rank aggregation using Spearman's Footrule based measure is computationally easy. However, it is *open* whether the **RAPF** problem using Spearman's Footrule distance is computationally tractable. On the other hand, the **IPF** problem using Spearman's Footrule distance is tractable. We design a polynomial time algorithm for the **IPF** problem in Spearman's Footrule distance and use it to approximate the **IPF** problem in Kendall-Tau distance. Borda's method [10] is a "positional" method. It assigns a score corresponding to the position in which a candidate appears within each voter's ranked list of preferences, and the candidates are sorted by their total score. Rank aggregation using Borda's method is also computationally easy, however, it does not satisfy the Condorcet criterion. Since Borda's method does not induce a distance between rankings it is unclear how to extend it to satisfy the p-fairness constraint.

Proportionate Fairness. Based on the Chairman assignment problem [31], the idea of proportionate fairness (p-fairness) was studied in the context of resource scheduling [8]. The Chairman assignment problem simply studies how to select a chairman for a union from k states such that at any time the accumulated number of chairmen from each state is proportional to its weight. In [8], Baruha et al. propose an algorithm for generating the p-fair schedule. Then, [9] introduces a series of algorithms for different single resource p-fair scheduling problems. Note that p-fairness is a group fairness criteria that is close to statistical or demographic parity [15] studied in the context of group fairness. *We note that for the rank aggregation problem, p-fairness is more suitable and stronger than statistical parity, because it ensures statistical parity for every position in the ranked order. This makes the problem significantly harder and the existing solutions do not trivially adapt.*

Social Choice Theory. Various ranking methods have been studied in the field of social choice theory [4, 21, 23, 28, 29, 35]. Early social choice theory literature considered rank aggregation in the context of preference aggregation methods [23, 29, 35]. The social choice theory papers [4, 21] focus on Arrow's impossibility theorem. This theorem states that it is impossible to have a rank aggregation method that simultaneously satisfies several conditions some of which relate to fairness. The paper [28] seeks to identify rank aggregation methods that are "close" to satisfying Arrow's conditions, enabling decisions that are fairer in practice. However, the focus of these works is to propose *models*, whereas, our primary goal is to develop efficient computational framework by adapting some of these proposed models.

3.1 Fair Ranking Solutions

Several recent fair ranking studies focus on achieving fairness on a *single* rank [5, 13, 22, 36]. Celis et al. [13] introduce a top-k fairness measure that ensures a given upper and lower bound of the representation of each of the protected attribute values in the top-k, for fixed values of k. They use Spearman's footrule-like distance which is easier than Kendall-Tau distance since it can be modeled by a maximum weight perfect matching problem in a bipartite graph. They provide a dynamic programming exact algorithm, and efficient approximation algorithms. In [36], Zehlike et al. extend group fairness using the standard notion of protected groups and ensure that the proportion of protected candidates in every top-k ranking remains statistically above a given minimum (while not ensuring any upper bound). Asudeh et al. [5] propose sweep-line-based algorithms for a more general fairness ranking problem.

Next, we describe two related works in more detail: the first one is a recent work DetConstSort [22] that studies a variant of the **IPF** problem. The other one is FairILP [25], which to the best of our knowledge is the only recent work that studies some version of fair rank aggregation alas only with binary protected attributes and thus can be compared to **RAPF**.

3.1.1 DetConstSort. Geyik et al. [22] propose Algorithm DetConstSort to produce fairness-aware ranking given an input ranking. This algorithm ensures that for every protected attribute value p, and for every $k \in [1..n]$ the number of items with protected attribute value p among the top k ranked items in the output ranking is at least $\lfloor f(p) \cdot k \rfloor$, where $f(p)$ is the fraction of items with protected attribute value p, that is, $f(p) = \frac{1}{n} \sum_{v \in V} \mathbf{1}_{A(v)=p}$. Essentially, Algorithm DetConstSort produces a ranking that only satisfies the lower bound of p-fairness.

Example 3.1. **Statement:** DetConstSort **[22] does not produce the closest ranking that satisfies the p-fairness lower bound.** We simulate the running of Algorithm DetConstSort on the ranking given by Member 1 in Table 1 considering *seniority level* as the protected attribute. The algorithm scans the ranked items in descending order starting at the top ($k = 1$), and checks at each position, whether any value of the protected attribute becomes "tight" and thus an item with this value needs to be inserted to the tentative output ranking. For the ranking given by Member 1, no seniority level becomes tight at $k = 1, 2$. At $k = 3$, $\lfloor f(Senior) \cdot k \rfloor = \lfloor 5/12 * 3 \rfloor = 1$ and $\lfloor f(Mid\ career) \cdot k \rfloor = \lfloor 4/12 * 3 \rfloor = 1$. So, the top ranked *Senior* candidate (*Damien*) and the top ranked *Mid career* candidate (*Kim*) are inserted to the tentative output ranking. Since *Kim* is ranked higher than *Damien* in the input ranking, the tentative (ordered) output ranking is $[Kim, Damien]$. At $k = 4$, $\lfloor f(Junior) \cdot k \rfloor = \lfloor 3/12 * 4 \rfloor = 1$ and the top *Junior* candidate *Molly* needs to be inserted in the list. Since *Molly* is ranked higher than both *Kim* and *Damien* in the input ranking and since both *Kim* and *Damien* can be pushed to position 3 without violating the p-fairness lower bound, *Molly* is inserted into position 1 of the tentative output ranking which is now $[Molly, Kim, Damien]$. Continuing in the same manner, the final output ranking is

$$[Molly, Kim, Lee, Damien, Amy, Park,$$
$$Andres, Abigail, Aaliyah, Kabir, Kiara, Jazmine]$$

The Kendall-Tau distance between the Member 1 ranking and the output ranking is 12. However, consider the following ranking.

$$[Molly, Amy, Kim, Damien, Abigail, Lee,$$
$$Andres, Park, Aaliyah, Kabir, Kiara, Jazmine]$$

It also satisfies the p-fairness lower bound and the Kendall-Tau distance between it and the Member 1 ranking is only 8.

Example 3.2. **Statement:** DetConstSort **[22] does not produce a p-fair ranking.** The ranking produced by DetConstSort in Example 3.1 violates the upper bound of the p-fairness condition, since the seniority level of 2 out of the top 3 candidates is *Mid career* but $\lceil f(Mid\ career) \cdot 3 \rceil = \lceil 4/12 * 3 \rceil = 1 < 2$.

3.1.2 FairILP. Kuhlman and Rundensteiner [25] consider fairness aware rank aggregation in a setting of a *binary protected attribute*. To measure fairness they propose *pairwise statistical parity*.

Definition 3.3. **Pairwise statistical parity.** For a ranking σ with a binary protected attribute, let V_i be the set of items with protected attribute value i, we define $R_{par}(\sigma)$ as:

$$R_{par}(\sigma) = \frac{1}{|V_1||V_2|} \left| \sum_{\{u \in V_1\}} \sum_{\{v \in V_2\}} \left(\mathbf{1}_{\sigma(u)<\sigma(v)} - \mathbf{1}_{\sigma(v)<\sigma(u)} \right) \right|.$$

The ranking σ satisfies pairwise statistical parity if $R_{par}(\sigma) = 0$. The relaxed pairwise statistical parity requires that $R_{par}(\sigma) \leq \delta$, for a given $\delta \geq 0$. The *unnormalized* pairwise statistical parity is defined as $|V_1||V_2|R_{par}(\sigma)$.

Given m rankings $\rho_1, \rho_2, \ldots, \rho_m$, FairILP finds a ranking σ whose pairwise unnormalized statistical parity is bounded by a given $\delta \geq 0$ that is closest to the input rankings in Kemeny distance.

Example 3.4. **Statement:** FairILP **[25] is not necessarily p-fair even with** $\delta = 0$.

Consider the running example and assume that the (binary) protected attribute considered is gender.

