Ranking with Fairness Constraints

L. Elisa Celis
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Damian Straszak
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Nisheeth K. Vishnoi
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract

Ranking algorithms are deployed widely to order a set of items in applications such as search engines, news feeds, and recommendation systems. Recent studies, however, have shown that, left unchecked, the output of ranking algorithms can result in decreased diversity in the type of content presented, promote stereotypes, and polarize opinions. In order to address such issues, we study the following variant of the traditional ranking problem when, in addition, there are *fairness* or *diversity* constraints. Given a collection of items along with 1) the value of placing an item in a particular position in the ranking, 2) the collection of *sensitive* attributes (such as gender, race, political opinion) of each item and 3) a collection of fairness constraints that, for each k, bound the number of items with each attribute that are allowed to appear in the top k positions of the ranking, the goal is to output a ranking that maximizes the value with respect to the original rank quality metric while respecting the constraints. This problem encapsulates various well-studied problems related to bipartite and hypergraph matching as special cases and turns out to be hard to approximate even with simple constraints. Our main technical contributions are fast exact and approximation algorithms along with complementary hardness results that, together, come close to settling the approximability of this constrained ranking maximization problem. Unlike prior work on the approximability of constrained matching problems, our algorithm runs in linear time, even when the number of constraints is (polynomially) large, its approximation ratio does not depend on the number of constraints, and it produces solutions with small constraint violations. Our results rely on insights about the constrained matching problem when the objective function satisfies certain properties that appear in common ranking metrics such as discounted cumulative gain (DCG), Spearman's rho or Bradley-Terry, along with the nested structure of fairness constraints.

2012 ACM Subject Classification Information systems → Retrieval models and ranking, Theory of computation → Approximation algorithms analysis, Theory of computation → Discrete optimization

Keywords and phrases Ranking, Fairness, Optimization, Matching, Approximation Algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.28

Related Version A full version of this paper is available at [13], `https://arxiv.org/abs/1704.06840`.

1 Introduction

Selecting and ranking a subset of data is a fundamental problem in information retrieval and at the core of ubiquitous applications including ordering search results such (e.g., Google), personalized social media feeds (e.g., Facebook, Twitter or Instagram), ecommerce websites (e.g., Amazon or eBay), and online media sites (e.g., Netflix or YouTube). The basic

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 28; pp. 28:1–28:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

algorithmic problem that arises is as follows: There are m *items* (e.g., webpages, images, or documents), and the goal is to output a list of $n \ll m$ items in the order that is most *valuable* to a given user or company. For each item $i \in [m]$ and a position $j \in [n]$ one is given a number W_{ij} that captures the *value* that item i contributes to the ranking if placed at position j. These values can be tailored to a particular query or user and a significant effort has gone into developing models and mechanisms to learn these parameters [31]. In practice there are many ways one could arrive at W_{ij}, each of which results in a slightly different metric for the value of a ranking – prevalent examples include versions of discounted cumulative gain (DCG) [26], Bradley-Terry [6] and Spearman's rho [41]. Note that for many of these metrics, one does not necessarily need nm parameters to specify W and typically m "degrees of freedom" is enough (just specifying the "quality of each item"). Still, we choose to work with this general setting, and only abstract out the most important properties such a weight matrix W satisfies. Generally, for such metrics, W_{ij} is non-increasing in both i and j, and if we interpret $i_1 < i_2$ to mean that i_1 has better *quality* than i_2, then the value of the ranking can only increase by placing i_1 above i_2 in the ranking. Formally, such values satisfy the following property (known as monotonicity and the Monge condition)

$$W_{i_1j_1} \geq W_{i_2j_1} \quad \text{and} \quad W_{i_1j_1} \geq W_{i_1j_2} \quad \text{and} \quad W_{i_1j_1} + W_{i_2j_2} \geq W_{i_1j_2} + W_{j_1i_2} \tag{1}$$

for all $1 \leq i_1 < i_2 \leq m$ and $1 \leq j_1 < j_2 \leq n$. The *ranking maximization* problem is to find an assignment of the items to each of the n positions such that the total value obtained is maximized. In this form, the problem is equivalent to finding the maximum weight matching in a complete $m \times n$ bipartite graph and has a well known solution – the Hungarian algorithm.

However, recent studies have shown that producing rankings in this manner can result in one type of content being overrepresented at the expense of another. This is a form of *algorithmic bias* and can lead to grave societal consequences – from search results that inadvertently promote stereotypes by over/under-representing sensitive attributes such as race and gender [28, 5], to news feeds that can promote extremist ideology [18] and possibly even influence the results of elections [2, 3]. For example, [21] demonstrated that by varying the ranking of a set of news articles the voting preferences of undecided voters can be manipulated. Towards ensuring that no type of content is overrepresented in the context of the ranking problem as defined above, we introduce the *constrained ranking maximization* problem that restricts allowable rankings to those in which no type of content dominates – i.e., *to ensure the rankings are fair.*

Since fairness (and bias) could mean different things in different contexts, rather than fixing one specific notion of fairness, we allow the *user* to specify a set of *fairness constraints*; in other words, we take the constraints as input. As a motivating example, consider the setting in which the set of items consists of m images of computer scientists, each image is associated with several (possibly non-disjoint) sensitive attributes or *properties* such as gender, ethnicity and age, and a subset of size n needs to be selected and ranked. The user can specify an upper-bound $U_{k\ell} \in \mathbb{Z}_{\geq 0}$ on the number of items with property ℓ that are allowed to appear in the top k positions of the ranking, and similarly a lower-bound $L_{k\ell}$. Formally, let $\{1, 2, \ldots, p\}$ be a set of properties and let $P_\ell \subseteq [m]$ be the set of items that have the property ℓ (note that these sets need not be disjoint). Let x be an $m \times n$ binary assignment matrix whose j-th column contains a one in the i-th position if item i is assigned to position j (each position must be assigned to exactly one item and each item can be assigned to at most one position). We say that x satisfies the fairness constraints if for all

