FAiRDAS: Fairness-Aware Ranking as Dynamic Abstract System

Eleonora Misino[1,*], Roberta Calegari[1], Michele Lombardi[1] and Michela Milano[1]

[1]A M S –Università di Bologna, Italy

Abstract

AI has become increasingly prominent in online matchmaking and ranking systems, where individuals are paired, ranked and recommended based on their characteristics and preferences. The need for long-term fairness in these applications has become crucial to prevent biases and discrimination. To address this, fairness-aware algorithms are commonly employed, incorporating fairness constraints into the ranking process. These algorithms use metrics and models to ensure equitable treatment across user groups. However, studying the long-term fairness properties of these approaches can be complex, posing challenges in understanding their evolution and convergence. In this study, we propose an abstract dynamic system as a solution to design and ensure long-term fairness in ranking systems. This approach provides valuable insights into system behaviour, metric interactions, and overall dynamics. By considering the ranking system as a dynamic system, we can model the evolution and interaction of fairness metrics over time. Our proposed approach enables the analysis of system properties, trade-offs, and tensions that arise when optimizing multiple fairness metrics. To validate its effectiveness, we apply this approach to real-world use case scenarios, demonstrating its practical applicability.

Keywords

fair ranking, fair matchmaking, fairness in AI

1. Introduction

In recent years, artificial intelligence (AI) has gained significant prominence in the area of online matchmaking and ranking systems [1], which play a key role in diverse applications such as Airbnb or dating platforms like Tinder and Bumble[1] [2]. The objective of these systems is to pair individuals or resources based on their respective characteristics, skills, or preferences, expressed by a user query. These AI models can effectively learn from historical data, user preferences, and other relevant features to generate personalized recommendations and optimize the matching of individuals or items. While these systems strive to provide relevant and personalized recommendations, long-term fairness has emerged as a crucial aspect to consider. Long-term fairness refers to the equitable treatment of different groups of users over extended periods, aiming to avoid systemic biases or disadvantages [3, 4]. Long-term fairness appears to be an important feature to consider and enforce when we have multiple queries to a recommendation system or a ranking system. Given a sensitive attribute, while the single query

Aequitas 2023: Workshop on Fairness and Bias in AI | co-located with ECAI 2023, Kraków, Poland

*Corresponding author.

eleonora.misino2@unibo.it (E. Misino); roberta.calegari@unibo.it (R. Calegari); michele.lombardi2@unibo.it (M. Lombardi); michela.milano@unibo.it (M. Milano)

[1] Airbnb link, Tinder link, Bumble link.

does not exhibit any polarization, multiple repeated queries can be biased toward a specific value of the sensitive attribute.

The state of the art in addressing long-term fairness in online ranking systems involves the development of fairness-aware algorithms that explicitly incorporate fairness constraints into the ranking process [5, 6]. These algorithms leverage fairness metrics and mathematical models to ensure equitable treatment across user groups.

Fairness metrics in ranking models encompass diverse dimensions like demographic parity, equalized odds, or equal opportunity [7]. Each metric targets a specific fairness aspect, offering distinct insights. Considering and integrating multiple fairness metrics is common practice to attain a comprehensive understanding of fairness in rankings [8]. Thus, designing a theoretical framework that accommodates multiple metrics, possibly defined through parameters, becomes crucial.

Concerning algorithms, by formulating fairness constraints or objectives, the models ensure that the algorithm not only maximizes relevance or accuracy but also adheres to the desired fairness principles. These models often involve techniques from optimization, such as constrained optimization or multi-objective optimization, to balance fairness considerations with other objectives. One common approach is to introduce fairness constraints that enforce equal treatment or equal opportunity across different groups [4]. These constraints aim to ensure that the ranking algorithm does not favour or discriminate against specific demographic groups based on protected attributes such as gender, race, or age. By incorporating these constraints, the algorithm is encouraged to produce rankings that are fair and unbiased. Regularization techniques, such as fairness regularization, can also be employed to balance fairness considerations with other objectives [9], such as relevance or accuracy. Fairness regularization involves adding penalty terms to the optimization objective that explicitly encourage fair behaviour. The regularization terms act as a form of control, nudging the algorithm towards producing fair rankings while still optimizing for other desired properties.

Considering the factors mentioned above and given the complexities associated with studying the evolution and convergence of ranking systems, particularly when dealing with complex algorithms in existing state-of-the-art approaches, this paper proposes a novel solution. The main contribution of this paper is the formalization of **long-term fairness as an abstract dynamic system**, to model the evolution and interaction of fairness metrics over time.

This formalization enables a systematic analysis of system properties like stability, equilibrium, and convergence. It also facilitates the identification of critical points or attractors. Furthermore, it allows for the examination of how system changes, such as algorithm updates or user preference shifts, influence long-term fairness. Additionally, this formalization enables the exploration of trade-offs and tensions when optimizing multiple fairness metrics concurrently. It provides insights into how improvements in one metric may impact others and reveal the complex relationships between them. Through this approach, a deeper understanding of the dynamics between contrasting fairness metrics can be achieved, helping to identify potential synergies or conflicts among them.

It is important to note that the abstract dynamic system for long-term fairness is intentionally left abstract to accommodate customization based on specific case characteristics. Once the dynamic system parameters are selected, the abstract dynamic system can be tailored to the specific case under investigation. Experiments can then be conducted to observe system behaviour and evaluate the effectiveness of different approaches in achieving fairness goals. This formalization allows for better control and fine-tuning of each component of the system.