Table 4 shows three aggregated rankings for the running example, the first without fairness, with second subject to pairwise statistical parity with $\delta = 0$, and the third subject to p-fairness. Note that the first two rankings are identical, which implies that pairwise statistical parity does not imply p-fairness. Intuitively, the reason for this is that pairwise statistical parity just considers pairs of items with different protected attribute value in an aggregated manner and does not consider the actual positions of the items in the aggregated ranking.

In summary, **IPF** *is stronger than any of the existing fairness aware single rank problem [5, 13, 22, 36], because we consider proportionate representation considering both lower and upper bound of the protected attributes for every position. Similarly,* **RAPF** *promotes a stronger notion of fairness compared to* FairILP *[25], as well as consider both binary and multi-valued protected attribute.*

4 INDIVIDUAL P-FAIRNESS (IPF)

In this section, we describe our proposed solutions for the individual p-fairness or the **IPF** problem. First, we consider the binary case, denoted **BinaryIPF**, in which the protected attribute has two values, i.e., $\ell = 2$. We present an exact greedy algorithm GrBinaryIPF for **BinaryIPF**, prove its correctness, and analyze its running time. Then, we consider the general case of **IPF**, denoted **MultiValuedIPF**, when $\ell > 2$. We demonstrate that **MultiValuedIPF** cannot be solved using a greedy algorithm similar to the binary case, and present two solutions: a dynamic programming based exact algorithm ExactMultiValuedIPF, and an approximation algorithm ApproxMultiValuedIPF based on minimum weight matching. We analyze the running time and prove the correctness of both algorithms.

4.1 BinaryIPF

In this subsection we present an exact algorithm to **BinaryIPF** in which the protected attribute value of each of the items can take only two possible values. Algorithm GrBinaryIPF takes an input ranking ρ and the output is a p-fair ranking σ with the minimum Kendall-Tau distance to ρ. The algorithm builds on Lemma 4.1 that implies that if item u is the i-th item (counting from the top) with protected attribute value p in ranking ρ, then the same item is also the i-th item with protected attribute value p in ranking σ.

Baruah et al. [8] proved the following for any p-fair ranking. Consider the ranks of the items with protected attribute value p in the p-fair ranking. Then, for $i \in [1..f(p) \cdot n]$, the i-th such item has to be ranked within the interval

$$[\left\lfloor \frac{i-1}{f(p)} \right\rfloor + 1, \left\lceil \frac{i}{f(p)} \right\rceil].$$

Thus, for every item $v \in V$, we define the interval $[\texttt{top}(v), \texttt{bot}(v)]$ as the feasible positions of this item in any p-fair ranking that is closest to permutation ρ.

Algorithm GrBinaryIPF whose pseudo code is given in Algorithm 1 starts by computing $\texttt{bot}(v)$, for every $v \in V$ (Line 1). Then, it sorts the items according to their rank, and partitions the sorted list into two sub-lists, one for each protected attribute value (Line 3). The ranking σ is constructed from top to bottom. For each position i, Algorithm GrBinaryIPF considers the current top items u_1 and u_2 in each of the sub-lists. In case $\texttt{bot}(\cdot)$ of one of these two items is "tight", that is, equals i, Algorithm GrBinaryIPF assigns i to σ of this item. (In Lemma 4.2, we show that both items cannot be tight.) Otherwise, Algorithm GrBinaryIPF assigns i to σ of the item among u_1 and u_2 that is ranked higher in ρ. (Lines 5–12).

Algorithm 1 GrBinaryIPF

```
1:  compute bot(v) for each item v ∈ V
2:  sort the items according to the ranking ρ
3:  partition the sorted list into two sub-lists L1, L2, one for each
    protected attribute value
4:  for i = 1 to n do
5:      Let u1 and u2 be the current top items in L1 and L2
6:      if bot(u1) = i ∨ bot(u2) = i then
7:          v ← the tight item among u1 and u2
8:      else
9:          v ← the higher ranked item in ρ among u1 and u2
10:     end if
11:     σ(v) ← i
12:     remove v from its ordered list
13: end for
14: return σ
```

We demonstrate the algorithm considering as input the initial ranking provided by Member 2 in the running example and the binary protected attribute gender. The following two sub-lists are obtained:

$$L_{\text{female}} = [Amy, Molly, Abigail, Aaliyah, Kiara, Jazmine]$$
$$L_{\text{male}} = [Kim, Lee, Park, Kabir, Damien, Andres]$$

Note that $\texttt{bot}(Amy) = \texttt{bot}(Kim) = 2$. Amy is put into the first position in σ since Amy is ranked higher than Kim in ρ. Since $\texttt{bot}(Kim) = 2$, Kim is assigned be the second place in σ. By repeating this procedure, we end up with the ranking:

$$[Amy, Kim, Molly, Lee, Abigail, Park,$$
$$Kabir, Aaliyah, Damien, Kiara, Andres, Jazmine],$$

where $\mathcal{K}(\rho, \sigma) = 17$.

Running Time Analysis: It is straightforward to see that Algorithm GrBinaryIPF runs in $O(n)$ time, since all the computations can be done by a constant number of linear scans over the items.

As mentioned above the algorithm is based on the following lemma.

Lemma 4.1. *Consider two elements u and v with the same protected attribute value ($A(u) = A(v)$). If $\rho(u) < \rho(v)$ then $\sigma(u) < \sigma(v)$ in any p-fair ranking σ that is closest to permutation ρ,*

Proof Sketch: The proof is by contradiction. Assume that $\sigma(v) < \sigma(u)$. Consider the ranking σ' given by swapping $\sigma(v)$ and $\sigma(u)$, that is, $\sigma'(u) = \sigma(v)$, $\sigma'(v) = \sigma(u)$, and for every $w \in V \setminus \{u, v\}$, $\sigma'(w) = \sigma(w)$. The ranking σ' is also p-fair (since $A(u) = A(v)$), and we prove below that $\mathcal{K}(\sigma', \rho) < \mathcal{K}(\sigma, \rho)$; a contradiction.

Since $\rho(u) < \rho(v)$ and $\sigma(u) > \sigma(v)$ we have

$$(\sigma(u) - \sigma(v))(\rho(u) - \rho(v)) < 0$$
$$(\sigma'(u) - \sigma'(v))(\rho(u) - \rho(v)) > 0.$$

So, the pair (u, v) contributes 1 to $\mathcal{K}(\sigma, \rho)$ and 0 to $\mathcal{K}(\sigma', \rho)$. Since for $x, y \in V \setminus \{u, v\}$, we have $(\sigma(x) - \sigma(y)) = (\sigma'(x) - \sigma'(y))$ all such pairs (x, y) contribute the same to $\mathcal{K}(\sigma, \rho)$ and $\mathcal{K}(\sigma', \rho)$. Consider $w \in V \setminus \{u, v\}$, such that either $\sigma(w) > \sigma(u)$ or $\sigma(w) < \sigma(v)$. We have $(\sigma(w) - \sigma(u)) = (\sigma'(w) - \sigma'(u))$ and $(\sigma(w) - \sigma(v)) = (\sigma'(w) - \sigma'(v))$. Thus, the contribution of the pairs (u, w) and (v, w) is the same to $\mathcal{K}(\sigma, \rho)$ and $\mathcal{K}(\sigma', \rho)$.