$\ell \in [p]$ and $k \in [n]$, we have

$$L_{k\ell} \leq \sum_{1 \leq j \leq k} \sum_{i \in P_\ell} x_{ij} \leq U_{k\ell},$$

If we let $\mathcal{B}$ be the family of all assignment matrices x that satisfy the fairness constraints, the constrained ranking optimization problem is: Given the sets of items with each property $\{P_1, \ldots, P_p\}$, the fairness constraints, $\{L_{k\ell}\}$, $\{U_{k\ell}\}$, and the values $\{W_{ij}\}$, find

$$\arg\max_{x \in \mathcal{B}} \sum_{i \in [m], j \in [n]} W_{ij} x_{ij}.$$

This problem is equivalent to finding a maximum weight matching of size n *that satisfies the given fairness constraints* in a weighted complete $m \times n$ bipartite graph, and now becomes non-trivial – its complexity is the central object of study in this paper.

Beyond the fairness and ethical considerations, traditional *diversification* concerns in information retrieval such as query ambiguity (does "jaguar" refer to the car or the animal?) or user context (does the user want to see webpages, news articles, academic papers or images?) can also be cast in this framework. Towards this, a rich literature on diversifying rankings has emerged in information retrieval. On a high-level, several approaches redefine the objective function to incorporate a notion of diversity and leave the ranking maximization problem unconstrained. E.g., a common approach is to re-weight the w_{ij}s to attempt to capture the amount of diversity item i would introduce at position k conditioned on the items that were placed at positions $1, \ldots, k-1$ (see [7, 47, 17, 46, 48]), or casting it directly as an (unconstrained) multi-objective optimization problem [43]. Alternate approaches mix together or aggregate different rankings, e.g., as generated by different interpretations of a query [36, 20]. Diversity has also been found to be desirable by users [14], and has been observed to arise inherently when the ranking is determined by user upvotes [12]. Despite these efforts and the fact that all major search engines now diversify their results, highly uniform content is often still displayed – e.g., certain image searches can display results that have almost entirely the same attributes [28]. Further, [23] showed that no single diversification function can satisfy a set of natural axioms that one would want any fair ranking to have. In essence, there is a tension between relevance and fairness – if the w_{ij}s for items that have a given property are much higher than the rest, the above approaches cannot correct for overrepresentation. Hence the reason to cast the problem as a constrained optimization problem: The objective is still determined by the values but the solution space is restricted by fairness constraints.

Theoretically, the fairness constraints come with a computational price: The constrained ranking maximization problem can be seen to generalize various **NP**-hard problems such as independent set, hypergraph matching and set packing. Unlike the unconstrained case, even checking if there is a complete feasible ranking (i.e., $\mathcal{B} \neq \emptyset$) is **NP**-hard. As a consequence, in general, we cannot hope to produce a solution that does not violate any constraints. Some variants and generalizations of our problem have been studied in the TCS and optimization literature; here we mention the three most relevant. Note that some may leave empty positions in the ranking as opposed to selecting n elements to rank as we desire. [1] considered the bipartite perfect matching problem with poly(m) constraints. They present a polynomial time randomized algorithm that finds a near-perfect matching which violates each constraint additively by at most $O(\sqrt{m})$. [24] improved the above result to a $(1+\varepsilon)$-approximation algorithm; however, the running time of their algorithm is roughly $m^{K^{2.5}/\varepsilon^2}$ where K is the number of hard constraints and the output is a matching. [42] studied the approximability of the packing integer program problem which, when applied to our setting and gives an $O(\sqrt{m})$ approximation algorithm. In the constrained ranking maximization problem presented above, all of these results seem inadequate; the number of fairness constraints is $2np$ which would make the running time of [24] too large and an additive violation of $O(\sqrt{m})$ would render the upper-bound constraints impotent.

The main technical contributions of this paper are fast, exact and approximation algorithms for this constrained ranking maximization problem along with complementary hardness results which, together, give a solid understanding of the computational complexity of this problem. To overcome the limitations of the past work on constrained matching problems, our results often make use of two structural properties of such a formulation: A) The set of constraints can be broken into p groups; for each property $\ell \in [p]$ we have n (nested) upper-bound constraints, one for each $k \in [n]$, and B) The objective function satisfies the property stated in (1). Using properties A) and B) we obtain efficient – polynomial, or even linear time algorithms for this problem in various interesting regimes. Both these properties are natural in the information retrieval setting and could be useful in other algorithmic contexts involving rankings.

2 Our model

We study the following *constrained ranking maximization problem*

$$\underset{x \in R_{m,n}}{\arg\max} \sum_{i \in [m], j \in [n]} W_{ij} x_{ij} \qquad \text{s.t.} \quad L_{k\ell} \leq \sum_{1 \leq j \leq k} \sum_{i \in P_\ell} x_{ij} \leq U_{k\ell} \qquad \forall\ \ell \in [p], k \in [n], \tag{2}$$

where $R_{m,n}$ is the set of all matrices $\{0,1\}^{m \times n}$ which represent ranking m items into n positions. Recall that W_{ij} represents the profit of placing item i at position j and for every property $\ell \in [p]$ and every position k in the ranking, $L_{k\ell}$ and $U_{k\ell}$ are the lower and upper bound on the number of items having property ℓ that are allowed to be in the top k positions in the ranking. For an example, we refer to Figure 1.