The paper is organized as follows. Section 2 introduces FAiRDAS – Fairness-Aware Ranking as Dynamic Abstract System – and presents its mathematical foundation. The dynamic framework is explained in detail, along with the process of grounding it. Next, an empirical evaluation is conducted in 3, grounding the framework in the motivating example described above. The instantiation of the framework is evaluated, demonstrating the effectiveness of the proposed approach. Finally, in 4, concluding remarks are provided, along with a discussion on future works.

2. FAiRDAS

In this section, we present the problem formulation for the framework (Subsection 2.1). The main objective is to develop a comprehensive understanding of the problem at hand and establish a clear foundation for our proposed solution. Then, the FAiRDAS abstract dynamic system is introduced in Subsection 2.2. The main idea behind the work revolves around conceptualising long-term fairness as a dynamic system, allowing us to define the ideal behaviour and potentially conduct system property analysis. By mapping fairness to this abstract level, we establish a framework for understanding the desired dynamics of the system. After the definition of the dynamic system, we transition from the abstract level to the practical level by implementing concrete actions within the real system (Subsection 2.3). These actions are specifically designed to approximate the ideal behaviour defined at the abstract level. By taking tangible steps within the real system, we aim to align its behaviour as closely as possible with the envisioned fairness dynamics.

2.1. Problem formulation: ranking problem definition

We approach the problem of ranking a set of m resources $\mathcal{R}$ in response to a sequence of incoming queries $q_t{}_{t=1}^{\infty}$. Our focus lies on the ranking algorithm, which can be manipulated by adjusting a vector of parameters θ_t, referred to as the *action vector* (e.g. penalizing an over-exposed resource). Formally, we can represent the ranking algorithm as a function:

$$\rho : \mathcal{Q} \times \Theta \rightarrow \{r_k\}_{k=1}^{m} \tag{1}$$

where $\mathcal{Q}$ is the set of possible queries, Θ is the set of possible action vectors, and $\{r_k\}_{k=1}^{m}$ is the resources rank for query q_t.

At each time step t a vector of n metrics $x_t \in \mathbb{R}^n$ is defined as:

$$x_t = x(\rho(q_t, \theta_t)), \tag{2}$$

where q_t and θ_t are the query and the set of actions performed at time t, respectively; $\rho(\cdot, \cdot)$ is the ranking function performed on $\mathcal{R}$, and it is based on the given query and set of actions.

2.2. The FAiRDAS Approach

Evolution of Metrics as a Dynamic System Equation (2) defines how to evaluate metrics across all resources, but it has two significant limitations. First, since it considers a single query at a time, the equation requires an aggregation mechanism in order to be used for *measuring* fairness across multiple queries. Second, since our eventual goal is *enforcing* fairness, we are interested in assessing how much the current action vector θ_t affects all possible incoming queries.

We can address both issues by taking the expectation of the metrics over the query distribution Q_t at time t. This allows us to capture the average behaviour across all queries and account for the variations in query characteristics and preferences, as well as the dependency of such elements on time. Hence, we formalize the metric value as its expectation over the query distribution:

$$\bar{x}_t = \mathbb{E}_{q_t \sim Q_t} \left[x(\rho(q_t, \theta_t)) \right] \tag{3}$$

Note that Q_t is not directly observable, but it can be approximated, for example by checking the last N queries (Equation (8)).

In Equation (3), the set of actions θ_t is performed to approach the ideal behaviour, and depends on the expected behaviour of the queries from the previous step:

$$\theta_{t+1} = \phi(\bar{x}_t), \tag{4}$$

where the function $\phi(\cdot)$ represents the selection mechanism of the set of actions. At each time step, we have direct and explicit control over the selection of the actions, so that we can actively influence the ranking algorithm to adapt and improve its fairness performance over time.

By focusing just on the evolution of the metrics, we combine Equation (3) and Equation (4) to obtain:

$$\bar{x}_{t+1} = f_\phi(\bar{x}_t), \tag{5}$$

where $f_\phi(\cdot)$ represents the evolution function depending on the selection mechanism ϕ.

Defining an Ideal System Behavior The ultimate objective is to evolve the system in such a way that it ensures the pre-defined metrics remain below a vector of *user-defined thresholds* $\mu \in \mathbb{R}^n$.

The metrics of interest are application dependant and may include fairness indicators as well as other metrics. For example, the application domain may require fairness metrics as well as performance metrics to guarantee both system's fairness and predictive performance within an acceptable range.

By setting a threshold for each metric, the user establishes boundaries that the system should not surpass. For fairness metrics, the threshold represents the maximum limit beyond which the system would be considered unfair. On the other hand, for accuracy metrics, the threshold represents the maximum acceptable level of error in the prediction.

It is worth mentioning that metrics can be contrasting or even conflicting – like in the case of fairness and accuracy. In such case cases, it may not be possible to satisfy all of the thresholds simultaneously. For example, improving fairness metrics might result in a decrease in accuracy, or vice versa. In such scenarios, it becomes essential to strike a balance and determine a satisfactory compromise that aligns with the desired objectives and priorities.

