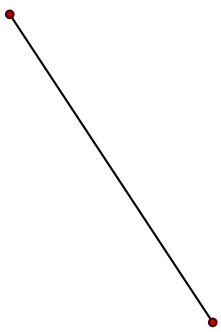
Pareto-Optimal Fairness-Utility Amortizations in Rankings with a DBN Exposure Model

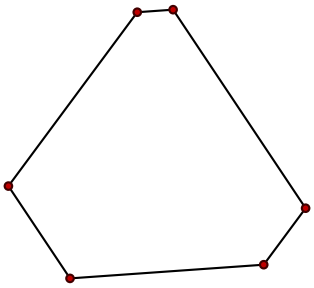
Till Kletti
till.kletti@naverlabs.com
Naver Labs Europe
France

Jean-Michel Renders
jean-michel.renders@naverlabs.com
Naver Labs Europe
France

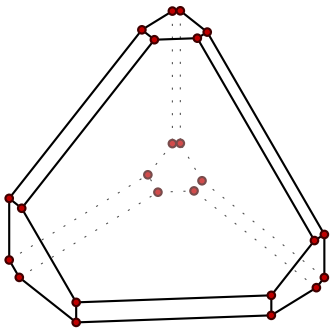
Patrick Loiseau
patrick.loiseau@inria.fr
Univ. Grenoble Alpes, Inria, CNRS,
Grenoble INP, LIG
France



(a) $n = 2$. The DBN-expohedron in $\mathbb{R}^2$ for a relevance vector $\boldsymbol{\rho} = (0.9, 0.1)$



(b) $n = 3$. The DBN-expohedron in $\mathbb{R}^3$ for a relevance vector $\boldsymbol{\rho} = (0.9, 0.5, 0.1)$



(c) $n = 4$. The DBN-expohedron in $\mathbb{R}^2$ for a relevance vector $\boldsymbol{\rho} = (0.9, 0.7, 0.5, 0.1)$

Figure 1: Examples of DBN-expohedra for $\gamma = 0.5, \kappa = 0.7$.

ABSTRACT

In recent years, it has become clear that rankings delivered in many areas need not only be useful to the users but also respect fairness of exposure for the item producers. We consider the problem of finding ranking policies that achieve a Pareto-optimal tradeoff between these two aspects. Several methods were proposed to solve it; for instance a popular one is to use linear programming with a Birkhoff-von Neumann decomposition. These methods, however, are based on a classical Position Based exposure Model (PBM), which assumes independence between the items (hence the exposure only depends on the rank). In many applications, this assumption is unrealistic and the community increasingly moves towards considering other models that include dependences, such as the Dynamic Bayesian Network (DBN) exposure model. For such models, computing (exact) optimal fair ranking policies remains an open question.

In this paper, we answer this question by leveraging a new geometrical method based on the so-called expohedron proposed recently for the PBM (Kletti et al., WSDM'22). We lay out the structure of a new geometrical object (the DBN-expohedron), and propose for it a Carathéodory decomposition algorithm of complexity $O(n^3)$, where n is the number of documents to rank. Such an algorithm enables expressing any feasible expected exposure vector as a distribution over at most n rankings; furthermore we show that we can compute the whole set of Pareto-optimal expected exposure vectors with the same complexity $O(n^3)$. Our work constitutes the first exact algorithm able to efficiently find a Pareto-optimal distribution of rankings. It is applicable to a broad range of fairness notions, including classical notions of meritocratic and demographic fairness. We empirically evaluate our method on the TREC2020 and MSLR datasets and compare it to several baselines in terms of Pareto-optimality and speed.

CCS CONCEPTS

• **Information systems** → **Probabilistic retrieval models**; • **Mathematics of computing** → **Permutations and combinations**;

KEYWORDS

fair ranking, multi-objective optimization, DBN, expohedron, GLS, Carathéodory, Pareto-optimal

ACM Reference Format:
Till Kletti, Jean-Michel Renders, and Patrick Loiseau. 2022. Pareto-Optimal Fairness-Utility Amortizations in Rankings with a DBN Exposure Model. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3477495.3532036

1 INTRODUCTION

Automatic ranking systems take an increasingly large role in our everyday lives, be it as the result of a simple web search, online job markets, or receiving recommendations of new songs to listen

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532036

to. Such systems connect item producers with item consumers and determine which items (webpages, songs, etc.) the consumers are exposed to, and which items remain hidden from sight. While it is generally desirable to give much visibility to items that are relevant to consumers, it is also important to ensure that no item producers are unfairly disadvantaged in terms of exposure.

Fairness has been intensely studied in many applications such as classification [10] or regression [1] tasks. Fairness in ranking, however, presents some particular challenges that do not appear in classification or regression [18, 27]. In particular since highly-ranked items are more visible, and consequently more often clicked than lowly-ranked ones, the commonly adopted policy of ordering the items by decreasing order of relevance, the Probability Ranking Principle (PRP), is the optimal solution for consumer utility (under some assumptions) [21]. On the other hand, the PRP generally yields exposures (i.e., visibilities) that are not adequately related to the items' *merit*, which is a type of unfairness we wish to avoid. Such an "adequate" relation between exposure and merit is called *fairness of exposure* [23]. In this paper our ambition is not to decide which exposures are fair, but to develop methods making it possible to achieve any feasible fair exposure. We will illustrate this on a commonly used notion of fairness, namely *meritocratic* fairness, for which the merit of an item is proportionate to its relevance [15], but it also applies on other notions such as demographic fairness.

Methods able to mitigate unfairness in rankings include *pre-processing*, the treatment of data before input into a ranking component; *in-processing*, the training of fair ranking models; and *post-processing*, a step that involves re-ranking a certain set of n items, given a query. Within the post-processing step it is possible to deliver different rankings for the same query, a process called *amortization*, because the unfairness of the initial ranking can be compensated in subsequent ones. This is usually done by finding an optimal distribution over rankings, from which rankings are sampled.

Most approaches aiming to find such an optimal distribution consider a Position Based Model (PBM), for which the exposure given to an item is assumed to only depend on its rank. For instance it is possible to use bistochastic matrices to represent distributions over permutations and optimize utility with a fairness constraint as a linear program [23]. Conversely it is possible to optimize fairness under a utility constraint as a linear program [3]. Morik et al. proposed to use a controller to make sure that the exposure is proportional to a notion of merit [15]. More recently a novel geometric method using a polytope called *expohedron* was proposed [13]. All these methods rely on using a PBM as exposure model. In a PBM, however, the items are assumed *independent*, such that the exposure of an item depends only on its rank. This is not a realistic assumption because the users, while browsing the list of results, may be influenced by the documents they see to continue or abandon the browsing.

In our work we consider a different range of exposure models encompassed by the Dynamic Bayesian Network (DBN) model [6], an exposure model that takes into account the relevance values from previously ranked items. As such the DBN, and its variants the Cascade Model (CM), Discrete Choice Model (DCM), Simplified Dynamic Bayesian Network (SDBN) and Click Chain Model (CCM) [7], are much more realistic browsing models. Yet they have received much less attention from the fair ranking community, in part because these models are more complex than simple PBMs. Indeed, using a DBN removes linearity from the problem, thereby making it impossible to solve with linear programming approaches such as [23]. The DBN has been the subject of recent interest, in particular in the TREC2020 Fair Ranking track [2] and in a paper proposing a controller [25] for it; but to date there is no method that provably computes an optimal distribution over rankings.

In this paper we draw upon the recent approach using the so-called *expohedron*, proposed by [13] for a PBM exposure model, to work out a method able to find distributions over rankings with optimal expected exposures for a DBN exposure model. As [13] strongly relies on the particular (simple) structure of the PBM, this entails a number of challenges, in particular to characterize theoretically the geometrical structure of the new DBN-*expohedron* that we obtain, illustrated in Figure 1. Then, however, it makes it possible, for any feasible target exposure, to find a distribution over rankings whose expected exposure is exactly the target exposure. Furthermore, it also enables to adapt the procedure in [13] to find the Pareto-front of a multi-objective fairness-utility optimization problem.

Our contributions can be summarized as follows:

(1) Drawing upon the ideas of [13], we define a new expohedron called *DBN-expohedron* based on the DBN model and we provide theoretical results concerning the geometry of this new more complicated structure.
(2) Using the above results, we devise an algorithm capable of finding the Carathéodory decomposition of any point inside a DBN-expohedron. This makes it possible to express any feasible target exposure as the expected exposure of a distribution over rankings. This algorithm has complexity $O(n^3)$, where n is the number of items to rank.
(3) Using the above results we give an algorithm capable of finding the Pareto-front of a Multi-Objective Optimisation (MOO) problem invovling a fairness objective and a utility objective. This algorithm has complexity $O(n^3)$, where n is the number of items to rank.
(4) We perform experiments on the TREC2020 Fairness Track dataset [2] and on the MSLR dataset [20], and compare our approach in terms of Pareto-optimality and speed with respect to 3 baselines: A Plackett-Luce (PL) policy [8], a heuristic controller [25] and a Learning to Rank (LTR) method [17].