We are left with the case $w \in V \setminus \{u, v\}$, such that $\sigma(w) \in (\sigma(v), \sigma(u))$. In this case

$$(\sigma(v) - \sigma(w)) = (\sigma'(u) - \sigma'(w)) < 0$$
$$(\sigma(u) - \sigma(w)) = (\sigma'(v) - \sigma'(w)) > 0.$$

Clearly, $\rho(v) - \rho(w) > \rho(u) - \rho(w)$. Thus

$$(\sigma(v) - \sigma(w))(\rho(v) - \rho(w)) < (\sigma'(u) - \sigma'(w))(\rho(u) - \rho(w))$$
$$(\sigma(u) - \sigma(w))(\rho(u) - \rho(w)) < (\sigma'(v) - \sigma'(w))(\rho(v) - \rho(w)),$$

which implies that the contribution of the pairs (u, w) and (v, w) to $\mathcal{K}(\sigma', \rho)$ is at most their contribution to $\mathcal{K}(\sigma, \rho)$. ■

Lemma 4.2. *For any iteration i of the algorithm one cannot have $\texttt{bot}(u_1) = i$ and $\texttt{bot}(u_2) = i$.*

Proof. Since $\texttt{bot}(\cdot)$ is nondecreasing as we iterate over the items, then for all items v that are added to σ up to iteration i we have $\texttt{bot}(v) \leq i$. Suppose that both $\texttt{bot}(u_1) = i$ and $\texttt{bot}(u_2) = i$. In this case thare are $i + 1$ items ($i - 1$ items from previous iterations together with u_1 and u_2) that need to be ranked in the top i places, which is infeasible, and thus in contradiction to the feasibility of p-fair ranking as proved in [8]. □

THEOREM 4.3. *Algorithm* GRBINARYIPF *returns the exact solution to the* **BinaryIPF** *problem.*

PROOF. To obtain a contradiction assume that μ is the p-fair ranking with minimum distance to ρ, and that $\mu \neq \sigma$. Let i be the top rank where μ and σ differ. Let u be the item ranked i in μ and v be the item ranked i in σ. Since the order of the items with the same value of the protected attribute has to be the same in both μ and σ, we must have that $A(u) \neq A(v)$. Without loss of generality assume that u is in sub-list L_1 and v is in sub-list L_2. Certainly $\mathtt{bot}(u) \neq i$ and $\mathtt{bot}(v) \neq i$ as otherwise either μ or σ would not be p-fair. Thus, according to our algorithm $\rho(v) < \rho(u)$. Let j be the rank of v in μ. Since i is the top rank where μ and σ differ we must have $j > i$. Also, all items ranked $i, \ldots, j-1$ in μ must be in sub-list L_1. As otherwise, the order of the items from L_2 would not be the same in both μ and σ.

Since $\rho(v) < \rho(u)$, then for item w ranked $j-1$ in μ, we also have $\rho(v) < \rho(w)$. Item w is ranked lower than $j-1$ in σ, thus $\mathtt{bot}(w) \geq j$. Since the rank of item v in σ is i, $\mathtt{top}(v) \leq i \leq j-1$. However, then the ranking μ' given by swapping the items w and v ranked $j-1$ and j in μ is p-fair and similar to Lemma 4.1 it can be shown to be closer than μ to the ranking ρ, a contraction. □

4.2 MultiValuedIPF

In this subsection we present an approximate and an exact algorithms for **MultiValuedIPF**. The input is a ranking ρ and the output is a p-fair ranking σ that minimizes the Kendall-Tau distance to ρ.

MultiValuedIPF is a harder problem. We begin this section by demonstrating that a simple greedy scheme is not adequate to solve **MultiValuedIPF** like in the binary case. Consider the following artificial example consisting of 20 items with four possible values of their protected attribute. There are 5 items with protected attribute value a, 10 items with protected attribute value b, 4 items with protected attribute value c, and 1 item with protected attribute value d. Note that we must have one item with protected attribute value a in each block of 4 ranked items starting from the top, one item with protected attribute value b in each block of 2 ranked items starting from the top, and one item with protected attribute value c in each block of 5 ranked items starting from the top.

Now, consider the ranking $\rho = 1, \ldots, 20$ (the identity permutation [30]). The value of the protected attribute for these items in given in the table 5.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a	a	b	b	b	d	c	b	c	b	a	b	c	b	a	b	c	b	a	b

Table 5: Protected attribute values

The greedy algorithm creates a p-fair ranking by starting with the highest ranked item 1 (with protected attribute value *a*), then the 2 items 3, and 4 (with value *b*), then the item 6 (with value *d*), next it must pick item 7 (with value *c*), and 5 (with value *b*), and only then item 2 (with value *a*). From this point on the p-fair ranking coincides with the original ranking; that is, items $8, \ldots, 20$ appear in order, and the resulting ranking is $(1, 3, 4, 6, 7, 5, 2, 8, \ldots, 20)$. The Kendall-Tau distance of this ranking from ρ is 7.

However, the optimal p-fair ranking is $(1, 3, 4, 7, 2, 5, 6, 8, \ldots, 20)$ which is closer to ρ with Kendall-Tau distance 5.

4.2.1 Approximation Algorithm. We first present an efficient algorithm APPROXMULTIVALUEDIPF for computing **IPF** that is based on minimum weight matching in a bipartite graph. Then, we proceed with the exact algorithm that is more complex.

The proposed algorithm APPROXMULTIVALUEDIPF whose pseudo code is given in Algorithm 2 considers as the underlying abstraction a weighted bipartite graph $G(V, Y, E)$, where $|V| = |Y| = n$. The nodes in V represent the items and the nodes in Y represent their potential position.

Recall that Baruah et al. [8] proved that in any p-fair ranking the position of an item $v \in V$ must be within $[\mathtt{top}(v)..\mathtt{bot}(v)]$. Consequently, an edge $e_{vy} \in E$ exists, if $y \in [\mathtt{top}(v)..\mathtt{bot}(v)]$. The algorithm assigns the weight $|\rho(v) - y|$ to every edge $e_{vy} \in E$ (Line 2), which is the Spearman's footrule distance between position of v in ρ and y. Then, APPROXMULTIVALUEDIPF finds a perfect matching in this graph (Line 4), and the output of the perfect matching induces a p-fair ranking σ (Line 6) that is closest to ρ in Spearman's footrule distance.

Algorithm 2 APPROXMULTIVALUEDIPF(G)

```
1: for e_vy ∈ E do
2:     weight(e_vy) ← |ρ(v) − y|
3: end for
4: Find M a minimum weight perfect matching in G
5: for e_vy ∈ M do
6:     v is set to be the item ranked y in σ
7: end for
8: return σ
```

We demonstrate the algorithm considering as input the initial ranking provided by Member 2 and the ternary protected attribute *seniority level*. The top ranked candidate of Member 2 is *Park* whose seniority level is *Mid career*. Note that $f(\textit{Mid career}) = \frac{4}{12} = \frac{1}{3}$. Thus, $\mathtt{top}(Park) = 1$ and $\mathtt{bot}(Park) = \lceil 1 \cdot \frac{3}{1} \rceil = 3$. Thus, there are 3 edges connecting to nodes in Y: $e_{Park,1}, e_{Park,2}, e_{Park,3}$ with weights 0, 1, 2, respectively. The rest of the edges of the bipartite graph are computed similarly. The minimum weight perfect matching in the created bipartite graph implies the following ranking, and the Spearman's footrule distance to the original ranking of Member 2 is 18:

$$[Park, Amy, Damien, Kabir, Andres, Molly,$$
$$Kim, Aaliyah, Abigail, Kiara, Lee, Jazmine].$$

Running Time: The running time of Algorithm APPROXMULTIVALUEDIPF is dominated by the running time of the minimum weight perfect matching which is $O(n^{2.5} \log n)$.

THEOREM 4.4. APPROXMULTIVALUEDIPF *admits a* 2*-approximation factor for the* **IPF** *problem.*

Proof Sketch: A lemma similar to Lemma 4.1 holds also for the Spearman's footrule distance, and thus any p-fair ranking that is closest in Spearman's footrule distance to the ranking ρ has to correspond to a perfect matching in G. It follows that the minimum weight perfect matching implies a p-fair ranking σ that is closest in Spearman's footrule distance to the ranking ρ. Let μ be a p-fair ranking that is closest (in Kendall-Tau distance) to ρ. Clearly, $\mathcal{S}(\rho, \sigma) \leq \mathcal{S}(\rho, \mu)$. Thus, by the inequalities in [17] $\mathcal{K}(\rho, \sigma) \leq \mathcal{S}(\rho, \sigma) \leq \mathcal{S}(\rho, \mu) \leq 2\mathcal{K}(\rho, \mu)$. We conclude that σ is a 2-approximation to a p-fair ranking that is closest (in Kendall-Tau distance) to the ranking ρ. ∎

4.2.2 Exact Algorithm. We present a dynamic programming based exact algorithm ExactMultiValuedIPF for the **MultiValuedIPF** problem. We prove that when ℓ, the number of different values of the protected attribute, is a constant (or even logarithmic in n), Algorithm ExactMultiValuedIPF computes the closest p-fair ranking in polynomial time. The running time of ExactMultiValuedIPF is exponential in ℓ, and thus when $\ell = \Omega(n)$, the running time of ExactMultiValuedIPF is exponential. Due to space constraints we just describe the intuition behind this algorithm.