Finding this trade-off involves carefully considering the relative importance of each metric and making informed decisions based on the context and requirements of the system. The most immediate outcome of this process is the definition of the thresholds, but in some cases it might be usefule to *rescale* different metrics to reflect their relative importance. At this stage, we model the whole process in a abstract fashion, by treating the thresholds as parameters and by viewing any rescaling operation as part of the definition of the metrics themselves.

Accordingly, to these considerations, a disarable discrete dynamical system that ensures the user objectives can be described as follows:

$$\bar{x}_{t+1} = A\min(0, \bar{x}_t - \mu) + \bar{x}_t, \tag{6}$$

where A is a $n \times n$ diagonal and positive definite matrix, and due to result from the theory of discrete, linear, dynamic systems μ represents the upper bound for the set of fixed points in the system. By defining A and μ, we establish a framework that guides the evolution of the system towards achieving fairness objectives.

Based on the choice of A, the expected behaviour of the system defined in Equation (6) can be schematized via the following three situations: (a) with eigenvalues close to zero, the system converges slowly, making softer and less drastic choices; (b) with eigenvalues close to 1, the behaviour is highly rapid, indicating more aggressive choices; (c) with eigenvalues greater than 1, the theoretical dynamic system oscillates. However, since the minimum limits the oscillations, a stable behaviour is anticipated. We report the theoretical results in Appendix 4.1.

Approximating the Ideal Behavior The behaviour of the system, i.e., the metrics evolution over time, described in Equation (5) may undergo changes over time due to two main factors: *i)* the drift in the query distribution (change or shift in the characteristics, preferences, or distribution of the incoming queries over time), and *ii)* the set of actions taken to enforce fairness. While the first factor is beyond our control this is not true for the second factor. We can leverage this control to shape the system's evolution and approximate the desired behaviour outlined in Equation (6), which represents the ideal dynamics of the system.

Let's define the distance between the approximate and the ideal system evolution as:

$$\mathcal{L}(\bar{x}_t) = \|f(\bar{x}_t) - A\min(0, \bar{x}_t - \mu) - \bar{x}_t\|_2^2. \tag{7}$$

Formally, we want to approximate the evolution of the system in Equation (6) by finding the set of actions θ_t to perform at time step t in order to minimize the distance to the ideal behaviour; thus, we have to solve:

$$\arg\min_{\theta_t} \mathcal{L}(\bar{x}_t). \tag{8}$$

The cost function of the system, represented by Equation (8), can be tailored to the specific scenario being addressed.

We can then expand $\bar{x}_t$ according to its definition, and we can rely on a sample approximation for the expectation. For example, we can assume that the distribution drift over time is limited and use as sample the queries from the last N time steps; when $N = 0$, our approximation is not impacted by drift effects, but it is more affected by sampling noise (i.e. we treat a single query as representative for the full distribution); increasing N allows to adjust this trade-off.

With this strategy, Equation (8) can be customised as follows[2]:

$$\arg\min_{\theta_t} \left\{ \mathcal{L}(x_t) + \frac{\beta}{N} \sum_{i=1}^{N} \mathcal{L}(x_{t-i}^t) \right\}, \text{ with } x_{t-i}^t = x(\rho(q_{t-i}, \theta_t)). \tag{9}$$

The rationale behind the semantics of Equation (9) can be summarized with the following intuition: to limit the impact of drift-effects, we consider only one query per time step, and we approximate the expectation in Equation (3) by considering the N previous queries before q_t (current query arrived at time t) and treat them as if they arrived at time t. Therefore, during the optimization phase, the cost function is computed by considering the impact of the actions not only on q_t but also on the N previous queries (Equation (9)). However, since the previous queries are passed, their contribution is weighted with a parameter $\beta \in [0, 1]$, to lower their importance. It is worth mentioning that by changing N we can also control the computational efficiency of the system.

2.3. FAiRDAS grounding

The proposed approach is applicable to a wide range of scenarios where the objective is to promote fairness in the outcomes of user queries over time. To tailor the abstract framework FAiRDAS to a specific application all the system parameters should be selected. Specifically, the abstract system incorporates the following parameters and actions to be grounded:

- *Specific metrics of interest, along with the thresholds and rescaling factors, definition.* This involves the selection of metrics relevant to assessing fairness, as well as other metrics of interest, including those related to performance. Appropriate rescaling factors are applied to ensure comparability among these metrics. It is essential to include weights in the metric definition to indicate the importance of each metric within the specific scenario.
- *Ideal dynamical system definition.* The matrix of the dynamic system and the threshold vector are defined to represent the desired behaviour of the system. This involves defining the matrix A and the threshold vector μ as depicted in Equation (6).
- *Actions for achieving fairness definition.* Potential actions or interventions are identified to promote fairness within the system.
- *Cost function definition.* A cost function is defined to quantify the trade-offs or penalties associated with different outcomes or actions. This is the function that guides the

[2] The cost function in Equation (9) represents one possible approximation of the query expectation; for example, other distance metrics can be defined depending on the application, and the factor β can be time-dependent instead of constant to weight each past contribution x_{t-i} differently.

optimization process, taking into account the approximation for the expectation of the fairness metrics as outlined in equation 3.
- *Optimization method definition.* Select the appropriate method to solve the optimization problem, considering factors such as computational efficiency, accuracy, and scalability.

