https://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–23
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

Fairness and Bias in Online Selection

José Correa,[a,*] Andrés Cristi,[b] Paul Dütting,[c] Ashkan Norouzi-Fard[c]

[a] Departamento de Ingenieria Industrial, Universidad de Chile, Santiago 8370439, Chile; [b] College of Management of Technology, Ecole Polytechnique Fédérale de Laussane, 1015 Lausanne, Switzerland; [c] Google Research, 8002 Zurich, Switzerland

*Corresponding author

Contact: correa@uchile.cl, https://orcid.org/0000-0002-3012-7622 (JC); andres.cristi@ing.uchile.cl, https://orcid.org/0000-0002-1227-2092 (AC); duetting@google.com, https://orcid.org/0000-0002-0635-6812 (PD); ashkannorouzi@google.com, https://orcid.org/0000-0002-2336-9826 (AN-F)

Received: October 14, 2021
Revised: June 12, 2023; September 26, 2024; August 8, 2025
Accepted: August 13, 2025
Published Online in Articles in Advance: September 24, 2025

Area of Review: Optimization

https://doi.org/10.1287/opre.2021.0662

Abstract. There is growing awareness and concern about fairness in machine learning and algorithm design. This is particularly true in online selection problems, where decisions are often biased: for example, when assessing credit risks or hiring staff. We address the issues of fairness and bias in online selection by studying multicolor versions of the classic secretary and prophet problems. In the multicolor secretary problem, we consider that each candidate has a color, and we can only compare candidates of the same color. In addition, we are given probabilities with which the best candidate of a given color is the best candidate overall. These probabilities but not the outcome of the random coin flip are known to both the online and offline algorithms. We characterize the optimal online algorithm and show that unlike the optimal offline algorithm, it enjoys very desirable fairness properties. In the multicolor prophet problem that we study, candidates can again be partitioned into groups of different colors. To counteract imbalanced selection, each color is associated with a target selection probability. We design fair online algorithms that conditional on stopping, select from the different colors with the given target probabilities and achieve optimal approximation guarantees against the fair offline optimum. We also study data-driven (sampling-based) variants of both the multicolor secretary problem and the multicolor prophet problem, and we provide an empirical evaluation of our algorithms.

Funding: J. Correa and A. Cristi were partially funded by the Centro de Modelamiento Matemático, Facultad de Ciencias Físicas y Matemáticas [Grant ANID FB210005]; the Millennium Institute for Research in Market Imperfections and Public Policy [Grant ICS13002]; the Instituto de Sistemas Complejos de Ingeniería [Grant AFB230002]; the Fondo Nacional de Desarrollo Científico y Tecnológico [Grant FONDECYT 1220054]; and Programa Formación de Capital Humano Avanzado (PFCHA) [Grant 2018-21180347].

Supplemental Material: All supplemental materials, including the code, data, and files required to reproduce the results, are available at https://doi.org/10.1287/opre.2021.0662.

Keywords: online selection • optimal stopping • fairness

1. Introduction

The sharp growth in data availability that characterizes modern society challenges our processing capabilities not only because of its massiveness but also because of the increasingly strict social norms that society seeks in the algorithms processing it. For instance, machine learning algorithms are now used to make credit and lending decisions, to estimate the success of a kidney transplant, to inform hiring decisions, and to recommend schools to pupils among others. Therefore, there is a founded concern over the use of algorithms that may violate social norms. Two basic such norms that are receiving significant attention are fairness and privacy, and although a formalization of the latter is relatively well established through the notion of differential privacy (Dwork et al. 2006), the former is much more unexplored from an algorithmic perspective (Kearns and Roth 2019).

In this paper, we are particularly interested in the study of fairness in (machine learning) algorithms in the context of sequential decision making under stochastic input. We consider two fundamental problems in fair online selection, both concerned with selecting a single candidate. In both problems, candidates are partitioned into different groups or *colors*. The candidates arrive sequentially, and upon the arrival of a candidate, we have to irrevocably decide whether we want to select the candidate or not.

1.1. The Multicolor Secretary Problem

In the *multicolor secretary problem*, candidates arrive in uniform random order; we can rank candidates within

a group, but we cannot compare candidates across groups. We are thus looking at a secretary problem with a (particular) partially ordered set of secretaries. The use of poset models as a versatile tool to evaluate fairness and bias in online selection was initiated by Salem and Gupta (2024). Related problems with poset structure were previously studied by Preater (1999), Georgiou et al. (2008), and Kumar et al. (2011).

In our model, there is also a prior probability that the best candidate from a group is the best candidate overall. We model this as an independent, random coin flip that occurs after the selection. The problem models situations in which different qualities of the candidates make them largely incomparable (e.g., performance in final examinations at two different schools A and B in two different countries), but there is some probabilistic "ground truth" that the best candidate of a group is the best overall (for example, it may be known that the best student of a year is more likely to come from school A than from school B). In situations like this, it is natural to evaluate the fairness of an algorithm by how closely its selection probabilities match the prior probabilities. Namely, if p_i is the probability with which the maximal candidate of group $i \in [k]$ is the best candidate overall, then we say that the algorithm is *α-fair* for $\alpha \geq 1$ if for all groups i, conditional on stopping, the algorithm selects a candidate of group i with probability in $[p_i/\alpha, \alpha \cdot p_i]$ (Definition 2). Intuitively, the closer α is to one, the better the algorithm reflects the prior probabilities, and the fairer it is.

Putting fairness considerations aside for the moment, we consider the goal of designing an online algorithm that maximizes the probability of selecting the best overall candidate and compare it with the *offline optimum* (that is oblivious to the random coin flip that determines the color of the best candidate). Note that here, the offline optimum simply picks the best candidate from the group of largest prior probability. So, for example, if there are two groups with priors 51% and 49%, then the optimum offline algorithm *always* chooses the best candidate from the first group. This is precisely the type of behavior that we seek to avoid. Indeed, in situations like this, the offline optimum is *not* α-fair for *any* $\alpha \geq 1$. One may think that the best-possible online algorithm is to mimic the offline optimum: namely, to select the group of largest prior probability and then, run the classic secretary algorithm on that group. We prove that this is not the case, and indeed, our main result is to obtain the best-possible online algorithm for the problem and to establish that it satisfies very desirable fairness properties. Hence, for this variant of online selection, fairness follows as a consequence of being online optimal.

Example 1. (Multicolor Secretary). Suppose you are hiring a professional for a position and have four candidates for filling it. Two come from school A, and two come from school B. You can compare candidates coming from the same school, but you cannot compare across schools. Suppose, in addition, that from previous experience, you know that 60% of the time, the best candidate comes from school A. If the process is offline and you can see all candidates simultaneously, the best strategy is to pick the best candidate from school A. This guarantees you a probability of picking the overall best of 0.6. On the contrary, if the process is online as in the secretary problem and candidates come in random order, the situation changes dramatically. A natural idea would be to simply ignore the candidates from school B and run the secretary algorithm on the candidates from school A. For $n = 2$, the secretary algorithm selects the best with probability $1/2$, so you end up selecting the overall best candidate with probability 0.3. You could instead do something that is more fair to the candidates from school B: wait until you see the second candidate of any of the two schools. If they are the best of their school, select them. If not, select a candidate of the other school. One can easily show that with this policy, you end up selecting the best candidate with probability 0.375. Indeed, the probability that the second candidate of a school arrives before the second of the other school is $1/2$, and this candidate is the best of their school with probability $1/2$; so, the probability of selecting the overall best is

$$\frac{1}{2}\left(\frac{1}{2}0.6 + \frac{1}{2}\cdot\frac{1}{2}0.4\right) + \frac{1}{2}\left(\frac{1}{2}0.4 + \frac{1}{2}\cdot\frac{1}{2}0.6\right) = \frac{3}{8} = 0.375.$$

Thus, in the online setting, we can take advantage of the fact that in some realizations, we observe both candidates from school B before the second candidate from school A. Our general result leverages this idea by skipping a fraction of the candidates of each color, where the fraction depends continuously on the prior probability that the overall best candidate is of that color.

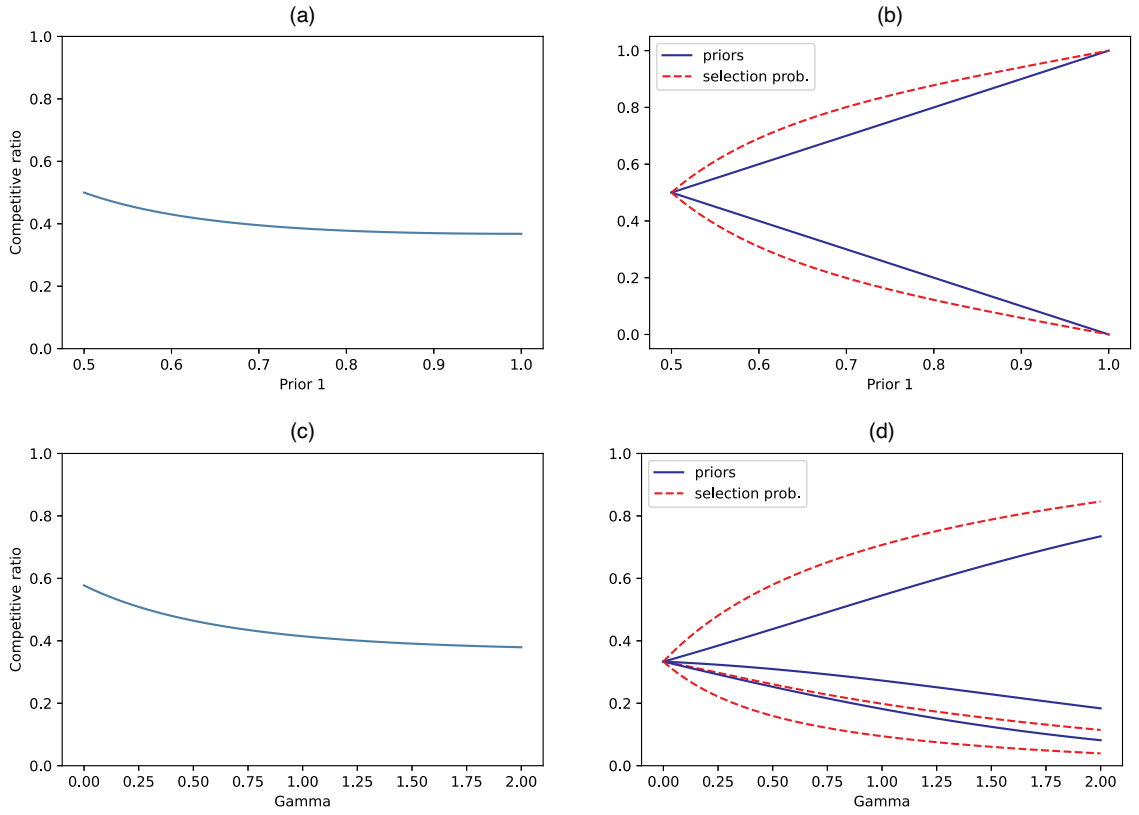
1.1.1. Our Results.

Our main result for the multicolor secretary problem (Theorem 1) characterizes the optimal online algorithm and gives a closed formula for its competitive ratio. We model the random order arrival by associating uniformly distributed arrival times with the candidates. An important ingredient in our proof is a lemma (Lemma 1) that establishes that the competitive ratio is decreasing as the number of candidates n_i of any given color i increases. We show that the online optimum algorithm employs time-dependent thresholds for each color that depend on the number of groups k and the prior probabilities $\mathbf{p} = (p_1, \ldots, p_k)$ but not on the number of candidates from each group. The algorithm then accepts the first candidate who arrives after the threshold for its color and is the best candidate

from that color so far. We visualize the approximation guarantees of the optimal online algorithm for different k and a range of prior probabilities $\mathbf{p} = (p_1, \ldots, p_k)$ in panels (a) and (c) of Figure 1. In the case where there are k groups and the maximum is equally likely to come from any of these groups, our algorithm achieves a competitive ratio of $k^{1/(k-1)}$ (Corollary 1). This is two for $k = 2$, $\sqrt{3}$ for $k = 3$, and $1 + O(\log k/k)$ as $k \to \infty$. With regard to fairness, we show that for equal priors over k groups and arbitrary group sizes, the optimal online algorithm does not choose from all groups with equal probability (a property that we coin 1-fairness) but approaches this property exponentially fast in the minimum group size (Theorem 2). For general priors over k groups, we show that when two groups j, j' have a similar prior $p_j > p_{j'} > (1 - \epsilon)p_j$, then the probability that the optimal online algorithm selects color j and the probability that it selects color j' are within ϵ of each other (Theorem 3). To exemplify this bound, consider the case where there are two groups, men and women, and the prior is such that the top candidate is a woman with probability 60% and a man with probability 40%. This translates into having $\epsilon = 1/3$ in the theorem statement, and thus, the optimal online algorithm will pick a woman at most 33% more often than a man. A pictorial view of the actual selection probabilities of the optimal online algorithm compared with the priors is shown in panels (b) and (d) of Figure 1.

We also study a data-driven version of the multicolor secretary problem. Intuitively, this model interpolates between a model with adversarial qualities, as assumed so far, and one in which the qualities are sampled from known distributions. As before, we assume that the ground set of candidates is partitioned into k groups (or colors) and that we can only compare candidates of the same color. We use C_i to denote the candidates of color i. We write n_i for the number of candidates of color i and $n = \sum_{i=1}^{k} n_i$ for the total number of candidates. Each candidate chooses to be in S (the set of samples) independently with probability q and to be part of V otherwise. The samples are visible at the outset. Afterward, the candidates in V arrive in uniform random order, and we can select one of them. As in our base model,

Figure 1. (Color online) For $k = 2$, $k = 3$, and Varying Priors, We Show the Competitive Ratio of the Optimal Algorithm Along with the Probabilities That It Selects a Candidate from Each Color Conditional on Stopping



Notes. For $k = 2$, we vary p_1 linearly in $[1/2, 1]$ (and $p_2 = 1 - p_1$). For $k = 3$, we set priors following a power law with varying exponent $\gamma \in [0, 2]$ (i.e., we set (p_1, p_2, p_3) proportional to $(1, 1/2^{\gamma}, 1/3^{\gamma})$). Observe that the competitive ratios start at $k^{\frac{1}{k-1}}$ when the priors are equal and tend to $1/e$ as the largest prior gets closer to one. Observe also that the conditional selection probabilities follow very closely the priors when they are not far from being balanced. (a) Competitive ratio for $k = 2$. (b) Conditional selection probabilities for $k = 2$. (c) Competitive ratio for $k = 3$. (d) Conditional selection probabilities for $k = 3$.

we again assume that the best candidate in $V \cap C_i$ is the best candidate overall with probability p_i. Our goal is to maximize the probability of selecting the best candidate in V, where we compare ourselves with the best offline algorithm. For $q = 0$, this model reduces to our base model, and as $q \to 1$, the model approaches a setting where ranks are drawn from a known distribution. We show that the competitive ratio is again decreasing with the number of candidates n_i of any given color i (Lemma 6). We also give a closed formula for the competitive ratio of any online algorithm that sets group-specific thresholds $\mathbf{t} = (t_1, \ldots, t_k)$ (Theorem 4), and we argue how this can be used to computationally solve for the optimal such algorithm. We visualize the resulting competitive ratios for $k = 1, 2, 3$ groups and equal priors $p_i = 1/k$ as a function of the sampling rate q in Figure 2. We note that in the special case where $k = 1$ and $q \to 1$, our result recovers the classic result of Gilbert and Mosteller (1966).

