The Initial Screening Order Problem

Jose M. Alvarez
University of Pisa
Pisa, Italy

Antonio Mastropietro
University of Pisa
Pisa, Italy

Salvatore Ruggieri
University of Pisa
Pisa, Italy

ABSTRACT

We investigate the role of the initial screening order (ISO) in candidate screening tasks, such as employee hiring and academic admissions, in which a screener is tasked with selecting k candidates from a candidate pool. The ISO refers to the order in which the screener searches the candidate pool. Today, it is common for the ISO to be the product of an information access system, such as an online platform or a database query. The ISO has been largely overlooked in the literature, despite its potential impact on the optimality and fairness of the chosen k candidates, especially under a human screener. We define two problem formulations describing the search behavior of the screener under the ISO: the best-k, where the screener selects the k best candidates; and the good-k, where the screener selects the k first good-enough candidates. To study the impact of the ISO, we introduce a human-like screener and compare it to its algorithmic counterpart, where the human-like screener is conceived to be inconsistent over time due to fatigue. In particular, our analysis shows that the ISO, under a human-like screener solving for the good-k problem, hinders individual fairness despite meeting group level fairness, and hampers the optimality of the selected k candidates. This is due to position bias, where a candidate's evaluation is affected by its position within the ISO. We report extensive simulated experiments exploring the parameters of the best-k and good-k problems for the algorithmic and human-like screeners. The simulation framework is flexible enough to account for multiple screening settings, being an alternative to running real-world candidate screening procedures. This work is motivated by a real-world candidate screening problem studied in collaboration with an European company.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**; • **Human-centered computing → Interaction design theory, concepts and paradigms**.

KEYWORDS

Fair set selection, position bias, search user behavior

1 INTRODUCTION

Candidate screening is a complex, human-dependent task. It consists of a decision-maker or user, which we refer to as the *screener*, tasked with choosing k candidates from a candidate pool. Common candidate screening processes include the evaluation of resumes for a job interview [38] or application packages for college admission [51]. The screener usually evaluates the candidate pool using limited information and under strict time constraints. Information access systems, such as online platforms like LinkedIn and database queries like Taleo, play a central role today in candidate screening by allowing screeners to search more efficiently a candidate pool. Enabled by Machine Learning (ML) [40], these information access systems often present candidates according to an estimated relevance or, at least, according to a relevant characteristic chosen by the screener [5]. An industry around algorithmic candidate screening has emerged in recent years [53], though poor fairness results [19, 30, 33, 42, 43, 48, 49, 54] and calls to consider the behavior of the human user [2, 9, 12, 45] continue to drive research on the social impact of these information access systems.

In this paper, we investigate the role of the *initial screening order* (ISO) in the fair set selection problem implicit in the task of candidate screening. The ISO refers to the order in which the candidates appear in the candidate pool. It can be chosen by or provided to the screener enabled via an information access system [17]. We develop a utility-based framework to understand the search behavior of the screener when going over the ISO, and use it to implement extensive simulations that study the influence of the ISO. We find that the ISO can impact the optimality (i.e., choosing the best candidates) and fairness (i.e., treating similar candidates similarly) of the selected set of k candidates, especially when the screener is human. This is mainly because of the *position bias* inherent to the ISO. Here, position bias refers to the penalty (or premium) a candidate experiences due to where it falls on the ISO, as humans are predisposed to favor the items placed at the top of a list [3, 4]. We motivate the ISO problem further in Section 1.1 based on our collaboration with an European company.

In Section 2, we describe how the screener searches the ISO for an optimal and fair set of k candidates w.r.t. a representational quota q of protected candidates. We define two problem formulations. In the best-k the screener selects the k top candidates, whereas in the good-k the screener selects the k first good-enough candidates fitting some minimum candidate quality measure. We devise algorithmic solutions for both problems. The good-k is noteworthy as it allows the screener to partially search the candidate pool; indeed, the set selection problem formulation often assumes a full search. In Section 3, we analyse the algorithmic and human-like screeners to understand the impact of the ISO when a human is involved. The former refers to a consistent screener; the latter refers to an inconsistent screener whose evaluation of candidates suffers over time due to the fatigue of performing a repetitive task [28, 29]. In

Section 4, we enhance our analysis of these two screeners through simulations that mimic multiple screening settings. Our results confirm the role of position bias inherent to the ISO and raise new fairness concerns. For instance, we find that the human-like screener can violate individual fairness by not evaluating similar candidates similarly [15] while still meeting q, and that the algorithmic screener can miss the best candidate depending on its search procedure. In Section 5, we conclude by discussing the limitations and extension of our work.

Our work is the first to formalize the ISO problem. Our *main contributions* are threefold. *First*, we formalize the role of the ISO in two search behaviors of the screener with the best-k and good-k problems. *Second*, we introduce a human-like screener and compare it theoretically and experimentally to its algorithmic counterpart. *Third*, we provide a flexible simulation tool for studying the ISO problem able to inform practitioners without needing to run real-world screening scenarios.

1.1 Qualitative Background

This work borrows from a previous collaborative effort at a European Fortune Global 500. We refer to this company as G. The purpose was to study G's hiring process as an algorithmic fairness problem. We worked closely with Human Resources (HR), focusing on candidate screening. In this phase of G's hiring process, an HR officer reduces the candidate pool for a job opening into a smaller pool of suitable candidates based on each candidate's profile. The candidate pool was stored in Oracle's Taleo, a hiring platform used by HR officers to, among other things, obtain the ISO. For more details on our collaboration, see Appendix A.

The following *five stylized facts* summarize key practices by the HR officers (henceforth, screeners) that motivated the ISO problem. ***G1*** *Varying ISOs.* Screeners chose the ISO. The choice was restricted by the sorting fields of the hiring platform, such as using the candidates' last name. ***G2*** *Two ways to search the candidate pool.* Two search practices became apparent: full or partial search of the candidate pool. ***G3*** *Meeting the set of minimum basic requirements.* Screeners were able to differentiate candidates relative to each other, but their focus was on finding candidates that met these requirements. Order within the selected k candidates was not necessarily important. ***G4*** *Diverse suitable candidates.* Fairness goals already existed in the form of representation quotas, often around gender, that were enforced by the screeners. ***G5*** *A consistent notion of time.* Screeners aimed at spending one minute per candidate.

Although G1-5 are specific to G, they highlight salient aspects of real-world candidate screening problems likely to hold in similar settings involving humans searching a pool of candidates (see, e.g., [16, 22, 36, 37, 44]). G4-5 are standard to the fair set selection problem, especially under an algorithmic screener, while G1-3 introduce new considerations to such problem formulation, especially under a human screener. We come back to G1-5 in Sections 2 and 3.

1.2 Related Work

This work focuses on the screener's search behavior given an ISO. The creation of the ISO itself can be modeled as a fair set selection or, if order matters, fair ranking problem [39, 57, 58] in which the goal is to learn a fair ISO from data using, e.g., probability-based [55, 56] and exposure-based [4, 26] methods. Similarly, other works, e.g., [23, 31, 32] study how information access systems, like online job markets, enable the creation of candidate pools and, in turn, the ISO. Instead, motivated by our experience at G, we are interested in how a screener, particularly a human one, searches the ISO. With some exceptions [16, 37], most fairness works avoid studying the user of the ISO, treating, e.g., position bias as a technical bias. Our screener-centric approach is similar to web search click models that were the first to formalize [14] and test [22, 36, 44] how users search over an ISO. Different from these works, we consider a user that "clicks" on more than one item, and formalize such user under a utility-maximizing framework with fairness constraints, relating the insights from these works to candidate screening as well as to past fair set selection works [50].

The works on click models provide empirical evidence for the position bias, though they precede considerably the fairness literature. Notably, two works provide recent empirical evidence for position bias in candidate screening due the ISO with a focus on individual fairness [15]. Echterhoff et al. [16] collaborate with a college to study anchoring bias [52] in admissions officers. They find that the same applicant is better off if it is preceded by worst rather than better applicants, and propose an algorithm that balances out the anchoring bias when presenting applications to the admissions officer. Pei et al. [37] also collaborate with a college to study how the platforms used by professors when evaluating homeworks affects the students' grades. They run experiments varying the order in which the homeworks are presented by the platform, and show that the default alphabetical order unfairly rewards students with the same work quality due to their last names. We differ from these two works by defining the ISO as a parameter in the problem formulation. We also do not work with empirical data, and instead provide a simulations framework flexible enough to capture multiple screening scenarios, including those in [16, 37].

Our best-k and good-k formulations (Section 2.2) belong to the fair set selection literature [10]. The reference problem is the secretary problem (SP) [21] where we select a candidate in a randomly ordered sequence committing irrevocably to the acceptance or rejection decision after each evaluation.[1] The past literature has already analyzed the fairness implication of the SP, even in the k-choice extension [50]. However, our best-k and good-k formulations, differently from the SP, focus on the screening process, not the interview phase. Formally, we assume an offline set selection, meaning each candidate is individually evaluated by the screener and each decision is not irrevocable. Such setting emphasizes the role of the ISO. Notably, past works focusing on the SP online setting assume the additive utility we employ in the best-k formulation [34, 50]. This fact shows how peculiar the good-k problem formulation is relative to the previously analyzed settings.

2 SEARCHING THE POOL OF CANDIDATES

We formulate the fair set selection problem, in which a decision-maker selects a set of items from a population, by considering the *initial screening order* (ISO). Here, the candidates for a job represent the items and the screener evaluating their profiles represents the

[1] When order matters, the reference problem is the top-k selection [20].

decision-maker. Let the ISO be the product of an information access system specific to hiring; we treat it as given.