By following these steps, we can customize FAiRDAS to a specific application or domain, allowing for the effective enforcement of fairness or other metrics over time in the outcomes of user queries. The flexibility and adaptability of our approach enable its application in various contexts, providing a framework for addressing fairness concerns in query-based systems. The parameterization of the system in FAiRDAS empowers us to navigate the complexities and intricacies of rankings while ensuring fairness. It offers a versatile framework that accommodates the inherent non-smoothness of rankings, enabling effective and customizable handling of fairness considerations.

3. Empirical Evaluation

3.1. Case Study

In this section, we present a real-world case study [3] that focuses on a web platform providing a matchmaking service for AI experts. The platform aims to bridge the gap between customers (such as companies or other entities) seeking AI expertise to solve specific problems. Users of the platform describe the issues they need to address, along with any desired expertise. The platform then generates a ranked list of AI experts that best align with the user's query, providing them with a tailored response. The ranking process takes into consideration the relevance and suitability of the experts' expertise to the user's specific scenario. This ensures an optimal match between the user's requirements and the available AI experts.

Within this context, fairness is a crucial aspect that is given due consideration. Specifically, the sensitive attribute of country is acknowledged, prompting the development of fair algorithms to address this concern. In generating the ranked list of AI experts, our goal is to ensure fairness with respect to the country attribute. This entails considering the expertise and qualifications of the AI experts while giving proper weight to their skills in relevant AI subfields. Additionally, measures are implemented to mitigate any potential biases associated with the country attribute. These measures ensure that experts from all countries have an equal opportunity to be included in the ranked list. Formally, let each resource be characterized by two score vectors: $s_L \in \mathbb{R}^l$ and $s_C \in \mathbb{R}^c$. The vector s_L captures the expertise of AI experts across l subfields of AI. Each component of s_L represents the skill level of the expert in a specific subfield, and its value is confined within the range of $[-1, 1]$. On the other hand, s_C represents the country of the AI expert, encoded in a one-hot format, and serves as a protected attribute within our use case to guarantee fair nationality distribution. The queries are described by the same vectors of scores which reflect the user's requirements. It is important to note that the query includes both AI subfields (l) and preferred country (c) in $\mathbb{R}^{l+c}$ due to the specific nature of the case at hand. In this scenario, when selecting AI experts for a particular use case, it might be necessary to specify the language in which we can communicate with the expert. However, in typical cases,

[3] StairwAI Project.

the sensitive attribute cannot be directly provided by the user. It is crucial to understand that this particularity does not impact the mathematical foundation of the model; on the contrary, it shows its flexibility.

Dataset Generation For the experimentation phase, synthetic data was generated to facilitate better control over the level of an imbalance concerning the country attribute, allowing for the manipulation of bias levels. To achieve this, the synthetic data was designed to represent various countries in a controlled manner. By adjusting the parameters of the data generation process, the desired level of imbalance or bias in terms of country representation could be defined.

We generate three datasets for the experimental valuation, each consisting of 40 resources and 100 queries. Each data sample is defined by 9 AI subfields and belongs to one of 5 countries. The score vectors of the AI subfields are sampled from a uniform distribution, while the country vectors are sampled from a categorical distribution with $p_1, \ldots, p_5$ where the event probabilities were normalized. The synthetic datasets differ in the event probabilities of the resources country distribution to represent three levels of bias. In Figure 1, we show the scores distribution of the generated data. Figure 1(a) represents the distribution of the query scores, which is the same for all three datasets. The resources of the first dataset are uniformly distributed among the 5 classes, i.e., $p_i = 0.2 \ \forall \ i$ (Figure 1(b)); we refer to this dataset as *Balanced*. In the second dataset, hereinafter *Mild Unbalanced* dataset, we introduce a bias towards the attribute `Country3` by setting $p_3 = 0.4$ (Figure 1(c)), and in the third dataset, named *Strong Unbalanced*, we increase the bias with $p_3 = 0.8$ (Figure 1(d)).

Ranking Algorithm Given an incoming query $q \in \mathbb{R}^{l+c}$, the ranking algorithm computes the cosine similarity between q and each resource vector; the resources are ranked based on the resulting scores.

Metrics of Interest In this case study, we are interested in enforcing fairness over the protected attribute while preserving the ranking accuracy. As fairness metric we use the *Disparate Impact Discrimination Index* [10]. Given a sample $\{x_i, y_i\}_{i=1}^n$ including values for a protected attribute x and a continuous target value y, the Disparate Impact Discrimination Index is defined as:

$$\mathrm{DIDI}(x, y) = \sum_{v \in \mathcal{X}} \left| \frac{\sum_{i=1}^n y_i I(x_i = v)}{\sum_{i=1}^n I(x_i = v)} - \frac{1}{n} \sum_{i=1}^n y_i \right| \tag{10}$$

where $\mathcal{X}$ is the domain of x and $I(\psi)$ is the indicator function for the logical formula ψ.