We distinguish two important special cases of the problem: when only the upper-bound constraints are present, and when only the lower-bound constraints are present. These variants are referred to as the *constrained ranking maximization problem (U)* and the *constrained ranking maximization problem (L)* respectively, and to avoid confusion we sometimes add (LU) when talking about the general problem with both types of constraints present. Furthermore, most our results hold under the assumption that the weight function W is monotone and satisfies the Monge property (1), whenever these assumptions are not necessary, we emphasize this fact by saying that *general weights* are allowed.

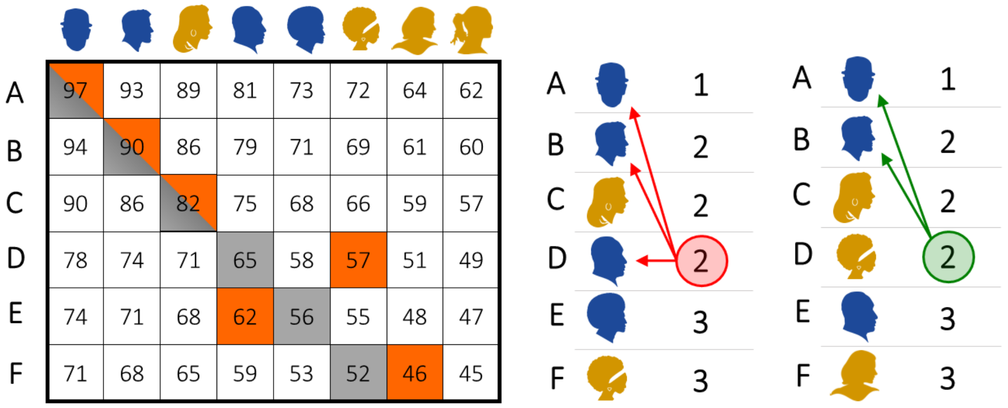
3 Our results

In this section, we present an overview of our results. The statements of theorems here are informal for the ease of readability and for the formal statements our results, we refer the reader to the full version of the paper [13].

Let the *type*

$$T_i := \{\ell \in [p] : i \in P_\ell\}$$

of item i be the set of properties that the item i has. Our first result is an exact algorithm for solving the constrained ranking maximization problem whose running time is polynomial if the number of distinct T_is, denoted by q, is constant.



A	97	93	89	81	73	72	64	62
B	94	90	86	79	71	69	61	60
C	90	86	82	75	68	66	59	57
D	78	74	71	65	58	57	51	49
E	74	71	68	62	56	55	48	47
F	71	68	65	59	53	52	46	45

(a) An example of a value matrix W. The values corresponding to the optimal (unconstrained) ranking in (b) and the optimal constrained ranking in (c) are depicted by gray and orange respectively. (Note that, for clarity of the rank order, the above is the transpose of the matrix referred to as W in the text.)

(b) The optimal unconstrained ranking. The upper-bound constraint at position D is violated as there are 3 men, but only 2 are allowed.

(c) The optimal constrained ranking. The upper-bound constraint at position D is no longer violated; in fact all constraints are satisfied.

Figure 1 A simple example of our framework: In (a) a matrix of W_{ij}s is presented. Here, the options are people who are either male (blue) or female (yellow), and 6 of them must be ranked. We assume that there is a single upper-bound constraint for each position in the ranking which is applied to both genders as depicted in figures (b) and (c). The constraints are satisfied in the latter, but not the former. The weights of these two rankings are depicted in figure (a).

▶ **Theorem 3.1** (Exact dynamic programming-based algorithm). *There is an algorithm that solves the constrained ranking maximization problem (LU) in $O(pqn^q + pm)$ time when the values W satisfy property* (1)*.*

This algorithm combines a geometric interpretation of our problem along with dynamic programming and proceeds by solving a sequence of $q-$dimensional sub-problems. When q is allowed to be large, the problem is **NP**-hard; see Theorem 3.5.

Generally, we may not be able to assume that q is a constant and, even then, it would be desirable to have algorithms whose running time is close to $(m+n)p$, the size of the input. Towards this we consider a natural parameter of the set of properties: The size of the largest T_i, namely

$$\Delta := \max_{i \in [m]} |T_i|.$$

The complexity of the constrained ranking maximization problem turns out to show interesting behavior with respect to Δ (note that $\Delta \leq p$ and typically $p \ll q$). The case when $\Delta = 1$ corresponds to the simplest practical setting where there are p disjoint properties, i.e., the properties partition the set of items. For instance, a set of images of humans could be partitioned based on the ethnicity or age of the individual. Note that even though $q = p$ for $\Delta = 1$, this q could still be large and the previous theorem may have a prohibitively large running time.

When $\Delta = 1$ we prove that the constrained ranking maximization problem (LU) is polynomial time solvable even when the matrix W does not satisfy the property (1).