2.1 Setting

Let us onsider a *candidate pool* C of n candidates, where each *candidate* c is described by the *vector of* p *attributes* $\mathbf{X}_c \in \mathbb{R}^p$ and the *protected attribute* W_c. We assume that W is binary, such that $W_c = 1$ if c belongs to the protected group and $W_c = 0$ otherwise; we can relax this assumption if needed. The candidates are evaluated by a *screener* $h \in \mathcal{H}$, where $\mathcal{H}$ denotes the set of available screeners. The following variables refer to a specific h. The goal of h is to obtain a *set of* k *selected candidates* $S^k \in [C]^k$, with $[C]^k$ denoting the set of k-subsets of C, based on each candidate's application profile as summarized by the tuple $(\mathbf{X}_c, W_c)$. Candidate evaluation occurs when h uses an *individual scoring function* $s\colon \mathbb{R}^p \rightarrow [0,1]$, such that $s(\mathbf{X}_c)$ returns the score of c, and h cannot use W_c when scoring c. The higher the score, the better c fits the job.

The screener h explores the candidate pool C in a specific order. We denote the *set of total orderings of candidates* in C by Θ. An *order* $\sigma \in \Theta$ maps an integer $i \in \{1, \dots n\}$ to a candidate $c \in C$, indicating that c occupies the i-th position according to σ, with notation $\sigma(i) = c$ and vice-versa $\sigma^{-1}(c) = i$. Importantly, the screener explores C under the ISO $\theta \in \Theta$, which represents the order chosen by or, alternatively, provided to h before searching C (recall, G1 in Section 1.1) via an information access system. The screener is not required to explore the entirety of C, meaning h can either fully or partially explore C given θ (recall, G2 in Section 1.1). We assume that the screener *respects* θ, meaning:

$$c_1 \in C \text{ is evaluated before } c_2 \in C \text{ only if } \theta^{-1}(c_1) < \theta^{-1}(c_2). \quad (1)$$

2.2 Two Problem Formulations

We formulate two utility-based fair set selection problems for h with the shared objective of achieving an optimal and fair set S^k. Under the *best-k* formulation, S^k represents *the fair best k candidates* in C according to h; we denote it as S^k_{best}. Under the *good-k* formulation, S^k represents *the fair first good-enough k candidates* in C according to h; we denote it as S^k_{good}. The key difference between the two is that the best-k requires a full search of C while the good-k allows for a partial search of C under the ISO θ.

How we define optimality, as shown in Sections 2.2.1 and 2.2.2, determines the best-k and good-k problems. For fairness, we define the *representational quota* $q \in [0,1]$ as the desired fraction of protected candidates in S^k, and use the fraction $f(S^k) \in [0,1]$:

$$f(S^k) = \frac{\left|\{c \in S^k \text{ s.t. } W_c = 1\}\right|}{k} \quad (2)$$

for h to meet q when deriving S^k by satisfying the condition $f(S^k) \geq q$. The unconstrained version is achieved by $q = 0$. We view q as a policy enforced by h to achieve a diverse S^k (recall, G4 in Section 1.1). It is a statement on the composition of S^k, not a statement on the ordering of protected candidates within S^k.[2]

2.2.1 *Best-k.*

The screener h finds the set of best k candidates in C given the fairness constraint q while respecting the ISO θ (1). Here, h needs to evaluate the complete C since it must score and rank all candidates according to the individual scoring function s before choosing the ones with the highest scores and that satisfy q.

We view the goal in terms of maximizing a utility for h. We define *utility* as the benefit derived by h from selecting k candidates. Formally, utility is a function $U^k\colon [C]^k \times \Theta \rightarrow \mathbb{R}$. The simplest expression for U^k is to add the scores of the selected candidates:

$$U^k_{add}(S^k, \theta) = \sum_{c \in S^k} s(\mathbf{X}_c) \quad (3)$$

rationalizing that h maximizes its utility by selecting the k most suitable candidates given θ. Notice, though, that θ in (3) does not affect the evaluation of S^k due to the commutative property of addition. Under (3), we define **the best-*k* problem** as:

$$\begin{aligned} \underset{S^k \in [C]^k}{\arg\max} \quad & U^k_{add}(S^k, \theta) \\ \text{s. t.} \quad & f(S^k) \geq q \end{aligned} \quad (4)$$

with its solution as S^k_{best}. In the presence of tied scores, S^k_{best} may not be unique. In such a case, we consider any solution. We emphasize that (3) is not the only possible model for the utility of h and alternative models, such as exposure discounting [47], can be considered for (4). We leave this for future work.

2.2.2 *Good-k.*

The screener h finds k candidates in C that meet a set of *minimum basic requirements* ψ (recall, G3 in Section 1.1) given the fairness constraint q while respecting the ISO θ (1). We represent ψ as a *minimum score*, such that h deems a candidate $c \in C$ as eligible for being selected if $s(\mathbf{X}_c) \geq \psi$. Unlike the best-$k$ formulation, here h is not required to evaluate the whole C as it is enough to find the first k candidates that are good enough according to ψ and that satisfy q.

We still view the goal in terms of maximizing a utility for h. We need to, however, define an alternative utility function to (3) that ensures h stops searching C after finding the *k-th* good-enough candidate according to ψ. We define the following expression:

$$U^k_{\psi}(S^k, \theta) = \begin{cases} k - \sum_{c \in S^k} p(c, S^k, \theta) & \text{if, } \forall c \in S^k,\ s(\mathbf{X}_c) \geq \psi \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

with the *penalty function* defined as:

$$\begin{aligned} p(c, S^k, \theta) = \mathbb{1}\{\exists\, c' \in C \setminus S^k \\ \text{s.t. } \theta^{-1}(c') < \theta^{-1}(c) \wedge s(\mathbf{X}_{c'}) \geq \psi \wedge W_{c'} = W_c\}. \end{aligned} \quad (6)$$

Under (5), h wants to find as quickly as possible the *k-th* suitable candidate without wanting to check whether the *k-th* + 1 candidate is also suitable. This is because, for a candidate c, (6) looks for another candidate of the same group as c and meeting ψ, who occurs before c under θ but who has not been selected into S^k. It models the "wasted effort" in choosing a candidate occurring after another one meeting all the same requirements. At worst, there are k penalties. Under (5), we define **the good-*k* problem** as:

$$\begin{aligned} \underset{S^k \in [C]^k}{\arg\max} \quad & U^k_{\psi}(S^k, \theta) \\ \text{s. t.} \quad & f(S^k) \geq q \end{aligned} \quad (7)$$

[2] For $k = 10$ and $q = 0.5$, e.g., the fair screener would need to derive S^k with 50% of protected candidates though in no particular order within S^k.

```
Algorithm 1 ExaminationSearch
Require: $n, \theta, k, q$
Ensure: $S^k_{best}$
1:  $q^* \leftarrow \texttt{round}(q \cdot k); r^* \leftarrow k - q^*$
2:  $scores \leftarrow [s(\mathbf{X}_c)\ for\ c = \theta(1), \ldots, \theta(n)]$
3:  $\tau \leftarrow \texttt{argsortdesc}(scores)$
4:  $i \leftarrow 1; k^* \leftarrow 0; Q \leftarrow \{\}; R \leftarrow \{\}$
5:  while $k^* < k$ do
6:      $c \leftarrow \theta(\tau(i)); i \leftarrow i + 1$
7:      if $W_c == 0$ and $\texttt{len}(R) == r^*$ then
8:          continue
9:      $k^* \leftarrow k^* + 1$
10:     if $W_c == 1$ and $\texttt{len}(Q) < q^*$ then
11:         $Q \leftarrow Q \cup \{c\}$
12:     else
13:         $R \leftarrow R \cup \{c\}$
    return $Q \cup R$
```

```
Algorithm 2 CascadeSearch
Require: $n, \theta, k, q, \psi$
Ensure: $S^k_{good}$
1:  $q^* \leftarrow \texttt{round}(q \cdot k); r^* \leftarrow k - q^*$
2:  $i \leftarrow 1; k^* \leftarrow 0; Q \leftarrow \{\}; R \leftarrow \{\}$
3:  while $k^* < k$ do
4:      $c \leftarrow \theta(i); i \leftarrow i + 1$
5:      if $W_c == 0$ and $\texttt{len}(R) == r^*$ then
6:          continue
7:      $Y_c \leftarrow s(\mathbf{X}_c)$
8:      if $Y_c \geq \psi$ then
9:          $k^* \leftarrow k^* + 1$
10:         if $W_c == 1$ and $\texttt{len}(Q) < q^*$ then
11:             $Q \leftarrow Q \cup \{c\}$
12:         else
13:             $R \leftarrow R \cup \{c\}$
    return $Q \cup R$
```

Figure 1: The search procedures for, respectively, the best-k and good-k problems. Due to the lack of fatigue, they represent the algorithmic screener h_a. For the human-like screener h_h, we only need to track the accumulated Φ, draw ϵ from it, and add ϵ when computing the individual scores in line 2 for Algorithm 1, and line 7 for Algorithm 2.

with its solution as $S^k_{good}(\psi)$ or, if there is no ambiguity on ψ, simply as S^k_{good}. If the fairness constraint is strengthened to a fixed quota, $f(S^k) = q$, the solution is unique; in the general case, $f(S^k) \geq q$, there can be two solutions but with different fractions of the protected group. See Example B.1 in Appendix B.1 for details. We emphasize that (5) is not the only utility model for (7). Other models are possible as long as they describe the partial search. For more details, we motivate (5) in Appendix B.1 by presenting a simpler utility model without (6) and showing its failure to stop h.