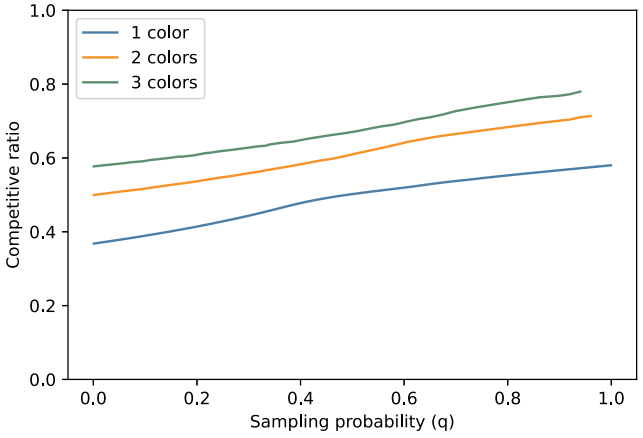
Our results for the multicolor secretary problem are summarized in Table 1. In Section 2.4, we provide additional numerical evaluations. We first explore the effect of a wrong prior $\mathbf{p}' \neq \mathbf{p}$ on the competitive ratio, showing that the competitive ratio degrades gracefully. Then, using a simple model of belief update dynamics, we show how wrong priors can be corrected over time. For this, it is important that our algorithm (unlike the offline optimum) selects from all groups with positive probability.

1.2. The Multicolor Prophet Problem

In our second problem, which we call the *multicolor prophet problem*, candidates have values drawn independently from given distributions and arrive in an arbitrary order. The goal is to maximize the expectation of the value of the selected candidate while selecting from each color with probability proportional to a prescribed vector. We compete with the optimal offline algorithm satisfying the same fairness constraint. More formally, for a given partition of the candidates into colors C_i for $i \in [k]$ such that $|C_i| = n_i$ and a vector of probabilities $\mathbf{p} = (p_1, \ldots, p_k)$, we denote by FairOpt the expected value of the optimal offline algorithm that selects from color i with probability p_i (see Section 3.1). We say that an online algorithm is fair if conditioned on stopping, it stops at color i with probability p_i for all $i \in [k]$ (see Definition 3). In other words, in a fair online algorithm conditioned on the algorithm making a selection, the probability that a candidate is picked from group i is exactly p_i.

We seek to bound the ratio between the expected value achieved by FairOpt and the best fair online algorithm. So, the underlying paradigm here is that although we can compare, we understand that selection probabilities can be unfair or biased (for example, because of differences in the value distributions or because of different arrival positions), and we want to correct this using the prior probabilities.

Figure 2. (Color online) Competitive Ratio of Our Algorithm for the Sample-Driven Multicolor Secretary Problem for $k = 1, 2, 3$ and Balanced Priors



Note. This was obtained by numerically optimizing Equation (3).

Table 1. Competitive Ratios for the Multicolor Secretary Problem

Equal priors	General priors	General priors, sample based
$k^{1/(k-1)}$ (Corollary 1)	Optimal guarantee (Theorem 1)	Optimal threshold-based guarantee (Theorem 4)

Note. The parameter k denotes the number of colors.

Example 2. (Multicolor Prophet). Consider again the problem of hiring one of four professionals for a position. Now, our goal is to maximize the expected grades of the selected candidate. For simplicity, assume that they are i.i.d. realizations of a Uniform[0, 1] distribution. A natural fairness constraint in this case is that we should select candidates with the same probability. We can achieve this in our framework by making each candidate a group on its own and letting $p_i = 1/4$ for $i \in \{1, 2, 3, 4\}$. The optimal *offline* algorithm for this problem is perfectly fair. It selects each candidate with probability 1/4 and achieves an expected value of $4/5 = 0.8$. We can also calculate the optimal *online* policy via dynamic programming (DP); a candidate should be accepted if their grades are above the expectation of what comes next if we skip them. If the threshold for candidate $i + 1$ is z, the threshold for candidate i is $T(z) = \max\{z, X\} = z + \mathbf{E}[(X - z)_+]$, where X is a Uniform[0, 1] random variable. With this, $T(z) = z + (1 - z)^2/2$. The threshold for the last candidate is, of course, zero. So, for the third candidate, it is $T(0) = 1/2$. For the second, it is $T(1/2) = 5/8$, and for the first one, it is $T(5/8) = 89/128 \approx 0.695$. This policy gives an expected grade of $T(89/128) \approx 0.741$, but the

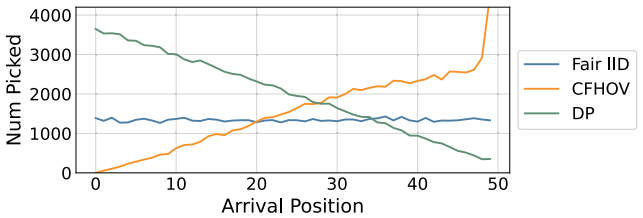
first candidate is accepted with probability 0.305, which is rather unfair to the other three candidates. We can instead calculate thresholds so that all candidates are selected with equal probability, namely $\alpha \cdot 1/4$, for some $\alpha \in [0,1]$. For the uniform distribution, setting $\alpha = 1$ gives in expectation ≈ 0.739. We can thus regain fairness while losing relatively little value. In Figure 3, we visualize the selection frequencies as a function of the arrival position for the same problem with 50 candidates. We plot the selection frequencies for the optimal online algorithm obtained via dynamic programming, the worst-case optimal algorithm of Correa et al. (2021b), and the optimal fair online algorithm that we develop for this case. The optimal online algorithm obtained via dynamic programming chooses much more often from the beginning, whereas the worst-case optimal algorithm chooses mostly from the end. In contrast, our algorithm selects candidates with equal probability. Our algorithm obtains an expected value of 0.661, whereas the fair offline algorithm has a value of $50/51 \approx 0.98$. If the distributions are more skewed, the asymmetry of the selection probabilities in the DP becomes much sharper, but we show in Theorem 6 that if we set $\alpha = 2/3$, then the ratio between the optimal fair offline value and the value of a fair online algorithm stays below 3/2 for any distribution.

1.2.1. Our Results.

We derive the optimal online algorithms and prove tight bounds on their competitive ratio for a range of settings. Our general approach is to mimic the selection probabilities q_i with which FairOpt selects candidate i. We then set up our online algorithms so that they select candidate i with probability $\alpha \cdot q_i$ for some $\alpha \in [0,1]$. In the most general version, we prove that an approximation factor of two is best possible (Theorem 5), whereas improved factors can be obtained by making natural assumptions on the prior probabilities and the arrival order. This includes a tight factor 3/2 approximation for the special case of i.i.d. random variables (Theorem 6) and a tight $1/(2-\sqrt{2}) \approx 1.707$ approximation for the case where random variables are only i.i.d. within each group but arrive in uniform random order (Theorem 7). In Section 3.3, we consider a version of the multicolor prophet problem with samples. We argue that if we are given the probabilities $q_1, \ldots, q_n$, then $O_\varepsilon(\log n)$ samples from each distribution suffice to recover the factor two approximation for the general case up to an additive error of ε, whereas $O_\varepsilon(n)$ samples suffice for the i.i.d. result.

We summarize our findings for the multicolor prophet problem in Table 2. In Section 3.4, we provide further empirical evaluations of our fair online algorithms and compare their performance and fairness guarantees with additional online algorithms that have been proposed in the literature.

Figure 3. (Color online) We Plot the Selection Frequencies as a Function of Arrival Position for the i.i.d. Uniform[0, 1] Case for $n = 50$ Candidates Aggregated Over 100,000 Runs for the Optimal Online Algorithm via DP, the Worst-Case Optimal Online Algorithm of Correa et al. (2021b) (CFHOV), and the Optimal Fair Online Algorithm (Fair IID)



1.3. Related Work

An important precursor to our work is Buchbinder et al. (2014). Their starting point is the observation that the optimal policy for the classic secretary problem introduces incentives for candidates to arrive late. Indeed, that optimal policy skips the first $1/e$ fraction of candidates and then, selects the first candidate who is best so far. With this in mind, Buchbinder et al. (2014) look for *incentive-compatible* policies: that is, policies in which the acceptance probabilities for each of the arrival positions have to be the same. This can also be interpreted as a fairness constraint, according to which selection probabilities should not depend on the time of arrival. Khatibi and Jacobson (2018) use this paradigm to study fairness in the online allocation of tasks to workers. They model this problem through the weighted secretary problem with k positions and design incentive-compatible algorithms for this problem. The incentive-compatibility constraint studied in these two papers has some similarities with the quota approach that we take in the multicolor prophet problem, with the difference that we aim to balance selection probabilities based on the candidates' identities rather than their arrival times.

The poset approach for modeling biased evaluations in secretary problems was initiated by Salem and Gupta (2024). In their model, every secretary has a score, which induces a total order on the secretaries. The online algorithm only has access to a partial order that is consistent with the total order. The goal is to hire d secretaries whose weight is competitive with that of the d highest-scoring secretaries. As a special case of their model, they consider the *group bias model*, which coincides with our model, where secretaries are partitioned

Table 2. Competitive Ratios for the Multicolor Prophet Problem

General	i.i.d.	Random order
2 (Theorem 5)	3/2 (Theorem 6)	$1+\sqrt{2}/2 \approx 1.707$ (Theorem 7)

Note. These guarantees hold for any number of colors.

into k groups and can only be compared with secretaries in their own group. They insist on their algorithms to satisfy *ordinal fairness*; secretaries with the same rank within their group should be accepted with the same probability, and these probabilities should decrease with rank. They consider both an adversarial model and a stochastic model. In the adversarial model, both the scores and the group membership are decided adversarially. In the stochastic model, each secretary selects a group independently with probability $q = (q_1, \ldots, q_k)$, and scores are sampled independently from a common distribution. They show how to parallelize algorithms for the d-choice secretary problem to obtain a $k(1 + o(1))$-competitive algorithm for the adversarial model and a $2e(1 + o(1))$-competitive algorithm for the stochastic setting (as $d \to \infty$). For the single-choice problem, they obtain $O(k)$ and $O(1)$ competitive algorithms. They also show a $\Omega(k)$ lower bound for the adversarial setting, which follows by considering the case where all of the value comes from one group, and the ordinal fairness constraint forces the algorithm to select low scores. In summary, although the models considered by Salem and Gupta (2024) share some features of our problems—comparisons are only possible within groups, like in the multicolor secretary problem, or candidates have scores drawn from a distribution, like in the multicolor prophet problem—they also differ from our problems in significant ways. In particular, the notion of ordinal fairness is based on ranks, whereas our notions of fairness are based on selection probabilities by group.

Kumar et al. (2011) and Feldman and Tennenholtz (2012) do not take a fairness perspective but study secretary problems that share features of our model. Kumar et al. (2011) consider the problem of selecting a maximal secretary from a partially ordered set of candidates. Their algorithm skips the first $\tau(k)$ elements, where k is the number of maximal secretaries. Afterward, it takes any undominated candidate, provided that among the candidates seen so far, there are at most k undominated secretaries. This latter condition may make them pass on undominated candidates who arrive early in the sequence, but it will never pass on the last maximal candidate in the permutation. They show that this algorithm succeeds with probability at least $k^{-k/(k-1)}((1 + \log k^{1/(k-1)})^k - 1)$. This approaches $1/e$ as $k \to 1$ and one as k grows large. For the special case where the poset consists of k chains, they show that their algorithm succeeds with probability $k^{-1/(k-1)} - O(k/n)$, whereas no online algorithm can achieve a better success probability than $k^{-1/(k-1)} - o(1)$ (for $k = o(\sqrt{n})$ and $n \to \infty$). This final result also applies to our problem but only in the special case where the best secretary is equally likely to come from any of the groups. Our result in Corollary 1 strengthens the bounds for this case by showing that for all n, the *tight* bound for this case is $k^{-1/(k-1)}$. This improvement is enabled by Lemma 1, which shows that for all k, the worst case is when $n \to \infty$.

Feldman and Tennenholtz (2012) study the problem of selecting secretaries in parallel; each of n candidates in total is randomly assigned to one of Q queues, and the order of candidates within each queue is random as well. Candidates can only be compared with other candidates in the same queue. Only the first D candidates in each queue can be hired. The objectives are to select d secretaries and to hire as many of the top d secretaries as possible. For a given parameter $k \in \mathbb{N}$, they show that if $d = 1$ and $D = n/k$, then with $Q = 1$ queue, the best-possible ratio is $(de)^{-1}$, whereas with $Q = k$ queues, it is possible to achieve a ratio of $k^{-k/(k-1)} - o(1)$. The connection to the poset model and our model is that the uniform random assignment to queues can be interpreted as assigning each secretary one of Q colors uniformly at random, with all secretaries assigned to the same color forming a chain. The best secretary in a chain is the best secretary overall with probability $1/Q$. So, this model is again restricted to the case where the best secretary overall is equally like to come from any of the groups.

In follow-up work, Arsenis and Kleinberg (2022) study the single-choice prophet inequality problem with individual fairness constraints. They consider two fairness conditions. The *identity-independent fairness* (IIF) condition requires that the probability that the algorithm hires a candidate is independent of the candidate's identity. The *time-independent fairness* (TIF) condition requires that the probability that the algorithm hires a candidate is independent of their arrival time. They show that the gap between the best online IIF algorithm and the best offline (IIF or not) algorithm is $1/2$ and that the gap between the best online TIF algorithm and the best offline algorithm is $1/2$. These results are related because they also consider fairness in a prophet inequality setting, but the approach is more closely related to Buchbinder et al. (2014). Indeed, the IIF and TIF requirements are two natural variations of the incentive-compatibility constraint studied in this earlier work.

A different approach to modeling fairness in online allocation problems is that of Lien et al. (2014). They consider the online allocation of a scarce resource but not from a revenue optimization viewpoint. Rather, they study the fill rate—the ratio of the allocated amount to observed demand—and seek algorithms maximizing the minimum fill rate. This approach captures a natural notion of fairness that seeks to make the least happy customer as happy as possible, but it is rather different from the approaches discussed so far.

Alpern and Baston (2017) study a variant of the classic secretary problem, in which candidates are evaluated independently by two committee members with different objectives. In their model, each candidate is

described by a pair of scores (x, y), where x and y are distributed uniformly and independently on [0, 1] and observable by both selectors. One committee member is interested in x, and the other is interested in y. The utility that the firm derives from a candidate is assumed to be $(x + y)/2$. Unanimous hiring decisions are respected, whereas candidates with a split decision are hired with probability p. A consensus cost c is deducted from the utility of a selector who has rejected a candidate who is nevertheless hired. The main result is that each stage game has a unique (symmetric) equilibrium, which involves setting two thresholds $z < v$. A selector recommends to hire if her value is at least v or her value is at least z and the other value is at least v. They then examine how p and c affect the utilities of the selectors and the firm at equilibrium. So, this work studies "biased" committee members and characterizes optimal recommendation strategies for a fixed conflict resolution scheme.

A final set of related works considers secretary and/or prophet problems with restrictions on which information is available to the decision maker without considering whether this leads to fair decisions. Bhattacharjya and Deleris (2014), for example, consider an optimal stopping problem where the decision maker gets imperfect information about random variables presented to her one by one, such as the expected reward or some multidimensional signal and the expected reward conditional on that signal. Similarly and closely related to our results for the multicolor secretary problem and the multicolor prophet problem with samples is the strand of research studying online selection problems with samples. This line of research was initiated by Azar et al. (2014), who considered combinatorial variants of the prophet inequality problem where instead of having full distributional knowledge, we only have access to one sample from each distribution. The single-choice problem with samples was first studied in more detail in Correa et al. (2022). In subsequent work, Rubinstein et al. (2020) establish that one sample from each distribution is enough to achieve the optimal constant factor prophet inequality for the single-selection problem with nonidentical distributions. Related models have been recently studied for the secretary problem. Kaplan et al. (2020) consider a dependent sampling model in which k of n candidates are first sampled and then, their relative rankings are revealed. Here, we adopt the somewhat simpler independent sampling model of Correa et al. (2021a), in which each element is observed with probability q. For a general unifying framework of single-selection problems with sampling (advice), we refer the reader to the work of Dütting et al. (2021). This framework captures cases where the quality of the candidate is drawn from a known distribution or from an unknown distribution, and the designer receives some samples from it. It also encompasses cases where the online decision maker has learned a binary classifier that directly recommends whether the current secretary is the best overall or not.