Consider an index $1 \leq k < n$. Suppose that we wish to break the problem of computing the closest p-fair ranking into two subproblems. One is computing the top k items of the p-fair ranking and the other is computing the bottom $n - k$ items. Let's concentrate on computing the bottom $n - k$ items, that is, which items are in positions $k + 1, \ldots, n$ and their order. Certainly, this depends on which items are ranked in the top k (but it does not depend on the order of these top k items). The algorithm is based on the observation that the amount of this information is limited. Note that for each item u if $\mathtt{bot}(u) \leq k$ then u must be in the top k and if $\mathtt{top}(u) > k$ then u must be in the bottom $n - k$. The only ambiguity is regarding the items u for which $\mathtt{top}(u) \leq k$ and $\mathtt{bot}(u) \geq k + 1$. It follows that for all these items u we have $k, k + 1 \in [\mathtt{top}(u), \mathtt{bot}(u)]$. By the definition of $\mathtt{top}(\cdot)$ and $\mathtt{bot}(\cdot)$, for any two items u and v such that $A(u) = A(v)$, the intersection of $[\mathtt{top}(u), \mathtt{bot}(u)]$ and $[\mathtt{top}(v), \mathtt{bot}(v)]$ consists of no more than a single item. It follows that for each of the ℓ possible values of the protected attribute we have exactly one item u with this value for which $k, k + 1 \in [\mathtt{top}(u), \mathtt{bot}(u)]$. Since each such item can be either in the top k or in the bottom $n - k$, the number of possibilities is bounded by 2^{ℓ}.

The dynamic programming based exact algorithm ExactMultiValuedIPF for the **MultiValuedIPF** problem works as follows. The algorithm works in n iterations. For $k = 1, \ldots, n$, and for every subset L_k of the items u for which $k, k + 1 \in [\mathtt{top}(u), \mathtt{bot}(u)]$, it computes the optimal rank of the top k elements of the closest p-fair ranking subject to the constraint that the items in L_k are in the bottom $n - k$. (Note that there may not be a feasible solution for some of these subsets.) The computation is done using the optimal ranking of the top $k - 1$ elements computed for all possible subsets L_{k-1}. It is not difficult to see that each such computation can be implemented in $O(\ell 2^{\ell})$ time, and thus the algorithm is polynomial as long as $\ell = O(\log n)$.

Theorem 4.5. *The running time complexity of* ExactMultiValuedIPF *is linear when ℓ is constant and polynomial when ℓ is $O(\log n)$.*

5 RANK AGGREGATION SUBJECT TO P-FAIRNESS (RAPF)

In this section, we present two scalable solution frameworks for the **RAPF** problem both for binary and multi-valued protected attribute. Algorithm RandAlgRAPF is randomized, highly scalable, but the approximation factor is in expectation. Algorithm AlgRAPF, on the other hand, produces a deterministic approximation factor, but less scalable than its randomized counterpart.

5.1 Randomized Algorithm

We start with the randomized algorithm RandAlgRAPF. Input to this algorithm are m rankings $\rho_1, \rho_2, \ldots, \rho_m$ and the output is σ, the aggregated p-fair ranking. Algorithm RandAlgRAPF randomly selects one of the $\rho_1, \rho_2, \ldots, \rho_m$ input rankings uniformly, denoted $\rho_{randInd}$. Then, Algorithm RandAlgRAPF calls the subroutine AlgIPF(ρ), with the parameter $\rho_{randInd}$. The subroutine AlgIPF(ρ) computes the p-fair ranking that is closest to this selected ranking $\rho_{randInd}$ (either exactly or approximately). The resulting p-fair ranking σ is the output of Algorithm RandAlgRAPF.

The subroutine AlgIPF(ρ) can invoke any of the algorithms described in Section 4. Recall that GrBinaryIPF produces an exact p-fair solution of the binary **IPF** problem. For multi-valued **IPF**, the highly scalable Algorithm ApproxMultiValuedIPF produces a 2 approximation factor, whereas, the dynamic programming based solution Algorithm ExactMultiValuedIPF is more expensive but produces an exact solution for the multi-valued **IPF** problem. Depending on the underlying **IPF** problem, any of these could be used inside AlgIPF.

We prove that *in expectation* the approximation factor of the aggregate ranking returned by this incredibly simple Algorithm RandAlgRAPF is 2+ the approximation factor of the algorithm for the **IPF** problem invoked in AlgIPF(ρ).

Running Time: The running time of Algorithm RandAlgRAPF is the same as the running time of Algorithm AlgIPF.

Theorem 5.1. *Let α be the approximation factor of the algorithm for the* **IPF** *problem invoked in* AlgIPF(ρ). *The expected Kemeny distance of the ranking returned by Algorithm* RandAlgRAPF *to $\rho_1, \rho_2, \ldots, \rho_m$ is bounded by $\alpha + 2$ times the minimum Kemeny distance of any p-fair ranking to $\rho_1, \rho_2, \ldots, \rho_m$.*

Proof. Let OPT_U be the optimal (unconstrained) aggregate ranking of $\rho_1, \rho_2, \ldots, \rho_m$, and let OPT_F be the optimal p-fair aggregate ranking. For $i \in [1..m]$, let σ_i = AlgIPF(ρ_i). Note that for $i \neq randInd$, we do not compute σ_i; it is just used in the proof. We have

$$\kappa(OPT_U, \rho_1, \rho_2, \ldots, \rho_m) \leq \kappa(OPT_F, \rho_1, \rho_2, \ldots, \rho_m).$$

Since for $i \in [1..m]$, $\mathcal{K}(\sigma_i, \rho_i) \leq \alpha \mathcal{K}(OPT_F, \rho_i)$ we also have

$$\sum_{i=1}^{m} \mathcal{K}(\sigma_i, \rho_i) \leq \alpha \kappa(OPT_F, \rho_1, \rho_2, \ldots, \rho_m).$$

By the triangle inequality for any $i \in [1..m]$ we have

$$\begin{aligned} &\kappa(\sigma_i, \rho_1, \rho_2, \ldots, \rho_m) \\ &\leq \sum_{j=1}^{m} \left[\mathcal{K}(\sigma_i, \rho_i) + \mathcal{K}(\rho_i, OPT_U) + \mathcal{K}(OPT_U, \rho_j)\right] \\ &= m\left[\mathcal{K}(\sigma_i, \rho_i) + \mathcal{K}(\rho_i, OPT_U)\right] + \kappa(OPT_U, \rho_1, \rho_2, \ldots, \rho_m) \end{aligned}$$

Summing over all $i \in [1..m]$ we get

$$\sum_{i=1}^{m} \kappa(\sigma_i, \rho_1, \rho_2, \ldots, \rho_m)$$
$$= m \sum_{i=1}^{m} [\mathcal{K}(\sigma_i, \rho_i) + \mathcal{K}(\rho_i, OPT_U)]$$
$$+ m \cdot \kappa(OPT_U, \rho_1, \rho_2, \ldots, \rho_m)$$
$$= m \sum_{i=1}^{m} \mathcal{K}(\sigma_i, \rho_i) + 2m \cdot \kappa(OPT_U, \rho_1, \rho_2, \ldots, \rho_m)$$
$$\leq m \cdot \alpha \cdot \kappa(OPT_F, \rho_1, \rho_2, \ldots, \rho_m) + 2m \cdot \kappa(OPT_U, \rho_1, \rho_2, \ldots, \rho_m)$$
$$\leq m \cdot (\alpha + 2) \cdot \kappa(OPT_F, \rho_1, \rho_2, \ldots, \rho_m)$$

The expected distance is $\frac{1}{m}$ of this sum, that is

$$E\left[\kappa(\sigma_{randInd}, \rho_1, \rho_2, \ldots, \rho_m)\right] \leq (\alpha + 2) \cdot \kappa(OPT_F, \rho_1, \rho_2, \ldots, \rho_m).$$

□

5.2 Deterministic Algorithm

We proceed to describe the deterministic algorithm AlgRAPF. Input to this algorithm are m rankings $\rho_1, \rho_2, \ldots, \rho_m$ and the output is σ, the aggregated p-fair ranking. Algorithm AlgRAPF whose pseudo code is given in Algorithm 3 runs in two steps. (1) Invoke the subroutine AlgIPF(ρ) to solve the **IPF** problem for each of the m input rankings (Line 2); (2) out of the m fair rankings produced in step 1, find the ranking that minimizes the Kemeny distance to the input rankings and output it as the aggregated p-fair ranking (Lines 4–11).