Remark 1. *The ISO θ can influence the screening process under the good-k problem (7) due to the potential partial search of C by h, affecting which candidates are selected. This remark holds without assuming anything about h.*

To observe Remark 1, let $k = 1$ and assume two candidates such that $s(\mathbf{X}_{c_1}) \geq \psi$ and $s(\mathbf{X}_{c_2}) \geq \psi$. A θ such that $\theta^{-1}(c_1) = 1$ and $\theta^{-1}(c_2) = 2$ would imply that c_1 is considered eligible and is selected before h even evaluates c_2. Conversely, a reverse θ such that $\theta^{-1}(c_1) = 2$ and $\theta^{-1}(c_2) = 1$ would imply the opposite.

2.3 Two Search Procedures

We present two procedures describing the baseline search behavior of the screener under the best-k and good-k problems.

The ***ExaminationSearch*** procedure, or **Algorithm 1**, solves the best-k problem, returning S^k_{best} for given n (candidates) and θ (ISO), and parameters k (subset size) and q (group fairness constraint). First, line 2 calculates the minimum number q^* of candidates from the protected group to be selected, and the maximum number of candidates r^* not in that quota. Then, candidates are considered by descending scores, using the `argsortdesc` procedure (lines 2-3). The loop in lines 5-13 iterates until k candidates are found. The loop adds candidates to the sets Q and R: Q are candidates in the quota of the protected group; R are candidates not in that quota tat can be non-protected or protected. A non-protected candidate can be only added to the R set, thus line 7 checks if there is still room in R to do this. A protected candidate is added to the quota set Q if there is room (lines 10-11) or to the other set R otherwise (lines 12-13). Finally, the procedure returns the candidates in the quota set Q or in the other set R. The result of the *ExaminationSearch* procedure maximizes (4), as candidates are added in decreasing score, while keeping the fairness constraint through the quota management.

The ***CascadeSearch*** procedure, or **Algorithm 2**, solves the good-k setting, returning S^k_{good} for given n and θ, and parameters k, q and ψ (minimum basic requirement). The difference with the *ExaminationSearch* procedure consists in strictly following θ (line 4) and checking ψ (line 8) before adding a candidate to the quota set Q or to the other set R. The result of the *CascadeSearch* procedure maximizes (7), as no penalty is accumulated in the loop. This is because, a non-protected candidate ($W_c = 0$) is not added only if there is no room in R (and R never gets smaller to allow for more room later on), while a protected candidate ($W_c = 1$) is not added only if it does not meet ψ in line 8 and, thus it cannot be counted for the penalty.

Our aim is not to provide novel optimal algorithms, but to study the the screener's search behavior of C under θ. Therefore, we move from the optimality analysis of the two algorithms (e.g., [20]), and focus on modeling the search behavior of h when solving the two problems. Following from Section 1.2, we note that both algorithms are inherently sequential and can be applied online because we aim to model a human-like screener (described in the next section) that operates sequentially. Yet, in both settings, the applicants are disclosed according to θ and not to the score, differently from past set selection works (e.g., [50]).

3 THE HUMAN-LIKE SCREENER

To study the human interaction with the *initial screening order* (ISO), we distinguish two kinds of screeners h based on the proneness to error when evaluating the candidate pool: h is an *algorithmic screener*, denoted by $h_a \in \mathcal{H}_a$, if it can consistently evaluate C;

whereas h is a *human-like screener*, denoted by $h_h \in \mathcal{H}_h$, if its fatigue hinders the consistency of its evaluation of C.

3.1 Fatigue and Fatigued Scores

We first introduce a *time component* to study these two screeners. Let t denote the discrete unit of time that represents how long h takes to evaluate a candidate $c \in C$. We assume that t is constant (recall G5 Section 1.1), implying that time itself cannot be optimized by h. We track time along θ, meaning h evaluates the first candidate that appears in θ at time $t = 1$, and so on. Time t, thus, ranges from 0 to n at maximum. We then introduce a *fatigue component* $\phi(t)$ specific to h_h as a function of t and model the *accumulated fatigue* $\Phi\colon \{0, \dots, n\} \to \mathbb{R}$, with $\Phi(0) = 0$. The discrete derivative of Φ, that is, $\phi(i) = \Phi(i) - \Phi(i-1)$, defined for $t \geq 1$, is the effort of h_h to examine the t-th candidate. How we define Φ conditions the effect of fatigue on our analysis of h_h. We make the simplest modeling choice for ϕ by assuming that *fatigue accumulates linearly over time*, or $\phi(t) = \lambda$ so that $\Phi(t) = \lambda \cdot t$, meaning h_h becomes tired over time at a constant pace.

How does fatigue materialize for h_h? We model the effect of fatigue on h_h through the *fatigued score*:

$$s_{h_h}(\mathbf{X}_c) + \epsilon \tag{8}$$

where ϵ is a random variable dependent on Φ that quantifies the deviation from the *truthful score* $s_h(\mathbf{X}_c)$. We model ϵ using *two modeling choices* at a given t. ***First modeling choice:*** ϵ_1 is a centered Gaussian, and the fatigue affects only its variance. Formally, $\epsilon_1 \sim \mathcal{N}(0,\, v(\Phi(t-1)))$, where $v\colon \mathbb{R} \to \mathbb{R}$ defines the variance of ϵ_1 as an increasing function of Φ. ***Second modeling choice:*** ϵ_2 as an uncentered Gaussian, whose mean is a decreasing function of the fatigue. Formally, $\epsilon_2 \sim \mathcal{N}(\mu(\Phi(t-1),\, v(\Phi(t-1)))$, where $\mu\colon \mathbb{R} \to \mathbb{R}$ is a decreasing function rather than a constant of Φ.

Intuitively, under ϵ_1, h_h tends to overscore or underscore candidates over time, introducing both negative and positive bias (i.e., fatigue as “less attention” when evaluating more candidates) over time; under ϵ_2, instead, h_h tends to underscore the candidates (i.e., fatigue as “less effort” when evaluating more candidates) over time, introducing always a negative bias. With both ϵ_1 and ϵ_2, we capture two realistic biased settings driven by the ISO θ. We assume that h_h is unaware of its fatigue, representing an unconscious bias due to, e.g., performing a repetitive tasks over time [28, 29].

Remark 2. *Φ implies that h_h evaluates identical candidates c_1 and c_2 differently under θ at t_1 and t_2, as long as $\Phi(t_1) \neq \Phi(t_2)$ and regardless of whether h_h is solving for the best-k or good-k problem.*

Algorithms 1 and 2 represent h_a as there is no notion of biased scores. To represent the human-like screener h_h, we must track Φ over time and draw ϵ_1 (or ϵ_2) to compute the fatigued scores (8) of h_h at time t. The only changes are to line 2 in Algorithm 1 and line 7 in Algorithm 2, where the score computed for candidate c is biased by ϵ_1 (or ϵ_2). We present the human-like versions of the search procedures in Appendix B.2 as Algorithms 3 and 4; though these two can be observed using Figure 1.

3.2 Position Bias Implications

No two candidates occupy the same position in the ISO θ. With this fact in mind, we now analyse the fairness and optimality implications of the position bias implicit to θ. For concreteness, we make *two assumptions*. ***A1****: We assume that θ is independent of the protected attribute W*, meaning that how candidates appear in θ contains no information about W. ***A2****: We assume that the individual scoring function s is able to evaluate any candidate c fairly and truthfully*, meaning $s(\mathbf{X}_c)$ captures no information about W_c and only information about the suitability of c. Under *A1* and *A2*, we can control for other biases, such as measurement error in s, and focus on the position bias coming from θ.

We start with the fairness implications for both best-k and good-k problems. Given θ, it is important to distinguish between the group fairness constraining h (i.e., the quota q) and the individual fairness violation when h fails to evaluate similar candidates similarly [15]. Regarding group-level fairness, both h_a and h_h are fair in solving for (4) and (5) by satisfying $f(S^k) \geq q$. This point is clear for h_a in Algorithms 1 and 2 as there is no fatigue involved. The same holds for h_h in Algorithms 3 and 4 because the error on the score does not affect the evaluation of q. Here, we have that the expected error $\mathbb{E}[\epsilon \mid W_c = 1, \theta] = \mathbb{E}[\epsilon \mid W_c = 0, \theta]$, regardless of ϵ_1 or ϵ_2 for h_h. The fatigue and, thus, the fatigued scores are, on average, shared across protected and non-protected candidates.

Distinguishing between h_a and h_h becomes important under individual-level fairness because h_a ensures it, while h_h violates individual fairness. A candidate's position in θ influences the amount of error made by h_h when evaluating that candidate. Similar candidates will not be evaluated similarly due to the unequal accumulation of fatigue experienced by h_h when searching θ. For ϵ_2, e.g., even if $\mathbf{X}_{c_j} = \mathbf{X}_{c_i}$ but $j > i$, the score of j is less, on average, than the one of i, and i has an unfair premium over j from θ.

We now consider the optimality implications for both best-k and good-k problems. Recall that each problem, due to its own utility model, has different optimal solutions. It follows that h_a reaches the optimal solution in both problems as the absence of fatigue enables h_a to consistently judge suitable candidates. The opposite holds for h_h due to the inconsistent scoring of candidates ascribed by the accumulated fatigue. The biased scores not only violate individual fairness, but also lead h_h to misjudge candidates, eventually choosing the wrong ones when searching θ.