2. The Multicolor Secretary Problem

In the *multicolor secretary problem*, n candidates arrive in uniform random order. Candidates are partitioned into k groups $C = \{C_1, \dots, C_k\}$. We write $\mathbf{n} = (n_1, \dots, n_k)$ for the vector of group sizes (i.e., $|C_j| = n_j$) for all $1 \le j \le k$. We identify each of the groups with a distinct color and denote by $c(i)$ the color of candidate i. We can compare candidates of the same color, but we cannot compare candidates across groups. We assume that comparisons are strict and use $i \succ i'$ to denote that candidate i is better than candidate i'. We write $\max C_j$ for the best candidate of color j and $\max C$ for the best candidate overall. A natural assumption is that the best candidate from a group is the best candidate overall with equal probability $1/k$, but we can also consider the case where these probabilities are different. We denote the probabilities with which the best candidate of group j is the best candidate overall by p_j, and we write $\mathbf{p} = (p_1, \dots, p_k)$ for the vector of these probabilities. Because candidates are incomparable across groups, our decision is independent of which group contains the best candidate. Therefore, it is equivalent to assume that the group that the best candidate belongs to is realized *after the fact* (i.e., if we select the best candidate of color j, the selected candidate is the overall best with probability p_j, independently of everything else). The goal is to design an online algorithm that maximizes the probability of selecting the best candidate overall.

2.1. Key Definitions

2.1.1. Competitive Ratio. We evaluate online algorithms by means of their competitive ratio. Consider some online algorithm ALG. The algorithm selects the best candidate overall if, on the one hand, it selects the best candidate of a given color, and, on the other, this color has the best candidate overall. For an instance of the multicolor secretary problem with group sizes $\mathbf{n}$ and probabilities $\mathbf{p}$, we denote by $\mathrm{ALG}(\mathbf{n}, \mathbf{p}) \in \{1, \dots, n\} \cup \{\phi\}$ the random index at which the algorithm stops, where ϕ denotes the case when the algorithm does not stop. The success probability of ALG is the probability that it picks the best candidate overall: namely, $\mathbf{E}[\mathbf{1}_{\mathrm{ALG}(\mathbf{n},\mathbf{p})=\max C}] = \mathbf{E}[p_{c(\mathrm{ALG}(\mathbf{n},\mathbf{p}))} \cdot \mathbf{1}_{\mathrm{ALG}(\mathbf{n},\mathbf{p})=\max C_{c(\mathrm{ALG}(\mathbf{n},\mathbf{p}))}}]$. We compare this with the optimal offline algorithm **OPT** (i.e., the best algorithm that can select a candidate after all candidates have arrived). An optimal strategy is to choose a color j with maximum p_j and then, choose the best candidate of that color. We denote by $\mathbf{OPT}(\mathbf{n}, \mathbf{p}) \in \{1, \dots, n\}$ the random index that OPT selects. The success probability of OPT is $\mathbf{E}[\mathbf{1}_{\mathbf{OPT}(\mathbf{n},\mathbf{p})=\max C}] = \max\{p_1, \dots, p_k\}$.

Definition 1. (Competitive Ratio). Fix k and $\mathbf{p} = (p_1, \ldots, p_k)$. An online algorithm ALG is $\beta(k, \mathbf{p})$-*competitive* if for all input lengths n and partition sizes $\mathbf{n} = (n_1, \ldots, n_k)$,

$$\frac{\mathbf{E}[\mathbf{1}_{\mathrm{OPT}(\mathbf{n},\mathbf{p})=\mathrm{maxC}}]}{\mathbf{E}[\mathbf{1}_{\mathrm{ALG}(\mathbf{n},\mathbf{p})=\mathrm{maxC}}]} \le \beta(k, \mathbf{p}).$$

Note that $\beta(k, \mathbf{p}) \ge 1$ and that the smaller $\beta(k, \mathbf{p})$ is, the better the approximation guarantee is.

2.1.2. Unbiased Selection.

We also examine the extent to which online or offline algorithms are biased where ideally, selection should be unbiased. One way to measure this is by quantifying how much the probability of selecting from any given color class j can differ from the corresponding probability p_j. The goal of abiding to the probabilities $\mathbf{p}$ as much as possible can be viewed as a form of meritocratic fairness (Joseph et al. 2016).

Definition 2. (Fairness). Fix k and $\mathbf{p} = (p_1, \ldots, p_k)$. An offline or online algorithm ALG is $\alpha(\mathbf{n}, \mathbf{p})$-*fair*, where $\alpha(\mathbf{n}, \mathbf{p}) \ge 1$, if for all colors $j \in [k]$,

$$\frac{p_j}{\alpha(\mathbf{n}, \mathbf{p})} \le \mathbf{P}(c(\mathrm{ALG}(\mathbf{n}, \mathbf{p})) = j \mid \mathrm{ALG}(\mathbf{n}, \mathbf{p}) \ne \phi) \le \alpha(\mathbf{n}, \mathbf{p}) \cdot p_j.$$

2.1.3. Uniform Arrival Times.

We model *uniform random arrival order* through *uniform random arrival times*. For this, we sample n independent realizations of the Uniform[0, 1] distribution and denote them by $\tau_1 < \tau_2 < \cdots < \tau_n$ indexed in increasing order.

2.2. Optimal Online Algorithm

We derive the optimal online algorithm (without fairness considerations) and observe that—in sharp contrast to the optimal offline algorithm—it is robustly fair and provides an "equal treatment of equals" (Aristotle) guarantee.

2.2.1. The Algorithm.

We show that the optimal online algorithm is from the class of algorithms given by Algorithm 1. Algorithms from this class receive as input a vector of thresholds $\mathbf{t} = (t_1, \ldots, t_k)$, one for each color $j \in [k]$. When a candidate i arrives, the algorithm first checks if the candidate arrived after the time threshold for its color $t_{c(i)}$, and if it did, then it accepts the candidate if it is the best candidate of that color so far.

Algorithm 1 (GroupThresholds($\mathbf{t}$))

Input: $\mathbf{t} \in [0,1]^k$, a threshold in time for each group
Output: $i \in [n]$, index of chosen candidate
/* assuming arrival times $\tau_1 < \cdots < \tau_n$ */
for $i \leftarrow 1$ **to** n **do**
  if $\tau_i > t_{c(i)}$ then
    if $i > \max\{i' \mid \tau_{i'} \le \tau_i, c(i') = c(i)\}$ **then**
      return i
    end
  end
end

Notice that the time-based arrival model considered in this section is equivalent to the random order arrival model and is used for the sake of simplicity of presentation and proofs. If we are given an algorithm in the time-based model (such as Algorithm 1), then we can translate it into the random arrival model by having the algorithm draw n arrival times from Uniform[0, 1] and assign the ith smallest arrival time to the ith candidate in the input stream. If, on the other hand, we are given an algorithm in the uniform arrival model, then we can translate it into the time-based model by just ignoring the time component and just using that the candidate who arrived at τ_i was the ith candidate to arrive.

Therefore, any algorithm in one model can be easily used in the other model with identical properties.

2.2.2. Competitive Ratio.

Surprisingly, we can show that for any probabilities $\mathbf{p} = (p_1, \ldots, p_k)$, there exist optimal thresholds $\mathbf{t}^* = (t_1^*, \ldots, t_k^*)$ that achieve the best competitive ratio. Later on, we show how these thresholds can be computed explicitly (see Lemma 4). Using these thresholds in Algorithm 1 results in the promised optimal online algorithm. Let us start by presenting the success probability of our algorithm for general probabilities and then, for the special case that $\mathbf{p} = (1/k, \ldots, 1/k)$. Afterward, we provide an overview of the proof of these results.

Theorem 1. (Competitive Ratio, General Probabilities). *Fix k and $\mathbf{p} = (p_1, \ldots, p_k)$. Assume w.l.o.g. that $p_j \ge p_{j+1}$ for all $j < k$. Then, there exist thresholds $\mathbf{t}^* = (t_1^*, \ldots, t_k^*)$ such that $t_j^* \le t_{j+1}^*$ for all $j < k$ that depend only on the number of colors k and the probabilities $\mathbf{p}$ but not on the number of candidates n or the partition sizes $\mathbf{n} = (n_1, \ldots, n_k)$ such that Algorithm 1 with thresholds $\mathbf{t}^*$ succeeds with probability at least*

$$\sum_{j=1}^{k} \int_{t_j^*}^{t_{j+1}^*} \left(\sum_{j'=1}^{j} p_{j'} \right) \frac{T_j^*}{\tau^j} \, d\tau,$$

where $T_j^* = \prod_{j'=1}^{j} t_{j'}$. For all k and $\mathbf{p} = (p_1, \ldots, p_k)$, no online algorithm can achieve a better competitive ratio in the worst case over all numbers of candidates n and partition sizes $\mathbf{n} = (n_1, \ldots, n_k)$.

For the special case where $\mathbf{p} = (1/k, \ldots, 1/k)$, we obtain the following corollary. It shows that in this case, we can set a single threshold, and it also provides a simpler-to-parse formula for the competitive ratio.

Corollary 1. (Competitive Ratio, Equal Probabilities). *Fix k and $\mathbf{p} = (1/k, \ldots, 1/k)$. Then, there exists a single threshold t^* such that Algorithm 1 with thresholds $\mathbf{t}^* = (t^*, \ldots, t^*)$*

achieves a competitive ratio of

$$k^{\frac{1}{k-1}}.$$

This is two for $k=2$, $\sqrt{3}$ *for* $k=3$, *and* $1+O(\frac{\log k}{k})$ *as* $k\to\infty$. *For all* k *and* $\mathbf{p}=(1/k,\dots,1/k)$, *no online algorithm can achieve a better competitive ratio in the worst case over all numbers of candidates* n *and partition sizes* $\mathbf{n}=(n_1,\dots,n_k)$.

The main difficulty in proving Theorem 1 and Corollary 1 is that in the point-wise optimal online algorithm, which can be obtained by backward induction, thresholds depend on the number of candidates of each color that have already arrived. This dependency leads to a blowup in algorithm complexity and complicates the analysis of the success probability. Our high-level approach is to argue that in the worst case, all n_j's are large and that in this case, the point-wise optimal online algorithm is well approximated by the optimal algorithm from the class of algorithms described in Algorithm 1, which simply sets time-dependent thresholds. So, we can optimize over these.

A first ingredient in our proof is Lemma 1, which shows that for the class of algorithms in Algorithm 1, for any vector of thresholds $\mathbf{t}$, the worst case arises when all n_j's are large.

Lemma 1. (Monotonicity). *Fix the probabilities* $\mathbf{p}=(p_1,\dots,p_k)$ *and a vector of thresholds* $\mathbf{t}\in[0,1]^k$. *For all* $j=1,\dots k$, *the success probability of* GroupThresholds($\mathbf{t}$) *is decreasing in* n_j.

Proof. By symmetry, it is enough to prove the lemma for $j=1$. Fix values for $n_1,n_2,\dots,n_k$, and consider an instance with these sizes. We prove that the success probability of GroupThresholds(t) in this instance is lower than the success probability in the instance with n_1-1 candidates of group 1 and n_j candidates of groups $j>1$ that results from removing the worst candidate of group 1. In fact, we can couple the realizations of the arrival times of the two instances by taking the arrival times of the smaller instance and sampling the arrival time of the worst candidate of group 1. Consider a realization for the smaller instance where GroupThresholds($\mathbf{t}$) fails. Then, either it never accepts a candidate, or it accepts a candidate who is not the overall maximum. In any of the two cases, adding the worst candidate of group 1 does not alter the relative ranks of other candidates, so the only possible difference is that the algorithm now selects him. Because he cannot be the best candidate, the algorithm also fails. □

Our next pair of lemmas, Lemma 2 and Lemma 3, allows us to bound the success probability of the point-wise optimal online algorithm by the limit success probability of the best algorithm, which sets time-dependent thresholds (Algorithm 1).

Lemma 2. *For given probabilities* $\mathbf{p}$ *and thresholds* $\mathbf{t}$ *that satisfy* $\min_j t_j\geq c>0$, *there is a value* $\mathbf{GT}(\mathbf{p},\mathbf{t})$ *such that the success probability of* GroupThresholds($\mathbf{t}$) *is at least* $\mathbf{GT}(\mathbf{p},\mathbf{t})$ *and at most* $\mathbf{GT}(\mathbf{p},\mathbf{t})+k\cdot(1-c)^z$, *where* $z=\min_j n_j$. *We call this value the limit success probability.*

Proof. Fix a vector of probabilities $\mathbf{p}$. Consider a vector $\mathbf{t}$ such that $\min_j t_j\geq c$ and two vectors of sizes $\mathbf{n},\mathbf{n}'$ such that $n_j<n_j'$ for all $j\in[k]$. Denote by $GT(\mathbf{p},\mathbf{t},\mathbf{n})$ and $GT(\mathbf{p},\mathbf{t},\mathbf{n}')$ the success probabilities of GroupThresholds($\mathbf{t}$) when the instance is of size $\mathbf{n}$ and $\mathbf{n}'$, respectively.

Couple the arrival times of the two instances in the following way. For each color j, identify the n_j candidates of color j of the smaller instance with the best n_j candidates of color j in the larger instance. Now, we run in parallel GroupThresholds($\mathbf{t}$) in the two coupled instances. This means that in the smaller instance, we will be ignoring the smallest $n_j'-n_j$ candidates of the larger instance. Note that the smallest elements do not alter the relative rank of the largest elements. Thus, if the algorithm in the smaller instance stops, then the algorithm running in the larger instance either stops with the same candidate or stops earlier with one of the smallest $n_j'-n_j$ candidates of some color j. Now, for a given color j, if one of the best n_j candidates arrives before t_j, then the algorithm in the larger instance will never select any of the smallest $n_j'-n_j$ candidates. Therefore, if for all colors j, at least one of the largest n_j candidates of color j arrives before t_j, the two algorithms stop with the same candidate. In other words, if the two algorithms stop with different candidates necessarily for some color j, the best n_j candidates arrived all after t_j. Using a union bound, the latter event happens with probability at most $\sum_{j=1}^k(1-t_j)^{n_j}$, which in turn, is at most $k\cdot(1-c)^z$, where $z=\min_j n_j$. Therefore,

$$GT(\mathbf{p},\mathbf{t},\mathbf{n})-k\cdot(1-c)^z\leq GT(\mathbf{p},\mathbf{t},\mathbf{n}').$$

Note that only the right-hand side depends on $\mathbf{n}'$, so we can take a limit when $\min_j n_j'$ tends to infinity, which by Lemma 1, exists, and conclude the result. □

Lemma 3. *For any vector of probabilities* $\mathbf{p}$ *and sizes* $\mathbf{n}$, *denote by* $\mathbf{ON}(\mathbf{n},\mathbf{p})$ *the optimal success guarantee of an online algorithm. Then, for every* $\mathbf{p},\mathbf{n}$, *there exists a vector* $\mathbf{t}$ *such that* $\mathbf{On}(\mathbf{n},\mathbf{p})\leq\mathbf{GT}(\mathbf{p},\mathbf{t})+o(1)$, *where* $\min_j t_j\geq 1/(2e)$.

Proof. Consider the optimal online algorithm obtained doing backward induction. This is when facing a candidate i that is the best seen so far of color $c(i)$ and having seen r_j candidates of color j, for $j\in[k]$, accept candidate i if the probability that i is the overall best is larger than or equal to the probability that we select the best overall if we do not stop with i and continue using the optimal policy. Denote by $b(c(i),\mathbf{r})$ the former probability and by $B(\mathbf{r})$ the latter probability. Thus, the optimal algorithm stops if $b(c(i),\mathbf{r})\geq B(\mathbf{r})$.