As in the randomized case, the subroutine AlgIPF(ρ) inside AlgRAPF computes an approximation to the closest p-fair of ρ by invoking any of the algorithms described in Section 4. The resulting approximation factor is 2+ the approximation factor of the chosen algorithm.

Algorithm 3 AlgRAPF($\rho_1, \rho_2, \ldots, \rho_m$)

```
1: for i ∈ [1..m] do
2:     σ_i ←AlgIPF(ρ_i)
3: end for
4: min ← m · n^2                ▷ upper bound on the distance
5: for i ∈ [1..m] do
6:     if κ(σ_i, ρ_1, ρ_2, ..., ρ_m) < min then
7:         min ← κ(σ_i, ρ_1, ρ_2, ..., ρ_m)
8:         minInd ← i
9:     end if
10: end for
11: return σ_minInd
```

We demonstrate Algorithm AlgRAPF using the running example. It first calls subroutine AlgIPF to find the p-fair rankings that are closest to the rankings of each of the 4 members. The Kendall-Tau distances between the resulting p-fair rankings and the original rankings are 6, 3, 4, and 9, respectively. Next, Algorithm AlgRAPF outputs the ranking among these 4 p-fair rankings that minimizes the Kemeny distance to original rankings. The output is the one closest to the ranking of member 2, shown below, and its Kemeny distance to the original rankings is 50.

$$[Park, Amy, Molly, Kabir, Abigail, Damien,$$
$$Kim, Aaliyah, Andres, Kiara, Lee, Jazmin]$$

Running Time: The running time of Algorithm AlgRAPF is m times the running time of algorithm AlgIPF plus $O(m^2 n \log n)$. Note that the Kendall-Tau distance between two rankings can be computed in $O(n \log n)$ time using a binary search tree.

Theorem 5.2. *Let α be the approximation factor of the algorithm for the* **IPF** *problem invoked in* AlgIPF(ρ)*. The aggregate ranking returned by Algorithm* AlgRAPF *is an $\alpha + 2$ approximation of the closest p-fair aggregate ranking of $\rho_1, \rho_2, \ldots, \rho_m$.*

Proof. In Theorem 5.1 we proved that

$$\sum_{i=1}^{m} \kappa(\sigma_i, \rho_1, \rho_2, \ldots, \rho_m) \leq m \cdot (\alpha + 2) \cdot \kappa(OPT_F, \rho_1, \rho_2, \ldots, \rho_m).$$

It follows that the minimum distance is bounded by $\frac{1}{m}$ of this sum, and thus

$$\kappa(\sigma_{minInd}, \rho_1, \rho_2, \ldots, \rho_m) \leq (\alpha + 2) \cdot \kappa(OPT_F, \rho_1, \rho_2, \ldots, \rho_m).$$

□

Lemma 5.3. *Algorithms* RandAlgRAPF *and* AlgRAPF *admit* 3*,* 3*, and* 4 *approximation factors for the* **RAPF** *problem when* GrBinaryIPF*,* ExactMultiValuedIPF*, and* ApproxMultiValuedIPF*, respectively, are used as the underlying solutions for the* **IPF** *problem.*

6 EXPERIMENTAL EVALUATIONS

The goal of this study is to evaluate the quality and scalability of our proposed solutions, designed for **IPF** and the **RAPF** problems. We also compare our solutions with multiple state-of-the-art solutions [22, 25] to demonstrate how our studied problems promote stronger notion of fairness for the rank aggregation problem.

All algorithms are implemented in Python 3.8. All experiments are conducted on a cluster server machine with 32GB RAM memory, OS: Scientific Linux release 7.8 (Nitrogen), CPU: Intel(R) Xeon(R) CPU E3-1245 v6 @ 3.70GHz. All numbers are presented as an average of 10 runs. For brevity, we present a subset of results that are representative. The code and the data is available at [1].

6.1 Dataset Description

We perform evaluations considering 3 real world datasets. (a) Fantasy players choose real athletes for their fantasy teams and generate scores based on the athlete's real performance. Rankings of the athletes are provided by real human voters. We use rankings of National Football League (NFL) players for 16 weeks of the 2019 football season from the top 25 experts. (b) German Credit Score: This is a publicly available dataset in the UCI repository. It is based on credit ratings generated by Schufa, a German private credit agency based on a set of variables for each applicant, including age, gender, and marital status, among others. Schufa Score is an essential determinant for every resident in Germany when it comes to evaluating credit rating before getting a phone contract, a long-term apartment rental or almost any loan. We use the credit-worthiness

[1] https://github.com/MouinulIslamNJIT/Rank-Aggregation_Proportionate_Fairness.git

as scores just it is done in [34], and create a protected attribute with 4 different values. (c) MoveLens Dataset: We use MovieLens 25 million movie dataset to select a set of movies that are all rated by the same set of users. The individual user rating is used to create individual ranking. We use the movie genres as the protected attribute. Table 6 has further details.

6.1.1 Synthetic dataset. We generate large scale synthetic data [25, 34] using Mallows' Model [26]. The Kemeny rank aggregation has been shown to be a maximum likelihood estimator for this model [34]. It contains two parameters - (i) θ that controls the degree of consensus among the rankings (higher values shows more agreement); (ii) p that dictates the probability of elements of the first group to be ranked higher than elements in the Second group. We refer to [25] for further details. The θ and p are set to 0.9 and 0.7 respectively in our experiments.

6.2 Implemented Algorithms

DetConstSort [22] is a fairness-aware ranking algorithm designed towards mitigating algorithmic bias for a single rank. DetConstSort only ensures the lower bound of proportionate representation. As shown in Section 3.1, it neither guarantees smallest Kendall-Tau distance nor ensures p-fairness. We implement this for **IPF**.

FairILP [25] finds the closest aggregate ranking that satisfies a bound on the pairwise statistical parity. The original implementation of FairILP is specified for a binary protected attribute. To adapt it for multi-valued protected attribute we ensure that for each value of the protected attribute, the bound on the pairwise statistical parity is satisfied between the items with this value and the rest of the items. In our experiments we set $\delta = 1$ as the (unnormalized) bound on the pairwise statistical parity. We note that due to the definition of pairwise statistical parity, it may be infeasible in many instances to find a solution for $\delta = 0$.

OptIPF is the exact solution for **IPF** produced by solving an Integer Linear Programming (ILP) model using Gurobi Optimizer 9.1. The optimizer does not scale and thus exact solutions cannot be computed for large-scale datasets.

OptRAPF is the exact solution for **RAPF** produced by solving an ILP model using Gurobi Optimizer 9.1. Again, the optimizer only produces the optimal solution on small datasets.

OptRA is the exact solution for rank aggregation without considering fairness, and is produced by solving an ILP model.

Measures. For quality evaluation we use the following measures. (i) Kendall-Tau and Kemeny Distances, (ii) percentage of items satisfying p-fairness, and (iii) approximation factors. For scalability evaluation, we measure the running time.