▶ **Theorem 3.2** (Polynomial time algorithm for $\Delta = 1$). *The constrained ranking maximization problem (LU) for general weights and $\Delta = 1$ can be solved in $\widetilde{O}(n^2 m)$ time.*

The above is obtained by reducing this variant of the ranking maximization problem to the minimum cost flow problem, that can be solved efficiently (the network is acyclic). We note that even though the running time is polynomial in m, it might be still not satisfactory for practical purposes. With the aim of designing faster – linear time algorithms, we focus on the case when only upper-bound constraints are present. For this case, we analyze a natural linear programming (LP) relaxation for the constrained ranking maximization problem (U). It reveals interesting structure of the problem and motivates a fast greedy algorithm. Formally, the relaxation considers the set $\Omega_{m,n}$ defined as

$$\Omega_{m,n} := \left\{ x \in [0,1]^{m\times n} : \sum_{j=1}^{n} x_{ij} \leq 1 \text{ for all } i \in [m], \quad \sum_{i=1}^{m} x_{ij} = 1, \text{ for all } j \in [n] \right\}$$

and the following linear program

$$\max_{x \in \Omega_{m,n}} \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}x_{ij} \qquad \text{s.t.} \qquad \sum_{i\in P_\ell}\sum_{j=1}^{k} x_{ij} \leq U_{k\ell}, \ \forall\, \ell \in [p],\, k \in [n]. \tag{3}$$

Observe that in the absence of fairness constraints, (3) represents the maximum weight bipartite matching problem – it is well known that the feasible region of its fractional relaxation has integral vertices and hence the optimal values of these two coincide. However, in the constrained setting, even for $\Delta = 1$, it can be shown that the feasible region is no longer integral – it can have fractional vertices. For this reason, it is not true that maximizing any linear objective results in an integral solution. Surprisingly, we prove that for $\Delta = 1$ the cost functions we consider are special and never yield optimal fractional (vertex) solutions.

▶ **Theorem 3.3** (Exact LP-based algorithm for $\Delta = 1$). *Consider the linear programming relaxation* (3) *for the constrained ranking maximization problem (U) when* $\Delta = 1$ *and the objective function satisfies* (1)*. Then there exists an optimal solution with integral entries and hence the relaxation is exact. Further, there exists a greedy algorithm to find an optimal integral solution in* $O(np + m)$ *time.*

The proof relies on a combinatorial argument on the structure of tight constraints that crucially uses the assumption that $\Delta = 1$ and the property (1) of the objective function. Note that the result of Theorem 3.3 implies in particular that whenever the linear program (3) is feasible then there is also an integer solution – a feasible ranking. This can be also argued for the general (LU) variant of the problem and its corresponding LP relaxation. However, extending Theorem 3.3 to this case seems more challenging and is left as an open problem.

When trying to design algorithms for larger Δ, the difficulty is that the constrained ranking *feasibility* problem remains **NP**-hard (in fact, even hard to approximate when feasibility is guaranteed) for $\Delta \geq 3$; see Theorem 3.5. Together, these results imply that unless we restrict to feasible instances of the constrained ranking problem, it is impossible to obtain any reasonable approximation algorithm for this problem. In order to bypass this barrier, we focus on the (U) variant of the problem and present an *algorithmically verifiable* condition for feasibility and argue that it is natural in the context of information retrieval. For each $1 \leq k \leq n$, we consider the set

$$S_k := \{l \in [p] : U_{(k-1)\ell} + 1 \leq U_{k\ell}\}$$

of all properties whose constraints increase by at least 1 when going from the $(k-1)$st to the kth position. We observe that the following *abundance of items* condition is sufficient for feasibility:

$$\forall k \text{ there are at least } n \text{ items } i \text{ s.t. } T_i \subseteq S_k. \tag{4}$$

Intuitively, this says that there should be always at least a few ways to extend a feasible ranking of $(k-1)$ items to a ranking of k items. Simple examples show that this condition can be necessary for certain constraints $\{U_{k\ell}\}$. In practice, this assumption is almost never a problem – the available items m (e.g., webpages) far outnumber the size of the ranking n (e.g., number of results displayed in the first page) and the number of properties p (i.e., there are only so many "types" of webpages).

We show that assuming condition (4), there is a linear-time algorithm that achieves an $(\Delta+2)$-approximation, while only slightly violating the upper-bound constraints. This result does not need assumption (1), rather only that the W_{ij}s are non-negative. This result is near-optimal; we provide an $\Omega\left(\frac{\Delta}{\log \Delta}\right)$ hardness of approximation result (see Theorem 3.5 and the full version of the paper [13] for more details).

▶ **Theorem 3.4** (($\Delta+2$)-approximation algorithm)**.** *For the constrained ranking maximization problem (U), under the assumption* (4)*, there is an algorithm that in linear time outputs a ranking x with value at least $\frac{1}{\Delta+2}$ times the optimal one, such that x satisfies the upper-bound constraints with at most a twice multiplicative violation, i.e.,*

$$\sum_{i \in P_\ell} \sum_{j=1}^{k} x_{ij} \leq 2U_{k\ell}, \quad \textit{for all } \ell \in [p] \textit{ and } k \in [n].$$

One can construct artificial instances of the ranking problem, where the output of the algorithm indeed violates upper-bound constraints with a 2-multiplicative factor. However, these violations are caused by the presence of high-utility items with a large number of properties. Such items are unlikely to appear in real-life instances and thus we expect the practical performance of the algorithm to be better than the worst-case bound given in Theorem 3.4 suggests. Lastly we summarize our hardness results for the constrained ranking problem.

▶ **Theorem 3.5** (Hardness Results – Informal)**.** *The following variants of the constrained ranking feasibility (U) and constrained ranking maximization (U) problem are* **NP***-hard.*

1. *Deciding feasibility for the case of $\Delta \geq 3$.*
2. *Under the feasibility condition* (4)*, approximating the optimal value of a ranking within a factor $O\left(\Delta/\log \Delta\right)$, for any $\Delta \geq 3$.*
3. *Deciding feasibility when only the number of items m, number of positions n, and upper-bounds u are given as input; the properties are fixed for every m.*
4. *For every constant c, deciding between whether there exists a feasible solution or every solution violates some constraint by a factor of c.*

Organization of the rest of the paper

In Section 4 we discuss other related work. Section 5 contains an overview of the proofs of our main results. For complete proofs, we refer the reader to the full version of the paper [13]. In Section 6 we provide a discussion of possible directions for future work and open problems.