On the one hand, we can calculate exactly $b(c(i),\mathbf{r})$. Let $j=c(i)$. It is the probability that the best of the first r_j candidates of color j is the best candidate overall. Now, the best of the first r_j candidates of color j is the best of color j with probability r_j/n_j, and the best of color j is the best overall with probability p_j. Therefore, $b(c(i),\mathbf{r})=b(j,\mathbf{r})=p_j\cdot r_j/n_j$, which is an increasing function of r_j. On the other hand, $B(\mathbf{r})$ is not as easy to calculate, but it is easy to see that it is a decreasing function of r_j, for all $j\in[k]$.

Suppose we are considering a candidate arriving at time t. If $\min_{j'} n_{j'}$ is large, we have that $r_j/n_j\approx t$. Taking advantage of this fact, we approximate the optimal online algorithm with the following that we denote by $\mathbf{ON}'(\mathbf{n},\mathbf{p})$. We accept a candidate of color j that arrives at time t if he is the best of color j seen so far, and $b(j,\lfloor t\mathbf{n}\rfloor)\geq B(\lfloor t\mathbf{n}\rfloor)$, where $\lfloor t\mathbf{n}\rfloor:=(\lfloor tr_1\rfloor,\dots,\lfloor tr_k\rfloor)$. By the monotonicity of b and B, for each color j, there is a value t'_j such that $b(j,\lfloor t\mathbf{n}\rfloor)\geq B(\lfloor t\mathbf{n}\rfloor)$ if and only if $t\geq t'_j$. Thus, it turns out that for $\mathbf{t}'$ defined this way, $\mathbf{ON}'(\mathbf{n},\mathbf{p})$ is actually GroupThresholds($\mathbf{t}'$).

We prove first that $\min_j t'_j\geq 1/(2e)$. In fact, note that the success probability of the optimal algorithm must be at least $\frac{1}{e}\max_j p_j$ because we can always apply the regular secretary algorithm in the color with highest p_j. Also, the probability that the optimal online algorithm selects the overall best candidate before time $1/(2e)$ is at most $\frac{1}{2e}\max_j p_j$ because the best of a color arrives before time $1/(2e)$ with probability $1/(2e)$. Therefore, the probability that the optimal algorithm finds the overall best candidate after time $1/(2e)$ is at least $\frac{1}{2e}\max_j p_j$, and therefore, $B(\lfloor\mathbf{n}/(2e)\rfloor)\geq\frac{1}{2e}\max_j p_j$. Recall that $b(j,\mathbf{r})=p_j r_j/n_j$, so $b(j,\lfloor\mathbf{n}/(2e)\rfloor)\leq p_j/(2e)$. Thus, we conclude that $t'_j\geq 1/(2e)$ for every $j\in[k]$. Using Lemma 2, we have that $\mathbf{ON}'(\mathbf{n},\mathbf{p})\leq\mathbf{GT}(\mathbf{p},\mathbf{t}')+k\cdot(1-\frac{1}{2e})^{\min_j n_j}$.

Now, we prove that $\mathbf{ON}'(\mathbf{n},\mathbf{p})$ approximates well $\mathbf{ON}(\mathbf{p},\mathbf{n})$. Consider some small $\varepsilon>0$. We show that (1) for large $\min_j n_j$, the probability that either of the two algorithms stops with a color j in $[t'_j-\varepsilon,t'_j+\varepsilon]$ is small and that (2) outside that interval, the two algorithms make the same decisions with high probability.

For the first fact, note that both algorithms stop only with a candidate who is the best seen so far of the same color. Now, if the best candidate of color j in the interval $[0,t'_j+\varepsilon]$ arrives in $[0,t'_j-\varepsilon]$, the algorithm will not stop with a candidate of color j in $[t'_j-\varepsilon,t'_j+\varepsilon]$. Thus, either of the two algorithms stops in $[t'_j-\varepsilon,t'_j+\varepsilon]$ with a candidate of color j for some $j\in[k]$, with probability at most $\sum_{j=1}^{k}\frac{2\varepsilon}{t'_j}\leq 4e\varepsilon k$.

For the second fact, we prove that with high probability, for each $j\in[k]$, at time $t=t'_j-\varepsilon$, $\mathbf{r}\leq t'_j\mathbf{n}$ and at time $t=t'_j+\varepsilon$, $\mathbf{r}\geq t'_j\mathbf{n}$, where the comparisons are element wise. We use the following standard Chernoff bounds. If X is a binomial random variable and $0<\delta<1$, then

$$\mathbf{P}(X\geq(1+\delta)\mathbf{E}(X))\leq e^{-\frac{1}{3}\delta^2\mathbf{E}(X)},\text{ and}$$

$$\mathbf{P}(X\leq(1-\delta)\mathbf{E}(X))\leq e^{-\frac{1}{3}\delta^2\mathbf{E}(X)}.$$

Because r_j at time t distributes as a Binomial(n_j,t), we can use these bounds to get that at time $t=t'_j-\varepsilon$,

$$\mathbf{P}(r_j\geq t'_j n_j)\leq\exp\left(-\frac{1}{3}\varepsilon^2 n_j(t'_j-\varepsilon)\right)$$

and at time $t=t'_j+\varepsilon$,

$$\mathbf{P}(r_j\leq t'_j n_j)\leq\exp\left(-\frac{1}{2}\varepsilon^2 n_j(t'_j+\varepsilon)\right).$$

Because $t'_j\geq 1/(2e)$ for all j, we can take $\varepsilon=20\frac{\log n_j}{\sqrt{n_j}}$ and use a union bound to obtain that with probability $1-O\left(\frac{k}{\min_{j'} n_{j'}}\right)$, for each $j\in[k]$, at time $t=t'_j-\varepsilon$, $\mathbf{r}\leq t'_j\mathbf{n}$ and at time $t=t'_j+\varepsilon$, $\mathbf{r}\geq t'_j\mathbf{n}$. From the monotonicity of b and B, we have that with probability $1-O\left(\frac{k}{\min_{j'} n_{j'}}\right)$, for all $j\in[k]$, before time $t'_j-\varepsilon$, it holds that $b(j,\mathbf{r})<B(\mathbf{r})$, and after time $t'_j+\varepsilon$, it holds that $b(j,\mathbf{r})>B(\mathbf{r})$.

Putting together facts (1) and (2), we get that with probability $1-O\left(k\frac{\log(\min_j n_j)}{\sqrt{\min_j n_j}}\right)$, $\mathbf{ON}'(\mathbf{n},\mathbf{p})$ and $\mathbf{ON}(\mathbf{n},\mathbf{p})$ select the same candidate. Thus, when $\min_j n_j$ tends to infinity,

$$\mathbf{ON}(\mathbf{n},\mathbf{p})\leq\mathbf{ON}'(\mathbf{n},\mathbf{p})+o(1)\leq\mathbf{GT}(\mathbf{p},t')+o(1).$$

This concludes the proof of the lemma. □

The final ingredient is the following pair of lemmas, Lemma 4 and Lemma 5; they solve for the optimal time-dependent thresholds and give a formula for evaluating the limit success probability in terms of these thresholds.

Lemma 4. *Consider a vector* $\mathbf{p}$ *such that* $p_j\geq p_{j+1}$ *for all* $j<k$. *The optimal thresholds* $\mathbf{t}^*$ *are given by*

$$t^*_k=(1-(k-1)p_k)^{\frac{1}{k-1}},$$

$$t^*_j=t^*_{j+1}\cdot\left(\frac{\sum_{r=1}^{j}\frac{p_r}{j-1}-p_j}{\sum_{r=1}^{j}\frac{p_r}{j-1}-p_{j+1}}\right)^{\frac{1}{j-1}}\text{ for }2\leq j\leq k-1$$

$$t^*_1=t^*_2\cdot e^{\frac{p_2}{p_1}-1}.$$

Proof. From the formula in Lemma 5, we can check that in the optimal vector $\mathbf{t}^*$, it holds that $t^*_j\leq t^*_{j'}$ if $p_j\geq p_{j'}$ by simply interchanging consecutive t_j's. With this in hand, we can assume w.l.o.g. that $p_1\geq p_2\geq\cdots\geq p_k$

and $t_1^* \le t_2^* \le \cdots \le t_k^*$. We impose first-order conditions to obtain the recursive formula.

Consider $1 \le j \le k$. We have that

$$\frac{\partial}{\partial t_j}\mathbf{GT}(\mathbf{p},\mathbf{t}) = \frac{1}{t_j}\sum_{\ell=j}^{k}\int_{t_\ell}^{t_{\ell+1}}\left(\sum_{r=1}^{\ell}p_r\right)\frac{T_\ell}{\tau^\ell}\,d\tau - p_j\frac{T_j}{t_j^j}.$$

Setting this derivative equal to zero, we obtain the equation

$$p_j\frac{T_j^*}{t_j^{*j-1}} = \sum_{\ell=j}^{k}\int_{t_\ell^*}^{t_{\ell+1}^*}\left(\sum_{r=1}^{\ell}p_r\right)\frac{T_\ell^*}{\tau^\ell}\,d\tau \tag{1}$$

for all $1 \le j \le k$.

For the case of $j = k$, note that $\sum_{r=1}^{k} p_r = 1$, so we obtain the following formula for t_k^*:

$$\frac{p_k}{t_k^{*k-1}} = \frac{1}{k-1}\left(\frac{1}{t_k^{*k-1}} - 1\right) \iff t_k^* = (1-(k-1)p_k)^{\frac{1}{k-1}}.$$

For $1 \le j \le k-1$, we can subtract Equation (1) for two consecutive indices, obtaining

$$\begin{aligned} p_j\frac{T_j^*}{t_j^{*j-1}} - p_{j+1}\frac{T_{j+1}^*}{t_{j+1}^{*j}} &= \int_{t_j^*}^{t_{j+1}^*}\left(\sum_{r=1}^{j}p_r\right)\frac{T_j^*}{\tau^j}\,d\tau \iff \\ \frac{p_j}{t_j^{*j-1}} - \frac{p_{j+1}}{t_{j+1}^{*j-1}} &= \int_{t_j^*}^{t_{j+1}^*}\left(\sum_{r=1}^{j}p_r\right)\frac{1}{\tau^j}\,d\tau. \end{aligned} \tag{2}$$

For $i \ge 2$, this is equivalent to

$$\begin{aligned} &\frac{p_j}{t_j^{*j-1}} - \frac{p_{j+1}}{t_{j+1}^{*j-1}} = \frac{1}{j-1}\left(\sum_{r=1}^{j}p_r\right)\left(\frac{1}{t_j^{*j-1}} - \frac{1}{t_{j+1}^{*j-1}}\right) \iff \\ &t_j^* = t_{j+1}^*\left(\frac{\sum_{r=1}^{j}\frac{p_r}{j-1} - p_j}{\sum_{r=1}^{j}\frac{p_r}{j-1} - p_{j+1}}\right)^{\frac{1}{j-1}}. \end{aligned}$$

For $j = 1$, Equation (2) becomes

$$\begin{aligned} &p_1 - p_2 = p_1\int_{t_1^*}^{t_2^*}\frac{1}{\tau}d\tau \iff p_1 - p_2 = p_1\log(t_2^*/t_1^*) \iff \\ &t_1^* = t_2^*\exp\left(\frac{p_2}{p_1} - 1\right). \end{aligned}$$

This concludes the proof of the lemma. □

Lemma 5. *Consider vectors of probabilities* $\mathbf{p}$ *and thresholds* $\mathbf{t}$*, and assume* $t_i \le t_{i+1}$ *for all* $i < k$*. The limit success probability of* GroupThresholds($\mathbf{t}$) *is given by*

$$\mathbf{GT}(\mathbf{p},\mathbf{t}) = \sum_{j=1}^{k}\int_{t_j}^{t_{j+1}}\left(\sum_{j'=1}^{j}p_{j'}\right)\frac{T_j}{\tau^j}\,d\tau,$$

where $T_j = \prod_{j'=1}^{j} t_{j'}$.

Proof. Because we are interested in the limit probability, we can assume that in every interval, at least one candidate of each color arrives. First, note that the algorithm does not stop before time τ if and only if for every color j, the best candidate who arrives in $[0,\tau]$ arrives before t_j. This happens with probability

$$\prod_{j=1}^{k}\frac{\min\{\tau,t_j\}}{\tau}.$$

Now, the algorithm stops with the best candidate of color j' if this candidate arrives at a time $\tau \ge t_{j'}$ and the algorithm does not stop before time τ. Therefore, conditioning on τ, the probability that the algorithm selects the best candidate of color j' is

$$\int_{t_{j'}}^{1}\prod_{j=1}^{k}\frac{\min\{\tau,t_j\}}{\tau}d\tau.$$

If the algorithm stops with the best candidate of color j', the algorithm succeeds with probability $p_{j'}$. Therefore, in total, the algorithm succeeds with probability

$$\begin{aligned} \sum_{j'=1}^{k}p_{j'}\int_{t_{j'}}^{1}\prod_{j=1}^{k}\frac{\min\{\tau,t_j\}}{\tau}d\tau &= \sum_{j'=1}^{k}p_{j'}\sum_{j=j'}^{k}\int_{t_j}^{t_{j+1}}\frac{T_j}{\tau^j}d\tau \\ &= \sum_{j=1}^{k}\int_{t_j}^{t_{j+1}}\left(\sum_{j'=1}^{j}p_{j'}\right)\frac{T_j}{\tau^j}\,d\tau, \end{aligned}$$

where $t_{j+1} := 1$ and $T_j = \prod_{r=1}^{j} t_r$. □

Putting together these lemmas yields Theorem 1.

Proof of Theorem 1. From Lemma 1, we have that the success probability of the algorithm is at least its limit. Lemma 4 characterizes the optimal thresholds; from Lemma 3, we have that in the limit, the optimal online algorithm has the same success guarantee as GroupThresholds(t^*), so no other algorithm can have a better worst-case guarantee. □

2.2.3. Fairness. The optimal offline algorithm is 1-fair for $\mathbf{p} = (1/k,\ldots,1/k)$, but as soon as probabilities are unbalanced, it will choose only from the colors that have maximum p_j. In the worst case, $|p_j - p_{j'}| < \epsilon$ for all j,j', but the optimal offline algorithm is forced to

choose from the unique color j, which has maximum p_j. We show that in the case where $p_j = 1/k$ for all j, the optimal online algorithm is not exactly 1-fair but approaches 1-fairness exponentially fast in the minimum group size $\min_j n_j$.

Theorem 2. (Fairness Result, Equal Probabilities). *For any k and $\mathbf{p} = (1/k, \dots, 1/k)$, Algorithm 1 with the optimal single threshold t^* is $1 + O(k^2(1-\frac{1}{e})^{\min_j n_j})$-fair.*

Proof. From the first-order conditions, we have that

$$\frac{\partial}{\partial t_1}\mathbf{GT}(\mathbf{p}, \mathbf{t}^*) = 0 \quad \Longleftrightarrow$$
$$\frac{1}{t_1^*}\mathbf{GT}(\mathbf{p}, \mathbf{t}^*) - p_1 = 0 \Longleftrightarrow \mathbf{GT}(\mathbf{p}, \mathbf{t}^*) = p_1 \cdot t_1^*.$$

Because the optimal success probability is at least p_1/e, we have that $t_1^* \geq 1/e$.

To see that the algorithm is $1 + O(k^2(1-1/e)^{\min_j n_j})$-fair, consider an instance of sizes $n_1' = n_2' = \cdots = n_k' = \max_j n_j$. Certainly in this other instance, the algorithm is 1-fair because all of the thresholds are equal. We couple the two instances by identifying the best n_j candidates of color j of the larger instance with the candidates of color j of the smaller instance, and we run the algorithm in both in parallel. In color j, the worst $n_j' - n_j$ candidates do not alter the relative rank of the best n_j candidates, and if one of the best n_j candidates arrives before time t_j^*, then the algorithm in the larger instance will not select one of the worst $n_j' - n_j$ candidates. Therefore, in order for the two algorithms to select a different candidate, we need that for some color j, all of the best n_j candidates arrive after t_j^*. This happens with probability at most $\sum_{j=1}^k (1-t_j^*)^{n_j} \leq k(1-\frac{1}{e})^{\min_j n_j}$. In the larger instance, the algorithm stops with each color with probability $\Theta(1/k)$, so $k(1-\frac{1}{e})^{\min_j n_j}$ is an $O(k^2(1-\frac{1}{e})^{\min_j n_j})$ fraction of it; therefore, the algorithm in the smaller instance is $1 + O(k^2(1-\frac{1}{e})^{\min_j n_j})$-fair. □

Moreover, we show that the optimal online algorithm is robust and degrades gracefully as we move away from perfectly balanced probabilities.