2 RELATED WORK

In this section we review closely related work and provide some detailed background that will be used throughout the paper.

Learning Plackett-Luce policies. In a recent paper [17], Oosterhuis proposed a Learning to Rank (LTR) approach using Plackett-Luce (PL) [19] distributions as ranking policy. More specifically, the rankings are generated by a Plackett-Luce (PL) distribution whose log-scores are given by a trained model using features associated to the (query,document) pairs as inputs.

The loss function used to train this model by policy gradient can be any convex combination of utility and fairness, which makes it possible to choose a loss function that is a scalarization of a multi-objective optimization problem. In their paper, [17] consider a PBM for the exposure, but the approach can be adapted to a DBN, without great difficulty.

Optimization over bistochastic matrices. A popular method to find optimal distributions over rankings is to represent them as bistochastic matrices [23, 24, 26]. The so-called Birkhoff-von Neumann (BvN) algorithm can express any bistochastic matrix as the expected value of a distribution over permutations matrices [9]. When a PBM is used, this allows to run a linear program to find an optimal bistochastic matrix, because with a PBM, the exposure of a ranking is a linear function of the bistochastic matrix. When a DBN exposure model is used, such a linear relationship does not exists and using bistochastic matrices no longer makes sense. Indeed the elements of a bistochastic matrix represent the marginal probabilities for the items of ending up at a certain rank; but as we will show later (see (2)), this marginal information does not suffice to deduce the expected exposure of an item. Therefore another method is needed.

Expohedron. Recently, Kletti et al. [13] introduced a polytope called *expohedron*, whose vertices represent all exposure vectors attainable with one ranking with a PBM and whose interior is the convex hull of the vertices. The points of this polytope represent all expected exposure vectors of distributions over rankings. The authors propose an algorithm capable of expressing any point inside the expohedron as the expected exposure of a distribution over permutations; this is called a *Carathéodory decomposition*. Furthermore they propose an algorithm able to find the set of all Pareto-optimal exposure vectors in the expohedron. Both these algorithms have complexity $O(n^2 \log(n))$, where n is the number of items to rank. However their approach strongly relies on the particular structure of PBM, which does not naturally extend to the DBN exposure model.

Controller. In [2020], Thonet and Renders [25] proposed a heuristic controller that starts by delivering a PRP ranking, looks at how much each item is advantaged or disadvantaged w.r.t. its merit and corrects this in the subsequent ranking by artificially increasing or decreasing the relevance values of the items and delivering the next ranking by ordering the items according to the corrected relevance values. This method is compatible with a DBN exposure method and will be used as a baseline.

3 MODEL

Setting. We assume a single or anonymous user that issues a query q many times and we are not preoccupied about being fair towards users, but only to the provider side. We suppose that for such a query q we are provided n items and estimations of their relevance value w.r.t. the query, between 0 (irrelevant) and 1 (relevant). We suppose that those relevance estimations are unbiased and represent the true relevances. This is obviously an unrealistic assumption, but it has the merit of enabling us to study the amortization problem independently of the problem of fairly estimating relevance values. Given a query q, our task is to deliver a sequence of rankings that is optimal in terms of utility provided to the user and in terms of fairness of item exposure. The remainder of this section is dedicated to the formal definitions of ranking, exposure, user-side utility, item-side fairness and to the formal introduction of our optimization problem.

Ranking. We formalize a ranking as a permutation $\pi \in \mathcal{S}_n$ of n items such that if item i is at rank k we have $i = \pi(k)$. Here $\mathcal{S}_n$ represents the set of permutations of size n. Given a relevance vector $\boldsymbol{\rho} \in \mathbb{R}^n$, we denote π_{PRP} an arbitrary ranking such that

$$\boldsymbol{\rho}_{\pi_{\text{PRP}}(1)} \geq \cdots \geq \boldsymbol{\rho}_{\pi_{\text{PRP}}(n)}. \tag{1}$$

A ranking π_{PRP} orders the items by decreasing relevance values.

Exposure model. We use the generic Dynamic Bayesian Network (DBN) exposure model [6]. Unlike with the simpler Position Based Model (PBM) used in [13, 17, 23], the exposures of a DBN depend on the actual relevances of the items in the ranking, thereby making it a more realistic model. The DBN model has two parameters: $\gamma \in [0, 1]$ (the continuation probability) and $\kappa \in [0, 1]$ (the satisfaction probability of a relevant item). Furthermore it depends on the relevance vector $\boldsymbol{\rho} \in \mathbb{R}^n$. Formally the expression of the exposure $\boldsymbol{\mathcal{E}}_i$ of an item i for a ranking $\pi \in \mathcal{S}_n$ is

$$\boldsymbol{\mathcal{E}}_i(\pi, \gamma, \kappa, \boldsymbol{\rho}) := \gamma^{\pi^{-1}(i)-1} \prod_{l=1}^{\pi^{-1}(i)-1} \left(1 - \kappa \boldsymbol{\rho}_{\pi(l)}\right). \tag{2}$$

Here $\pi^{-1}(i)$ is the rank of item i. Equation (2) has an intuitive interpretation: the user runs through the list and at each rank l, the user has a probability $1 - \gamma$ of simply abandoning his search and a probability $\kappa \boldsymbol{\rho}_{\pi(l)}$ of being satisfied with the item $\pi(l)$ and as a result stopping their search. In this paper we often omit the heavy notation involving $\gamma, \kappa, \boldsymbol{\rho}$ and write the exposure vector simply as $\boldsymbol{\mathcal{E}}(\pi) = (\boldsymbol{\mathcal{E}}_1(\pi), \ldots, \boldsymbol{\mathcal{E}}_n(\pi))^\top$.

The DBN model contains as particular cases many other click models, such as the Cascade Model, the Simplified Dynamic Bayesian Network model, the Discrete Choice Model and the Click Chain Model [7], see Appendix B.

Given a distribution over rankings $\mathcal{D}$, the exposure vector of $\mathcal{D}$ is the expectation

$$\boldsymbol{\mathcal{E}}(\mathcal{D}) := \mathbb{E}_{\pi \sim \mathcal{D}}[\boldsymbol{\mathcal{E}}(\pi)]. \tag{3}$$

User side utility. We define the utility of a ranking as the scalar product of exposure with relevance, as this corresponds to the Expected Reciprocal Rank (ERR) metric [5] that was developed jointly with the DBN model [6]. Formally we denote

$$\mathrm{U}(\pi) := \boldsymbol{\rho}^\top \boldsymbol{\mathcal{E}}(\pi), \tag{4}$$

where $\cdot^\top$ is the transpose operator. Given a distribution over rankings $\mathcal{D}$, the utility of $\mathcal{D}$ is

$$\mathrm{U}(\mathcal{D}) = \mathbb{E}_{\pi \sim \mathcal{D}}[\mathrm{U}(\pi)]. \tag{5}$$

In particular the utility is a linear function of the exposure vector. In fact this definition is analogous to the popular DCG metric [12] with the difference that a PBM exposure is used for the DCG. Similarly to the relationship that DCG bears with NDCG, we define the *normalized utility* and denote nU the utility divided by the utility obtained with a PRP ranking:

$$\mathrm{nU}(\pi) := \frac{\mathrm{U}(\pi)}{\mathrm{U}(\pi_{\text{PRP}})}. \tag{6}$$

This gives an adimensional metric that is useful to be able to aggregate utilities across different queries in a meaningful way.

(Un)fairness. We assume that a decision-maker has decided upon a vector of merits $\boldsymbol{\mu} \in \mathbb{R}^n$ with non-negative components. The fact that the vector of merits is a free parameter makes our setting very flexible in terms of covered fairness notions. This vector of merits can be set to be equal to the relevance vector so as to get a notion of *meritocratic* fairness, or it can be set to $\mathbf{1}$, so as to get a notion of *demographic* fairness. An exposure vector $\boldsymbol{\mathcal{E}}$ is said to be fair if it is proportional to the vector of merits $\boldsymbol{\mu}$, i.e., if there exists a positive real number $k \in \mathbb{R}_+$ such that $\boldsymbol{\mathcal{E}} = k\boldsymbol{\mu}$. As we will see in Section 4, there exists at most one exposure vector proportional to $\boldsymbol{\mu}$ such that it is achievable by a distribution over rankings, i.e., such that there exists a distribution over rankings $\mathcal{D}$ with $\boldsymbol{\mathcal{E}}(\mathcal{D}) \propto \boldsymbol{\mu}$. When such a feasible vector exists, we denote it by $\boldsymbol{\mathcal{E}}^*$ and call it the *target exposure*. When such a vector does not exist, we relax the proportionality relationship into an affine one, by adding a constant value to every element of $\boldsymbol{\mu}$ until the corresponding vector is feasible, i.e., we use the merit vector $\boldsymbol{\mu}' = \boldsymbol{\mu} + K\mathbf{1}$ for minimal K, and we define the target exposure as $\boldsymbol{\mathcal{E}}^* \propto \boldsymbol{\mu}'$. This definition of target exposure is to a certain degree arbitrary and we acknowledge that other definitions are possible such as projecting $\boldsymbol{\mu}$ on the expohedron with a euclidean distance.