To quantify the ranking accuracy we measure the cosine distance $\mathrm{d_{cos}}$ between the true rank vector r resulting from applying the ranking algorithm with no actions vector, and the rank vector r_θ computed by the ranking algorithm subject to an actions vector θ.

$$\mathrm{d_{cos}}(r, r_\theta) = 1 - \frac{r \cdot r_\theta}{|| \, r \, || \ \ || \, r_\theta \, ||} \tag{11}$$

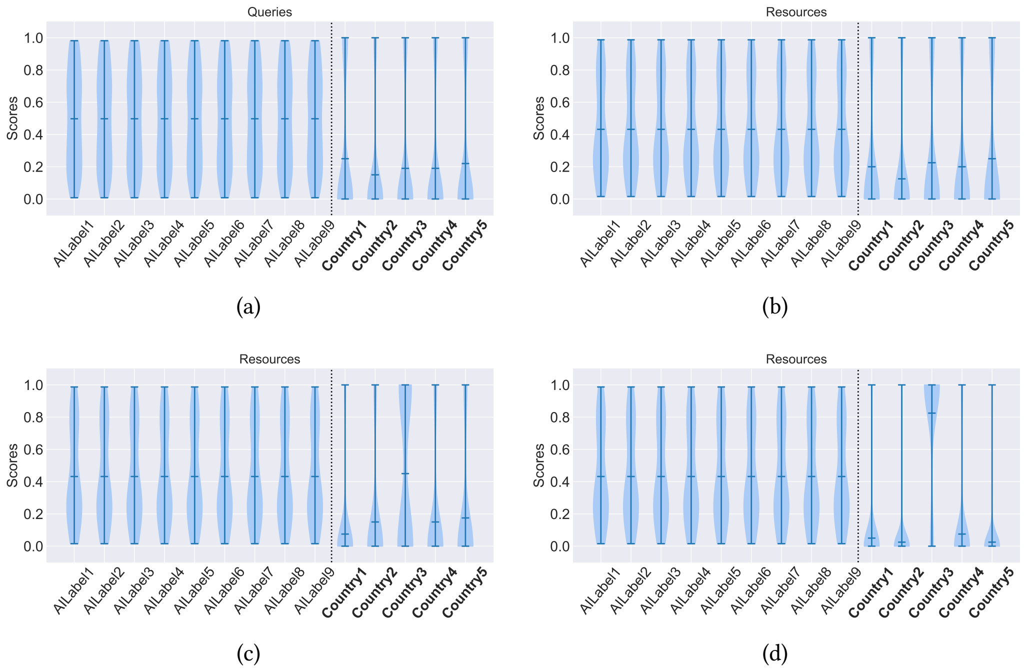


Figure 1: Query and resource scores distribution in the three datasets. The protected attribute is in bold. The query scores distribution common to all three datasets (a); while the resource scores distribution varies among *Balanced* (b), *Mild Unbalanced* (c) and *Strong Unbalanced* (d) datasets.

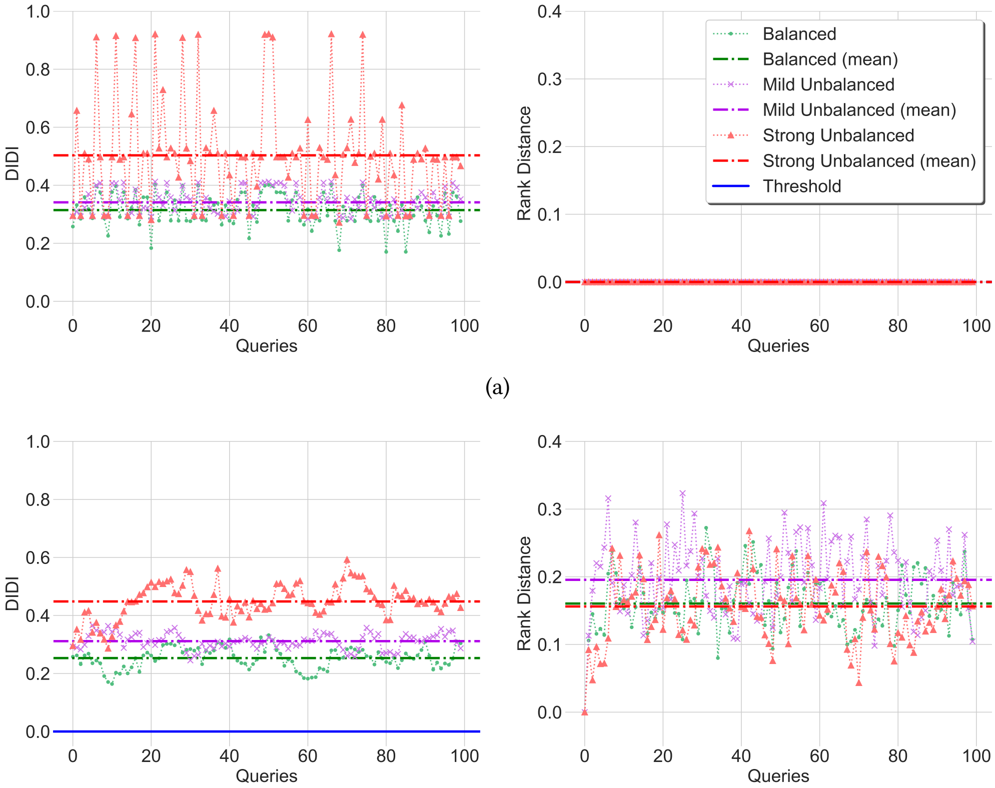
Ideal Dynamical System We adopt the dynamical system in Equation (6) with A a 2×2 diagonal matrix with eigenvalues equal to 0.5[4], and we vary the threshold components μ_i in the interval $[0, 1]$ to force different behaviours.

Cost Function As cost function, we adopt the $L2$-norm defined in Equation (7), and we approximate the metrics expectation by considering $N = 5$ previous queries and weighting their contribution with $\beta = 0.2$.