Theorem 3. (Fairness Result, General Probabilities). *Fix k and $\mathbf{p} = (p_1, \dots, p_k)$. Algorithm 1 with the optimal choice of thresholds $\mathbf{t}^* = (t_1^*, \dots, t_k^*)$ ensures that if $p_j = p_{j'}$, then $t_j^* = t_{j'}^*$. Moreover, $\mathbf{t}^*$ is a continuous function of $\mathbf{p}$. So, if p_j and $p_{j'}$ are close, so are t_j^* and $t_{j'}^*$, and so is the probability of selection. More precisely, if $p_j > p_{j'} > (1-\epsilon)p_j$, then $t_{j'}^* > t_j^* > (1-\epsilon)t_{j'}^*$, and furthermore,*

$$0 < \mathbf{P}(\textsc{GroupThresholds}(\mathbf{t}^*) \text{ selects color } j) - \mathbf{P}(\textsc{GroupThresholds}(\mathbf{t}^*) \text{ selects color } j') < \epsilon.$$

Proof. The facts that $p_j = p_{j'}$ implies $t_j^* = t_{j'}^*$ and that $\mathbf{t}^*$ is continuous in $\mathbf{p}$ follow directly from the formulas in Lemma 4.

We prove now the more precise bound. Assume $p_1 \geq p_2 \geq \cdots \geq p_k$ and that $p_{j+1} \geq (1-\varepsilon)p_j$. We will prove first that $t_j^* \geq e^{-\varepsilon} t_{j+1}^*$. From Lemma 4, it holds trivially for $j = 1$. For $j \geq 2$, we have that t_j^*/t_{j+1}^* equals

$$\left(\frac{\sum_{r=1}^j \frac{p_r}{j-1} - p_j}{\sum_{r=1}^j \frac{p_r}{j-1} - p_{j+1}}\right)^{\frac{1}{j-1}} \geq \left(\frac{\sum_{r=1}^j \frac{p_j}{j-1} - p_j}{\sum_{r=1}^j \frac{p_j}{j-1} - p_{j+1}}\right)^{\frac{1}{j-1}}$$
$$\geq \left(\frac{\frac{j}{j-1} - 1}{\frac{j}{j-1} - (1-\varepsilon)}\right)^{\frac{1}{j-1}}$$
$$= \left(\frac{1}{1+(j-1)\varepsilon}\right)^{\frac{1}{j-1}}$$
$$= \exp\left(-\frac{\log(1+(j-1)\varepsilon)}{j-1}\right)$$
$$\geq e^{-\varepsilon}.$$

Consider now nonconsecutive $j < j'$. Assume $p_{j'} = (1-\varepsilon)p_j$, and for $j \leq r \leq j'-1$, define ε_r such that $(1-\varepsilon_r) = \frac{p_{r+1}}{p_r}$. This means that $(1-\varepsilon) = \prod_{r=j}^{j'-1}(1-\varepsilon_r)$. Now, we have that

$$\frac{t_j^*}{t_{j'}^*} = \prod_{r=j}^{j'-1} \frac{t_r^*}{t_{r+1}^*} \geq \exp\left(-\sum_{r=j}^{j'-1} \varepsilon_r\right) \geq \prod_{r=j}^{j'-1}(1-\varepsilon_r) = (1-\varepsilon).$$

To bound the difference between the probabilities of selecting colors j and j' when $p_j \geq p_{j'} \geq (1-\varepsilon)p_j$, note first that conditional on that the algorithm does not stop before $t_{j'}^*$, it stops with either of the two colors with equal probability. Thus, the difference is the probability that the algorithm stops with color j in the interval $[t_j^*, t_{j'}^*]$. Now, this is upper bounded by the probability that the best candidate of color j from those who arrive in $[0, t_{j'}^*]$ arrives in $[t_j^*, t_{j'}^*]$, which is at most

$$\frac{t_{j'}^* - t_j^*}{t_{j'}^*} = 1 - \frac{t_j^*}{t_{j'}^*} \leq 1 - (1-\varepsilon) = \varepsilon,$$

concluding the proof of the theorem. □

To exemplify the conclusion of the last theorem, consider that we have two colors, say men and women, and that the prior is such that the top candidate is a woman with probability 60% and a man with probability 40%. This translates into having $\epsilon = 1/3$ in the statement of the theorem, which implies that the algorithm will pick a woman at most 33% more often than a man. See Section 2.4 for more examples and empirical validations of these results.

To wrap up the section, observe that for the case of equal probabilities (i.e., $\mathbf{p} = (1/k, \dots, 1/k)$), Corollary 1

and Theorem 2 imply that Algorithm 1 is $1+o(1)$-competitive and $1+o(1)$-fair. Unfortunately, these two properties cannot be simultaneously achieved for a general $\mathbf{p}$. Indeed, consider an instance where $p_1 = 1/\sqrt{k}$ and $p_i = (1-p_1)/(k-1)$ for all $2 \le i \le k$. Let ALG be an α-fair algorithm. Its success probability is

$$\sum_{i=1}^{k} p_i \cdot \mathbf{P}(ALG \text{ selects the best of color } i) \le \sum_{i=1}^{k} \alpha p_i^2 = \frac{\alpha}{k} + \alpha \frac{(1-p_1)^2}{k-1} \le \frac{2\alpha}{k}.$$

On the other hand, the optimal offline algorithm always selects from color 1 and therefore, gets the best candidate with probability $p_1 = 1/\sqrt{k}$. So, the competitive ratio of any α-fair algorithm is not better than $\sqrt{k}/(2\alpha)$. In particular, if we want an algorithm that is β-competitive and γ-fair, then $\max\{\beta, \gamma\} \ge k^{1/4}/2$.

2.3. Sample-Driven Multicolor Secretary Problem

In this section, we formulate a sample-driven version of our multicolor secretary problem inspired by Correa et al. (2021a). This model interpolates between the case where candidates have adversarially chosen values (when $q = 0$) and the case where candidate values are independent draws from a known distribution (as $q \to 1$). In this version of the problem, we have n candidates partitioned into k groups $C = \{C_1, \ldots, C_k\}$ of sizes $\mathbf{n} = (n_1, \ldots, n_k)$. Each of the n candidates is placed in the set S (the samples) independently with a given probability q, and in the set V otherwise. We get to observe all candidates in S before the selection process starts. That is, we get to see the colors of all of these candidates and the relative ranks within their color. Then, the candidates in V arrive one by one in uniform random order, and we can select one of them. For each new candidate, we observe the candidate's color and the candidate's relative rank within the candidates of the same color that we have already observed (including those in S). As before, we are given prior probabilities $(p_1, \ldots, p_k)$, where p_j is the probability that the best candidate of V is of color j. We want to maximize the probability of selecting the best candidate of V. In other words, if we denote by V_j the set of candidates of color j in V, then we want to choose an algorithm ALG that maximizes

$$\sum_{j=1}^{k} p_j \cdot \mathbf{P}(ALG \text{ selects best of } V_j),$$

which we call the success probability of ALG. To avoid some technical issues, when $V_j = \emptyset$, we interpret "ALG selects the best of V_j" as true. Notice that for fixed q, as n_j grows, the probability that $V_j = \emptyset$ decays exponentially fast to zero.

We extend the definition of GroupThresholds in the following way. First, we use independent Uniform[0, 1] arrival times to model both the arrival order *and* the partition into S and V; given arrival times $\tau_1 < \tau_2 < \cdots < \tau_n$, we place all candidates i with arrival times $\tau_i < q$ in S and the rest in V. Our algorithm SD-GroupThresholds($\mathbf{t}$) is parameterized by thresholds $\mathbf{t} = (t_{j,\ell})_{1 \le j \le k, 1 \le \ell}$ in $[q, 1]$. Upon observing a candidate of color j at time τ that is best so far in V_j, the algorithm computes the candidate's relative rank with respect to the set of candidates of color j with arrival time in $[0, \tau]$. If the relative rank is ℓ, the candidate is selected if $t_{j,\ell} \le \tau$.

As for the regular version of the algorithm, we can show that its success probability decreases with n_j, for all j. A proof of this lemma appears in Appendix A.

Lemma 6. (Monotonicity, with Samples). *For fixed q, $(p_1, \ldots, p_k)$, and $\mathbf{t}$, the success probability of the algorithm* SD-GroupThresholds($\mathbf{t}$) *is decreasing with n_j for all $j = 1 \ldots k$.*

Now, similar to Lemma 5, we can calculate the limit success probability of SD-GroupThresholds($\mathbf{t}$). Detailed calculations that lead to the formula in the following theorem can be found in Appendix A.

Theorem 4. (Competitive Ratio, with Samples). *The limit success probability of SD-*GroupThresholds($\mathbf{t}$) *is*

$$\sum_{j=1}^{k} p_j \int_q^1 \sum_{\ell : t_{j,\ell} \le \tau} q^{\ell-1} \cdot \left(1 - \sum_{s \ge \ell : t_{j,s} \le \tau} \left(\frac{q}{\tau}\right)^{s-\ell} \left(\frac{\tau - \max\{q, t_{j,s}\}}{\tau}\right)\right) \cdot \prod_{j' \ne j} \left(1 - \sum_{s : t_{j',s} \le \tau} \left(\frac{q}{\tau}\right)^{s-1} \left(\frac{\tau - \max\{q, t_{j',s}\}}{\tau}\right)\right) d\tau. \tag{3}$$

2.3.1. Computation of the Optimal Thresholds.

For fixed $q < 1$ and given $\varepsilon > 0$, it is possible to approximate the optimal solution numerically by setting $t_{j,\ell} = 1$ for $\ell > \log_q(\varepsilon) = O(1/\varepsilon(1-q))$ and optimizing over the finitely many remaining thresholds. Doing this results in a reduction in the success probability that is not larger than $\sum_{\ell > \log_q(\varepsilon)} (1-q) q^{\ell-1} = O(\varepsilon)$.

2.3.2. Numerical Evaluation.

We provide a numerical evaluation of the competitive ratio and fairness properties of our sample-based algorithm in Figure 2 and Appendix B. Note that for the special case of $k = 1$ and as $q \to 1$, our result recovers the classic result of Gilbert and Mosteller (1966).

2.4. Additional Empirical Evaluation

We conclude this section by returning to the basic setting without samples and providing two additional sets of numerical evaluations that illustrate how our algorithms perform in situations where the prior probabilities are initially incorrect.

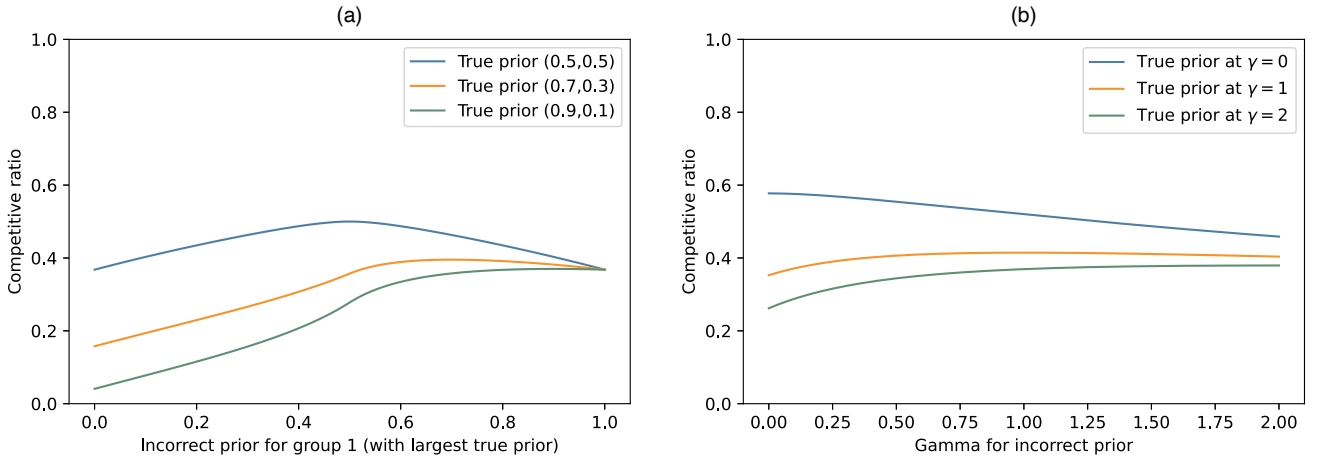
In Figure 4, we examine the effect of incorrect priors $\mathbf{p}' \neq \mathbf{p}$ on the competitive ratio of our fair online algorithms. In Figure 4(a), we look at $k = 2$ groups and three different true priors for the two groups ($(0.5, 0.5)$, $(0.7, 0.3)$, and $(0.9, 0.1)$) as a function of the assumed prior for group 1 (i.e., p_1') ranging from zero to one. The competitive ratio peaks when the assume prior coincides with the actual prior and deteriorates more quickly when the largest true prior is underestimated. In Figure 4(b), we look at three groups where priors follow a power law with varying exponent $\gamma \in [0, 2]$ (i.e., we set (p_1, p_2, p_3) proportional to $(1, 1/2^{\gamma}, 1/3^{\gamma})$). Figure 4(b) shows the effect of running the algorithm with an assumed γ' that is different from the true underlying γ. Again, the competitive ratio peaks when the assumed prior coincides with the actual prior and deteriorates away from it. The deterioration effect is not as pronounced as in the setup of Figure 4(a) because in Figure 4(b), all priors are affected proportionally.

In our second experiment, we investigate how, in the long run, our algorithm might help mitigate the bias generated by incorrectly set priors. For this, we consider a Bayesian model, in which candidates have a true value drawn independently from an exponential distribution with parameters λ_1 and λ_2 for each of the two groups, but these parameters are unknown to the decision maker. The decision maker has a prior belief that we assume is the gamma distribution (conjugate of the exponential). The decision maker only observes ordinal information until a candidate is selected. Only once a candidate is selected is the true value observed (and only of that candidate). Under these assumptions, every time a candidate is selected, an easy update rule can be derived to obtain a posterior belief. In the experiment, we draw all of the true values from an Exponential(1) (i.e., $\lambda_1 = \lambda_2 = 1$). For group 1, we initialize the parameters of the gamma distribution (the belief of the decision maker) so that it matches the truth, with some uncertainty. For group 2, we initialize the belief with worse parameters so that the decision maker believes that the probability of containing the best candidate is significantly higher for group 1, around $2/3$ versus $1/3$. Because the belief for group 1 is consistent with $\lambda_1 = 1$, selecting only from group 1 will not modify the relation between the two groups. In Figure 5, we show the results of 1 run and 100 runs of this experiment using our algorithm, with sequences of 500 iterations. We observe that the priors get closer to 0.5 as the number of iterations grows, so in fact, our algorithm allows the decision maker to learn the true priors.