We measure the unfairness of a distribution over rankings as the euclidean distance of its expected exposure to the target exposure:

$$\mathrm{F}(\mathcal{D}) = \left\| \mathbb{E}_{\pi\sim\mathcal{D}}[\boldsymbol{\mathcal{E}}(\pi)] - \boldsymbol{\mathcal{E}}^* \right\|_2. \tag{7}$$

As for the normalized utility, we define the *normalized unfairness* as the unfairness divided by the unfairness obtained with a PRP ranking:

$$\mathrm{nF}(\mathcal{D}) \coloneqq \frac{\|\mathbb{E}_{\pi\sim\mathcal{D}}[\boldsymbol{\mathcal{E}}(\pi)] - \boldsymbol{\mathcal{E}}^*\|_2}{\|\boldsymbol{\mathcal{E}}_{\mathrm{PRP}} - \boldsymbol{\mathcal{E}}^*\|_2}. \tag{8}$$

This normalization has the advantage of giving a metric with values between 0 and 1 (for Pareto-optimal distributions) independently of the number of documents n Equation (8) assumes that the target exposure is always different from the exposure obtained with a PRP ranking. This is not a very restrictive assumption, because when the two exposure vectors coincide, the problem becomes trivial, as it suffices to deliver a PRP ranking to get both maximal utility and minimal unfairness.

Optimization problem. The normalized unfairness and the normalized utility are the two objectives of the Multi-Objective Optimisation (MOO) problem:

$$\max_{\mathcal{D}} \mathrm{nU}(\mathcal{D}), \quad \min_{\mathcal{D}} \mathrm{nF}(\mathcal{D}). \tag{9}$$

Our goal is to find a set of distributions $\mathcal{D}$ of rankings that is Pareto-optimal for the two objectives of (9). Note that in (5) and (8), nF and nU depend on $\mathcal{D}$ only through the expected value $\boldsymbol{\mathcal{E}}(\mathcal{D})$. Therefore it is possible to decompose the MOO (9) into two subproblems:

(1) Finding all Pareto optimal vectors $\boldsymbol{\mathcal{E}} \in \mathbb{R}^n$ that are the expectation of some distribution $\mathcal{D}$,
(2) Given a Pareto-optimal exposure vector $\boldsymbol{\mathcal{E}} \in \mathbb{R}^n$, finding a distribution $\mathcal{D}$ such that $\boldsymbol{\mathcal{E}} = \mathbb{E}_{\pi\sim\mathcal{D}}[\boldsymbol{\mathcal{E}}(\pi)]$.

In the remainder of the paper we draw on the ideas developed by Kletti et al. [13], see Section 2. We first derive several properties of the so-called expohedron resulting from a DBN exposure model. We show that these properties make it possible to design a Carathéodory decomposition algorithm in the DBN-expohedron, thereby making it possible to find a distribution $\mathcal{D}$ such that $\boldsymbol{\mathcal{E}}(\mathcal{D}) = \boldsymbol{\mathcal{E}}$ for any $\boldsymbol{\mathcal{E}}$ for which such a distribution exists. Then we show that using these properties we can recover the whole set of Pareto-optimal exposure vectors for the MOO (9). Both these algorithms have complexity $O(n^3)$, where n is the number of documents to be ranked.

4 THE DBN-EXPOHEDRON

In this section we formally define the DBN-expohedron and provide several properties about its geometry. In previous work [13], the expohedron has been defined as the convex hull of the exposure vectors achieved with a ranking, using a PBM. In this paper we use the same definition, but using a DBN exposure model instead of a PBM.

The DBN-expohedron. Given a DBN exposure model parametrized by $\gamma \in [0, 1)$, $\kappa \in [0, 1]$ and given an exposure vector $\boldsymbol{\rho} \in [0, 1]^n$, we define the DBN-expohedron as

$$\Pi(\gamma, \kappa, \boldsymbol{\rho}) \coloneqq \mathrm{Conv}\left(\left\{\boldsymbol{\mathcal{E}}(\pi, \gamma, \kappa, \boldsymbol{\rho}) \mid \pi \in \mathcal{S}_n\right\}\right). \tag{10}$$

The DBN-expohedron is the convex hull of all exposure vectors $\boldsymbol{\mathcal{E}}(\pi)$ with $\pi \in \mathcal{S}_n$ and $\mathcal{S}_n$ is the set of permutations of size n. As such it contains exactly those exposure vectors that are expected values of distributions over rankings. This is why we say that vectors inside the expohedron are *feasible*.

Properties. A first important property is that the DBN-expohedron is actually contained in a hyperplane of $\mathbb{R}^n$, i.e., it is an object of dimension $n - 1$.

Proposition 1. *The polytope $\Pi(\gamma, \kappa, \boldsymbol{\rho})$ is contained in a hyperplane with normal vector $\boldsymbol{\nu} = \mathbf{1} + \frac{\gamma\kappa}{1-\gamma}\boldsymbol{\rho}$.*

See proof in Appendix (page 11).

From Proposition 1, it follows that for every vector of merits $\boldsymbol{\mu}$, there exists at most one feasible expected exposure $\boldsymbol{\mathcal{E}}^*$ vector such that $\boldsymbol{\mathcal{E}}^* \propto \boldsymbol{\mu}$, because the arrow in the direction $\boldsymbol{\mu}$ intersects the hyperplane at most once.

Remark 1. The fact that the normal vector is not $\mathbf{1}$, as would be the case for a PBM expohedron [13], implies that minimizing $\|\boldsymbol{\mathcal{E}}\|$, as in [8], does not lead to equal exposures, as is illustrated in Figure 2.

Before further characterizing the faces of our expohedron, let us recall the definition of *zone* $Z(\pi)$ from [13] as the subset of $\mathbb{R}^n$ of vectors $\mathbf{x}$ such that

$$\mathbf{x}_{\pi(1)} \geq \ldots \geq \mathbf{x}_{\pi(n)}. \tag{11}$$

The following proposition gives a characterization of the faces of the DBN-expohedron that is similar to the characterization in [13]. However the normal vectors to the faces take a very different expression with a DBN model.

Proposition 2. *Every face F of $\Pi(\gamma, \kappa, \boldsymbol{\rho})$ is characterized by*

(1) A zone $Z(\pi)$ with $\pi \in \mathcal{S}_n$,
(2) A subset S of $\{1, \ldots, n\}$ called set of splits.

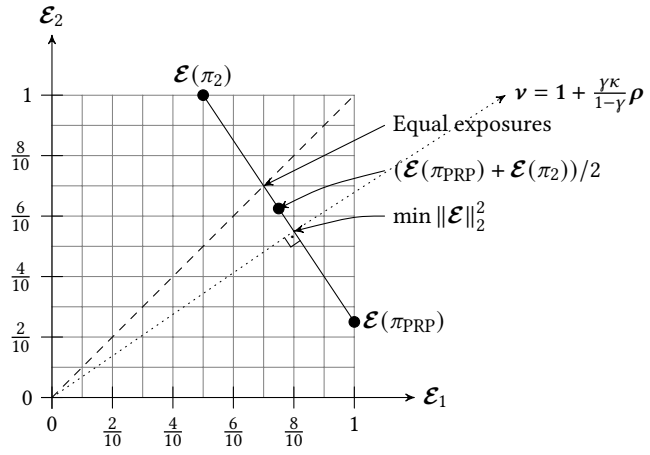


Figure 2: A DBN-expohedron for $n = 2$ documents of relevances $\boldsymbol{\rho} = (1, 0)$ with parameters $\gamma = \kappa = 0.5$. It is apparent that minimizing $\|\mathcal{E}\|_2^2$ does not lead to any interpretable notion of equity.

We denote a face by $F = (\pi, S)$. In the basis in which π is the identity, the $|S|$ normal vectors of a face are given by

$$\boldsymbol{\nu}_s = \mathbf{1}(s) + \frac{\gamma\kappa}{1-\gamma}(\boldsymbol{\rho} \odot \mathbf{1}(s)), \quad \forall s \in S, \tag{12}$$

where

$$\mathbf{1}(s)_i = \begin{cases} 1, \text{ if } i \leq s, \\ 0, \text{ else}, \end{cases} \tag{13}$$

and where $\odot$ is the element-wise product. Furthermore the dimension of a face $F = (\pi, S)$ is $\dim(F) = n - |S|$.