Set of Actions Our set of actions applies directly to the rank returned by the ranking algorithm: each resource in $\mathcal{R}$ can be either put at the bottom of the rank or kept in the current position based on the value of a binary variable. In particular, we associate a binary variable b_i to each resource indicating whether the $i - th$ resource should ($b_i = 1$) or should not ($b_i = 0$) be placed at the bottom of the ranking list. The optimizer seeks to determine which resources to place at the bottom by exploring the space $\{0, 1\}^m$, where m represents the number of resources. In other words, the algorithm searches for the optimal assignment of the m binary variables representing the action “place at the bottom”.

[4] The eigenvalues of 0.5 were selected as they fall within the stable range [0,2] discussed in the previous section. Further studies will be dedicated to finding the optimal eigenvalues for the specific scenario at hand.

Optimization Method To solve the optimization problem we use a random walk to search in the hyperspace $\Omega \subseteq \{0,1\}^m$. Starting with a random generated point the algorithm searches for a better solution among a set of candidates recursively. The new candidate points are generated by flipping each component of the current best solution with a decreasing probability p.



(a)

(b)

Figure 2: DIDI and $\mathrm{d}_{\cos}$ values for the three datasets without performing any action (a) and by setting the threshold component for the DIDI to be equal to 0 (b).

3.2. Experiments

The presented case study was applied to three datasets, and we varied the threshold components μ_i to enforce different behaviours[5]. To establish a reference benchmark, we computed the DIDI and $\mathrm{d}_{\cos}$ metrics for each query in the three datasets without taking any further action. Figure 2(a) illustrates the resulting curves, while Table 1 summarizes the mean values of the two metrics across all queries. As expected, the *Strong Unbalanced* dataset exhibited a higher average DIDI compared to the *Balanced* and *Mild Unbalanced* datasets. Furthermore, the fairness metrics displayed a more unstable behaviour in the *Strong Unbalanced* dataset when compared to the

[5] The source code to generate the datasets and reproduce the experiments is available at https://github.com/EleMisi/FAiRDAS under MIT license.

Table 1
Mean of the two metrics computed over the incoming of queries for each of the three datasets. Column 3, shows the mean of DIDI and $\mathrm{d_{cos}}$ computed without performing any action on the rank. Columns 4-7 refer to the experiments with four different thresholds.

		Thresholds $(\{\mathrm{DIDI},\mathrm{d_{cos}}\})$				
Dataset	**Metrics**	$\{-,-\}$	$\{0.0, 1.0\}$	$\{0.5, 0.2\}$	$\{0.4, 0.2\}$	$\{0.1, 0.1\}$
Balanced	DIDI	0.322	0.256	0.322	0.286	0.256
	$\mathrm{d_{cos}}$	0.0	0.162	0.0	0.098	0.162
Mild Unbalanced	DIDI	0.341	0.312	0.341	0.304	0.312
	$\mathrm{d_{cos}}$	0.0	0.186	0.0	0.129	0.186
Strong Unbalanced	DIDI	0.516	0.431	0.424	0.454	0.431
	$\mathrm{d_{cos}}$	0.0	0.158	0.162	0.177	0.158

other datasets.