3. The Multicolor Prophet Problem

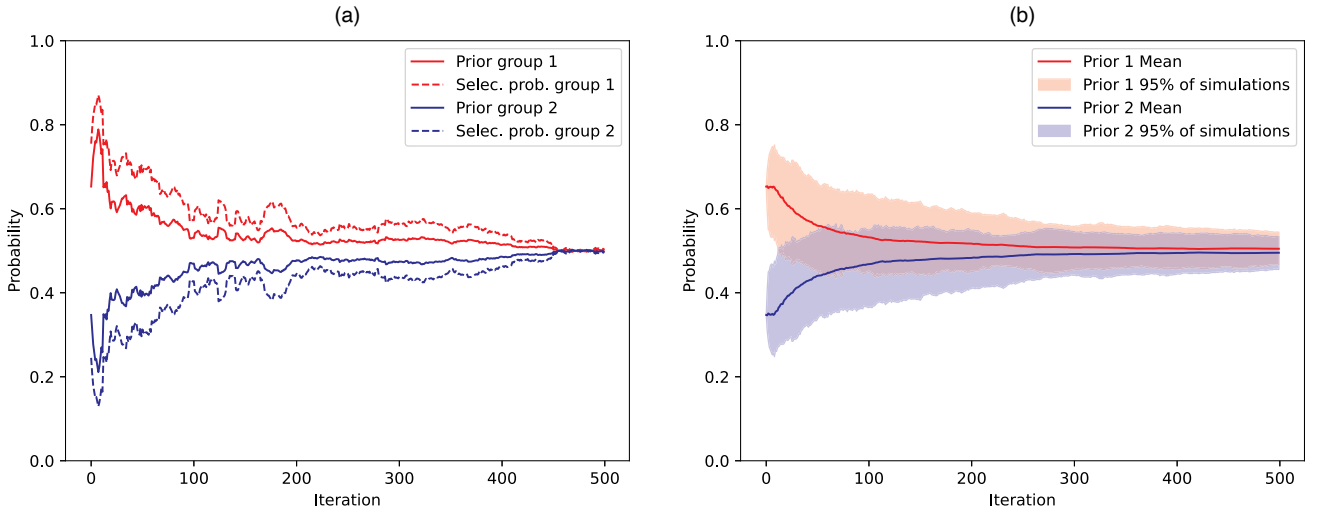
We next consider the following *multicolor prophet problem*. In this model, n candidates arrive one by one, and the arrival order is either a fixed arbitrary permutation or a uniformly random permutation. Candidates are partitioned into k groups $C = \{C_1, \ldots, C_k\}$. We write $\mathbf{n} = (n_1, \ldots, n_k)$ for the vector of group sizes (i.e., $|C_j| = n_j$) for all $1 \leq j \leq k$. We identify each of the groups with a distinct color and let $c(i), v_i$ denote the color and value of candidate i, respectively. The value v_i is revealed upon the arrival of i and is drawn independently from a given distribution F_i. We use $\mathbf{F} = (F_1, \ldots, F_n)$ to refer to the vector of distributions. We are also given a probability vector $\mathbf{p} = (p_1, \ldots, p_k)$. The goal is to select a candidate in an online manner in order to maximize the expectation of the value of the selected candidate while selecting from each color with probability proportional to $\mathbf{p}$. We distinguish between the *basic setting*, in which p_j is the proportion of candidates who belong to group j (i.e., $p_j = n_j/n$), and the *general setting*, in which $\mathbf{p}$ is arbitrary. We compare ourselves with the *fair optimum*, which is the optimal offline algorithm that respects the p_j's.

Figure 4. (Color online) Competitive Ratios for $k = 2$ and 3 When We Use the Optimal Algorithm for an Incorrect Prior



Notes. For $k = 3$, we use the same parameterization as in Figure 1. (a) Competitive ratios for $k = 2$ and incorrect priors. (b) Competitive ratios for $k = 3$ and incorrect priors.

Figure 5. (Color online) Experiment Evaluating How the Priors Evolve as We Select Candidates from Both Groups When We Start with Incorrect Priors



Notes. (a) One run of the experiment. (b) Average of 100 runs of the experiment.

3.1. Key Definitions

3.1.1. Fair Optimum. We define $\textsc{FairOpt}(n, C, \mathbf{F}, \mathbf{p})$ as the optimal offline algorithm that selects a candidate of group j with probability p_j for all j, and we write $\mathbf{E}[\textsc{FairOpt}(n, C, \mathbf{F}, \mathbf{p})]$ for the expected value that it achieves. More precisely, among the class of randomized rules to select a candidate that (a) chooses a candidate from each color j with probability p_j and (b) can observe the realizations of all values, $\textsc{FairOpt}(n, C, \mathbf{F}, \mathbf{p})$ is the one that maximizes the expectation of the value of the selected candidate.

Intuitively, one can think of FairOpt as the limit of the following experiment. We draw m times, with $m \gg 1$, an independent sample of the vector $(v_1, \dots, v_n)$, so we obtain $\{(v_{i,s})_{i=1}^n\}_{s=1}^m$. In each of the vectors, we select a candidate $i^*(s)$ so that $\frac{1}{m}\sum_{s=1}^m v_{i^*(s),s}$ is maximized and $i^*(s)$ belongs to color j in $m \cdot p_j$ of the vectors.

3.1.2. Ex Ante Relaxation. We denote by q_i the probability with which $\textsc{FairOpt}(n, C, \mathbf{F}, \mathbf{p})$ selects candidate i. Using these probabilities, we can obtain the following upper bound on the performance of FairOpt, which is known in the prophets literature as the *ex ante relaxation* (see, for example, Feldman et al. 2016):

$$\textsc{ExAnte}(n, C, \mathbf{F}, \mathbf{p}) = \sum_{i=1}^{n} q_i \cdot \mathbf{E}(v_i \mid v_i \geq F_i^{-1}(1 - q_i)).$$

3.1.3. Fair Selection. We say that an online algorithm ALG is fair if it selects a candidate of each color j with probability proportional to p_j.

Definition 3. (Fair Online Algorithm). We say that an online algorithm ALG is fair if

$$\mathbf{P}(c(\text{ALG}) = j | \text{ALG } \mathbf{stops}) = p_j \quad \forall 1 \leq j \leq k.$$

Note that this is analogous to being 1-fair in Definition 2.

3.1.4. Approximation Ratio. Our goal is to find the fair online algorithm FairAlg with the best-possible approximation ratio with respect to FairOpt. To formally define this, let $\mathbf{E}[\textsc{FairAlg}(n, C, \mathbf{F}, \mathbf{p})]$ denote the expected value achieved by FairAlg.

Definition 4. (Approximation Ratio). We say that online algorithm FairAlg provides an α-approximation if

$$\sup_{n, C, \mathbf{F}, \mathbf{p}} \frac{\mathbf{E}[\textsc{FairOpt}(n, C, \mathbf{F}, \mathbf{p})]}{\mathbf{E}[\textsc{FairAlg}(n, C, \mathbf{F}, \mathbf{p})]} \leq \alpha.$$

Note that the smaller $\alpha \geq 1$, the better. Specifically, if $\alpha = 1$, then the expected value achieved by the fair online algorithm matches that of the fair offline algorithm.

3.2. Optimal Online Algorithms

We develop optimal fair online algorithms with surprisingly small competitive ratio under different assumptions on the setting. In the first setting (Section 3.2.1), we consider an arbitrary fixed order of the candidates, nonidentical distributions, and general probabilities $\mathbf{p}$. In the second setting, we assume that variables are i.i.d. and make the natural additional assumption that the p_j's are proportional to the group sizes (Section 3.2.2). In the third and final setting (Section 3.2.3), we relax the i.i.d. assumption to hold only within groups and assume that candidates arrive in uniform random order.

Our high-level approach is the following. We design online algorithms that accept each candidate i with probability $\alpha \cdot q_i$, where $\mathbf{q} = (q_1, \dots, q_n)$ are the marginal probabilities with which the optimal fair offline algorithm FairOpt accepts candidate $i = 1, \dots, n$. Note that for a fixed choice of α, this uniquely determines

thresholds $\mathbf{t} = (t_1, \dots, t_n)$ that we have to set for candidate $i = 1, \dots, n$. We are still free to choose the parameter α, and we choose it to optimize the worst-case approximation ratio.

Intuitively, choosing a smaller α makes us accept less frequently, but conditional on stopping, we choose higher values. The right trade-off between these two forces and hence, the right choice of α turn out to be different in each of the three settings. We find that in each of the three settings, the optimal approximation ratio is equal to $1/\alpha^*$, where α^* is the optimal choice of α.

Our algorithms and analysis share features of several papers in the prophets inequality literature that obtain competitive algorithms through comparison with the ex ante relaxation (e.g., Alaei 2014, Lee and Singla 2018, Ezra et al. 2020). A novel aspect of our work is that we compare with the fair offline optimum and provide bounds on the worst-case loss from ensuring proportional selection probabilities in an online algorithm.

3.2.1. General Distributions. We start by considering the setting in which candidates arrive in any fixed order, candidate values are drawn from not necessarily identical distributions, and the probabilities $\mathbf{p}$ can be arbitrary.

Our algorithm for this case (Algorithm 2) receives as input the probabilities $q_1, \dots, q_n$ with which FairOpt accepts candidate $1, \dots, n$. It then sets thresholds so that it accepts each of the candidates with probability $q_i/2$.

Note that for $i = 1$, we can achieve this by setting the threshold to $t_1 = F_1^{-1}(1 - q_1/2)$. For $i = 2$, we have to set a slightly lower threshold than $F_2^{-1}(1 - q_2/2)$ because with some probability, namely $q_1/2$, we stop at $i = 1$. Indeed, if we set the threshold to $t_2 = F_2^{-1}(1 - \frac{q_2/2}{1-q_1/2})$, we reach candidate $i = 2$ with probability $1 - q_1/2$, and conditional on reaching it, we accept it with probability $\frac{q_2/2}{1-q_1/2}$; so, we accept it with probability exactly $q_i/2$ as desired. Continuing like this yields the thresholds used in the algorithm.

Algorithm 2 (Fair General Prophet)

Input: Distributions $F_1, \dots, F_n$ and $q_1, \dots, q_n$
Output: $i \in [n]$, index of chosen candidate
$s \leftarrow 0$
for $i \leftarrow 1$ **to** n **do**
  if $v_i \geq F_i^{-1}(1 - \frac{q_i/2}{1-s/2})$ **then**
    return i
  end
  $s \leftarrow s + q_i$
end

We show that this algorithm is fair and that it achieves an optimal approximation guarantee. Fairness follows quite directly from the fairness of FairOpt because our algorithm accepts with the same marginal probabilities just scaled down by $1/2$. For the approximation guarantee, we compare the expected value collected by the online algorithm with the expected value achieved by the ex ante relaxation, which is constrained to use the marginal probabilities $\mathbf{q}$ of FairOpt. Because the latter is only higher than the expected value achieved by FairOpt, it also implies an approximation guarantee with respect to FairOpt. We formalize this discussion in the following theorem. Note that the bound of two in this theorem is incomparable with the well-known factor 2 in the regular prophet inequality (Samuel-Cahn 1984), and indeed, we prove it via a substantially different method.

Theorem 5. *For general settings and general distributions, Algorithm 2 is fair and achieves a 2-approximation to FairOpt. No fair online algorithm can achieve a better approximation ratio.*

Proof. We prove first that for all $i \in [n]$, Algorithm 2 selects candidate i with probability $q_i/2$. In fact, for $i = 1$, the algorithm stops if $v_1 \geq F_1^{-1}(1 - q_1/2)$, which happens with probability $1 - F_1(F_1^{-1}(1 - q_1/2)) = q_1/2$. Inductively, assume that it holds true for candidates $1, \dots, i-1$. Because the values are independent, the probability that the algorithm selects candidate i is the probability that the algorithm does not stop before i times the probability that v_i is above the threshold. By the inductive argument, the former is $1 - \sum_{i'=1}^{i-1} q_{i'}/2$, and from the definition of the threshold, we get that the latter is $(q_i/2)/(1 - \sum_{i'=1}^{i-1} q_{i'}/2)$. Multiplying the two numbers, we get that the algorithm selects candidate i with probability $q_i/2$. This concludes the inductive argument. Now, notice that FairOpt satisfies the fairness constraint, and therefore, the algorithm also satisfies it.

To prove that the algorithm is a 2-approximation with respect to FairOpt, note first that because $\sum_{i'=1}^{n} q_{i'} \leq 1$, we have that $1 - \sum_{i'=1}^{i} q_{i'}/2 \geq 1/2$ for all i, and then, $1 - \frac{q_i/2}{1-\sum_{i'=1}^{i} q_{i'}/2} \geq 1 - q_i$ for all i. With this in hand, we can now compare the expectation of the value of the selected candidate:

$$\begin{aligned}
\mathbf{E}(\mathrm{ALG}) &= \sum_{i=1}^{n} \frac{q_i}{2} \mathbf{E}\left(v_i \mid v_i \geq F_i^{-1}\left(1 - \frac{q_i/2}{1 - \sum_{i'=1}^{i} q_{i'}/2}\right)\right) \\
&\geq \frac{1}{2} \sum_{i=1}^{n} q_i \mathbf{E}(v_i \mid v_i \geq F_i^{-1}(1 - q_i)) \\
&\geq \mathbf{E}(\mathrm{FairOpt}).
\end{aligned}$$

The last step follows from the ex ante relaxation, assuming that q_i is the probability that FairOpt selects candidate i.

To see that there is no fair online algorithm with a better guarantee, it is enough to adapt the tight bound

for the classic prophet inequality. Consider an instance with two candidates, candidates 1 and 2, of colors 1 and 2, respectively. Their values are $v_1 = 1$ with probability 1; for some small $\varepsilon > 0$, $v_2 = 0$ with probability $1-\varepsilon$, and $v_2 = 1/\varepsilon$ with probability ε. We set the p_i's to be the probability that each candidate is the maximum (i.e., $p_1 = 1-\varepsilon$ and $p_2 = \varepsilon$). With this, $\mathbf{E}(\textsc{FairOpt}) = 2-\varepsilon$, but any online algorithm (not necessarily fair) has expected value at most one. □

3.2.2. Identical Distributions. We next consider the i.i.d. setting, where all values v_i are independent samples from a common distribution F. In this case, $p_j = n_j/n$ for all groups j is a natural assumption because this is the probability with which the maximum overall is from group j. Also, note that in this case, the optimal offline algorithm is fair and chooses each element with probability $1/n$. That is, $q_i = 1/n$ for all i.

Algorithm 3 tries to mimic the optimal fair offline algorithm but aims at slightly lower marginal acceptance probabilities of $2/(3n)$. The derivation of the thresholds $\mathbf{t}$ that achieve this follows the same logic as in our algorithm for general distributions.

Algorithm 3 (Fair IID Prophet)
Input: Distributions F
Output: $i \in [n]$, index of chosen candidate
for $i \leftarrow 1$ **to** n **do**
  if $v_i \geq F^{-1}\left(1 - \frac{2/(3n)}{1-2(i-1)/(3n)}\right)$ **then**
    return i
  end
end

We prove that this algorithm is fair and achieves an optimal approximation ratio of 3/2. To show fairness, we again exploit that the algorithm accepts each candidate i with a rescaled version of the marginal probability $q_i = 1/n$ with which FairOpt accepts a candidate. In the previous proof, it was key that the threshold was always larger than $F_i^{-1}(1-q_i)$, which is not the case here. For $i > n/2$, the threshold is, in fact, smaller. To circumvent this, we use a stochastic dominance argument to establish the approximation factor.

Theorem 6. *For basic settings and i.i.d. distributions Algorithm 3 is fair and achieves a 3/2-approximation to FairOpt. No fair online algorithm can achieve a better approximation ratio.*

Proof. We prove first that Algorithm 3 selects each candidate with probability $2/(3n)$. For candidate 1, it is clear that $\mathbf{P}(v_1 \geq F^{-1}(1-1/(3n))) = 2/(3n)$. Inductively, assume that candidates $1,\dots,i-1$ are selected each with probability $2/(3n)$. Because the values are independent, the probability that the algorithm selects i is the probability that it does not stop before i times the probability that v_i surpasses the threshold. The former happens with probability $1-(i-1)\frac{2}{3n}$, and the latter happens, by definition of the threshold, with probability $\frac{2/(3n)}{1-2(i-1)/(3n)}$. If we multiply these two quantities, we obtain that the probability of selecting i is $2/(3n)$. Therefore, the algorithm stops with probability $2/3$, and a candidate of color j is selected with probability exactly $n_j\frac{2}{3n} = p_j\frac{2}{3}$; so, it is fair.