6.3 Summary of Results

Our first observation is that, consistent with our theoretical analysis, p-fairness promotes stronger notion of fairness, by ensuring proportionate representation of each of the protected attribute values for every position in the ranked order. Naturally, incorporating p-fairness inside rank aggregation comes with a cost - the Kendall-Tau and Kemeny distances are typically higher (albeit not substantially worse) for the p-fair rank aggregation than that of OptRA. Second, our experimental results demonstrate that our proposed model and solutions satisfy the fairness criteria proposed in state-of-the-art solutions [22, 25] - however, these aforementioned existing solutions do not extend to satisfy p-fairness. Third, our experimental results corroborate our theoretical results, that is, GrBinaryIPF is exact, ApproxMultiValuedIPF admits a solution that is no more than twice the optimal for **MultiValuedPF**, and AlgRAPF in conjunction with ApproxMultiValuedIPF admits tighter approximation factor compared to our proposed theoretical bound 4. Finally, our scalability results indicate that our proposed solutions are scalable considering very large number of items $(1,000,000)$ and ranks $(10,000)$. In fact, RandAlgRAPF is insensitive to the number of ranks. We extend our experiments and consider relaxed p-fairness varying $\delta \geq 0$ values as defined in Definition 2.5.

6.4 Quality Experiments

In this section we describe the results of our qualitative analysis.

6.4.1 BinaryIPF Results. Figures 1a and 2a compare the fairness of GrBinaryIPF and DetConstSort. These results clearly indicate that GrBinaryIPF consistently satisfies p-fairness, whereas, DetConstSort does not.

Figure 3a compares the Kendall-Tau distance between the input ranking and the ranking computed by OptIPF, GrBinaryIPF, and DetConstSort. Consistent with our theoretical analysis OptIPF and GrBinaryIPF always produce the same distance. At times DetConstSort computes a ranking with a smaller distance. This can indeed happen, as DetConstSort does not necessarily compute a p-fair ranking.

Figure 4a plots the Kendall-Tau distance of the ranking computed by GrBinaryIPF as we relax the p-fairness using $\delta \geq 0$ values. We note that for a small value of δ the relaxed output is the same as input unfair ranking, and the Kendall-Tau distance is 0.

6.4.2 MultiValuedIPF Results. We use the MovieLens and German Credit Score datasets to demonstrate the effectiveness of our proposed solution ApproxMultiValuedIPF and compare it with DetConstSort. Figures 1b, 1c, 2b, and 2c demonstrate that also in this case ApproxMultiValuedIPF consistently satisfies p-fairness whereas DetConstSort fails to satisfy p-fairness. Figures 3b, 3c compares the Kendall-Tau distance between the input ranking and the ranking computed by ApproxMultiValuedIPF and DetConstSort. Again, at times DetConstSort computes a ranking with a smaller distance since DetConstSort does not necessarily compute a p-fair ranking.

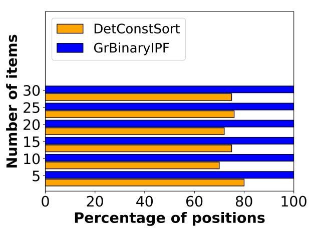
Figures 4b, 4c plot the Kendall-Tau distance of the rankings by ApproxMultiValuedIPF, as we relax the p-fairness using $\delta \geq 0$. Unsurprisingly, for large δ, the Kendall-Tau values become 0.

6.4.3 RAPF Results. Next, we evaluate the **RAPF** problem by studying the effectiveness of our proposed AlgRAPF using GrBinaryIPF (Fantasy football) and ApproxMultiValuedIPF (MovieLens), and compare it with FairILP [25] and OptRAPF, whenever appropriate.

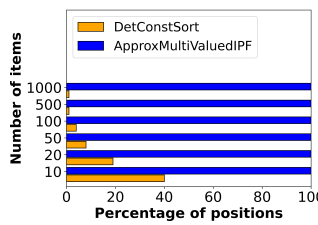
Figures 6a and 6b demonstrate that AlgRAPF consistently satisfies p-fairness whereas FairILP fails to satisfy p-fairness. Figures 7a and 7b compare the Kemeny distance between the input rankings and the aggregate ranking produced by AlgRAPF, RandAlgRAPF, FairILP, and OptRA. As expected OptRA achieves the smallest distance, followed by FairILP, since it does not require p-fairness,

Dataset	#records (n)	# ranks (m)	protected attributes (ℓ)
Fantasy football ranking	55	25	American Football Conference (AFC):**proportion: 50%** , National Football Conference (NFC): **proportion: 50%**
German Credit Score	1000	1	age<35 & sex = female: **proportion: 33.5%** , age≥35 & sex = female: **proportion:** 35.5% age<35 & sex = male: **proportion:** 21.3%, age≥35 & sex = male: **proportion: 9.7%**
MovieLens	268	7	Thriller: **proportion: 2.24%**, Western: **proportion: 6.72%**, Documentary: **proportion: 3.36%**, Comedy: **proportion: 21.64%**, Horror: **proportion: 4.85%**, Musical: **proportion: 0.37%**, Film-Noir: **proportion: 1.49%**, Drama: **proportion: 59.33%**

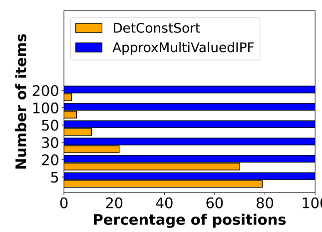
Table 6: Real world datasets



(a) Fantasy football: GrBinaryIPF **vs** DetConstSort

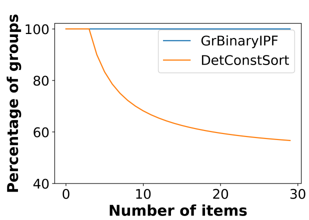


(b) German Credit: ApproxMultiValuedIPF **vs** DetConstSort

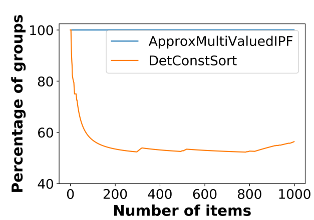


(c) MovieLens: ApproxMultiValuedIPF **vs** DetConstSort

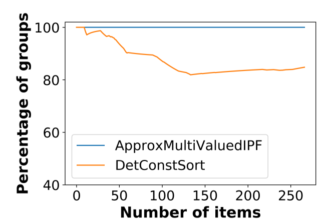
Figure 1: Percentage of positions satisfying p-fairness (IPF)



(a) Fantasy football: GrBinaryIPF **vs** DetConstSort

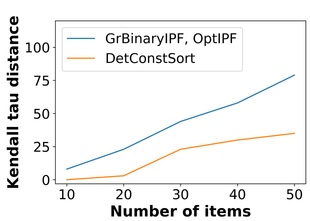


(b) German Credit: ApproxMultiValuedIPF **vs** DetConstSort

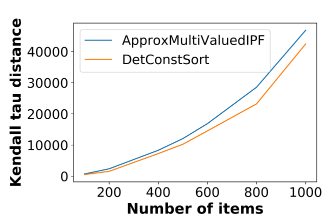


(c) MovieLens: ApproxMultiValuedIPF **vs** DetConstSort

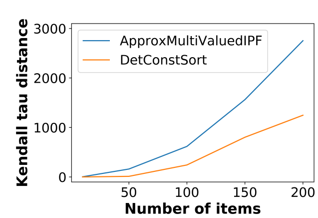
Figure 2: Percentage of groups satisfying p-fairness (IPF)