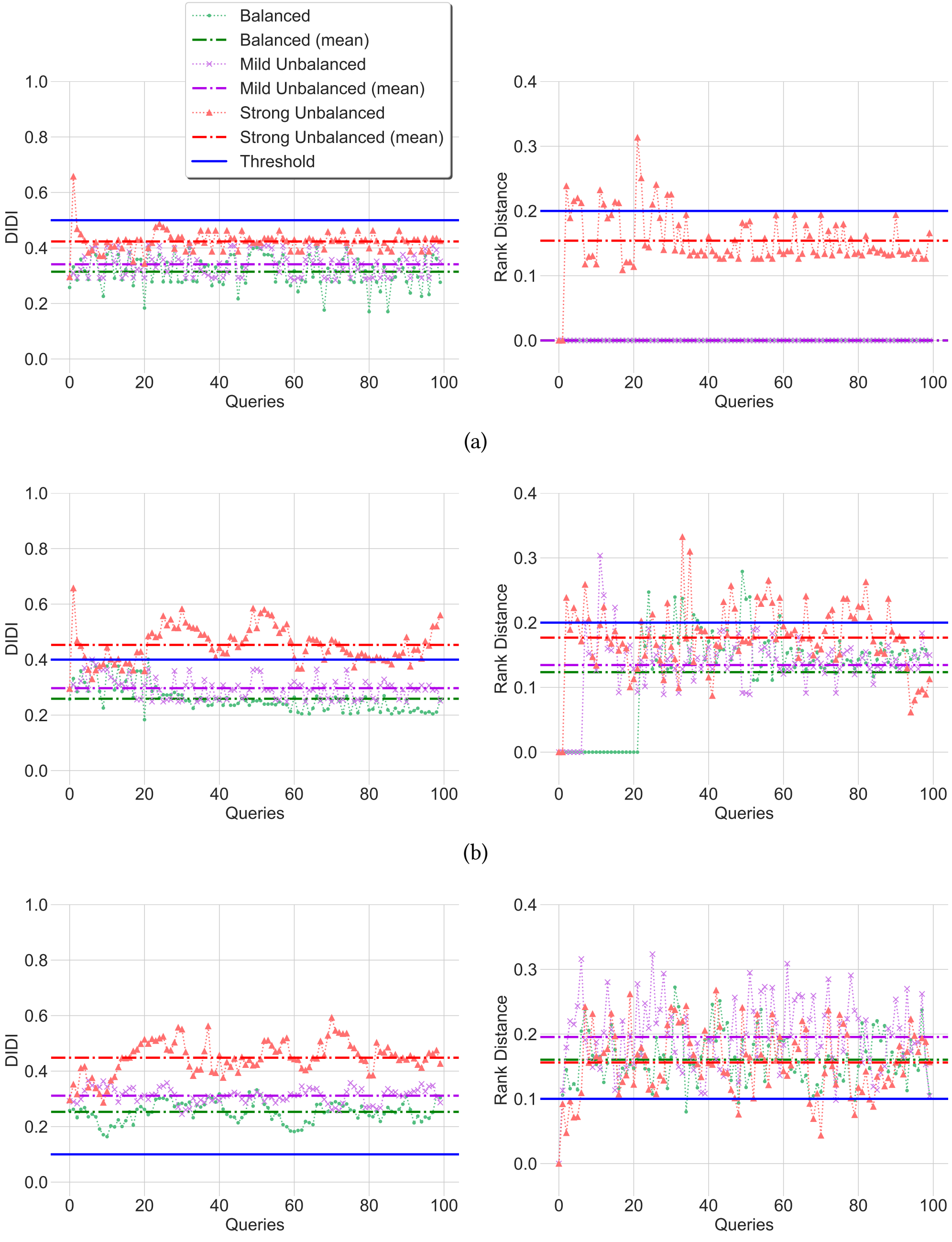
As an initial step, we aim to determine the minimum average DIDI achievable by employing FAiRDAS. For this purpose, we set the threshold component for DIDI to 0, while keeping the component for ranking accuracy at 1. By doing so, we allow FAiRDAS to prioritize the reduction of DIDI without imposing any constraint on the rank accuracy metrics. Notice that the opposite setting, where we want to maximize the rank accuracy without any requirements on the fairness metrics, is equivalent to not taking any actions, i.e., it is equivalent to the reference benchmark (Figure 2(a)). Figure 2(b) visualizes the resulting curves. As anticipated, applying FAiRDAS with $\mu = \{0, 1\}$ leads to an increase in $\mathrm{d_{cos}}$ for all datasets, while the DIDI decreases and exhibits a more stable behaviour.

In Figure 3, we present a comparison of the impact of FAiRDAS when applying three different thresholds. Specifically, in Figure 3(a), we depict the scenario where we set the thresholds for DIDI and $\mathrm{d_{cos}}$ as 0.5 and 0.2, respectively [6]. Since these thresholds exceed the mean values of DIDI and $\mathrm{d_{cos}}$ for the *Balanced* and *Mild Unbalanced* datasets, FAiRDAS does not take any action on incoming queries, and the rank quality remains unaffected. Conversely, in the case of the *Strong Unbalanced* dataset, the rank quality declines as FAiRDAS must implement measures to reduce DIDI.

In Figure 3(b), we maintain a constant threshold for $\mathrm{d_{cos}}$ while reducing the threshold for DIDI to 0.4. This new threshold is now lower than the average value of DIDI achieved by applying FAiRDAS in the initial experiment on the *Strong Unbalanced* dataset (Figure 2(b), red line). As a result, the system reaches an equilibrium above the threshold for the *Strong Unbalanced* dataset. Furthermore, FAiRDAS needs to take actions on the incoming queries of the *Balanced* and *Mild Unbalanced* datasets to ensure that the resulting rankings do not violate the fairness threshold. Consequently, the value of $\mathrm{d_{cos}}$ is affected in all the datasets.

In our final experiment, we examine the behavior of the system when faced with unattainable thresholds—both threshold components set to 0.1. As expected, the system is unable to meet

[6] It is important to note that these thresholds represent the maximum allowed values for identifying unfair behaviour and the maximum acceptable error in accuracy, respectively.



(a)

(b)

(c)

Figure 3: DIDI and $d_{\cos}$ values for the three datasets by applying three different vectors of thresholds.

the requirements but instead settles into an equilibrium state near the desired thresholds. The average DIDI values exhibited by the system align with the initial bias present in the three datasets: *Strong Unbalanced* being the most biased, while *Balanced* and *Mild Unbalanced* are closer in terms of DIDI. However, as FAiRDAS takes actions on the incoming queries to approach the desired fairness threshold, the quality of rankings deteriorates for all the datasets.

4. Conclusion

In this study, we proposed a novel approach for addressing long-term fairness in matchmaking and ranking systems by formalizing it as an abstract dynamic system. By modelling the evolution and interaction of fairness metrics over time, valuable insights into system behaviour, metrics interactions, and overall dynamics can be gained. This formalization enables a systematic analysis of system properties, such as stability, equilibrium, and convergence, and facilitates the exploration of trade-offs and tensions when optimizing multiple fairness metrics simultaneously.

One of the advantages of the proposed abstract dynamic system is its flexibility and customizability. It can be tailored to specific cases, allowing for targeted analysis and experimentation. Different metrics – including both fairness and performance metrics – can be considered simultaneously and easily parameterized within the system. Furthermore, the study includes a discussion and evaluation of the proposed approach in a real-world use case. By grounding the framework in a practical scenario, the effectiveness and applicability of the approach can be demonstrated, providing valuable insights and practical guidance for implementing fair ranking systems.