See proof in Appendix (page 11).

With Proposition 2 it becomes possible to check whether an arbitrary point $\mathbf{x} \in \mathbb{R}^n$ is inside $\Pi(\gamma, \kappa, \boldsymbol{\rho})$ using Algorithm 1. This non-trivial result is necessary in Section 5 to design our Carathéodory decomposition and to find the Pareto-front of our MOO problem.

Algorithm 1 Check whether a point $\mathbf{x} \in \mathbb{R}^n$ is inside $\Pi(\gamma, \kappa, \boldsymbol{\rho})$

```
procedure is_inside(Input: x ∈ ℝ^n, Π(γ, κ, ρ))
    π ← argsort(x)                          ▷ Identify the zone of x
    v ← E(π)                                ▷ Create the zone's vertex
    is_inside ← True
    if ν^⊤(x − v) ≠ 0 then
        is_inside ← False
    end if
    for s ∈ {1, ..., n − 1} do
        if ν_s^⊤(x − v) > 0 then
            is_inside ← False
        end if
    end for
end procedure Output: is_inside
```

Lemma 1. *Algorithm 1 returns* `True` *if and only if $\mathbf{x} \in \Pi(\gamma, \kappa, \boldsymbol{\rho})$. It operates with complexity $O(n^2)$.*

See proof in Appendix (page 11).

With Proposition 2 it also becomes possible to identify the lowest-dimensional face a point of the expohedron is contained in, using Algorithm 2.

Algorithm 2 Identify the smallest face in which a point is contained

```
procedure face_id(Input: x ∈ Π(γ, κ, ρ))
    π ← argsort(x)                          ▷ Identify the zone of x
    v ← E(π)                                ▷ Create the zone's vertex
    S ← ∅
    for s ∈ {1, ..., n} do
        if ν_s^⊤(v − x) = 0 then
            S ← S ∪ {s}
        end if
    end for
end procedure Output: The face F = (π, S)
```

Lemma 2. *Algorithm 2 finds the smallest face in which a point $\mathbf{x} \in \Pi(\gamma, \kappa, \boldsymbol{\rho})$ is contained. It operates with complexity $O(n^2)$.*

See proof in Appendix (page 11).

Furthermore with Proposition 2 it becomes possible to project vectors on the lowest-dimensional subspace containing a certain face $F = (\pi, S)$, since all normal vectors are known. This can be done by building an orthonormal projection matrix from the set $\{\boldsymbol{\nu}_s \mid s \in S\}$ using for instance the Gram-Schmidt orthonormalization process.

Thanks to these results, we can design a Carathéodory decomposition algorithm in the expohedron and find the set of Pareto-optimal solutions to (9), as we will see in Section 5.

5 OUR PROPOSED ALGORITHMS

5.1 Carathéodory decomposition

In order to be able to express any feasible target exposure as the expected exposure of a distribution over rankings, we need a method able to express any point inside the DBN-expohedron as a convex combination of its vertices. Such a combination that uses at most n vertices is called a *Carathéodory decomposition*, after a famous theorem of Carathéodory [4, 16].

In this section we expose such a method by showing how one can adapt the generic GLS method, named after Grötschel, Lovász and Schrijver [11] to the DBN-expohedron. The GLS method is a generic Carathéodory decomposition algorithm that needs to be adapted to the particular polytope on which it is applied. In recent work [13] the GLS method was adapted to the PBM-expohedron, but does not naturally extend to the DBN-expohedron, for which such a decomposition algorithm remains to be found.

The two steps in the GLS method that need adaptation are the following:

(1) We need to find the intersection of a half-line starting from inside the polytope, with the border of the polytope.
(2) Given a point on a face of the polytope, we need to find a method able to find a vertex of the same face.

Step (1) can be solved with a bisection search, provided one has a method for checking whether a point is inside the polytope or

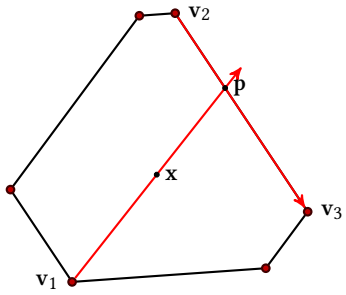


Figure 3: A DBN-expohedron for $n = 3$ documents with $\rho = (0.1, 0.5, 0.9)$, $\gamma = 0.5$ and $\kappa = 0.7$. The red arrows illustrate the GLS procedure for the Carathéodory decomposition of point x. The point x is decomposed as a convex sum of vertices v_1, v_2, v_3.

not [13]. For a PBM, this can be easily checked using the so-called majorization criterion [13, 14]. For the DBN we can use Algorithm 1.

Step (2) is much easier. Given a point $\mathbf{v} \in Z(\pi)$ on a face F of the expohedron, the vertex $\boldsymbol{\mathcal{E}}(\pi)$ is also on the face F, because there is only one vertex in each zone. Therefore Step 2 has complexity $O(n \log(n))$.

Algorithm 3 gives the pseudo-code of our Carathéodory decomposition in the DBN-expohedron.

Algorithm 3 Our Carathéodory decomposition in the DBN-expohedron

procedure GLS(INPUT: $\mathbf{x} \in \Pi(\gamma, \kappa, \boldsymbol{\rho})$)
  $\mathbf{v}_1 \leftarrow \boldsymbol{\mathcal{E}}(\texttt{argsort}(\mathbf{x}))$ ▷ Choose the initial vertex
  $\alpha_1 \leftarrow 1$ ▷ Set the initial vertex's weight to 1
  $\mathbf{p}_1 \leftarrow \mathbf{x}$
  for $i \in \{1, \dots, n\}$ **do**
    $\mathbf{p}_{i+1} \leftarrow \max\{\lambda > 0 \mid \mathbf{x} + \lambda(\mathbf{x} - \mathbf{v}_i) \in \Pi(\gamma, \kappa, \boldsymbol{\rho})\}$ ▷ Find the intersection with the border of the expohedron using a bisection search with `is_inside`
    $\alpha_{i+1} \leftarrow \alpha_i - \frac{\|\mathbf{p}_i - \mathbf{p}_{i+1}\|}{\|\mathbf{p}_{i+1} - \mathbf{v}_i\|}\alpha_i$ ▷ Update convex coefficients
    $\alpha_i \leftarrow \frac{\|\mathbf{p}_i - \mathbf{p}_{i+1}\|}{\|\mathbf{p}_{i+1} - \mathbf{v}_i\|}\alpha_i$
    $\mathbf{v}_{i+1} \leftarrow \boldsymbol{\mathcal{E}}(\texttt{argsort}(\mathbf{p}_{i+1}))]$
  end for
end procedure OUTPUT: $\alpha_1, \dots, \alpha_n, \mathbf{v}_1, \dots, \mathbf{v}_n$

Theorem 1. *Algorithm 3 finds a Carathéodory decomposition of any point in the DBN-expohedron with complexity $O(n^3)$*

See proof in Appendix (page 12).

5.2 Pareto-front identification

Our MOO problem (9) can be expressed in the expohedron as

$$\max_{\boldsymbol{\mathcal{E}} \in \Pi} \boldsymbol{\rho}^\top \boldsymbol{\mathcal{E}}, \quad \min_{\boldsymbol{\mathcal{E}} \in \Pi} \left\| \boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{E}}^* \right\|_2. \tag{14}$$

One objective is linear, the other objective is quadratic. We can apply a similar procedure as in [13] to geometrically build the Pareto-frontier. We start from one extremity, the point $\boldsymbol{\mathcal{E}}^*$, the only one for which the unfairness is minimal. The level curves of the unfairness are hyperspheres centered in $\boldsymbol{\mathcal{E}}^*$, so this objective

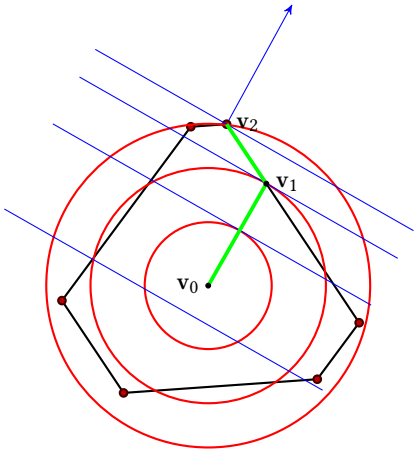


Figure 4: Illustration of Algorithm 4 on a DBN-expohedron with $\rho = (0.1, 0.5, 0.9)$, $\gamma = 0.5$ and $\kappa = 0.7$. The red circles are the level curves of the fairness objective, the blue lines are the level curves of the utility objective. The blue arrow is the the direction of steepest ascent for the utility objective: the vector ρ projected on the expohedron's hyperplane. The green curve is the set of Pareto-optimal exposure vectors for demographic fairness. The point v_0 is the point of pure fairness, where all exposures are identical and the point v_2 corresponds to a PRP ranking

deteriorates in the same manner in every direction. Because of this, the direction in which we move from here is the one where we increase utility the most within the constraint of staying inside the expohedron. This direction is the projection of the vector $\boldsymbol{\rho}$ onto the hyperplane containing the expohedron. This direction is followed until the border of the expohedron is reached: we are on a face of the expohedron. Then the same reasoning can be repeated on the current face of the expohedron and we need to follow the direction of the projection of $\boldsymbol{\rho}$ onto the hyperplane containing the current face. Each face has at least one less dimension than the previous one, so we are sure to end up at the PRP vertex or at a point on a face with maximal utility

Theorem 2. *Algorithm 4 finds all Pareto-optimal exposure vectors in the expohedron $\Pi(\gamma, \kappa, \boldsymbol{\rho})$.*

See proof in Appendix (page 12).