To prove the approximation factor, we prove an approximate stochastic dominance between the algorithm and FairOpt. More specifically, we prove that for all $q \in [0,1]$, it holds that

$$\frac{\mathbf{P}(\text{ALG} \geq F^{-1}(q))}{\mathbf{P}(\textsc{FairOpt} \geq F^{-1}(q))} \geq \frac{2}{3}. \tag{4}$$

If this is true, then $\frac{3}{2}\mathbf{P}(\text{ALG} \geq x) \geq \mathbf{P}(\textsc{FairOpt} \geq x)$ for all $x \geq 0$. Thus, by simply integrating on both sides, we obtain that $\frac{3}{2}\mathbf{E}(\text{ALG}) \geq \mathbf{E}(\textsc{FairOpt})$.

Now, to calculate the left-hand side of Equation (4), note that because the values are i.i.d. and $p_j = n_j/n$ for all colors j, we have $\textsc{FairOpt} = \max_i v_i$. For the case of the algorithm, we can condition on which candidate is selected. The algorithm selects candidate i w.p. $2/(3n)$, and conditional on that, the probability that the value v_i is above $F^{-1}(q)$ is just the ratio between $1-q$ and the probability that v_i is above the corresponding threshold or one if the ratio is larger that one. So, we obtain that the left-hand side of Equation (4) is equal to

$$\frac{\sum_{i=1}^n \frac{2}{3n}\min\left\{1, \frac{1-q}{\frac{2/(3n)}{1-(i-1)2/(3n)}}\right\}}{1-q^n} = \frac{\sum_{i=1}^n \min\left\{\frac{2}{3n}, (1-q)\left(1-\frac{2(i-1)}{3n}\right)\right\}}{1-q^n}. \tag{5}$$

Consider a given $q \in [0,1]$ such that for all $i \in [n]$,

$$\frac{2}{3n} \leq (1-q)\left(1-\frac{2(i-1)}{3n}\right).$$

Then, Equation (5) is equal to $\frac{2}{3(1-q^n)} \geq \frac{2}{3}$.

If, in turn, q is such that for all $i \in [n]$,

$$\frac{2}{3n} \geq (1-q)\left(1-\frac{2(i-1)}{3n}\right),$$

then Equation (5) can be rewritten as

$$\begin{aligned}\frac{(1-q)}{(1-q^n)}\sum_{i=1}^n\left(1-\frac{2(i-1)}{3n}\right) &= \frac{1}{\sum_{\ell=0}^{n-1}q^\ell}\left(n-\frac{2}{3}\cdot\frac{1}{n}\cdot\frac{(n-1)n}{2}\right)\\ &= \frac{1}{\sum_{\ell=0}^{n-1}q^\ell}\left(n-\frac{n-1}{3}\right)\\ &\geq \frac{1}{n}\left(n-\frac{n-1}{3}\right)\\ &\geq \frac{2}{3}.\end{aligned}$$

Finally, consider all other cases for q. Because $(1-q)\left(1-\frac{2(i-1)}{3n}\right)$ is decreasing in i, this means that q is such that for $i=1$, $\frac{2}{3n}$ is the minimum in Equation (5), and for $i=n$, $(1-q)\left(1-\frac{2(i-1)}{3n}\right)$ is the minimum. In

other words,

$$\frac{2}{3n} < (1-q),$$

and

$$\frac{2}{3n} > (1-q)\left(1-\frac{2(n-1)}{3n}\right),$$

which is equivalent to

$$1-\frac{2}{n+2} < q < 1-\frac{2}{3n}.$$

Let us set $q = 1-\frac{x}{n}$, where $x \in (\frac{2}{3}, 2-\frac{4}{n+2})$. Note that the interval is empty if $n=1$, so we assume $n \geq 2$. We can rewrite Equation (5) as follows:

$$\begin{aligned}
&\frac{\sum_{i=1}^{n} \min\left\{\frac{2}{3}, x\left(1-\frac{2}{3}\cdot\frac{i-1}{n}\right)\right\}\cdot\frac{1}{n}}{1-\left(1-\frac{x}{n}\right)^n} \\
&= \frac{\int_0^1 \min\left\{\frac{2}{3}, x\left(1-\frac{2}{3}\cdot\frac{\lfloor tn\rfloor}{n}\right)\right\} dt}{1-\left(1-\frac{x}{n}\right)^n} \\
&\geq \frac{\int_0^1 \min\left\{\frac{2}{3}, x\left(1-\frac{2}{3}t\right)\right\} dt}{1-\left(1-\frac{x}{n}\right)^n} \\
&= \frac{\frac{2}{3}\left(\frac{3}{2}-\frac{1}{x}\right)+\int_{\frac{3}{2}-\frac{1}{x}}^1 x\left(1-\frac{2}{3}t\right) dt}{1-\left(1-\frac{x}{n}\right)^n} \\
&= \frac{1-\frac{2}{3x}+x\left(1-\frac{3}{2}+\frac{1}{x}\right)-\frac{2}{3}x\cdot\frac{1}{2}\left(1-\left(\frac{3}{2}-\frac{1}{x}\right)^2\right)}{1-\left(1-\frac{x}{n}\right)^n} \\
&= \frac{1-\frac{1}{12}x-\frac{1}{3x}}{1-\left(1-\frac{x}{n}\right)^n}.
\end{aligned}$$

This last expression is increasing in n, so we show that for $n=2$ and $x \in (\frac{2}{3}, 2)$, it is at least $2/3$. This would be equivalent to

$$\begin{aligned}
&\frac{1-\frac{1}{12}x-\frac{1}{3x}}{1-(1-\frac{x}{2})^2} \geq \frac{2}{3} \text{ for } x \in \left(\frac{2}{3}, 2\right) \\
\iff \quad & 2x^3-9x^2+12x-4 \geq 0 \text{ for } x \in \left(\frac{2}{3}, 2\right) \\
\iff \quad & (x-2)^2(2x-1) \geq 0 \text{ for } x \in \left(\frac{2}{3}, 2\right).
\end{aligned}$$

The last statement is clearly true, so we conclude that the algorithm is a $3/2$-approximation.

To finish the proof of the theorem, we have to show that no fair online algorithm can achieve an approximation ratio better than $2/3$. In fact, consider the instance where the value of every candidate distributes as

$$v_1 = \begin{cases} \frac{2}{3}\cdot\frac{1}{\varepsilon} & \text{with probability } \frac{\varepsilon}{n} \\ \frac{1}{3} & \text{with probability } 1-\frac{\varepsilon}{n}. \end{cases}$$

On the one hand, in this instance, $\mathbf{E}(\textsc{FairOpt}) = 1 + O(\frac{\varepsilon}{n})$. On the other hand, a fair algorithm must select each candidate with equal probability. Consider a fair algorithm ALG, and let β/n be this probability for some $\beta \in [0,1]$. We have that

$$\mathbf{E}(\text{ALG}) = \frac{1}{3}\mathbf{P}\left(\text{ALG} = \frac{1}{3}\right) + \frac{2}{3\varepsilon}\mathbf{P}\left(\text{ALG} = \frac{2}{3\varepsilon}\right).$$

Because ALG stops with probability β, we have that $\mathbf{P}(\text{ALG} = \frac{1}{3}) \leq \beta$. Now, $\mathbf{P}(\text{ALG} = \frac{2}{3\varepsilon})$ is maximized if the algorithm stops whenever it sees a value of $\frac{2}{3\varepsilon}$. Because a candidate has value $\frac{2}{3\varepsilon}$ with probability $\frac{\varepsilon}{n}$ and the algorithm reaches the ith candidate with probability $1-(i-1)\frac{\beta}{n}$, we obtain that

$$\begin{aligned}
\mathbf{P}\left(\text{ALG} = \frac{2}{3\varepsilon}\right) &\leq \sum_{i=1}^{n} \frac{\varepsilon}{n}\left(1-\beta\frac{i-1}{n}\right) \\
&= \varepsilon\int_0^1 1-\beta t\, dt + O(\varepsilon/n) \\
&= \varepsilon\left(1-\frac{\beta}{2}\right) + O(\varepsilon/n).
\end{aligned}$$

Therefore,

$$\mathbf{E}(\text{ALG}) \leq \frac{1}{3}\beta + \frac{2}{3}\left(1-\frac{\beta}{2}\right) + O(1/n) = \frac{2}{3} + O(1/n).$$

Making n large and ε small, we can conclude that no fair algorithm can be better than a $3/2$-approximation. □

3.2.3. Random Order. Finally, we consider a setting where values are only i.i.d. within each group but may differ across groups. We assume that candidates arrive in uniform random order and that $p_j = n_j/n$ for all colors j.

Algorithm 4 (Fair Random Order Prophet)
Input: Distributions $F_1, \ldots, F_k$
Output: $i^* \in [n]$, index of chosen candidate
for $i \leftarrow 1$ **to** n **do**
  if $v_i \geq F_{c(i)}^{-1}\left(1-\frac{1/(n\sqrt{2})}{1-(i-1)/(n\sqrt{2})}\right)$ **then**
    return $i^* = i$
  end
end

For this case, we show that Algorithm 4, which sets thresholds to achieve an acceptance probability of $1/\sqrt{2}$ that is fair and achieves an optimal approximation ratio of $1/(2-\sqrt{2}) \approx 1.707$. The proof of this result appears in Appendix C.

Theorem 7. *For basic settings with i.i.d. valuations within groups and uniform arrival order, there is a* $1/(2-\sqrt{2})$*-approximation with respect to FairOpt.*

3.3. Sample-Driven Multicolor Prophet Problem

If instead of the distributions of the random variables, we only have sample access to them, we still can obtain approximate versions of our results if we are given the probabilities of stopping with each variable. The

analysis is similar to that of Rubinstein et al. (2020) for the i.i.d. prophet inequality with samples.

For the case of general distributions, if we are given the probabilities $q_1, \ldots, q_n$, we can obtain an approximate version of Algorithm 2. Using $M_i = O(\log(n/\varepsilon)/(\varepsilon^2 q_i))$ samples of F_i, we can replace the threshold $F_i^{-1}(1 - \frac{q_i/2}{1-s/2})$ with the $\lfloor M_i \cdot (1-\varepsilon) \cdot \frac{q_i/2}{1-s/2} \rfloor$th largest sample. Let τ_i be the resulting threshold. A simple Chernoff bound indicates that the probability that $(1 - F_i(\tau_i))$ is in the interval $[(1-\varepsilon) \cdot \frac{q_i/2}{1-s/2}, \frac{q_i/2}{1-s/2}]$ is at least $1 - \varepsilon/n$. This means that with probability at least $1 - \varepsilon$, all of the thresholds are larger than the thresholds of Algorithm 2, but for every variable, the probability that it is larger than its threshold is only a factor $(1 - \varepsilon)$ smaller.

For the case of i.i.d. random variables, $1/q_i = O(n)$, so in total, we need $O(n^2 \log(n/\varepsilon)/\varepsilon^2)$ samples to repeat the same argument. Now, because in Algorithm 3, we only want to approximate quantiles $\frac{2/3n}{1-2(i-1)/3n}$, which range from $2/3n$ to $2/n$, we can reduce the number of samples. In fact, we only need to approximate well the $O(1/\varepsilon)$ elements of the grid $\{2/3n, (1+\varepsilon) \cdot 2/3n, (1+\varepsilon)^2 \cdot 2/3n, \ldots, 2/n\}$. Therefore, $M = O(n \log(1/\varepsilon)/\varepsilon^2)$ samples are enough.

3.4. Additional Empirical Evaluation

We conclude this section with an empirical comparison of our multicolor prophet algorithms (Algorithm 2 (Fair PA) and Algorithm 3 (Fair IID)) against other standard algorithms. We focus on the case where values are distributed i.i.d. and each candidate is a group on its own. We compare with the following baselines.

- SC algorithm (Samuel-Cahn 1984). This algorithm sets a single threshold so that the maximum is above this threshold with probability exactly 1/2. It achieves an optimal 2-approximation for possibly nonidentical independent distributions and arbitrary arrival order.
- EHKS algorithm (Ehsani et al. 2018). This algorithm sets a single threshold so that an individual candidate is accepted with probability $1/n$. It achieves an approximation of $(e+1)/e \approx 1.58$ for possibly nonidentical independent distributions and random arrival order.
- CFHOV algorithm (Correa et al. 2021b). This algorithm sets a sequence of thresholds based on acceptance probabilities that result from solving a differential equation. It achieves an optimal 1.342-approximation for IID distributions.
- DP algorithm (e.g., Chow et al. 1971). This algorithm is the optimal threshold algorithm for the prophet problem, where thresholds are obtained by backward induction. This algorithm is optimal, even when distributions are different and candidates arrive in arbitrary order.

We consider two settings. In the first one, the input stream consists of 50 samples from the uniform distribution in the range [0, 1], and in the second one, the input consists of 1,000 samples from the binomial distribution with 1,000 trials and 1/2 probability of success of a single trial. For better comparability with existing algorithms, in both cases, we assume that each candidate is a group on its own. We run each algorithm 100,000 times.

In Figure 6, we compare the number of times that our algorithms, SC, EHKS, CFHOV, and DP pick from each position of the stream. We observe that the SC, EHKS, and DP baselines pick candidates more from the first half of the stream compared with the second half (by more than a factor of 1.75), whereas CFHOV picks mostly from the second half of the stream (by more than a factor of 4). So, all of these algorithms are unfair. In contrast, our algorithms select the same number of candidates throughout the stream. The average values of the chosen candidate for Algorithm 2 (Fair PA), Algorithm 3 (Fair IID), SC, EHKS, CFHOV, and DP for the uniform distribution are 0.501, 0.661, 0.499, 0.631, 0.752, and 0.964, respectively, whereas for the binomial distribution, they are 298.34, 389.24, 277.63, 363.97, 430.08, and 548.94, respectively.

In conclusion, for both settings, both of our algorithms (Algorithm 2 and Algorithm 3) provide perfect fairness while giving 51.97% and 68.57% (for the uniform case), respectively, and giving 54.35% and 70.91% (for the binomial case), respectively, of the value of the optimal but unfair online algorithm.

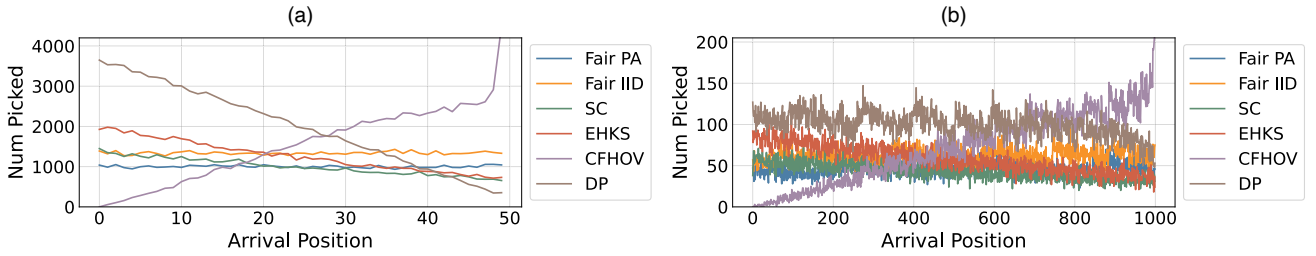
It should be noted that the fact that DP picks mostly from the first half is not a general phenomenon. It heavily depends of the distribution from which the input is drawn. On the contrary, SC and EHKS always pick more often from the early parts of the stream because their conditional stopping probability is constant.

4. Conclusion and Open Problems

In this work, we explored questions of fairness and bias in natural multicolor variants of the two canonical problems of online selection: the secretary problem and the prophet problem. We designed optimal fair online algorithms for these problems and provided a comprehensive empirical evaluation of these algorithms and their properties.