To summarize, h_a *reaches the optimal and fair solution for both best-k and good-k problems.* Moreover, h_a *guarantees individual fairness in both problems.* Instead, h_h *reaches the fair but sub-optimal solutions for both best-k and good-k problems.* Moreover, h_h *does not guarantee individual fairness in both settings.* Significantly, under the h_h, the position of a candidate in θ matters. It impacts whether a candidate, depending on the search strategy, is evaluated or not (Remark 1) and, if so, is evaluated fairly or not (Remark 2). These results only worsen when relaxing *A1* and *A2*. We explore these insights further by considering multiple hiring settings for h_a and h_h in the next section.

4 AN EXPERIMENTAL FRAMEWORK

We introduce a flexible experimental framework in R [41], based on Monte Carlo simulations, to study the implications of the *initial*

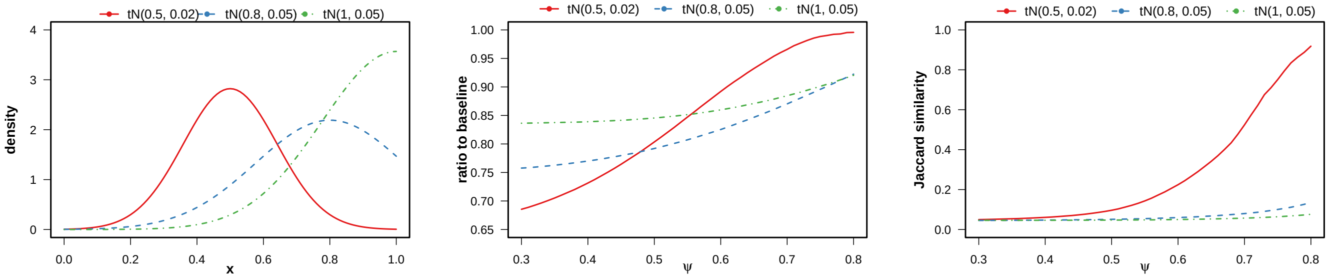

Figure 2: Left: density of experimental candidate score distributions, with scores ranging from 0 to 1. Center: RtB for different score distributions and the setting $n = 120$, $k = 6$, $q = 0.5$, and $\theta \perp\!\!\!\perp s$ at the variation of the minimum score ψ. Right: JdS for different score distributions and the setting $n = 120$, $k = 6$, $q = 0.5$, and $\theta \perp\!\!\!\perp s$ at the variation of the minimum score ψ.

screening order (ISO) for the best-k and good-k problems. It can handle different screening scenarios under the algorithmic h_a or human-like h_h screeners. Visit the following GitHub repository for the code: https://github.com/cc-jalvarez/initial-screening-order-problem/tree/main.

4.1 Setup

4.1.1 Generating the sample. We assume a sample consisting of n triplets $\{(s(\mathbf{X}_{c_i}), \theta(c_i), W_{c_i})\}_{i=1}^{n}$ drawn from a probability distribution with domain $\mathcal{G}_n \times \mathbb{R}^n \times \{0,1\}^n$, where $\mathcal{G}_n$ is the set of all permutations of $\{1, \ldots, n\}$. Each sample represents a candidate pool C sorted according to an ISO θ.

For the candidate scores, $s(\mathbf{X}_{c_i})$, we consider *three distributions* to model scenarios in which top candidates, i.e., top scores, occur with different probabilities in C. All three distributions use the truncated normal, $tN(\mu, \sigma)$ [7], with values bounded in $[0, 1]$. These scenarios, illustrated in Figure 2 (left), are:

- *Symmetric distribution of scores* (in red) defined by $\mu = 0.5$ and $\sigma = 0.02$, implying that top candidates occur with a very low probability in C.
- *Asymmetric distribution of scores* (in blue) defined by $\mu = 0.8$ and $\sigma = 0.05$, implying that top candidates occur with a higher probability and median value (≈ 0.75) in C compared to the previous scenario.
- *Increasing distribution of scores* (in green) defined by $\mu = 1$ and $\sigma = 0.05$, implying that top candidates occur with an even higher probability and median value (≈ 0.85) in C compared to the previous scenarios.

These thee scenarios have implications, in particular, for the good-k problem where we set the minimum score ψ and the screener is not required to explore all of C under θ (Remark 1). Setting a large ψ makes the screening process highly selective in the first scenario, less selective in the second scenario, and not selective at all in the third scenario, representing candidate pools with different candidate quality on average.

For the ISOs, $\theta(c_i)$, we consider *two settings* in which a given θ relates or not to $s(\mathbf{X}_{c_i})$, allowing us to explore θ as a product of different information access systems:

- *θ is generated randomly and independently from the candidate scores.* Formally, $\theta \perp\!\!\!\perp s$.
- *θ is generated randomly with a correlation ρ with the candidate scores*, where ρ is the Spearman's rank correlation of the pairs $\{(\theta(c_i), s(\mathbf{X}_{c_i}))\}_{i=1}^{k}$.[3] Formally, $\theta \not\perp\!\!\!\perp s$.

Under $\theta \perp\!\!\!\perp s$, θ carries no information about the candidate quality in C. It captures settings where the screener uses the information access system to sort alphabetically or perform a random shuffle of the candidates. Under $\theta \not\perp\!\!\!\perp s$, in particular for $\rho = -1$, θ sorts candidates by descending scores. It captures settings where the screener obtains a ranked (i.e., informative) list of candidates from the information access system.

Finally, with regard to protected candidates, i.e., $W = 1$, we initially consider the sample of candidates drawn from $Ber(pr)$ such that $pr = 0.2$ is the fraction of protected candidates in C. The sample is independently drawn from both the scores and the ISO, according to assumptions *A1* and *A2* in Section 3.2. We then increase pr to study a more diverse C and its effect on the screener reaching the representational quota q.

4.1.2 Fatigued scores. Beyond the triplet, for the fatigued scores of h_h, we fix $\lambda = 1$, hence $\Phi(t) = t$, and define:

- $\epsilon_1 \sim \mathcal{N}(0,\ (0.005 \cdot (t-1))^2)$, i.e., with constant expectation and standard deviation of $0.005 \cdot (t-1)$; and
- $\epsilon_2 \sim \mathcal{N}(-0.005 \cdot (t-1),\ (0.001 \cdot (t-1))^2)$, i.e., with decreasing expectation and smaller standard deviation than ϵ_1.

Assumptions *A1*, *A2*, and the fact that we define fatigue as a linear term in the final score (8) make the scoring function trivial in this setup. We purposely take for granted the truthful evaluation of the candidates in C to focus on how θ and the fatigue of h_h affect the selected set of k candidates.

4.1.3 Evaluation metrics. We consider the solution S^{k}_{best} of the best-k problem (4) for h_a (Algorithm 1) as *the baseline solution*. We compare it with the the solutions for the good-k problem (7) under h_a (Algorithm 2), and the solutions for both the best-k (4) and good-k (7) problems under h_h (respectively, Algorithms 3 and 4). We define two metrics to capture how close is the compared solution to the baseline solution:

- *Ratio to baseline* (RtB) is the ratio of U^{k}_{add} between the compared solution and the baseline solution. When calculating the utility of h_h, we use the truthful scores, not the fatigued

[3] To generate the correlated θ, we rely on copulas [18, Section 3.4].

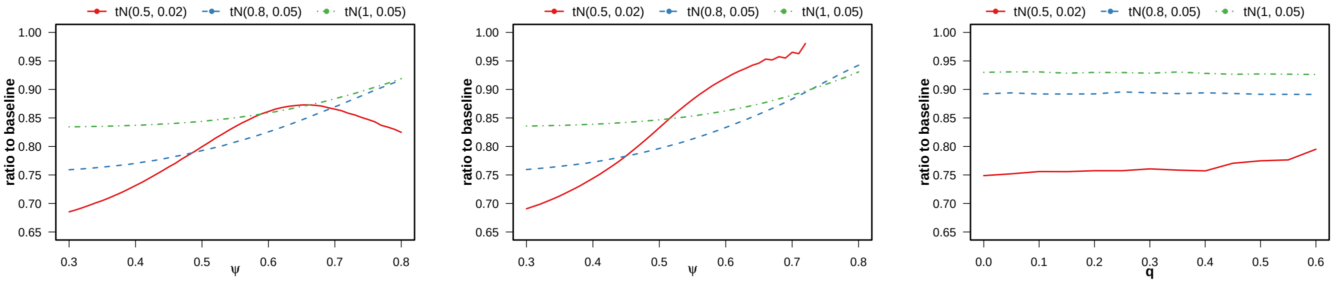

Figure 3: RtB for different score distributions with $n = 120$, $k = 6$, $\theta \perp\!\!\!\perp s$ setting. Left: good-k solution for fatigue with ϵ_1 and setting $q = 0.5$ at the variation of the minimum score ψ. Center: good-k solution for fatigue with ϵ_2 and setting $q = 0.5$ at the variation of the minimum score ψ. Right: best-k solution for fatigue with ϵ_1 at the variation of the representational quota q.

scores, to compare to the baseline. For the solution S^k_{good} under h_a, e.g., it is $U^k_{add}(S^k_{good})/U^k_{add}(S^k_{best})$.

- *Jaccard similarity* (JdS) is the proportion of candidates in both the compared and baseline solutions over those in at least one of the two solutions. For the solution S^k_{good} under h_a, e.g., it is $|S^k_{good} \cap S^k_{best}| \,/\, |S^k_{good} \cup S^k_{best}|$.

The RtB captures whether the compared solution achieves the same utility as the baseline solution as measured by U^k_{add}, while the JdS captures the overlap in candidates between the compared solution and the baseline solution. For both metrics, the closer the ratio is to 1, the better the compared solution approximates the baseline solution in terms of, respectively, utility and composition.