6 EXPERIMENTS

We perform experiments using two datasets: TREC and MSLR. Our code will be made available online.[1] Our experiments were performed on a laptop with an Intel®Core™i7-8650U CPU @ 1.90GHz processor.

6.1 Datasets

We use the TREC2020 Fairness Track dataset[2] and the MSLR dataset [20]. The TREC data is actually made up of two datasets: the training set and the test set. Since our own method does not require any training, we evaluate it on the test set only, but we use the

[1] https://github.com/naver/expohedron
[2] https://fair-trec.github.io/2020/index.html

Algorithm 4 An algorithm able to build the Pareto from of MOO problem (14) in the DBN-expohedron

procedure Pareto(INPUT: Expohedron $\Pi(\gamma, \kappa, \boldsymbol{\rho})$, target exposure $\mathbf{v}^{(0)} = \boldsymbol{\mathcal{E}}^*$)
  $F = (\pi, S) \leftarrow$ face_id($\mathbf{v}^{(0)}$) ▷ Initialize splits
  $P \leftarrow$ Gram-Schmidt($\{\boldsymbol{\nu}_s \mid s \in S\}$) ▷ Build the projection matrix P on the linear subspace of F
  $\boldsymbol{\rho}^{(0)} \leftarrow \boldsymbol{\rho} - P\boldsymbol{\rho}$ ▷ Project $\boldsymbol{\rho}$ on the subspace
  $l \leftarrow 0$
  while $\boldsymbol{\rho}^\top \mathbf{v}^{(l)} < \boldsymbol{\rho}^\top \boldsymbol{\mathcal{E}}(\pi_{\text{PRP}})$ **do** ▷ While utility is not maximal
    $k \leftarrow \max\{k \geq 0 \mid \mathbf{v}^{(l)} + k\boldsymbol{\rho}^{(l)} \in \Pi\}$ ▷ Do a bisection search with is_inside
    $\mathbf{v}^{(l+1)} \leftarrow \mathbf{v}^{(l)} + k\boldsymbol{\rho}^{(l)}$
    $F = (\pi, S) \leftarrow$ face_id($\mathbf{v}^{(l+1)}$)
    $P \leftarrow$ Gram-Schmidt($\{\boldsymbol{\nu}_s \mid s \in S\}$) ▷ Build the projection matrix on the linear subspace of the new face F
    $\boldsymbol{\rho}^{(l+1)} \leftarrow \boldsymbol{\rho} - P\boldsymbol{\rho}$ ▷ Project $\boldsymbol{\rho}$ on the new face
    $l \leftarrow l + 1$
  end while
end procedure
OUTPUT: A sequence of at most n points $(\mathbf{v}^{(j)})_{j\in\{0,\dots,l\}}$. The Pareto-curve is the union of line segments of endpoints $[\mathbf{v}^{(j)}, \mathbf{v}^{(j+1)}]$.

training set on LTR , one of our baselines (see below). The training set contains 200 queries with a variable number of documents ranging from 10 to 171. For each query-document pair, we estimated relevances scores with values in $[0, 1]$ ourselves. The details of how we computed them will be made available in our github repository.

For the MSLR dataset, we randomly separated it into a training set and a test set of 434 queries each. We use the test set for evaluation and we use the training set only for the baseline method that requires one. For each query-document pair, a graded relevance value in $\{0, 1, 2, 3, 4\}$ is provided by the dataset. We divide those values by 4 in order to conform to our setting that requires relevance values to be in $[0, 1]$. Thus for MSLR, the relevance vector $\boldsymbol{\rho}$ is in $\{0, 0.25, 0.5, 0.75, 1\}^n$.

6.2 Metrics and baselines

We use a DBN exposure model as described by (2), with parameters $\gamma = 0.5$ and $\kappa = 0.7$. These parameters are purely arbitrary, and they correspond to the parameters used in the TREC2020 challenge. We consider the merit of a document to be its relevance $\boldsymbol{\mu} = \boldsymbol{\rho}$, i.e., we work with meritocratic fairness. Therefore the exposure vector guaranteeing fairness is the only $\boldsymbol{\mathcal{E}}^* \in \Pi(\gamma, \kappa, \boldsymbol{\rho})$ such that $\boldsymbol{\mathcal{E}}^* \propto \boldsymbol{\rho}$. If it does not exists we take $\boldsymbol{\mu}' = \mu + k\mathbf{1}$ with $k > 0$ as small as possible. We compute this value using a bisection search with Algorithm 1.

We analyze 4 methods in total. For each method, $T = 1\,000$ rankings are delivered. Exposures cumulate over time and we average the final exposures by the time horizon T, i.e., the total number of rankings delivered for each query, so as to get a metric that does not depend on T. For the methods requiring a scalarized objective we use

$$\min \alpha(-\text{nU}) + (1-\alpha)\text{nF}^2. \tag{15}$$

We make α vary in $[0, 1]$ with a step 0.05, getting 21 values of α. We use the square nF^2 instead of nF, because this simplifies the gradient computation for the baseline LTR.

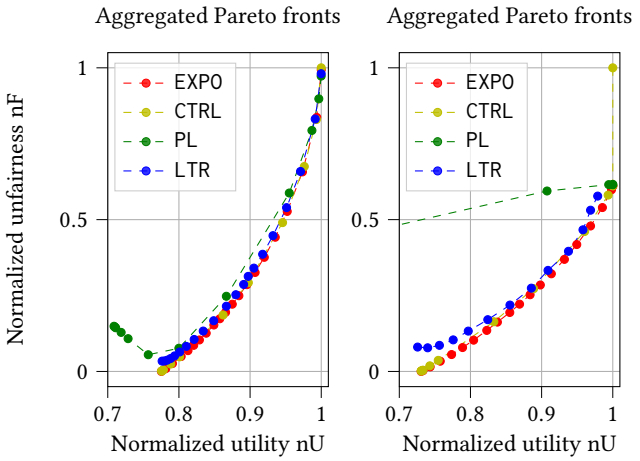
The methods we compare are the following:

(1) Our own method (EXPO) builds the Pareto frontier in the expohedron. It then chooses a point on this Pareto frontier corresponding to a trade-off α between nU and nF. It expresses it as the expected value of a distribution over permutations. Finally it delivers the sequence of T rankings using m-balanced words [13, 22]. Intuitively, m-balanced words are sequences of letters such that in every sub-string the frequencies of each letter are as equal as possible, given an overall proportion in the whole string. In our case using balanced words instead of randomly sampling from our distribution $\mathcal{D}$ allows to more accurately approach the expected value of $\mathcal{D}$ at any time.
(2) We execute a controller (CTRL) [25] and recover the fairness and utility at the last time step. This controller computes at each time step the current error with respect to the target exposure and defines the scores at time t as
$$\mathbf{y}_t = \boldsymbol{\rho} + g(\boldsymbol{\mathcal{E}}^* - \boldsymbol{\mathcal{E}}_t), \tag{16}$$
where $\boldsymbol{\mathcal{E}}_t$ is the exposure vector at time t and $g \geq 0$ is a trade-off parameter called *gain*. At time $t + 1$ the ranking is determined by decreasing scores. We make the gain vary in $[0, 1]$ on a logarithmic scale.
(3) We sample from a Plackett-Luce distribution (PL) as proposed by Diaz et al. [8] with varying temperature τ and with parameters $\boldsymbol{\rho}$. We make the temperature vary in $[0.001, 50]$ on a logarithmic scale.
(4) We use the approach by Oosterhuis [17] (LTR), that we adapted to the DBN exposure model. This baseline requires a training set to find an optimal function $\sigma : \mathbb{R} \to \mathbb{R}$ that to a relevance value $\boldsymbol{\rho}_i$ associates a score $\sigma(\boldsymbol{\rho}_i)$ to be used as log-scores of a PL distribution. The function σ is then used on the test set and the rankings are sampled by a policy $\text{PL}(\sigma(\boldsymbol{\rho}_1), \dots, \sigma(\boldsymbol{\rho}_n))$.