(a) Fantasy football: OptIPF, GrBinaryIPF , DetConstSort

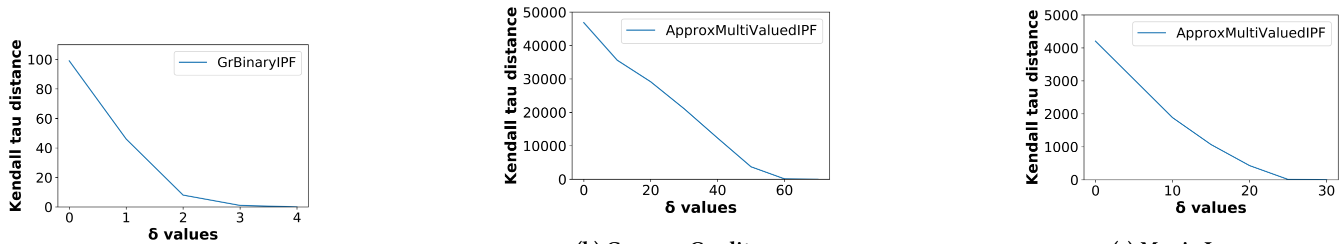


(b) German Credit: ApproxMultiValuedIPF **vs** DetConstSort



(c) MovieLens: ApproxMultiValuedIPF **vs** DetConstSort

Figure 3: Kendall-Tau distance IPF

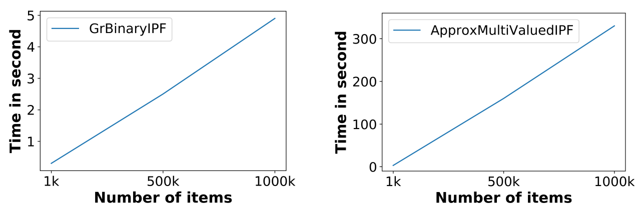


(a) Fantasy Football: GrBinaryIPF

(b) German Credit : ApproxMultiValuedIPF

(c) Movie Lens: ApproxMultiValuedIPF

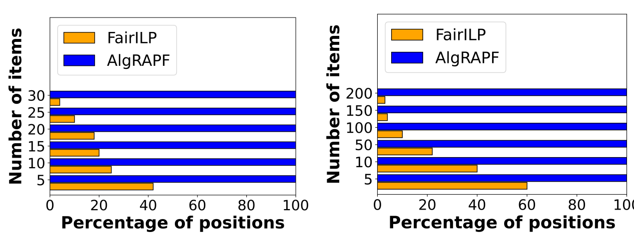
Figure 4: Varying δ analysis IPF



(a) GrBinaryIPF

(b) ApproxMultiValuedIPF

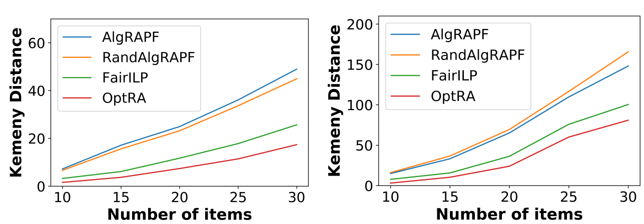
Figure 5: Running time analysis of IPF



(a) Fantasy football: p-fairness: AlgRAPF **vs** FairILP **[25]**

(b) MovieLens: p-fairness: AlgRAPF **vs** FairILP **[25]**

Figure 6: % of positions satisfying p-fairness (RAPF)



(a) Fantasy football: AlgRAPF **vs** RandAlgRAPF**vs** OptRA **vs** FairILP **[25]**

(b) MovieLens: AlgRAPF **vs** RandAlgRAPF**vs** OptRA **vs** FairILP **[25]**

Figure 7: Kemeny Distance RAPF

and then AlgRAPF and RandAlgRAPF. Algorithm RandAlgRAPF is inferior to AlgRAPF in practice, since its performance is same as the latter one only in expectation.

Number of items	10	15	20	25	30
GrBinaryIPF (Football)	1.0	1.0	1.0	1.0	1.0
ApproxMultiValuedIPF (MovieLens)	1.52	1.46	1.37	1.33	1.30
ApproxMultiValuedIPF (Credit Score)	1.8	1.76	1.60	1.57	1.52
AlgRAPF (Football)	2.86	2.76	2.15	2.14	2.01
AlgRAPF (MovieLens)	1.90	1.21	1.18	1.11	1.10
RandAlgRAPF (Football)	2.98	2.77	2.15	2.13	2.06
RandAlgRAPF (MovieLens)	2.10	1.71	1.6	1.70	1.60

Table 7: Approximation factor of the algorithms

Figures 9a and 9b plot the Kemeny distance of the ranking computed by OptRA, AlgRAPF, RandAlgRAPF as we relax the p-fairness using $\delta \geq 0$ values. Unsurprisingly, with large δ, our algorithms become very close to OptRA.

Finally, Table 7 presents the actual approximation factors of the different algorithms proposed in this work. Because of the exponential nature of the OptIPF this comparison could be conducted only on small datasets. As evident from the table the actual approximation factors are lower than the proven theoretical bounds.

6.5 Case Study

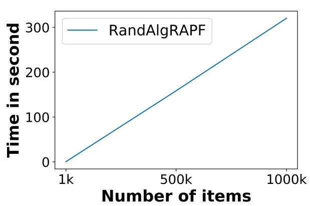
For the case study we use the 10 popular movies based on 5 different IMDB users. All these movies belong to 3 different genres (protected attribute): Drama, Western, Comedy. The proportion of these genres are 0.4, 0.3, and 0.3, respectively. The last two columns of the table 8 show the ranked order of the results based on FairILP [25] and our proposed OptRAPF, respectively. It is easy to notice that compared to FairILP, OptRAPF ranks the movies in a manner where different genres are proportionally distributed in all 10 ranked positions, thereby promoting improved user experience.

6.6 Scalability Experiment

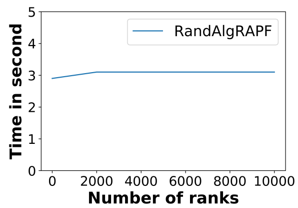
We present the running times of RAPF, RandAlgRAPF, GrBinaryIPF, ApproxMultiValuedIPF. We do not present these results wrt any other baselines because of two reasons: first, we have shown that the baselines DetConstSort [22] and FairILP [25] do not satisfy the p-fairness criteria; second, the baseline algorithm FairILP [25] is inherently not scalable. We use synthetically generated data using Mallows' model for this purpose. We vary n and m. Figures 5, and 8 show these results and demonstrate that our solution easily scale to 1 million items (n) and $10,000$ ranks (m).

Movie	User1	User2	User3	User4	User5	OptRAPF	FairILP	Genre
Bad News Bears, The (1976)	9	7	7	7	4	7	3	Comedy
True Grit (2010)	7	5	1	9	3	9	6	Western
My Darling Clementine (1946)	2	3	3	3	10	4	4	Western
Last Picture Show, The (1971)	4	1	5	1	5	5	1	Drama
Man with the Golden Arm, The (1955)	6	8	4	10	6	8	10	Drama
Heaven Can Wait (1978)	10	10	8	8	8	10	9	Comedy
Rio Bravo (1959)	1	4	6	5	7	1	5	Western
Elephant Man, The (1980)	5	2	2	4	2	6	2	Drama
Buddy Holly Story, The (1978)	3	6	10	6	9	2	8	Drama
Animal House (1978)	8	9	9	2	1	3	7	Comedy

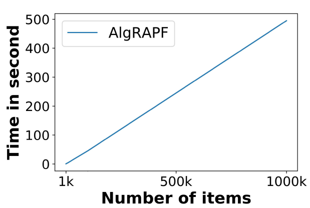
Table 8: Case study results on MovieLens dataset



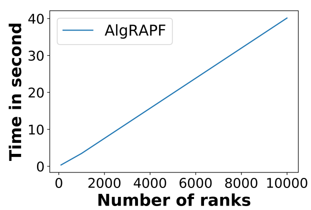
(a) Vary n**,** $m = 100$**:** RandAlgRAPF



(b) Vary m**,** $n = 1000$ **:** RandAlgRAPF

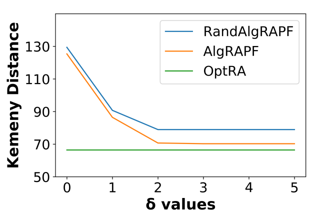


(c) Vary n**,** $m = 100$**:** AlgRAPF

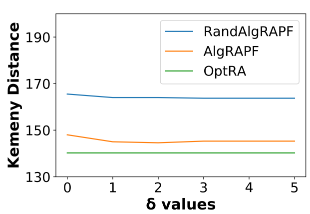


(d) Vary m**,** $n = 1000$ **:** AlgRAPF

Figure 8: Running time analysis



(a) Fantasy Football : AlgRAPF vs RandAlgRAPF varying δ



(b) Movie Lens : AlgRAPF vs RandAlgRAPF varying δ

Figure 9: Varying δ **analysis RAPF**

These results also corroborate our theoretical analysis and shows that the running time of RandAlgRAPF is not dependent on m.