4 Other related work

Information retrieval, which focuses on selecting and ranking subsets of data, has a rich history in computer science, and is a well-established subfield in and of itself; see, e.g., the foundational work by [39]. The probability ranking principle (PRP) forms the foundation of information retrieval research [32, 38]; in our context it states that *a system's ranking should order items by decreasing value.* Our problem formulation and solutions are in line with this – *subject to satisfying the diversity constraints.*

A related problem is diverse data summarization in which a subset of items with varied properties must be selected from a large set [35, 9], or similarly, voting with diversity constraints in which a subset of items of people with varied properties or attributes must be selected via a voting procedure [34, 10]. However, the formulation of these problem is considerably different as there is no need to produce a ranking of the selected items, and hence the analogous notion of constraints is more relaxed. Extending work on fairness in classification problems [45], the fair ranking problem has also been studied as an (unconstrained) multi-objective optimization problem, and various fairness metrics of a ranking have been proposed [43].

Combining the learning of values along with the ranking of items has also been studied [37, 40]; in each round an algorithm chooses an ordered list of k documents as a function of the estimated values W_{ij} and can receive a click on one of them. These clicks are used to update the estimate of the W_{ij}s, and bounds on the regret (i.e., learning rate) can be given using a bandit framework. In this problem, while there are different types of items that can affect the click probabilities, there are no constraints on how they should be displayed.

Recent work has shown that, in many settings, there are impossibility results that prevent us from attaining both *property* and *item* fairness [30]. Indeed, our work focuses on ensuring property fairness (i.e., no property is overrepresented), however this comes at an expense of item fairness (i.e., depending on which properties an item has, it may have much higher / lower probability of being displayed than another item with the same value). In our motivating application we deal with the ranking of documents or webpages, and hence are satisfied with this trade-off. However, further consideration may be required if, e.g., we wish to rank people as this would give individuals different likelihoods of being near the top of the list based on their properties rather than solely on their value.

5 Proof overviews

Overview of the proof of Theorem 3.1. We first observe that the constrained ranking maximization problem has a simple geometric interpretation. Every item $i \in [m]$ can be assigned a *property vector* $t_i \in \{0,1\}^p$ whose ℓ-th entry is 1 if item i has property ℓ and 0 otherwise. We can then think of the constrained ranking maximization problem as finding a sequence of n distinct items $i_1, i_2, \ldots, i_n$ such that $L_k \leq \sum_{j=1}^{k} t_{i_j} \leq U_k$ for all $k \in [n]$, where U_k is the vector whose ℓ-th entry is $U_{\ell k}$. In other words, we require that the partial sums of the vectors corresponding to the top k items in the ranking stay within the region $[L_{k1}, U_{k1}] \times [L_{k2}, U_{k2}] \times \cdots \times [L_{kn}, U_{kn}]$ defined by the fairness constraints.

Let $Q := \{t_i : i \in [m]\}$ be the set of all the different property vectors t_i that appear for items $i \in [m]$, and let us denote its elements by $v_1, v_2, \ldots, v_q$. A simple but important observation is that whenever two items $i_1, i_2 \in [m]$ (with say $i_1 < i_2$) have the same property vector: $t_{i_1} = t_{i_2}$, then in every optimal solution either i_1 will be ranked above i_2, only i_1 is ranked, or neither is used. This follows from the assumption that the weight matrix is monotone in i and j and satisfies the property as stated in (1).

Let us now define the following sub-problem that asks for the *property vectors* of a feasible solution: Given a tuple $(s_1, s_2, \ldots, s_q) \in \mathbb{N}^q$ such that $k = s_1 + s_2 + \cdots + s_q \leq n$, what is the optimal way to obtain a feasible ranking on k items such that s_j of them have property vector equal to v_j for all $j = 1, 2, \ldots, q$? Given a solution to this sub-problem, using the observation above, it is easy to determine which items should be used for a given property vector, and in what order. Further, one can easily solve such a sub-problem given the solutions to smaller sub-problems (with a smaller sum of s_js), resulting in a dynamic programming algorithm with $O(n^q)$ states and, hence, roughly the same running time.

Overview of the proof of Theorem 3.2. The main idea is to reduce ranking maximization to the minimum cost flow problem and then observe several structural properties of the resulting instance which allow one to solve it efficiently (in $\widetilde{O}(n^2m)$ time).

Given an instance of the constrained ranking maximization problem (U), we construct a weighted flow network $G = (V, E)$ such that every feasible ranking corresponds to a feasible flow of value n in G. Roughly, for every item i and every property ℓ a chain of n vertices is constructed so that placing item i (such that $i \in P_\ell$) at position k corresponds to sending one unit of flow through the chain corresponding to item i up to its kth vertex and then switching to the chain corresponding to property ℓ. Edge weights in these gadgets (chains) are chosen in such a way that the cost of sending a unit through this path is $-W_{i,k}$. The capacities in chains corresponding to properties implement upper-bound constraints. The lower-bound constraints can be also enforced by putting appropriate weights on edges of these chains.

The instance of the minimum cost flow problem we construct has $O(nm)$ vertices and $O(nm)$ edges and is acyclic, which allows to replace the application of the Bellman-Ford algorithm in the first phase of the Successive Shortest Path algorithm by a linear-time procedure. This then easily leads to an implementation in $O(n^2 m \log m)$ time.