4.1.4 Simulations. For each set of the parameters (n, k, q, ρ, ψ), we run 10,000 times the experiments by randomly generating n triplets at each run. The runs for which a solution of the problem does not exist are discarded. This mainly occurs in the good-k problem when there are not enough k candidates with scores greater or equal than ψ. **The plots report the mean output based on the evaluation metrics over all the runs**.

4.2 Experiments without Fatigue

We start by considering h_a to clarify the relation between the best-k (Algorithm 1) and good-k (Algorithm 2) solutions. Here, we are interested if their solutions differ due to θ, since the best-k requires a full search while the good-k allows for a partial search of C.

Let us study the impact of the score distributions at the variation of ψ. We consider **the case of $\mathbf{n = 120, k = 6, q = 0.5}$ and $\theta \perp\!\!\!\perp$ s**, with a focus on the good-k as ψ is specific to this problem. We find that, as ψ increases and screening becomes more selective, the good-k approximates the best-k solution, especially when there is a low probability of having top candidates in C. The symmetric distribution of scores (in red) in Figure 2 (center, right) illustrates this point. Having few top candidates forces h_a to explore more C under θ, especially as ψ increases and the k first good-enough candidates essentially become the k top candidates. The opposite holds for the other two distributions of scores, asymmetric (in blue) and increasing (in green), which are more resilient to ψ as each represents a higher concentration of top candidates in C. Having many top candidates makes it difficult for h_a to select the k top candidates under a partial search. As the RtB and JdS metrics show in Figure 2 (center, right), h_a still achieves significant utilities under the other two distributions but is unlikely to derive the same selected set of candidates under a partial search w.r.t. a full search of C despite having a highly selective ψ.

Figure 2, in short, illustrates how the two set selection problems materialize differently due to θ. Clearly, as noted in Remark 1, where the k top candidates appear in θ can determine if they are selected or not by h_a under a partial search. The position bias in the ISO becomes more prevalent under many top candidates as even the C's best candidate may never be selected by h_a under a partial search if it lies at the bottom of θ.

We also study **the case when $\theta \not\perp\!\!\!\perp$ s**, which we present in Appendix C.1. Here, we briefly discuss these results as they further illustrate the role of θ. We find that the good-k solution approximates quite well the best-k solution already for $\rho = -0.5$, while for $\rho = -1$, the two solutions are the same. These results are expected as θ essentially represents the best-k solution (or an approximation of it) depending on ρ's strength. Under $\rho = -1$, e.g., the θ searched by h_a is already sorted by the candidate scores and, in turn, the k first good-enough candidates are also the k best candidates in C. See Figure 6 (center, right) for details.

Further, for **both cases** we also study **the impact of changing the number of candidates n and selected candidates k**. The plots are shown in Appendix C.1 We find that under $\theta \perp\!\!\!\perp s$, the ratio k/n is positively correlated with the ability of the good-k to approximate best-k. It means that the more candidates we can select from C, the better the chance to include top ones under a partial search. Clearly, the influence of θ diminishes as k/n increases. See Figure 6 (left). Similarly, we study **the impact of changing the representational quota** q. Here, results are expected given our underlying *A1* and *A2* assumptions, finding that under $\theta \perp\!\!\!\perp s$, q does not affect the relative strengths of best-k and good-k solutions. See Figure 7 (all).

4.3 Experiments with Fatigue

We now focus on h_h, and start by studying whether fatigue impacts the utility w.r.t. the baseline solution (namely, Algorithm 1). We compare to such a baseline both the best-k with fatigue (Algorithm. 3) and good-k with fatigue (Algorithm. 4) solutions.

We again consider **the case of $\mathbf{n = 120, k = 6, q = 0.5}$ and $\theta \perp\!\!\!\perp$ s**. Figure 3 (left) shows the RtB metric for the three score distributions

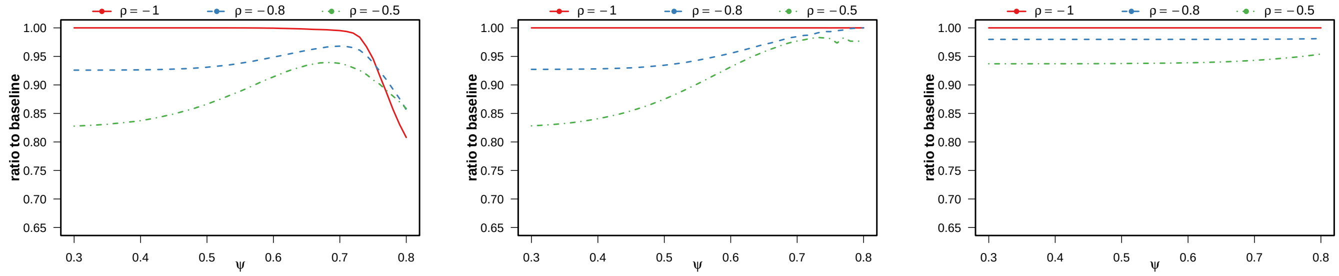

Figure 4: RtB for different ρ's with setting $n = 120$, $k = 6$, and $q = 0.5$ at the variation of the minimum score ψ. Left: good-k solution for fatigue with ϵ_1 and the $tN(0.5, 0.02)$ score distribution. Center: good-k solution for fatigue with ϵ_2 and the $tN(0.5, 0.02)$ score distribution. Right: good-k solution for fatigue with ϵ_1 and the $tN(1, 0.05)$ score distribution.

for the good-k solution with **fatigued scores due to** ϵ_1. Based on Figure 2 (center), for the asymmetric (in blue) and increasing (in green) score distributions, there is no considerable difference w.r.t. the case without fatigue. Instead, for the symmetric score distribution (in red), there is a considerable decrease under high ψ values, which can be attributed to the low number of top scores on which ϵ_1's has a large effect. For the other two score distributions, we have that there are enough top scores such that ϵ_1 does not change the top score distribution. Since the RtB metric captures achieving the utility of the baseline model, under a partial search h_h is still able to reach high utility solutions when C has many top candidates. As ψ increases and h_h becomes more selective, it also becomes more tired under θ as it needs to search more and more candidates to achieve k. Even as ψ increases and h_h becomes more selective, having enough top candidates reduces the need for h_h to continue searching C and, in turn, increase its fatigue. Figure 3 (center) is analogous to (left), but considers the **fatigued scores based on** ϵ_2. The effect for the symmetric distribution (in red) is not present in such a case, due to the lower standard deviation of ϵ_2. Under $\theta \perp\!\!\!\perp s$, variance appears more relevant than bias in the case of low probability for top scores. This result illustrates the importance of how we define fatigue.

For **the case when** $\theta \not\!\perp\!\!\!\perp$ **s**, Figure 3 (right) shows the RtB for the **best-k solution at the variation of the quota** q. There is a considerable and constant loss in utility under fatigue, which is more consistent for the symmetric score distribution (in red). Note that the RtB is lower than in the case of the good-k with fatigue (see left) for $\psi \geq 0.5$. This result means that, for the symmetric distribution, the good-k solution with fatigue has better utility than the best-k solution with fatigue.

Finally, we consider **the impact of ρ on θ on the good-k solution**. Figure 4 (left) considers the symmetric score distribution (in red) where, for the lower half of ψ's, the lines are similar to the analogous case without fatigue shown in Figure 6 (center). For the higher half of ψ's, instead, there is a decrease in the metric. Again, these results are due to the low probability of top scores for which the effects of the bias due to ϵ_1 is not counter-balanced by ρ. Such an effect does not appear for ϵ_2 nor for ϵ_1 under the increasing score distribution. In fact, Figure 4 (center) and (right) closely resemble those in Figure 6 (center) and (right) respectively. It means that fatigue does not have an impact on utility of the good-k solution if there are sufficiently many top scores or a sufficiently small variability of the fatigue. Further, this last result points at the importance of providing a θ to the human screener with some information about candidate quality. Intuitively, under a partial search procedure and the threat of position bias materializing through θ, we would like to decrease h_h's fatigue by minimizing its need to search more of C, reinforcing the role of information access systems.

5 CONCLUSION

In this work, we presented the *initial screening order* (ISO) as a parameter of interest; defined two formulations under distinct utility models of the fair set selection problem, the best-k and the good-k, with their corresponding algorithmic implementations; and introduced a human-like screener to study the effects of the ISO on a human user. We also provided a simulation framework flexible enough to study and model multiple screening scenarios. Our analysis confirms the fairness and optimality impact of the ISO, motivated by the risk of position bias, on the set of k candidates selected by an algorithmic or human-like screener.

Extensive simulations showed a complex relations between best-k and good-k problems. Our results are limited by the functional assumptions made for formulating the two problems and screeners, in particular, the human-like screener. Future work should explore alternative utility models and fatigue terms, while still relying on the current experimental framework. An alternative formulation to fatigue, e.g., could involve a human-like screener that rests while searching over the ISO. We see recurrent survival models [13] well suited for this task. Future work should also explore theories on human decision-making (e.g., [1, 11, 25]), or the use of the simulations framework for testing for optimal parameters (e.g., deriving a minimum score ψ for which best-k and good-k problems coincide). That said, given that it is costly and time-consuming to run real candidate screening experiments, especially at the same scale (10000 runs per setting) of Section 4, we view our work as another example [6, 8, 24, 27, 46] of how simulations can be useful tools for studying the fairness and optimality of real-world decision-making processes involving ML-based systems [35].