6.3 Results

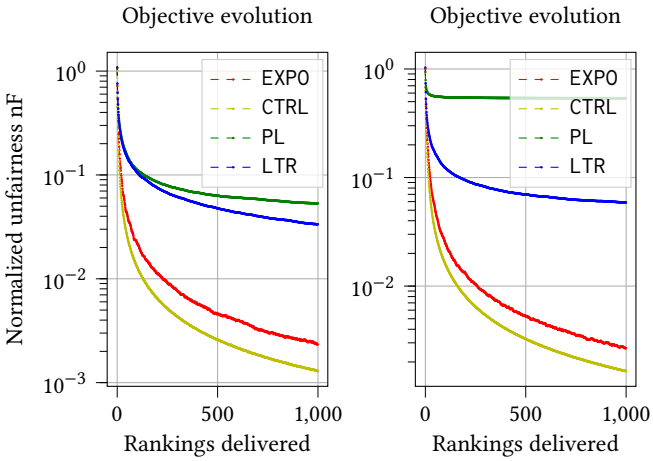
Does our method Pareto-dominate the baselines? From Theorem 2 we expect the method EXPO to yield Pareto-optimal expected exposure vectors. We experimentally verify this claim by comparing its performance to the baselines in terms of the two objectives nU, nF.

Figure 5 confirms that our method Pareto-dominates all the baselines. We observe that the CTRL baseline is particularly close to the EXPO method. This suggests that the CTRL baseline might be Pareto-optimal, in some cases; and it is an interesting empirical result in itself since this method was proposed as a heuristic and has as of yet no proof of optimality. However Figure 5b also shows that for some parameters, CTRL yields solutions that are not Pareto-optimal (point (1, 1)). Since we work with meritocratic fairness and, since with a very high temperature a PL policy delivers rankings in a uniform manner, it comes as no surprise that the curve obtained with the PL method does not attain zero unfairness and that the unfairness does not attain its minimum for an infinite temperature. The LTR method performs expectedly better than a naive PL policy,



(a) TREC (b) MSLR

Figure 5: For both the TREC and the MSLR dataset, and for our baselines the averaged normalized unfairness metric and the average normalized utility metric are plotted for various trade-off parameters after delivery of $T = 1\,000$ rankings.



(a) TREC (b) MSLR

Figure 6: For both the TREC and the MSLR dataset, and for our baselines both the unfairness metric and the utility metric are plotted for various parameters after delivery of $T = 1\,000$ rankings.

but it is not as close to the true Pareto-front as the CTRL method, especially at the pure fairness endpoint, which it does not reach.

The result for the MSLR dataset in Figure 5b present a particularity in that the PL, LTR and EXPO curves have a normalized unfairness lower than 1, while being of maximal utility, whereas CTRL reaches the point $(1, 1)$ for a gain of 0. In order to achieve nF < 1 and nU $= 1$, the CTRL method would seem to require a gain slightly larger than 0. This can be explained by the fact that the MSLR dataset contains a lot of duplicates relevance values, as the values are all in $\{0, 0.25, 0.5, 0.75, 1\}$. This means that there exist many different PRP rankings, and delivering several different PRP rankings can already reduce the unfairness by a large amount without reducing the utility. This happens naturally for a PL policy, because items with equal scores will have equal expected exposure, it happens for EXPO because it is Pareto-optimal, but it does not happen for the controller when the gain is set to 0.

How does the fairness evolve over time? Assume that we are in a situation in which our goal is to minimize the unfairness only, without consideration for the utility. We choose for each method the parameters α, τ or g that minimize the normalized unfairness. Then we look at how the average normalized unfairness metric evolves over the delivery of $T = 1\,000$ rankings, for our different methods.

The results, displayed in Figure 6, show that with the PL method, the metric does not converge to 0, as can be expected from Figure 5. Furthermore the CTRL method seems to converge more quickly to 0 than the EXPO method. This may be due to the fact that the CTRL method is at liberty of choosing from any possible ranking at each time step, while the EXPO method, delivered with balanced words, has a support of size at most n. Recall, however, that convergence to 0 of the controller (let alone "faster than EXPO") is merely an experimental observation, whereas it is theoretically guaranteeed with EXPO.

Is our method efficient? The time taken by the methods cannot be fairly compared by one number alone. For instance our EXPO method has a moderate initial computation cost for each query, but once an optimal distribution $\mathcal{D}$ is found, a policy can be delivered very efficiently. On the other hand the CTRL method has no initial cost, but some computations need to be done each time a new ranking is delivered. The LTR method has a very high initial cost of training, but once a mapping is found the cost of ranking delivery is low. The PL method has no initial cost and it can be delivered efficiently. To account for these subtleties, we report separately for each method the time for pre-computation (which is constant) and the time for delivery (which is proportional to the number T of rankings delivered).

In Table 1 we report the average time it takes to deliver $T = 1\,000$ rankings for each query, for both datasets. The results show the extraordinary efficiency of using an m-balanced word generator to deliver a ranking policy found with the EXPO method. They also show the efficiency of our Carathéodory decomposition algorithm and of our Pareto-front identification algorithm, especially w.r.t. a learning approach LTR. Finally we note that the CTRL method has no initial cost, but a higher delivery cost per ranking, than our EXPO method. Hence it may be faster for short time horizons T, but will be slower for longer horizons.

7 CONCLUSION

In this paper, we propose a geometric method to compute Pareto-optimal ranking policies to maximize both user utility and item producer fairness. Our method is the first that provably solves this key timely problem for a realistic DBN exposure model—while prior literature was limited to a PBM exposure model—; and it

Table 1: For both datasets and a given trade-off parameter, we report the time in seconds, that it took to deliver $T = 1\,000$ rankings. EXPO Pareto is the time it took to compute the whole Pareto-front using our EXPO method, averaged over all the queries. EXPO GLS is the time it took to compute the Carathéodory decomposition of the fairness endpoints, averaged over all the queries. EXPO delivery is the time it took to deliver a total of $T = 1\,000$ rankings using an m-balanced word generator, averaged over all the queries. LTR learning is the total training time of the whole training dataset, while LTR delivery it the average time it took to deliver a total of $T = 1\,000$ rankings by sampling a PL distribution. CTRL is the time it took to deliver a total of $T = 1\,000$ rankings using the CTRL method, averaged over all the queries.

	TREC	MSLR
EXPO Pareto	0.9962	0.1853
EXPO GLS	1.0591	4.350
EXPO delivery	0.0019	0.0054
CTRL	0.0568	0.0796
PL	0.0141	0.0630
LTR learning	468.5	1746.2
LTR delivery	0.0143	0.0631

works in reasonable time and space complexity. Our experiments show that it outperforms classical models based on PL distributions, which are not optimal. Interestingly though, we find that a simple heuristic controller does (almost) as good as our method in terms of optimality. This is interesting as it experimentally shows optimality of the controller in our experiments, and also because it offers an alternative to adapt to the situation at stake: the controller has no pre-computation time and is faster for delivering just a few rankings; whereas our method has a (small) offline pre-computation time, but is much faster per delivered ranking. Of course our method also has the advantage of the theoretical guarantee of optimality in any case.

ACKNOWLEDGMENTS

This work has been partially supported by MIAI@Grenoble Alpes, (ANR-19-P3IA-0003).

REFERENCES

[1] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. 2019. Fair Regression: Quantitative Definitions and Reduction-based Algorithms. *arXiv:1905.12843 [cs, stat]* (May 2019). http://arxiv.org/abs/1905.12843 arXiv: 1905.12843.

[2] Asia J. Biega, Fernando Diaz, Michael D. Ekstrand, and Sebastian Kohlmeier. 2020. Overview of the TREC 2020 Fair Ranking Track. In *The Twenty-Eighth Text REtrieval Conference (TREC 2020) Proceedings*.