Overview of the proof of Theorem 3.3. Unlike the $\Delta = 0$ case where the LP-relaxation (3) has no non-integral vertex (it is the assignment polytope) , even when $p = 1$, fractional vertices can arise (see the full version of the paper [13]). Theorem 3.3 implies that for $\Delta = 1$, although the feasible region of (3) is not integral in all directions, it is along the directions of interest. In the proof we first reduce the problem to the case when $m = n$ (i.e., when one has to rank all of the items) and w has the *strict* form of property (1) (i.e., when the inequalities in assumption (1) are strict). Our strategy then is to prove that for every fractional feasible solution $x \in \Omega_{m,m}$ there is a direction $y \in \mathbb{R}^{m\times m}$ such that the solution $x' := x + \varepsilon y$ is still feasible (for some $\varepsilon > 0$) and its weight is larger than the weight of x. This implies that every optimal solution is necessarily integral.

Combinatorially, the directions we consider correspond to 4-cycles in the underlying complete bipartite graph, such that the weight of the matching can be improved by swapping edges along the cycle. The argument that shows the existence of such a cycle makes use of the special structure of the constraints in this family of instances.

To illustrate the approach, suppose that there exist two items $i_1 < i_2$ that have the same property $\ell \in [p]$, and for some ranking positions $j_1 < j_2$ we have

$$x_{i_1 j_2} > 0 \quad \text{and} \quad x_{i_2 j_1} > 0. \tag{5}$$

Following the strategy outlined above, consider $x' = x + \varepsilon y$ with $y \in \mathbb{R}^{m\times m}$ to be zero everywhere except $y_{i_1 j_1} = y_{i_2 j_2} = 1$ and $y_{i_1 j_2} = y_{i_2 j_1} = -1$. We would like to prove that the weight of x' is larger than the weight of x and that x' is feasible for some (possibly small)

$\varepsilon > 0$. The reason why we gain by moving in the direction of y follows from property (1). Feasibility in turn follows because y is orthogonal to every constraint defining the feasible region. Indeed, the only constraints involving items i_1, i_2 are those corresponding to the property ℓ. Further, every such constraint is of the form[1] $\langle 1_{R_k}, x\rangle \le U_{k\ell}$ where 1_{R_k} is the indicator vector of a rectangle $R_k := P_\ell \times [k]$. Such a rectangle contains either all non-zero entries of y, two non-zero entries (with opposite signs), or none. In any of these cases, $\langle 1_{R_k}, y\rangle = 0$.

Using a reasoning as above, one can show that no configuration of the form (5) can appear in any optimal solution for i_1, i_2 that share a property ℓ. This implies that the support of every optimal solution has a certain structure when restricted to items that have any given property $\ell \in [p]$; this structure allows us to find an improvement direction in case the solution is not integral. To prove integrality we show that for every fractional solution $x \in \mathbb{R}^{m\times m}$ there exists a fractional entry $x_{ij} \in (0,1)$ that can be slightly increased without violating the fairness constraints. Moreover since the i-th row and the j-th column must contain at least one more fractional entry each (since the row- and column-sums are 1), we can construct (as above) a direction y, along which the weight can be increased. The choice of the corresponding entries that should be altered requires some care, as otherwise we might end up violating fairness constraints.

The second part of Theorem 3.3 is an algorithm for solving the constrained ranking maximization problem for $\Delta = 1$ in optimal (in the input size) running time of $O(np + m)$. We show that a natural greedy algorithm can be used. More precisely, one iteratively fills in ranking positions by always selecting the *highest value* item that is still available and does not lead to a constraint violation. An inductive argument based that relies on property 1 and the $\Delta = 1$ assumption gives the correctness of such a procedure.

Overview of the proof of Theorem 3.4. Let $\Delta > 1$ be arbitrary. The most important part of our algorithm is a greedy procedure that finds a large weight solution to a slightly relaxed problem in which not all positions in the ranking have to be occupied. It processes pairs $(i, j) \in [m] \times [n]$ in non-increasing order of weights W_{ij} and puts item i in position j whenever this does not lead to constraint violation.

To analyze the approximation guarantee of this algorithm let us first inspect the combinatorial structure of the feasible set. In total there are $p \cdot n$ fairness constraints in the problem and additionally $m + n$ "matching" constraints, saying that no "column" or "row" can have more than a single one in the solution matrix $x \in \{0,1\}^{m\times n}$. However, after relaxing the problem to the one where not all ranking positions have to be filled, one can observe that the feasible set is just an intersection of $p + 2$ matroids on the common ground set $[m] \times [n]$. Indeed, two of them correspond to the matching constraints, and are partition matroids. The remaining p matroids correspond to properties: for every property ℓ there is a chain of subsets $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_n$ of $[m] \times [n]$ such that

$$\mathcal{I}_\ell = \{S \subseteq [m] \times [n] : |S \cap S_k| \le U_{k\ell} \text{ for all } k = 1, 2, \ldots, n\}$$

is the set of independent sets in this (laminar) matroid. In the work [27] it is shown that the greedy algorithm run on an intersection of K matroids yields K-approximation, hence $(p + 2)$-approximation of our algorithm follows.

[1] By $\langle\cdot,\cdot\rangle$ we denote the inner product between two matrices, i.e., if $x, y \in \mathbb{R}^{m\times n}$ then $\langle x, y\rangle := \sum_{j=1}^{m}\sum_{i=1}^{n} x_{ij}y_{ij}$.