We conclude with two takeaways from our analysis for user search behaviour. First, defining the proper problem formulation is important for understanding the impact of the ISO on the selected candidates, which reflects on the search procedures of the screeners regardless of their kind. Second, once the search procedure is clear, it is important to understand how the screener behaves as it searches over the ISO. These takeaways have direct implications for practitioners. They might seem obvious ex-post, though they are supported by an extensive analysis that accounts for several factors that only become clear under modeling and experiments.

ACKNOWLEDGMENTS

Jose M. Alvarez and Salvatore Ruggieri received funding from the European Union's Horizon 2020 research and innovation program under Marie Sklodowska-Curie Actions (grant agreement number 860630) for the project "NoBIAS - Artificial Intelligence without Bias". Antonio Mastropietro and Salvatore Ruggieri received funding from the European Union's Horizon Europe (grant agreement number 101070212) for the FINDHR project. This work reflects only the authors' views and the European Research Executive Agency is not responsible for any use that may be made of the information it contains. No other funding was received by the authors.

ETHICAL CONSIDERATIONS

We did not face any ethical challenges when drafting the paper. Results are based on simulated data intended to illustrate our theoretical analysis. During our collaboration with company G, in particular, which occurred before the drafting of this paper, we followed G's ethical guidelines at all times. We concluded our collaboration with G with an internal report that we presented and discussed with all stakeholders. No sensitive data (or data at all) from company G was used for this work. At no point did we receive monetary compensation from G. Our research has been funded by public institutions. The views reflected are entirely our own.

We believe that this work shows the importance of considering the human user in the formulation of the candidate screening problem. We want to stress that our distinction between an algorithmic screener and a human-like screener was to show the importance of considering the latter kind and not to endorse the former kind. We strongly believe that candidate screening is a complex, human-dependent and human-centered process that should not be left as an automated decision-making problem.

REFERENCES

[1] Gediminas Adomavicius, Jesse C. Bockstedt, Shawn P. Curley, and Jingjing Zhang. 2013. Do Recommender Systems Manipulate Consumer Preferences? A Study of Anchoring Effects. *Inf. Syst. Res.* 24, 4 (2013), 956–975.

[2] José M. Álvarez, Alejandra Bringas Colmenarejo, Alaa Elobaid, Simone Fabbrizzi, Miriam Fahimi, Antonio Ferrara, Siamak Ghodsi, Carlos Mougan, Ioanna Papageorgiou, Paula Reyero Lobo, Mayra Russo, Kristen M. Scott, Laura State, Xuan Zhao, and Salvatore Ruggieri. 2024. Policy advice and best practices on bias and fairness in AI. *Ethics Inf. Technol.* 26, 2 (2024), 31.

[3] Susan Athey and Glenn Ellison. 2011. Position Auctions with Consumer Search. *The Quarterly Journal of Economics* 126, 3 (2011), 1213–1270.

[4] Ricardo Baeza-Yates. 2018. Bias on the web. *Commun. ACM* 61, 6 (2018), 54–61.

[5] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 1999. *Modern Information Retrieval.* ACM Press / Addison-Wesley.

[6] Eszter Bokányi and Anikó Hannák. 2020. Understanding inequalities in ride-hailing services through simulations. *Scientific reports* 10, 1 (2020), 1–11.

[7] Z. I. Botev. 2017. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society Series B* 79, 1 (2017), 125–148.

[8] Dimitrios Bountouridis, Jaron Harambam, Mykola Makhortykh, Mónica Marrero, Nava Tintarev, and Claudia Hauff. 2019. SIREN: A Simulation Framework for Understanding the Effects of Recommender Systems in Online News Environments. In *FAT.* ACM, 150–159.

[9] Alejandra Bringas Colmenarejo, Luca Nannini, Alisa Rieger, Kristen M Scott, Xuan Zhao, Gourab K Patro, Gjergji Kasneci, and Katharina Kinder-Kurlanda. 2022. Fairness in agreement with European values: An interdisciplinary perspective on ai regulation. In *AIES.* ACM, 107–118.

[10] Thomas Kleine Buening, Meirav Segal, Debabrota Basu, Anne-Marie George, and Christos Dimitrakakis. 2022. On Meritocracy in Optimal Set Selection. In *EAAMO.* ACM, 20:1–20:14.

[11] Ana Karina Caraban, Evangelos Karapanos, Daniel Gonçalves, and Pedro F. Campos. 2019. 23 Ways to Nudge: A Review of Technology-Mediated Nudging in Human-Computer Interaction. In *CHI.* ACM, 503.

[12] Gonçalo Carriço. 2018. The EU and artificial intelligence: A human-centred perspective. *European View* 17, 1 (2018), 29–36.

[13] Praveen Chandar, Brian St. Thomas, Lucas Maystre, Vijay Pappu, Roberto Sanchis-Ojeda, Tiffany Wu, Ben Carterette, Mounia Lalmas, and Tony Jebara. 2022. Using Survival Models to Estimate User Engagement in Online Experiments. In *WWW.* ACM, 3186–3195.

[14] Nick Craswell, Onno Zoeter, Michael J. Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *WSDM.* ACM, 87–94.

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *ITCS.* ACM, 214–226.

[16] Jessica Maria Echterhoff, Matin Yarmand, and Julian J. McAuley. 2022. AI-Moderated Decision-Making: Capturing and Balancing Anchoring Bias in Sequential Decision Tasks. In *CHI.* ACM, 161:1–161:9.

[17] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2022. Fairness in Information Access Systems. *Found. Trends Inf. Retr.* 16, 1-2 (2022), 1–177.

[18] Paul Embrechts, Filip Lindskog, and Alexander Mcneil. 2003. Modelling Dependence with Copulas and Applications to Risk Management. In *Handbook of Heavy Tailed Distributions in Finance*, Svetlozar T. Rachev (Ed.). Vol. 1. North-Holland, Amsterdam, 329–384.

[19] Alessandro Fabris, Nina Baranowska, Matthew J. Dennis, Philipp Hacker, Jorge Saldivar, Frederik J. Zuiderveen Borgesius, and Asia J. Biega. 2023. Fairness and Bias in Algorithmic Hiring. *CoRR* abs/2309.13933 (2023).

[20] Ronald Fagin, Amnon Lotem, and Moni Naor. 2003. Optimal aggregation algorithms for middleware. *J. Comput. Syst. Sci.* 66, 4 (2003), 614–656.

[21] Thomas S Ferguson. 1989. Who solved the secretary problem? *Statist. Sci.* 4, 3 (1989), 282–289.

[22] Artem Grotov, Aleksandr Chuklin, Ilya Markov, Luka Stout, Finde Xumara, and Maarten de Rijke. 2015. A Comparative Study of Click Models for Web Search. In *CLEF (Lecture Notes in Computer Science, Vol. 9283).* Springer, 78–90.

[23] Yael S Hadass. 2004. The effect of internet recruiting on the matching of workers and employers. *SSRN 497262* (2004).

[24] Stefania Ionescu, Anikó Hannák, and Kenneth Joseph. 2021. An Agent-based Model to Evaluate Interventions on Online Dating Platforms to Decrease Racial Homogamy. In *FAccT.* ACM, 412–423.

[25] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender Systems - An Introduction.* Cambridge University Press.

[26] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately Interpreting Clickthrough Data as Implicit Feedback. *SIGIR Forum* 51, 1 (2017), 4–11.

[27] Donald Martin Jr., Vinodkumar Prabhakaran, Jill Kuhlberg, Andrew Smart, and William S. Isaac. 2020. Extending the Machine Learning Abstraction Boundary: A Complex Systems Approach to Incorporate Societal Context. *CoRR* abs/2006.09663 (2020).

[28] Daniel Kahneman. 2011. *Thinking, Fast and Slow.* Farrar, Straus and Giroux.

[29] Daniel Kahneman, Olivier Sibony, and Cass Sunstein. 2021. *Noise: A Flaw in Human Judgment.* William Collins.

[30] Emre Kazim, Adriano Soares Koshiyama, Airlie Hilliard, and Roseline Polle. 2021. Systematizing Audit in Algorithmic Recruitment. *Journal of Intelligence* 9, 3 (2021).

[31] Marios Kokkodis. 2018. Dynamic Recommendations for Sequential Hiring Decisions in Online Labor Markets. In *KDD.* ACM, 453–461.

[32] Marios Kokkodis, Panagiotis Papadimitriou, and Panagiotis G. Ipeirotis. 2015. Hiring Behavior Models for Online Labor Markets. In *WSDM.* ACM, 223–232.

[33] Alina Köchling and Marius Claus Wehner. 2020. Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of HR recruitment and HR development. *Business Research* 13, 3 (2020).

[34] Anay Mehrotra and L Elisa Celis. 2021. Mitigating bias in set selection with noisy protected attributes. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency.* 237–248.

[35] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. 2023. Human-in-the-loop machine learning: A state of the art. *Artif. Intell. Rev.* 56, 4 (2023), 3005–3054.

[36] Bing Pan, Helene Hembrooke, Thorsten Joachims, Lori Lorigo, Geri Gay, and Laura A. Granka. 2007. In Google We Trust: Users' Decisions on Rank, Position,

and Relevance. *J. Comput. Mediat. Commun.* 12, 3 (2007), 801–823.

[37] Jiaxin Pei, Zhihan Helen Wang, and Jun Li. 2023. 30 Million Canvas Grading Records Reveal Widespread Sequential Bias and System-Induced Surname Initial Disparity. (2023).

[38] Elena Pisanelli. 2022. Your resume is your gatekeeper: Automated resume screening as a strategy to reduce gender gaps in hiring. *Economics Letters* 221 (2022), 110892.

[39] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2022. Fairness in rankings and recommendations: an overview. *VLDB J.* 31, 3 (2022), 431–458.