[3] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR '18*. ACM Press, Ann Arbor, MI, USA, 405–414. https://doi.org/10.1145/3209978.3210063

[4] C. Carathéodory. 1907. Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen. *Math. Ann.* 64, 1 (March 1907), 95–115. https://doi.org/10.1007/BF01449883

[5] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management (CIKM '09)*. Association for Computing Machinery, New York, NY, USA, 621–630. https://doi.org/10.1145/1645953.1646033

[6] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web (WWW '09)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/1526709.1526711

[7] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. Click Models for Web Search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 7, 3 (July 2015), 1–115. https://doi.org/10.2200/S00654ED1V01Y201507ICR043

[8] Fernando Diaz, Bhaskar Mitra, Michael D. Ekstrand, Asia J. Biega, and Ben Carterette. 2020. Evaluating Stochastic Rankings with Expected Exposure. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 275–284. https://doi.org/10.1145/3340531.3411962

[9] Fanny Dufossé and Bora Uçar. 2016. Notes on Birkhoff–von Neumann decomposition of doubly stochastic matrices. *Linear Algebra Appl.* 497 (May 2016), 108–115. https://doi.org/10.1016/j.laa.2016.02.023

[10] 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 on - ITCS '12*. ACM Press, Cambridge, Massachusetts, 214–226. https://doi.org/10.1145/2090236.2090255

[11] Martin Grötschel, Laszlo Lovasz, and Alexander Schrijver. 1993. *Geometric Algorithms and Combinatorial Optimization* (2 ed.). Springer-Verlag, Berlin Heidelberg. https://doi.org/10.1007/978-3-642-78240-4

[12] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 422–446. https://doi.org/10.1145/582415.582418

[13] Till Kletti, Jean-Michel Renders, and Patrick Loiseau. 2022. Introducing the Expohedron for Efficient Pareto-optimal Fairness-Utility Amortizations in Repeated Rankings. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22)*. ACM, Tempe, AZ, USA, 10. https://doi.org/10.1145/3488560.3498490

[14] Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. 2011. *Inequalities: Theory of Majorization and Its Applications*. Springer New York, New York, NY. https://doi.org/10.1007/978-0-387-68276-1

[15] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling Fairness and Bias in Dynamic Learning-to-Rank. *arXiv:2005.14713 [cs, stat]* (May 2020). https://doi.org/10.1145/3397271.3401100 arXiv: 2005.14713.

[16] Márton Naszódi and Alexandr Polyanskii. 2019. Perron and Frobenius meet Carathéodory. *arXiv:1901.00540 [math]* (Jan. 2019). http://arxiv.org/abs/1901.00540 arXiv: 1901.00540.

[17] Harrie Oosterhuis. 2021. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1023–1032. https://doi.org/10.1145/3404835.3462830

[18] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2021. Fairness in rankings and recommendations: an overview. *The VLDB Journal* (Oct. 2021). https://doi.org/10.1007/s00778-021-00697-y

[19] R. L. Plackett. 1975. The Analysis of Permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 24, 2 (1975), 193–202. https://doi.org/10.2307/2346567

[20] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR* abs/1306.2597 (2013). http://arxiv.org/abs/1306.2597

[21] Stephen Robertson. 1977. The Probability Ranking Principle in IR. *Journal of Documentation* 33 (Dec. 1977), 294–304. https://doi.org/10.1108/eb026647

[22] Shinya Sano, Naoto Miyoshi, and Ryohei Kataoka. 2004. m-Balanced words: A generalization of balanced words. *Theoretical Computer Science* 314, 1-2 (Feb. 2004), 97–120. https://doi.org/10.1016/j.tcs.2003.11.021

[23] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*. ACM Press, London, United Kingdom, 2219–2228. https://doi.org/10.1145/3219819.3220088

[24] Ashudeep Singh, David Kempe, and Thorsten Joachims. 2021. Fairness in Ranking under Uncertainty. *arXiv:2107.06720 [cs]* (July 2021). http://arxiv.org/abs/2107.06720 arXiv: 2107.06720.

[25] Thibaut Thonet and Jean-Michel Renders. 2020. Multi-grouping Robust Fair Ranking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 2077–2080. https://doi.org/10.1145/3397271.3401292

[26] Lequn Wang and Thorsten Joachims. 2021. User Fairness, Item Fairness, and Diversity for Rankings in Two-Sided Markets. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '21)*. Association for Computing Machinery, New York, NY, USA, 23–41. https://doi.org/10.1145/3471158.3472260

[27] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2021. Fairness in Ranking: A Survey. *arXiv:2103.14000 [cs]* (May 2021). http://arxiv.org/abs/2103.14000 arXiv: 2103.14000.

APPENDICES

A Proofs

Proof of Proposition 1. The idea is to show that the scalar product

$$\boldsymbol{v}^\top \boldsymbol{\mathcal{E}}(\pi) \tag{17}$$

does note change whenever the ranks of two items of adjacent ranks are exchanged. All rankings can be obtained by successively exchanging the ranks of adjacent items, starting from any initial ranking. Therefore it suffices to show that (17) remains unchanged by the exchange of the ranks of two adjacent elements. Given an arbitrary ranking π_0 the exposures of ranks k and $k+1$ are

$$\ldots, \underbrace{K}_{\text{rank } k, \text{ item } \pi_0(k)}, \underbrace{K\gamma(1-\kappa\boldsymbol{\rho}_{\pi_0(k)})}_{\text{rank } k+1, \text{ item } \pi_0(k+1)}, \ldots \tag{18}$$

for a certain constant K. When the items at ranks k and $k+1$ are exchanged, to give a new ranking π_1, these exposures become

$$\ldots, \underbrace{K}_{\text{rank } k, \text{ item } \pi_0(k+1)}, \underbrace{K\gamma(1-\kappa\boldsymbol{\rho}_{\pi_0(k+1)})}_{\text{rank } k+1, \text{ item } \pi_0(k)}, \ldots \tag{19}$$

while the exposures at all other ranks remain unchanged. Let us denote $i = \pi_0(k)$ and $j = \pi_0(k+1)$. With some algebraic manipulations, it is then easy to show that the scalar products (17) for rankings π_0 and π_1 satisfy

$$\begin{aligned}\boldsymbol{v}^\top \boldsymbol{\mathcal{E}}(\pi_1) - \boldsymbol{v}^\top \boldsymbol{\mathcal{E}}(\pi_0) &= \left(1 + \frac{\gamma\kappa}{1-\gamma}\boldsymbol{\rho}_i\right)\left(K\gamma(1-\kappa\boldsymbol{\rho}_j) - K\right) \\ &\quad + \left(1 + \frac{\gamma\kappa}{1-\gamma}\boldsymbol{\rho}_j\right)\left(K - K\gamma(1-\kappa\boldsymbol{\rho}_i)\right) \\ &= 0.\end{aligned} \tag{20}$$

This means that (17) remains unchanged by exchanging two items of adjacent ranks, and thus that (17) is constant in the whole expohedron. □

Proof of Proposition 2. The proof to this theorem is quite complex, we first introduce two lemmas to build an equivalence with a well-known polytope called *permutohedron*, then we show how the result can be obtained as a corollary of Proposition 1.

Lemma 3. *For every $\pi \in \mathcal{S}_n$, the point $\boldsymbol{\mathcal{E}}(\pi, \gamma, \kappa, \boldsymbol{\rho})$ is in the zone $Z(\pi)$.*

Proof. We need to show that

$$\boldsymbol{\mathcal{E}}_{\pi(1)} \geq \ldots \geq \boldsymbol{\mathcal{E}}_{\pi(n)}. \tag{21}$$

This is clearly the case, because the exposure decreases with increasing rank. □

Lemma 4. *The vertices of the the DBN-expohedron $\Pi(\gamma, \kappa, \boldsymbol{\rho})$ are exactly the $n!$ points $\boldsymbol{\mathcal{E}}(\pi, \gamma, \kappa, \boldsymbol{\rho})$ for $\pi \in \mathcal{S}_n$. Formally*

$$\mathbf{x} \in \Pi(\gamma, \kappa, \boldsymbol{\rho}) \text{ is a vertex} \iff \exists \pi \in \mathcal{S}_n : \mathbf{x} = \boldsymbol{\mathcal{E}}(\pi, \gamma, \kappa, \boldsymbol{\rho}). \tag{22}$$

Proof.

- " $\Longrightarrow$ " Every vertex of $\Pi(\gamma, \kappa, \boldsymbol{\rho})$ is a point of type $\boldsymbol{\mathcal{E}}(\pi, \gamma, \kappa, \boldsymbol{\rho})$, because $\Pi(\gamma, \kappa, \boldsymbol{\rho})$ is the convex hull of these points.
- " $\Longleftarrow$ " To prove that a point in a polytope is a vertex, it suffices to show that it cannot be expressed as a convex combination of other vertices. Let $\pi \in \mathcal{S}_n$ such that

$$\boldsymbol{\mathcal{E}}(\pi, \gamma, \kappa, \boldsymbol{\rho}) = \sum_{i=1}^{N} \alpha_i \boldsymbol{\mathcal{E}}(\pi_i, \gamma, \kappa, \boldsymbol{\rho}), \tag{23}$$

with $\sum_{i=1}^{N} \alpha_i = 1$ and $\alpha_i > 0$ for all $i \in \{1, \ldots, N\}$. One can show by recursion over $k \in \{1, \ldots, n\}$, that

$$\boldsymbol{\mathcal{E}}_{\pi(k)}(\pi_i, \gamma, \kappa, \boldsymbol{\rho}) = \boldsymbol{\mathcal{E}}_{\pi(k)}(\pi, \gamma, \kappa, \boldsymbol{\rho}),\ \forall i \in \{1, \ldots, N\}. \tag{24}$$

The highest component of $\boldsymbol{\mathcal{E}}(\pi, \gamma, \kappa, \boldsymbol{\rho})$ is the number 1 at index $\pi(1)$, so all other $\boldsymbol{\mathcal{E}}(\pi_i, \gamma, \kappa, \boldsymbol{\rho})$ must also have a 1 at index $\pi(1)$. If all $\boldsymbol{\mathcal{E}}(\pi_i, \gamma, \kappa, \boldsymbol{\rho})$ have equal components at indices $\pi(1), \ldots, \pi(k)$, then by the same reasoning, they must have equal components at indices $\pi(k+1)$. By recursion, they have equal indices at all components, which concludes the proof.