To obtain a better – $(\Delta+2)$-approximation bound, a more careful analysis is required. The proof is based on the fact that, roughly, if a new element is added to a feasible solution S, then at most $\Delta+2$ elements need to be removed from S to make it again feasible. Thus adding greedily one element can cost us absence of $\Delta+2$ other elements of weight at most the one we have added. This idea can be formalized and used to prove the $(\Delta+2)$-approximation of the greedy algorithm. This is akin to the framework of K-extendible systems by [33] in which this greedy procedure can be alternatively analyzed. Finally, we observe that since the problem solved was a relaxation of the original ranking maximization problem, the approximation ratio we obtain with respect to the original problem is still $(\Delta+2)$.

It remains to complete the ranking by filling in any gaps that may have been left by the above procedure. This can be achieved in a greedy manner that only increases the value of the solution, and violates the constraints by at most a multiplicative factor of 2.

Overview of the proof of Theorem 3.5. Our hardness results are based on a general observation that one can encode various types of packing constraints using instances of the constrained ranking maximization (U) and feasibility (U) problem. The first result (part 1. in Theorem 3.5) – **NP**-hardness of the feasibility problem (for $\Delta \geq 3$) is established by a reduction from the hypergraph matching problem. Given an instance of the hypergraph matching problem one can think of its hyperedges as items and its vertices as properties. Degree constraints on vertices can then be encoded by upper-bound constraints on the number of items that have a certain property in the ranking. The inapproximability result (part 2. in Theorem 3.5) is also established by a reduction from the hypergraph matching problem, however in this case one needs to be more careful as the reduction is required to output instances that are feasible.

Our next hardness result (part 3. in Theorem 3.5) illustrates that the difficulty of the constrained ranking optimization problem (U) could be entirely due to the upper-bound numbers $U_{k\ell}$s. In particular, even when the part of the input corresponding to which item has which property is fixed, and only depends on m (and, hence, can be *pre-processed* as in [22]), the problem remains hard. This is proven via a reduction from the independent set problem. The properties consists of all pairs of items $\{i_1, i_2\}$ for $i_1, i_2 \in [m]$. Given any graph $G=(V,E)$ on m vertices, we can set up a constrained ranking problem whose solutions are independent sets in G of a certain size. Since every edge $e=\{i_1,i_2\} \in E$ is a property, we can set a constraint that allows at most one item (vertex) from this property (edge) in the ranking.

Finally, part 4. in Theorem 3.5 states that it is not only hard to decide feasibility but even to find a solution that does not violate any constraint by more than a constant multiplicative factor $c \in \mathbb{N}$. The obstacle in proving such a hardness result is that, typically, even if a given instance is infeasible, it is easy to find a solution that violates *many* constraints by a small amount. To overcome this problem we employ an inapproximability result for the maximum independent set problem by [25] and an idea by [16]. Our reduction (roughly) puts a constraint on every $(c+1)-$clique in the input graph $G=(V,E)$, so that at most one vertex (item) is picked from it. Then a solution that does not violate any constraint by a multiplicative factor more than c corresponds to a set of vertices S such that the induced subgraph $G[S]$ has no c-clique. Such a property allows us to prove (using elementary bounds on Ramsey numbers) that G has a large independent set. Hence, given an algorithm that is able to find a feasible ranking with no more than a c-factor violation of the constraints, we can approximate the maximum size of an independent set in a graph $G=(V,E)$ up to a factor of roughly $|V|^{1-1/c}$; which is hard by [25].

6 Discussion and future work

In this paper, motivated by controlling and alleviating algorithmic bias in information retrieval, we initiate the study of the complexity of a natural constrained optimization problem concerning rankings. Our results indicate that the constrained ranking maximization problem, which is a generalization of the classic bipartite matching problem, shows fine-grained complexity. Both the structure of the constraints and the numbers appearing in upper-bounds play a role in determining its complexity. Moreover, this problem generalizes several hypergraph matching/packing problems. Our algorithmic results bypass the obstacles implicit in the past theory work by leveraging on the structural properties of the constraints and common objective functions from information retrieval. More generally, our results not only contribute to the growing set of algorithms to counter algorithmic bias for fundamental problems [19, 4, 44, 9, 11, 8, 29, 15], the structural insights obtained may find use in other algorithmic settings related to the rather broad scope of ranking problems.

Our work also suggests some open problems and directions. The first question concerns the $\Delta = 1$ case and its (LU) variant; Theorem 3.2 implies that it can be solved in $\widetilde{O}(n^2m)$ time, can this be improved to nearly-linear time, as we do for the (U) variant (Theorem 3.3)? Another question is the complexity of the constrained ranking maximization problem (in all different variants) when $\Delta = 2$ – is it in **P**? The various constants appearing in our approximation algorithms are unlikely to be optimal and improving them remains important. In particular, our approximation algorithm for the case of large Δ in Theorem 3.4 may incur a 2-multiplicative violation of constraints. This could be significant when dealing with instances where the upper bound constraints are rather large (i.e., $U_{k\ell} \gtrsim \frac{k}{2}$) in which case, such a violation effectively erases all the constraints. It is an interesting open problem to understand whether this 2-violation can be avoided, either by providing a different algorithm or by making different assumptions on the instance.

In this work we consider linear objective functions for the the ranking optimization problem, i.e., the objective is an independent sum of profits for individual item placements. While this model might be appropriate in certain settings, there may be cases where one would prefer to measure the quality of a ranking as a whole, and in particular the utility of placing a given item at the kth position should also depend on what items were placed above it [46]. Thus defining and studying a suitable variant of the problem for a class of objectives on rankings that satisfy some version of the diminishing returns principle (submodularity), is of practical interest.

A related question that deserves independent exploration is to study the complexity of sampling a constrained ranking from the probability distribution induced by the objective (rather than outputting the ranking that maximizes its value, output a ranking with probability proportional to its value). Finally, extending our results to the online setting seems like an important technical challenge which is also likely to have practical consequences.