[40] Chuan Qin, Le Zhang, Rui Zha, Dazhong Shen, Qi Zhang, Ying Sun, Chen Zhu, Hengshu Zhu, and Hui Xiong. 2023. A Comprehensive Survey of Artificial Intelligence Techniques for Talent Analytics. *CoRR* abs/2307.03195 (2023).

[41] R Core Team. 2024. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

[42] Manish Raghavan, Solon Barocas, Jon M. Kleinberg, and Karen Levy. 2020. Mitigating bias in algorithmic hiring: evaluating claims and practices. In *FAT**. ACM, 469–481.

[43] Alene K. Rhea, Kelsey Markey, Lauren D'Arinzo, Hilke Schellmann, Mona Sloane, Paul Squires, Falaah Arif Khan, and Julia Stoyanovich. 2022. An external stability audit framework to test the validity of personality prediction in AI hiring. *Data Min. Knowl. Discov.* 36, 6 (2022), 2153–2193.

[44] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *WWW*. ACM, 521–530.

[45] Salvatore Ruggieri, José M. Álvarez, Andrea Pugnana, Laura State, and Franco Turini. 2023. Can We Trust Fair-AI?. In *AAAI*. AAAI Press, 15421–15430.

[46] Thomas C Schelling. 1971. Dynamic models of segregation. *Journal of Mathematical Sociology* 1, 2 (1971), 143–186.

[47] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *KDD*. ACM, 2219–2228.

[48] Mona Sloane, Emanuel Moss, and Rumman Chowdhury. 2022. A Silicon Valley love triangle: Hiring algorithms, pseudo-science, and the quest for auditability. *Patterns* 3, 2 (2022), 100425.

[49] Keith E. Sonderling, Bradford J. Kelley, and Lance Casimir. 2022. The Promise and The Peril: Artificial Intelligence and Employment Discrimination. *U. Miami Law Review* 77, 1 (2022), 3.

[50] Julia Stoyanovich, Ke Yang, and H. V. Jagadish. 2018. Online Set Selection with Fairness and Diversity Constraints. In *EDBT*. OpenProceedings.org, 241–252.

[51] Poorna Talkad Sukumar, Ronald A. Metoyer, and Shuai He. 2018. Making a Pecan Pie: Understanding and Supporting The Holistic Review Process in Admissions. *Proc. ACM Hum. Comput. Interact.* 2, CSCW (2018), 169:1–169:22.

[52] Amos Tversky and Daniel Kahneman. 1974. Judgment Under Uncertainty: Heuristics and Biases. *Science* 185, 4157 (1974), 1124–1131.

[53] Paris Will, Dario Krpan, and Grace Lordan. 2023. People versus machines: introducing the HIRE framework. *Artif. Intell. Rev.* 56, 2 (2023), 1071–1100.

[54] Christo Wilson, Avijit Ghosh, Shan Jiang, Alan Mislove, Lewis Baker, Janelle Szary, Kelly Trindel, and Frida Polli. 2021. Building and Auditing Fair Algorithms: A Case Study in Candidate Screening. In *FAccT*. ACM, 666–677.

[55] Ke Yang and Julia Stoyanovich. 2017. Measuring Fairness in Ranked Outputs. In *SSDBM*. ACM, 22:1–22:6.

[56] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *CIKM*. ACM, 1569–1578.

[57] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2023. Fairness in Ranking, Part I: Score-Based Ranking. *ACM Comput. Surv.* 55, 6 (2023), 118:1–118:36.

[58] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2023. Fairness in Ranking, Part II: Learning-to-Rank and Recommender Systems. *ACM Comput. Surv.* 55, 6 (2023), 117:1–117:41.

A COLLABORATING WITH G

Candidate screening at G represented both a time-consuming, repetitive task prone to human error and a sensitive, high-risk task requiring human oversight. Therefore, the option of full automation was not possible. The focus was, thus, on understanding and modeling the search of the HR officer via the hiring platform. Importantly, under the realistic risk of position bias affecting candidate evaluations via the hiring platform, we wanted to study the influence of the *initial screening order* (ISO) on the set of suitable candidates chosen by the HR officer.

Overall experience. The collaboration lasted for four months. Due to the COVID-19 pandemic, it was a hybrid collaboration. During this time, we mostly interviewed the HR officers to understand their tasks, constraints, and methodologies, often shadowing them during screening sessions. We were also granted access to Taleo by Oracle, the platform used by HR for managing the hiring pipeline. This allowed us to experience for ourselves the patterns we observed among the HR officers. These patterns resulted in the five stylised facts in Section 1.1.

We interacted, in particular, with five HR officers specialized in screening applications for technical roles within G, such as the roles of data scientist and front-end developer. These HR officers had to process considerable amounts of information within a time constraint. Based on what we observed, most candidate pools (except for very senior profiles like, e.g., director of data science) consisted of hundred of applications. These HR officers were involved in screening multiple candidate pools with similar deadlines within the same week. It became apparent to us, specially for candidate screening, how time-consuming and human-dependent was the hiring process at G.

We discussed our observations, often on a bi-weekly basis, to members of both the HR and AA teams. We emphasize that we simply collaborated with these teams as equals, sharing the goal of understanding G's hiring process and whether it was suitable or not for (fair) automation. We specifically use the wording "stylized facts" in Section 1.1 to emphasize that we draw inspiration from the collaboration with G in formulating the ISO problem rather than "hard facts" about G derived from an observational study. In fact, the first draft of this paper occurred about one year after our collaboration with G had concluded.

Deliverables. We concluded the collaboration with a report to AA that formalized G's candidate screening process as a ranking problem, evaluated the potential fairness implications, and assessed the risk and benefits of automation. The report was discussed with a wider audience within G in a series of presentations hosted by the AA and HR teams. The report focused mainly on candidate screening. It is worth mentioning also that throughout the collaboration, we followed G's strict ethical guidelines at all times. Further, at no point did we receive monetary compensation from G. Our research has been funded by public European institutions. The views reflected in this paper are entirely our own. No sensitive data (or data at all) from G was used for this work. This paper is not a deliverable specific to G.

The hiring pipeline. Hiring at G consists mainly of three phases. With respect to the stylized facts, these are concerned with the candidate screening phase or phase two. These three phases are the following:

- In *phase one*, the HR builds a candidate pool for the job opening. Candidates submit their CVs, complete a multiple-choice questioner, and write a motivation letter. Sensitive information, such as gender, ethnicity, and age, is also provided or it can be inferred. The candidate pool is stored in a database platform.
- In *phase two*, the HR officer reduces the candidate pool into a smaller pool of suitable candidates. The HR officer determines candidate suitability based on each candidate's profile using a set of minimum basic requirements.

- In *phase three*, the chosen candidates are interviewed by HR and the team offering the job. It is common for the hiring team to also prepare a use case for the candidates. The best candidates receive an offer. If no candidates are hired, HR resorts to the runner-up candidates from phase two and repeat phase three.

Note that the above represent G's hiring pipeline during the collaboration. We do not know if this hiring pipeline is still the case today, though it is not important for the purpose of this paper.

B SUPPLEMENTARY MATERIAL

B.1 Additional Discussion on the Utility Model in Section 2.2.2

First, we present Example B.1 below. It shows that for the utility model (5) for the good-k problem, in the general case, i.e., $f(S^k) \geq q$, there can be two solutions, but with different fractions of the protected group.

Example B.1. Consider $n = 3, k = 2, q = 0.5$. Assume three eligible candidates and $\theta(1) = c_1, \theta(2) = c_2, \theta(3) = c_3$ with $W_1 = 0$ and $W_2 = W_3 = 1$. Both $S' = \{c_1, c_2\}$ and $S'' = \{c_2, c_3\}$ are solutions of (7) with $U^k_\psi(S', \theta) = U^k_\psi(S'', \theta) = 2$. However, $f(S') = 0.5$ and $f(S'') = 1$. Intuitively, S' is obtained by strictly iterating over θ.

We now motivate the alternative utility model (5) used in the good-k problem. As a simple alternative utility function in the good-k setting consider the following utility model:

$$\hat{U}^k_\psi(S^k, \theta) = \begin{cases} n - \max_{c \in S^k} \theta^{-1}(c) & \text{if } \forall c \in S^k \; s(\mathbf{X}_c) \geq \psi \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

where we use $\hat{U}^k_\psi$ (i.e., the U hat) to differentiate from the utility model U^k_ψ considered in (5).

Intuitively, in the above utility definition the screener wants to find as quickly as possible a set of k eligible candidates. Therefore, if S^k contains only eligible candidates, the utility of h selecting S^k under θ is expressed by the number of candidates past the last one who was screened, i.e., the "saved effort" of the screener h. Despite the simplicity of (9), the above utility model (9) is not suitable to properly account for the intended good-k problem. To observe this last point, consider Example B.2.

Example B.2. Let $n = 3, k = 2, q = 0.5$. Assume three eligible candidates and $\theta(1) = c_1, \theta(2) = c_2, \theta(3) = c_3$ with $W_1 = W_2 = 0$ and $W_3 = 1$. It turns out that both $S' = \{c_1, c_3\}$ and $S'' = \{c_2, c_3\}$ maximize the utility (9) and satisfy the fairness constraint q.

Following up on Example B.2, why should have been c_2 considered, and then returned in S'', if c_1 already meets the minimum basic requirement? A reason for doing that is a variant of our good-k problem in which the screener h keeps evaluating non-protected candidates in C, even if their quota is reached but the one of protected candidates is not yet reached, for the purpose of keeping the best ones found so far. We do not consider such a variant in this paper. For this reason, we introduce the penalty function (6) in (5) presented in Section 2.2.2.