Overall, this study has the potential to advance the field of fairness in matchmaking and ranking systems.

Future Research Directions The present work represents a preliminary exploration of the topic, and there are several avenues for further research. One key area of focus will be the optimal selection of eigenvalues for the matrix A. It is essential to conduct in-depth studies to determine the most effective approach for selecting these eigenvalues and their impact on the overall system dynamics.

Additionally, a more comprehensive investigation of the system is necessary in terms of balancing conflicting fairness metrics and identifying any unreachable points. It will be interesting to explore methods for a priori identification of unreachable thresholds and visually analyze the system's behavior using phase diagrams. This analysis can provide insights into the dynamics and trade-offs between different fairness metrics.

Furthermore, an important aspect to study is how to choose the thresholds for different metrics, select their weights, and perform rescaling to make them comparable. Investigating methods for threshold determination, weight selection, and normalization techniques will contribute to refining the fairness-aware ranking system and enhancing its effectiveness.

Another important direction for future research is the integration of contextual fairness in the ranking systems. User-centric fairness is also an area that warrants further investigation. Involving users in shaping the fairness objectives and constraints of ranking systems can lead to more personalized and user-centric fairness approaches. Future studies should explore methods for actively engaging users in the fairness design process, allowing them to define their fairness preferences and customize the ranking system accordingly. Specifically, we will work on adapting the system in real-time based on user feedback and interactions, going beyond basic fairness constraints.

Transparency and explainability are crucial aspects of fairness in ranking systems. Future research efforts should be dedicated to enhancing the transparency and interpretability of ranking algorithms. Developing methods to explain the ranking decisions to users and providing them with insights into the fairness criteria employed can foster trust and understanding. Transparency and explainability also enable users to hold the ranking system accountable for its fairness outcomes.

Long-term impact assessment is another important research direction. Evaluating the effectiveness of fairness-aware ranking systems over extended periods is critical to ensure equitable outcomes and minimize unintended consequences. Future studies should assess how these systems affect various stakeholders, including users, content providers, and platform operators, and examine any long-term biases or disparities that may arise.

Furthermore, since we are operating within an experimental framework, a potential area for future investigation would involve identifying the most equitable ML algorithm based on the defined constraints (boundaries). Taking it a step further, examining the boundaries using diverse, polarized data sets would provide insights into the threshold at which a particular algorithm becomes ineffective (algorithm breaking point).

By addressing these future research directions, we can advance the field of fairness in ranking systems, mitigate biases and discrimination, and ensure more equitable and transparent outcomes for users and stakeholders alike.

Acknowledgments

The work has been partially supported by the AEQUITAS project funded by the European Union's Horizon Europe Programme (Grant Agreement No. 101070363), by the European Horizon 2020 Project StairwAI (GA 101017142), and by PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 8 "Pervasive AI", funded by the European Commission under the NextGeneration EU programme.

References

[1] R. Akerkar, Artificial intelligence for business, Springer, 2019.

[2] A. E. Roth, Who gets what–and why: the new economics of matchmaking and market design, Houghton Mifflin Harcourt, 2015.

[3] E. Pitoura, K. Stefanidis, G. Koutrika, Fairness in rankings and recommenders: Models, methods and research directions, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), IEEE, 2021, pp. 2358–2361.

[4] A. Singh, T. Joachims, Fairness of exposure in rankings, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2219–2228.

[5] M. Zehlike, K. Yang, J. Stoyanovich, Fairness in ranking: A survey, arXiv preprint arXiv:2103.14000 (2021).

[6] M. Zehlike, K. Yang, J. Stoyanovich, Fairness in ranking, part i: Score-based ranking, ACM Computing Surveys 55 (2022) 1–36.

[7] B. Hutchinson, M. Mitchell, 50 years of test (un) fairness: Lessons for machine learning, in: Proceedings of the conference on fairness, accountability, and transparency, 2019, pp. 49–58.

[8] A. Raj, M. D. Ekstrand, Measuring fairness in ranked results: An analytical and empirical comparison, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 726–736.

[9] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, L. Pizzato, Multistakeholder recommendation: Survey and research directions, User Modeling and User-Adapted Interaction 30 (2020) 127–158.

[10] S. Aghaei, M. J. Azizi, P. Vayanos, Learning optimal and fair decision trees for non-discriminative decision-making, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 1418–1426.

Appendix

4.1. Stability Analysis

Let's consider a discrete-time linear system with state equation:

$$x(k+1) = Mx(k) + Bu(k), \tag{12}$$

with M and B real-valued $n \times n$ matrices. Given $\lambda_i \in \mathbb{R}$ the eigenvalues of M, it is known that:

$$\begin{aligned} \text{if } |\lambda_i| < 1 \quad \forall i &\implies \text{asymptotic stability,} \\ \text{if } |\lambda_i| > 1 \text{ for some } i &\implies \text{unstability.} \end{aligned}$$

In the dynamic system defined in Equation (6), we have $M = \mathbb{I} - A$; thus, assuming A to be diagonal and positive definite, and denoting $\lambda_i^A \in \mathbb{R}^+$ the eigenvalues of A, we have that:

$$\begin{aligned} \text{if } 0 < \lambda_i^A < 2 \quad \forall i &\implies \text{asymptotic stability,} \\ \text{if } \lambda_i^A > 2 \text{ for some } i &\implies \text{instability.} \end{aligned}$$