References

1 Sanjeev Arora, Alan M. Frieze, and Haim Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 21–30, 1996. doi:10.1109/SFCS.1996.548460.

2 Drake Baer. The 'Filter Bubble' Explains Why Trump Won and You Didn't See It Coming, November 2016. NY Mag.

3 Eytan Bakshy, Solomon Messing, and Lada A Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132, 2015.

4 S. Barocas and A.D. Selbst. *Big Data's Disparate Impact.* SSRN eLibrary, 2015.

5 Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357, 2016.

6 Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

7 Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

8 L. Elisa Celis, Amit Deshpande, Tarun Kathuria, Damian Straszak, and Nisheeth K. Vishnoi. On the complexity of constrained determinantal point processes. In *APPROX/RANDOM 2017*, pages 36:1–36:22, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.36.

9 L. Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K Vishnoi. How to be fair and diverse? *Fairness, Accountability and Transparency in Machine Learning*, 2016.

10 L. Elisa Celis, Lingxiao Huang, and Nisheeth K. Vishnoi. Multiwinner voting with fairness constraints. In *IJCAI-ECAI*, 2018.

11 L. Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth K. Vishnoi. Fair and diverse DPP-based data summarization. *CoRR*, abs/1802.04023, 2018. arXiv:1802.04023.

12 L. Elisa Celis, Peter M. Krafft, and Nathan Kobe. Sequential voting promotes collective discovery in social recommendation systems. In *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.*, pages 42–51, 2016. URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM16/paper/view/13160.

13 L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. *CoRR*, abs/1704.06840, 2017. arXiv:1704.06840.

14 L. Elisa Celis and Siddartha Tekriwal. What Do Users Want in Q&A Sites: Quality or Diversity? In *International Conference on Computational Social Science (IC2S2)*, 2017.

15 L. Elisa Celis and Nisheeth K. Vishnoi. Fair Personalization. *Fairness, Accountability, and Transparency in Machine Learning*, 2017.

16 Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM J. Comput.*, 35(3):713–728, 2005. doi:10.1137/S0097539700382820.

17 Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008.

18 Matthew Costello, James Hawdon, Thomas Ratliff, and Tyler Grantham. Who views online extremism? Individual attributes leading to exposure. *Computers in Human Behavior*, 63:311–320, 2016.

19 Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *ITCS*, New York, NY, USA, 2012. ACM. doi:10.1145/2090236.2090255.

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

21 Robert Epstein and Ronald E. Robertson. The search engine manipulation effect (SEME) and its possible impact on the outcomes of elections. *Proceedings of the National Academy of Sciences*, 112(33):E4512–E4521, 2015. doi:10.1073/pnas.1419828112.

22 Uriel Feige and Shlomo Jozeph. Universal factor graphs. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 339–350, 2012. doi:10.1007/978-3-642-31594-7_29.

23 Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, pages 381–390. ACM, 2009.

24 Fabrizio Grandoni, R Ravi, Mohit Singh, and Rico Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1-2):525–554, 2014.

25 J. Hastad. Clique is Hard to Approximate Within $n^{1-\varepsilon}$. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96. IEEE Computer Society, 1996.

26 Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

27 T.A. Jenkyns. The Efficacy of the 'greedy' Algorithm. In *Proc. of 7th S-E. Conf. on Combinatorics, Graph Theory and Computing*, pages 341–350, 1976.

28 Matthew Kay, Cynthia Matuszek, and Sean A Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3819–3828. ACM, 2015.

29 Keith Kirkpatrick. Battling algorithmic bias: how do we ensure algorithms treat us fairly? *Communications of the ACM*, 59(10):16–17, 2016.

30 Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *Innovations in Theoretical Computer Science*, 2017.

31 Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*. Cambridge university press Cambridge, 2008.

32 Melvin Earl Maron and John L Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3):216–244, 1960.

33 Julián Mestre. Greedy in approximation algorithms. In *Algorithms - ESA 2006, 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006, Proceedings*, pages 528–539, 2006. doi:10.1007/11841036_48.

34 Burt L Monroe. Fully proportional representation. *American Political Science Review*, 89(4):925–940, 1995.

35 Debmalya Panigrahi, Atish Das Sarma, Gagan Aggarwal, and Andrew Tomkins. Online selection of diverse results. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 263–272. ACM, 2012.

36 Filip Radlinski, Paul N Bennett, Ben Carterette, and Thorsten Joachims. Redundancy, diversity and interdependent document relevance. In *ACM SIGIR Forum*, volume 43, pages 46–52. ACM, 2009.

37 Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International conference on Machine learning*, pages 784–791. ACM, 2008.

38 Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304, 1977.

39 Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988. doi:10.1016/0306-4573(88)90021-0.

40 Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Ranked bandits in metric spaces: learning diverse rankings over large document collections. *Journal of Machine Learning Research*, 14(Feb):399–436, 2013.

41 Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.

42 Aravind Srinivasan. Improved approximations of packing and covering problems. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 268–276, 1995. doi:10.1145/225058.225138.

43 Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management, Chicago, IL, USA, June 27-29, 2017*, pages 22:1–22:6, 2017. doi:10.1145/3085504.3085526.

44 Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna Gummadi. Fairness Constraints: A Mechanism for Fair Classification. In *Fairness, Accountability, and Transparency in Machine Learning*, 2015. URL: http://www.fatml.org/cfp.html.

45 Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings of The 30th International Conference on Machine Learning*, pages 325–333, 2013.

46 Cheng Xiang Zhai, William W Cohen, and John Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 10–17. ACM, 2003.

47 Mi Zhang and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130. ACM, 2008.

48 Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.