□

From previous lemmas we can conclude that $\Pi(\gamma, \kappa, \boldsymbol{\rho})$ has the same combinatorial structure as the PBM-expohedron [13] and as the permutohedron. Therefore a face of $\Pi(\gamma, \kappa, \boldsymbol{\rho})$ is composed of vertices defined by a partial ordering of the items. Such a partial ordering can be defined by a permutation $\pi \in \mathcal{S}_n$ and a set of splits $S \subseteq \{1, \ldots, n\}$. Given a permutation $\pi \in \mathcal{S}_n$ and a set of splits $\{s_1, \ldots, s_N\} \subseteq \{1, \ldots, n\}$ with $s_N = n$, the partial ordering is such that, when working in the basis of $\mathbb{R}^n$ in which π is the identity, we have

$$\boldsymbol{\mathcal{E}}(\pi')_1, \ldots, \boldsymbol{\mathcal{E}}(\pi')_{s_1} \geq \ldots \geq \boldsymbol{\mathcal{E}}(\pi')_{s_{N-1}+1}, \ldots, \boldsymbol{\mathcal{E}}(\pi')_{s_N}, \tag{25}$$

for every vertex $\boldsymbol{\mathcal{E}}(\pi')$ of the face. In particular, if we ignore the ranks $s_k + 1, \ldots, n$ for $k \in \{1, \ldots, N\}$, we can apply Proposition 1 to the ranks $s_1, \ldots, s_k$ and get

$$\boldsymbol{v}_{s_k} = \mathbf{1}(s_k) + \frac{\gamma\kappa}{1-\gamma}(\boldsymbol{\rho} \odot \mathbf{1}(s_k)) \tag{26}$$

as a normal vector to the face, which is what there was to prove. □

Proof of Lemma 1. The expohedron can be partitioned into $n!$ pieces, each of which is contained by a zone. A point $\mathbf{x}$ in a zone $Z(\pi)$ is in the expohedron if and only if it is in the part of the expohedron contained by $Z(\pi)$. Within a zone, the expohedron is delimited by n hyperplanes: one hyperplane corresponds to the hyperplane of the whole expohedron (checked in line 5) and $n-1$ other hyperplanes corespond to the $n-1$ facets adjacent to the only vertex of zone $Z(\pi)$ (checked in line 9).

The determination of the zone of $\mathbf{x}$ has complexity $O(n \log(n))$. In the loop over $s \in \{1, \ldots, n-1\}$, each scalar product has complexity $O(n)$. Therefore the complexity of the whole algorithm is $O(n^2)$. □

Proof of Lemma 2. Once we know that a point $\mathbf{x}$ is in the zone $Z(\pi)$, it can be in at most $n-1$ facets of the expohedron at once, because there are $n-1$ facets adjacent to each unique vertex of a zone. Since every face of the expohedron is in the intersection of a certain number of facets belonging to the same zone, it suffices to check to which facets of zone $Z(\pi)$, the point $\mathbf{x}$ belongs in order to know to which faces he belongs. The lowest-dimensional such face is the one with the most normal vectors, which is what the algorithm finds. The complexity of the initial zone determination

is $O(n\log(n)$, and the loop performs n scalar products of vectors of size n, so the total complexity is $O(n^2)$. □

Proof of Theorem 1. Algorithm 3 implements the GLS, which is proven to yield the Carathéodory decomposition of any point inside the polytope on which it is applied [11]. The most complex operation in the loop has complexity $O(n^2)$ so the total complexity of the algorithm is $O(n^3)$. □

Proof of Theorem 2. The fact that the algorithm builds the Pareto front is sufficiently explained in the paragraph preceding the pseudo-code. The complexity of the most complex operation inside the loop is $O(n^2)$ and the loop has length at most n, so the total complexity of the algorithm is $O(n^3)$. □

B Click Models

We will refer tho the click models described in the book [7] and, in particular, use the same notations.

B.1 Dynamic Bayesian Network (DBN). The general recursive formulation of the DBN Exposure (a.k.a. Examination probability) for the query-document pair (q, d) at rank k is:

$$\epsilon_k = \epsilon_{k-1}\gamma(1 - \alpha_{q,d}\sigma_{q,d}) \tag{27}$$

where γ is the patience parameter, $\alpha_{q,d}$ is the attraction probability of document d with respect to q and $\sigma_{q,d}$ is the probability that the user will be satisfied with that document and stop looking further down in the list. In this model, the product $\alpha_{q,d}\sigma_{q,d}$ is the relevance probability i.e. $\mathbb{P}(r_{q,d} = 1) = \alpha_{q,d}\sigma_{q,d}$. This model is equivalent to our general expression in (2), with $\rho = \alpha_{q,d}\sigma_{q,d}$ and $\kappa = 1$.

B.2 Simplified Dynamic Bayesian Network (SDBN). The SDBN Model is the particular case of the DBN model with a patience parameter γ equal to one. Its general recursive formulation for the query-document pair (q, d) at rank k is:

$$\epsilon_k = \epsilon_{k-1}(1 - \alpha_{q,d}\sigma_{q,d}) \tag{28}$$

Define $\omega := \frac{1}{2}\min \alpha_{q,d}\sigma_{q,d}$. Assuming that $\alpha_{q,d}\sigma_{q,d}$ is always in $(0, 1)$, this model is equivalent to our general expression in (2), with $\gamma = 1 - \omega$, and $\kappa = 1$ and $\rho = 1 - \frac{1-\alpha_{q,d}\sigma_{q,d}}{1-\omega}$.

B.3 Cascade Model (CM). The Cascade Model is the particular case of the DBN model with a patience parameter γ equal to one and a satisfaction probability $\sigma_{q,d}$ also equal to one. In other words, it assumes that if a document snippet looks relevant, then the document is also relevant and the user will be fully satisfied after clicking on it; consequently, she will stop after one single click. Its general recursive formulation for the query-document pair (q, d) at rank k is:

$$\epsilon_k = \epsilon_{k-1}(1 - \alpha_{q,d}) \tag{29}$$

In this model, the term $\alpha_{q,d}$ is considered as the relevance probability i.e. $\mathbb{P}(r_{q,d} = 1) = \alpha_{q,d}$. This model is equivalent to the SDBN, with $\sigma_{q,d} = 1$ and therefore also equivalent to our general expression (2) if all $\alpha_{q,d} \in (0, 1)$.

B.4 Discrete Choice Model (DCM). The DCM generalizes the Cascade Model (CM) by still allowing a user to go down in the list even after clicking on one document. In other words, it allows the user to click on more than one document. The general recursive formulation of the DCM Exposure for the query-document pair (q, d) at rank k is:

$$\epsilon_k = \epsilon_{k-1}(1 - \alpha_{q,d}(1 - \lambda)) \tag{30}$$

where λ is the probability of continuing the examination of the next items in the list after clicking on a document. This model is equivalent to the SDBN with $\sigma_{q,d} = (1 - \lambda)$ and therefore also equivalent to our general expression (2) if all $\alpha_{q,d}(1 - \lambda) \in (0, 1)$.

Note that some versions of the DCM exist with a rank-dependent λ_k, instead of a single continuation parameter λ. These versions do not fit our general expression.

B.5 Click Chain Model (CCM). The CCM also generalizes the Cascade Model (CM), but in a much more flexible way. The general recursive formulation of the CCM Exposure for the query-document pair (q, d) at rank k is:

$$\epsilon_k = \epsilon_{k-1}(\alpha_{q,d}((1 - \alpha_{q,d})\tau_2 + \alpha_{q,d}\tau_3) + (1 - \alpha_{q,d})\tau_1) \tag{31}$$

which can be re-written as:

$$\epsilon_k = \epsilon_{k-1}\tau_1(1 - \kappa_{q,d}) \tag{32}$$

with $\kappa_{q,d} = \alpha_{q,d}\left(\frac{\tau_1-\tau_2}{\tau_1} + \alpha_{q,d}\frac{\tau_2-\tau_3}{\tau_1}\right)$. See [7] for the meaning of the τ_i parameters.

This model is equivalent to our general expression in (2), with $\gamma = \tau_1$ and $\rho = \alpha_{q,d}\left(\frac{\tau_1-\tau_2}{\tau_1} + \alpha_{q,d}\frac{\tau_2-\tau_3}{\tau_1}\right)$ and $\kappa = 1$.