7 CONCLUSION AND FUTURE WORK

We propose the **RAPF** problem to incorporate a group fairness criteria (p-fairness) considering binary and multi-valued protected attributes with the classical rank aggregation problem. We first study how to produce a p-fair ranking that is closest to a single input ranking (**IPF**). **IPF** can be solved exactly using a greedy technique when the protected attribute is binary. When the protected attribute is multi-valued such an approach fails. We then present two solutions for multi-valued **IPF**, ExactMultiValuedIPF is optimal and ApproxMultiValuedIPF admits 2 approximation factor. Next, we design two computational frameworks to solve **RAPF**: RandAlgRAPF and AlgRAPF that exhibit 3 and 4 approximation factors when designed using ExactMultiValuedIPF and ApproxMultiValuedIPF, respectively. The effectiveness of our proposed solutions is demonstrated by comparison to state-of-the-art solutions using multiple real world and large scale synthetic datasets.

Our work opens up several interesting research directions.

A. Alternative models. There exist alternative ways to incorporate p-fairness inside rank aggregation. As an example, one can study the problem of minimizing "weighted" Kemeny distance where the weights are derived considering p-fairness criteria. A slightly different problem is to ensure proportionate fairness not on every position, but for every x (given as input) positions. This problem would be important in applications where every x consecutive individuals in a ranked order are eligible to get the same preferable outcome (such as, top-5% of employees get 100% bonus of their base salary, etc). Studying **RAPF** considering Spearman's Footrule remains part of our ongoing investigation.

B. RAPF for Top-k **or considering incomplete information.** We are studying how to adapt **RAPF** to produce only top-k aggregated rank. This will require us to adapt Kendall-Tau and Kemeny Optimization for top-k results. One possible approach is to consider all items in the individual rank starting at place $k + 1$ as ties, and generalize Kemeny based on ties [2, 20]. We are also interested to study how to obtain an aggregate p-fair ranking when each member inputs only a partial ranking [2, 20].

C. Hardness of IPF. We note that **IPF** essentially finds a perfect matching in a convex bipartite graph while minimizing crossings. The problem of minimizing the number of crossings in a (geometric) bipartite matching is known to be NP-Hard for general bipartite graphs [1]. For convex bipartite graphs, we currently explore if and how existing works that aim at finding a maximum matching without any crossing [14, 27] can adapt to crossing minimization of a perfect matching.

ACKNOWLEDGMENTS

The work of Dong Wei, Md. Mouinul Islam, and Senjuti Basu Roy are supported by the NSF CAREER Award #1942913, IIS #2007935, IIS #1814595, PPoSS:Planning 2118458, and by the Office of Naval Research Grants No: N000141812838, N000142112966.

REFERENCES

[1] Akanksha Agrawal, Grzegorz Guspiel, Jayakrishnan Madathil, Saket Saurabh, and Meirav Zehavi. 2019. Connecting the dots (with minimum crossings). In *35th International Symposium on Computational Geometry (SoCG 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[2] Nir Ailon. 2010. Aggregation of partial rankings, p-ratings and top-m lists. *Algorithmica* 57, 2 (2010), 284–300.

[3] Nir Ailon, Moses Charikar, and Alantha Newman. 2008. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)* 55, 5 (2008), 1–27.

[4] Kenneth J Arrow. 1950. A difficulty in the concept of social welfare. *Journal of political economy* 58, 4 (1950), 328–346.

[5] Abolfazl Asudeh, HV Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing fair ranking schemes. In *Proceedings of the 2019 International Conference on Management of Data*. 1259–1276.

[6] Pieter Cornelis Baayen and Z Hedrlin. 1964. *On the existence of well distributed sequences in compact spaces*. Stichting Mathematisch Centrum. Zuivere Wiskunde.

[7] John Bartholdi, Craig A Tovey, and Michael A Trick. 1989. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and welfare* 6, 2 (1989), 157–165.

[8] Sanjoy K Baruah, Neil K Cohen, C Greg Plaxton, and Donald A Varvel. 1996. Proportionate progress: A notion of fairness in resource allocation. *Algorithmica* 15, 6 (1996), 600–625.

[9] Sanjoy K Baruah, Johannes E Gehrk, C Greg Plaxton, Ion Stoica, Hussein Abdel-Wahab, and Kevin Jeffay. 1997. Fair on-line scheduling of a dynamic set of tasks on a single resource. *Inform. Process. Lett.* 64, 1 (1997), 43–51.

[10] JC de Borda. 1784. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)* (1784).

[11] Bryan Brancotte, Bo Yang, Guillaume Blin, Sarah Cohen-Boulakia, Alain Denise, and Sylvie Hamel. 2015. Rank aggregation with ties: Experiments and analysis. *Proceedings of the VLDB Endowment (PVLDB)* 8, 11 (2015), 1202–1213.

[12] Gerth Stølting Brodal, Loukas Georgiadis, Kristoffer Arnsfelt Hansen, and Irit Katriel. 2007. Dynamic matchings in convex bipartite graphs. In *International Symposium on Mathematical Foundations of Computer Science*. Springer, 406–417.

[13] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. 2017. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840* (2017).

[14] Danny Z Chen, Xiaomin Liu, and Haitao Wang. 2015. Computing maximum non-crossing matching in convex bipartite graphs. *Discrete Applied Mathematics* 187 (2015), 50–60.

[15] Alexandra Chouldechova and Aaron Roth. 2018. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810* (2018).

[16] Vincent Conitzer, Andrew Davenport, and Jayant Kalagnanam. 2006. Improved bounds for computing Kemeny rankings. In *AAAI*, Vol. 6. 620–626.

[17] Persi Diaconis and Ronald L Graham. 1977. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 2 (1977), 262–268.

[18] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[19] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. 2001. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*. 613–622.

[20] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D Sivakumar, and Erik Vee. 2006. Comparing partial rankings. *SIAM Journal of Discrete Mathematics* 20, 3 (2006), 628–648.

[21] Peter C Fishburn. 1970. Arrow's impossibility theorem: Concise proof and infinite voters. *Journal of Economic Theory* 2, 1 (1970), 103–106.

[22] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. 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*. 2221–2231.

[23] John G Kemeny. 1959. Mathematics without numbers. *Daedalus* 88, 4 (1959), 577–591.

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

[25] Caitlin Kuhlman and Elke Rundensteiner. 2020. Rank aggregation algorithms for fair consensus. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2706–2719.

[26] Colin L Mallows. 1957. Non-null ranking models. I. *Biometrika* 44, 1/2 (1957), 114–130.

[27] Federico Malucelli, Thomas Ottmann, and Daniele Pretolani. 1993. Efficient labelling algorithms for the maximum noncrossing matching problem. *Discrete Applied Mathematics* 47, 2 (1993), 175–179.

[28] Christopher McComb, Kosa Goucher-Lambert, and Jonathan Cagan. 2017. Impossible by design? Fairness, strategy, and Arrow's impossibility theorem. *Design Science* 3 (2017).

[29] Iain McLean. 1990. The Borda and Condorcet principles: three medieval applications. *Social Choice and Welfare* 7, 2 (1990), 99–108.

[30] James Tanton. 2005. *Encyclopedia of mathematics*. Facts On File, inc.

[31] Robert Tijdeman. 1980. The chairman assignment problem. *Discrete Mathematics* 32, 3 (1980), 323–330.

[32] Anke Van Zuylen and David P Williamson. 2007. Deterministic algorithms for rank aggregation and other ranking and clustering problems. In *International Workshop on Approximation and Online Algorithms*. Springer, 260–273.

[33] Anke Van Zuylen and David P Williamson. 2009. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research* 34, 3 (2009), 594–620.

[34] Ke Yang and Julia Stoyanovich. 2017. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. 1–6.

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

[36] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1569–1578.