B.2 The Two Search Procedures under a Human-Like Screener

We update the *ExaminationSearch* and *CascadeSearch* and their corresponding Algorithm 1 and Algorithm 2 from Section 2.3 under the human-like screener h_h from Section 3. We incorporate the *fatigued scores* formulation from Section 3.1 into both algorithms, resulting in a human-like *HuamnExaminationSearch* (Algorithm 3) and a human-like *HumanCascadeSearch* (Algorithm 4). In comparison to the algorithmic screener h_a, the main difference here is that both algorithms compute the fatigued score for candidate $c \in C$:

$$Y_c = s(\mathbf{X}_c) + \epsilon \tag{10}$$

where ϵ is a random variable depending on the accumulated fatigue Φ of h_h. In both algorithms 3 and 4, by requiring the fatigue component Φ, we also require a specific modeling choice for ϵ which is a probabilistic function of the accumulated fatigue Φ. As discussed in Section 3.1, ϵ can be modeled as either ϵ_1 or ϵ_2.

We stress once again that other formulations for ϵ are possible. These formulations are compatible with Algorithms 3 and 4 as long as ϵ is treated as a random variable that is drawn each time h_h evaluates candidate c. These formulations can be implemented as with ϵ_1 and ϵ_2, meaning by providing the corresponding probability distribution for the intended accumulated fatigue Φ.

C ADDITIONAL EXPERIMENTS

C.1 Experiments without Fatigue

In this section, we present the additional figures and corresponding discussions relating to the experimental analysis of the algorithmic screener h_a from Section 4.2.

First, we consider the impact of the correlation ρ between θ and the scores. Recall that $\rho = -1$ means that the candidates are ordered by descending scores. Under such a condition, the good-k and best-k procedures return the same solution. This result is apparent in Figure 6 (center, right) where we report the ratio to baseline for the symmetric (left) and the increasing (right) score distributions. The plots show that even a moderate correlation of $\rho = -0.5$ leads the good-k solution to approximate the best-k one quite well. For the increasing distribution (right), the ratio to baseline is around 95%. In summary, initial orders that negatively correlate to the score greatly reduce the difference in utility between the good-k and best-k solutions.

Second, we consider the impact of the number n candidates in C and the number of k candidates to be selected. We focus only on the symmetric distribution (in red) and the RtB metric, but the results are similar for the other two distributions (in blue and green) and the JdS metric. Figure 6 (left) compares the case $n = 120, k = 6$ considered earlier to two other scenarios in terms of n and k. The first scenario increases $k = 20$, but leaves the ratio of selected $k/n = 0.05$ the same by also increasing $n = 400$. The second scenario, instead, leaves $k = 6$ the same, but it increases $k/n = 0.2$ by decreasing $n = 30$. The plot shows that changes in the ratio k/n affect the metric, in particular a larger ratio ($n = 30$, $k = 6$) leads good-k to better approximate best-k for the same ψ. In other words, the more candidates we can select from the pool, the better the chance to include top ones. In summary, under $\theta \perp\!\!\!\perp s$,

Algorithm 3 HumanLikeExaminationSearch

```
Require: $n, \theta, k, q, \Phi$
Ensure: $S^k_{best}$
$q^* \leftarrow \texttt{round}(q \cdot k)$; $r^* \leftarrow k - q^*$
$scores \leftarrow [s(\mathbf{X}_{\theta(t)}) + \epsilon(\Phi(t-1))\ for\ t = 1, \ldots, n]$
$\tau \leftarrow \texttt{argsortdesc}(scores)$
$i \leftarrow 1$; $k^* \leftarrow 0$; $Q \leftarrow \{\}$; $R \leftarrow \{\}$
while $k^* < k$ do
    $c \leftarrow \theta(\tau(i))$; $i \leftarrow i + 1$
    if $W_c == 0$ and $\texttt{len}(R) == r^*$ then
        continue
    $k^* \leftarrow k^* + 1$
    if $W_c == 1$ and $\texttt{len}(Q) < q^*$ then
        $Q \leftarrow Q \cup \{c\}$
    else
        $R \leftarrow R \cup \{c\}$
return $Q \cup R$
```

Algorithm 4 HumanLikeCascadeSearch

```
Require: $n, \theta, k, q, \psi, \Phi$
Ensure: $S^k_{good}$
$q^* \leftarrow \texttt{round}(q \cdot k)$; $r^* \leftarrow k - q^*$
$i \leftarrow 1$; $k^* \leftarrow 0$; $Q \leftarrow \{\}$; $R \leftarrow \{\}$; $t \leftarrow 1$
while $k^* < k$ do
    $c \leftarrow \theta(i)$; $i \leftarrow i + 1$
    if $W_c == 0$ and $\texttt{len}(R) == r^*$ then
        continue
    $Y_c \leftarrow s(\mathbf{X}_c) + \epsilon(\Phi(t-1))$; $t \leftarrow t + 1$
    if $Y_c \geq \psi$ then
        $k^* \leftarrow k^* + 1$
        if $W_c == 1$ and $\texttt{len}(Q) < q^*$ then
            $Q \leftarrow Q \cup \{c\}$
        else
            $R \leftarrow R \cup \{c\}$
return $Q \cup R$
```

Figure 5: The algorithmic implementations for the best-k and good-k problems revisited under the human-like screener.

the ratio k/n is positively correlated with the ability of best-k to approximate good-k.

Finally, let us now consider the quota parameter q, thus far set to $q = 0.5$ over a population with a fraction of protected candidates set to $pr = 0.2$. Since we assumed that W is independent from both scores and the ISO, the fraction of protected group in the solutions of best-k and good-k is, on average, $min\{q, pr\}$. Figure 7 (left) shows this result in the solution for good-k. A less trivial question is whether q is also not affecting the evaluation metrics: e.g., whether the quota q changes the ratio to baseline? Figure 7 (center, right) show that this is not the case in two experimental settings. Again, this result is theoretically implied by the independence of W with scores and initial order. In summary, under $\theta \perp\!\!\!\perp s$, the q in the best-k and good-k problem does not affect the relative strengths of their solutions.

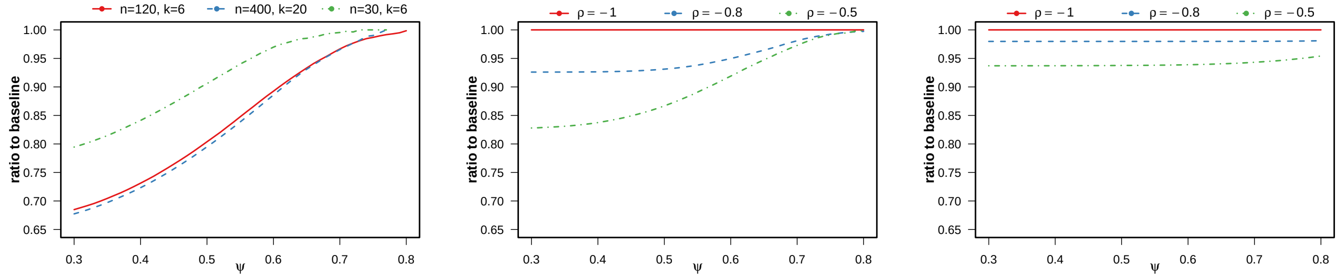

Figure 6: Left: RtB for different k/n ratios, for the $tN(0.5, 0.02)$ score distribution and with setting $q = 0.5$, $\theta \perp\!\!\!\perp s$. Center: RtB for different ρ's for the $tN(0.5, 0.02)$ score distribution and with setting $n = 120$, $k = 6$, and $q = 0.5$. Right: RtB for different ρ's for the $tN(1, 0.05)$ score distribution and with setting $n = 120$, $k = 6$, $q = 0.5$.

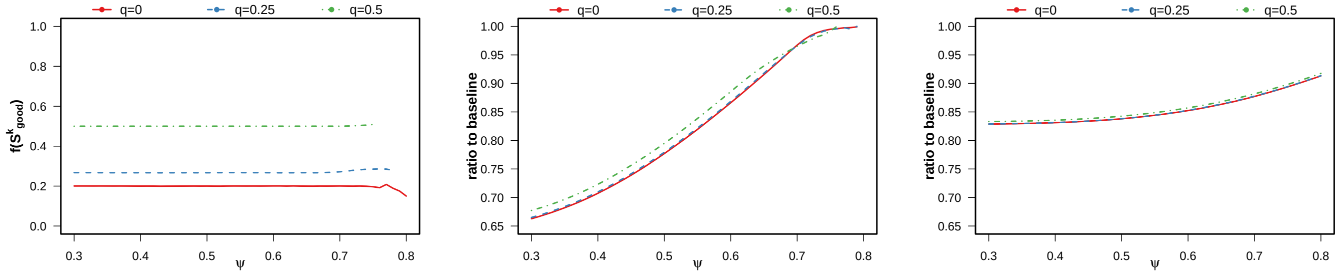

Figure 7: Left: fraction of protected candidates in the solution of good-k for different representational quotas q, for the $tN(0.5, 0.02)$ score distribution and with setting $n = 400$, $k = 20$, $\theta \perp\!\!\!\perp s$. Center: RtB for different representational quotas q, for the $tN(0.5, 0.02)$ score distribution and with setting $n = 400$, $k = 20$, $\theta \perp\!\!\!\perp s$. Right: RtB for different representational quotas q, for the $tN(1, 0.05)$ score distribution and with setting $n = 400$, $k = 20$, $\theta \perp\!\!\!\perp s$.