Specifically, we explored a multicolor variant of the secretary problem, where we assumed that comparisons within a group are accurate, whereas comparisons across groups are impossible. This assumption was motivated by situations where the evaluator makes biased evaluations of the candidates' abilities that are consistent within the groups but not across groups or where candidates take different tests (say at different institutions). In the psychology literature, this kind of bias has been referred to as a lack of *measurement equivalence* (Drasgow 1984). A mathematical model complying with this notion of bias arises naturally when for each candidate i, there is a ground truth X_i that is not directly

Figure 6. (Color online) In This Plot, We Present the Number of Times That Our Algorithms (Fair PA and Fair IID) and the Baselines (SC, EHKS, CFHOV, and DP) Pick from Each Position of the Input Prophet Problem Stream



Notes. In panel (a), the stream consists of 50 samples from the uniform distribution, and in panel (b), the stream consist of 1,000 samples from the binomial distribution. (a) Uniform distribution. (b) Binomial distribution.

observable. Through a test or an interview, we observe $f_c(X_i)$ if candidate i belongs to group c, where f_c is an unknown monotone function. Our work can be seen as capturing the case where the choice of these functions is adversarial, and hence, no comparisons across groups can be made. It would be interesting to relax this assumption and explore under which assumptions better approximation guarantees are possible.

Another crucial assumption in our multicolor secretary model was that we assumed that we are given probabilities, with which the best candidate of a color is the best candidate overall. Our algorithms were given these probabilities, but the actual coin flips were only realized after a candidate has been selected. This is in line with standard models of discrimination in labor markets (Becker 1971). This literature has found that although signals about candidates who are available in the hiring process often exhibit an implicit or even explicit bias (e.g., SAT scores), candidates have "true" unobservable qualities, which are only measurable via long-term outcomes. We analyzed numerically what happens with our algorithms if they are fed incorrect priors. It would, of course, also be interesting to explore the design of algorithms that optimally trade off approximation guarantees and robustness to inaccurate priors.

We also studied a multicolor variant of the prophet inequality problem and assumed that both online and offline solutions have to obey certain "quotas" for each color. Imposing quotas is one way to take *affirmative action* (Sowell 2004) but not the only one. It would be interesting to propose and study models that capture other more proactive forms of affirmative action, such as targeted outreach and recruiting or alternate hiring and admission practices.

Finally, there is ample opportunity to extend our work to combinatorial settings. We expect that building on the respective lines of work in the secretary, prophet, and optimal stopping literature in general could prove very fruitful. Particularly exciting directions include an extension to matching problems (Kesselheim et al. 2013, Gravin and Wang 2019, Ezra et al. 2020), allocation problems with matroid structure (Kleinberg and Weinberg 2012, Feldman et al. 2015b, Babaioff et al. 2018, Dütting et al. 2020b), or even general combinatorial allocation problems (Feldman et al. 2015a, Dütting et al. 2020a).

Acknowledgments

The authors are grateful to the review team whose insightful comments and detailed feedback greatly improved the presentation of the paper. This work was done while Andrés Cristi was an intern at Google Research in Zurich, Switzerland; a PhD student at Universidad de Chile; and a postdoctoral student in the Center for Mathematical Modeling at Universidad de Chile and at the Simons Institute for the Theory of Computing. An extended abstract of this work appeared in the Proceedings of the Thirty-Eighth International Conference on Machine Learning 2021.

Appendix A. Proofs Omitted from Section 2.3

Proof of Lemma 6. We take an instance with sizes $\mathbf{n} = (n_1, \dots, n_k)$ and increase it in one candidate. W.l.o.g. assume that the new candidate is of color 1, so we get an instance $\mathbf{n}' = (n_1 + 1, n_2, \dots, n_k)$. We couple the decisions of the algorithm in both instances by first drawing the arrival times of the best n_j candidates of each color j and separately drawing the arrival time of the $(n_1 + 1)$th candidate of color 1. We argue that in a realization of the arrival times where the algorithm fails with $\mathbf{n}$, it also fails with $\mathbf{n}'$. In fact, for such a realization, in the smaller instance $V_j \neq \emptyset$ for all j and for some j, the algorithm selects a candidate who is not the best of V_j or never stops. Adding the $(n_1 + 1)$th candidate of color 1 does not affect the relative ranks of the rest, so the only different action that the algorithm could do in the larger instance would be to select the $(n_1 + 1)$th candidate of color 1. But, V_1 was nonempty before adding this candidate, so she cannot be the best of V_1. Therefore, the algorithm also fails in the larger instance. □

Proof of Theorem 4. Within this proof, we refer to SD-GroupThresholds($\mathbf{t}$) simply as *ALG*. For a given color j, we want to calculate $\mathbf{P}(ALG \text{ selects best of } V_j)$. We say a candidate is acceptable if the algorithm would accept her if it had not stopped when she arrived. For a given time $\tau \in [q, 1]$, we define $N_{j,\tau}$ as the event that no candidate of color j who arrives in $[q, \tau)$ is acceptable. Note that $N_{j,\tau}$ only depends on the arrival times of candidates of color j, so these events are independent across colors. Now, in the limit, V_j is nonempty, so

we can condition on the arrival time of the best in V_j. Thus,

$$
\begin{aligned}
&\mathbf{P}(ALG \text{ selects best of } V_j)\\
&= \frac{1}{1-q}\int_q^1 \mathbf{P}(ALG \text{ selects best of } V_j \mid \text{best of } V_j \text{ arrives at } \tau)\, d\tau\\
&= \frac{1}{1-q}\int_q^1 \mathbf{P}(\cap_{j'=1}^{k} N_{j',\tau} \text{ and best of } V_j \text{ is acceptable} \mid \text{best of } V_j\\
&\qquad \text{arrives at } \tau) d\tau\\
&= \frac{1}{1-q}\int_q^1 \mathbf{P}(N_{j,\tau} \text{ and best of } V_j \text{ is acceptable} \mid \text{best of } V_j\\
&\qquad \text{arrives at } \tau)\cdot \prod_{j'\neq j}\mathbf{P}(N_{j',\tau})\, d\tau\\
&= \int_q^1 \sum_{\ell=1}^{\infty} q^{\ell-1}\cdot \mathbf{P}(N_{j,\tau} \text{ and best of } V_j \text{ is acceptable} \mid \text{best of}\\
&\qquad V_j \text{ arrives at } \tau \text{ and has rank } \ell)\cdot \prod_{j'\neq j}\mathbf{P}(N_{j',\tau})\, d\tau\\
&= \int_q^1 \sum_{\ell=1}^{\infty} q^{\ell-1}\cdot \mathbb{1}_{t_{j,\ell}\le\tau}\cdot \mathbf{P}(N_{j,\tau} \mid \text{best of } V_j \text{ arrives at } \tau\\
&\text{and has rank } \ell)\cdot \prod_{j'\neq j}\mathbf{P}(N_{j',\tau})\, d\tau\\
&= \int_q^1 \sum_{\ell=1}^{\infty} q^{\ell-1}\cdot \mathbb{1}_{t_{j,\ell}\le\tau}\cdot \left(1-\sum_{s=\ell}^{\infty}\mathbb{1}_{t_{j,s}\le\tau}\left(\frac{q}{\tau}\right)^{s-\ell}\left(\frac{\tau-\max\{q,t_{j,s}\}}{\tau}\right)\right)\\
&\cdot \prod_{j'\neq j}\left(1-\sum_{s=1}^{\infty}\mathbb{1}_{t_{j',s}\le\tau}\left(\frac{q}{\tau}\right)^{s-1}\left(\frac{\tau-\max\{q,t_{j',s}\}}{\tau}\right)\right) d\tau.
\end{aligned}
$$

Moving the indicator functions to the conditions on the indices of the sums, we obtain the formula of the lemma. □

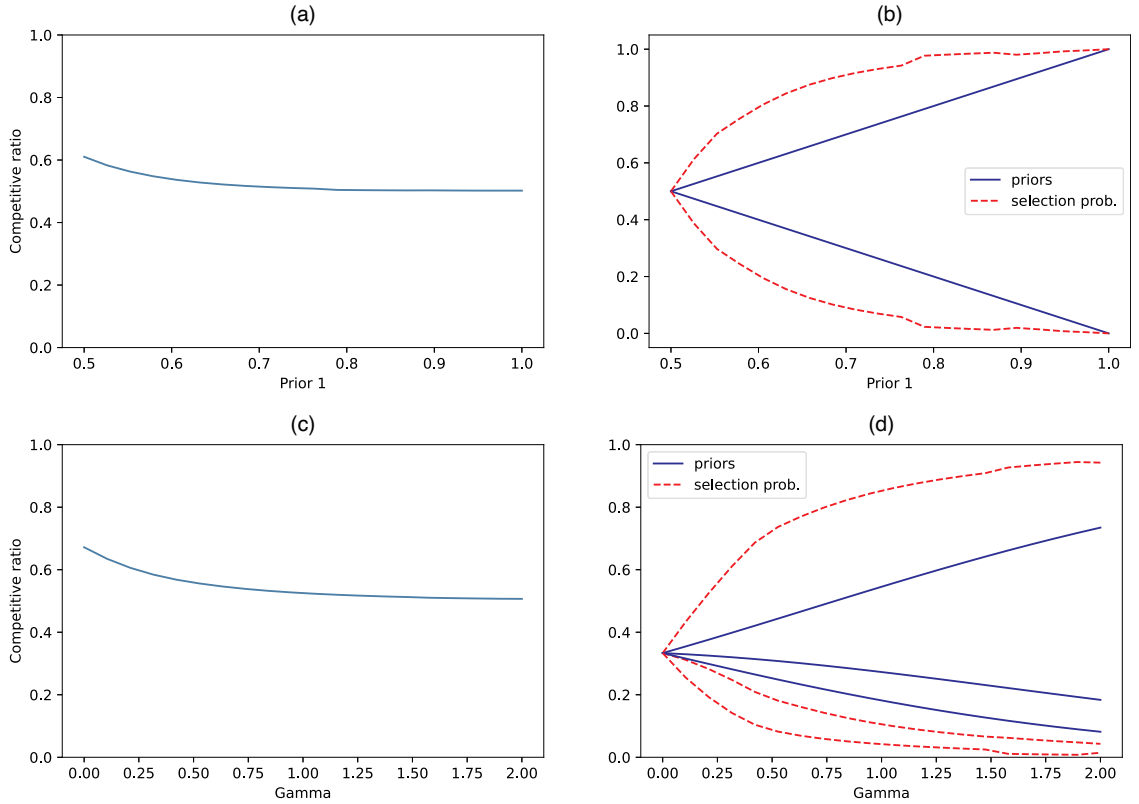
Appendix B. Additional Numerical Evaluation of SD-GroupThresholds

As we did in Figure 1 for the base model, we examine here the competitive ratio and fairness properties of our sample-based algorithm for the multicolor secretary problem. Unlike in the base model, we do not have an explicit formula for these values. However, we can efficiently compute them using the procedure described in Section 2.3. In Figure B.1, we consider the case for $k = 2$ and $k = 3$ groups with sampling rate $q = 1/2$. For Figure B.1(a), we look at $k = 2$ groups and vary the prior p_1 ($p_2 = 1 - p_1$) linearly in $[1/2, 1]$. In Figure B.1(b), we look at $k = 3$ groups in the power law setup described above, where we vary $\gamma \in [0, 2]$. We observe that the competitive ratio is declining in p_1 and γ. The algorithm achieves perfect fairness with balanced priors. As we move away from balanced priors, the selection probabilities first start to diverge from the priors, but then, they return back to them.

Appendix C. Proofs Omitted from Section 3.2.3

Proof of Theorem 7. Denote $\alpha = 1/\sqrt{2}$ and $r_i = \frac{\alpha/n}{1-\alpha(i-1)/n}$ for

Figure B.1. (Color online) For $k = 2$, $k = 3$, and Varying Priors, We Show the Competitive Ratio of Our Algorithm for the Sample-Driven Version of the Multicolor Secretary Problem with $q = 0.5$



Notes. These were obtained by numerically optimizing Equation (3). We also show the probabilities that it selects a candidate from each color conditional on stopping. For $k = 2$, we vary p_1 linearly in $[1/2, 1]$ (and $p_2 = 1 - p_1$). For $k = 3$, we set priors following a power law with varying exponent $\gamma \in [0, 2]$ (i.e., we set (p_1, p_2, p_3) proportional to $(1, 1/2^{\gamma}, 1/3^{\gamma})$). (a) Competitive ratio for $k = 2$. (b) Conditional selection probabilities for $k = 2$. (c) Competitive ratio for $k = 3$. (d) Conditional selection probabilities for $k = 3$.

$i \in [n]$. Note that Algorithm 4 stops with probability α and accepts a candidate i of color j if $v_i \geq F_j^{-1}(1-r_i)$. We have that

$$\begin{aligned}
\mathbf{E}(\mathrm{ALG}) &= \sum_{i=1}^{n} \mathbf{E}(v_i \mid ALG \text{ selects } i) \cdot \mathbf{P}(ALG \text{ selects } i) \\
&= \sum_{i=1}^{n} \frac{\alpha}{n} \cdot \mathbf{E}(v_i \mid ALG \text{ selects } i) \\
&= \sum_{i=1}^{n} \frac{\alpha}{n} \sum_{j=1}^{k} \frac{n_j}{n} \cdot \mathbf{E}(v_i \mid ALG \text{ selects } i \text{ and } c(i)=j) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{\alpha n_j}{n^2} \cdot \mathbf{E}(v_i \mid v_i \geq F_j^{-1}(1-r_i) \text{ and } c(i)=j) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{\alpha n_j}{n^2} \cdot \mathbf{E}(v_1 \mid v_1 \geq F_j^{-1}(1-r_i) \text{ and } c(1)=j).
\end{aligned}$$

Note that FairOpt selects each candidate with probability $1/n$, so we have that by the ex ante relaxation,

$$\mathbf{E}(\textsc{FairOpt}) \leq \sum_{j=1}^{k} \frac{n_j}{n} \cdot \mathbf{E}\left(v_1 \middle| v_1 \geq F_j^{-1}\left(1-\frac{1}{n}\right), \text{ and } c(1)=j \right).$$

Now, in order to compare $\mathbf{E}(\mathrm{ALG})$ and $\mathbf{E}(\textsc{FairOpt})$, note that

$$\begin{aligned}
&\mathbf{E}\left(v_1 \mid v_1 \geq F_j^{-1}(1-r_i), \text{ and } c(1)=j \right) \\
&\quad \geq \min\left\{1, \frac{1}{nr_i}\right\} \cdot \mathbf{E}\left(v_1 \middle| v_1 \geq F_j^{-1}\left(1-\frac{1}{n}\right), \text{ and } c(1)=j \right).
\end{aligned}$$

Therefore, we obtain that

$$\mathbf{E}(\mathrm{ALG}) \geq \mathbf{E}(\textsc{FairOpt}) \sum_{i=1}^{n} \frac{\alpha}{n} \min\left\{1, \frac{1}{nr_i}\right\}.$$

Using the definition of r_i, we get the following expression.

$$\begin{aligned}
\sum_{i=1}^{n} \frac{\alpha}{n} \min\left\{1, \frac{1}{nr_i}\right\} &= \sum_{i=1}^{n} \min\left\{\alpha, 1-\alpha\frac{i-1}{n}\right\} \frac{1}{n} \\
&\geq \int_0^1 \min\{\alpha, 1-\alpha t\}\, dt \\
&= \int_0^{\frac{1}{\alpha}-1} \alpha\, dt + \int_{\frac{1}{\alpha}-1}^{1} 1-\alpha t\, dt \\
&= 1-\alpha+2-\frac{1}{\alpha}-\frac{\alpha}{2}\left(1-\left(\frac{1}{\alpha}-1\right)^2\right) \\
&= 2-\alpha-\frac{1}{2\alpha}.
\end{aligned}$$

This last expression is maximized when $\alpha = 1/\sqrt{2}$, and its value is $2-\sqrt{2}$.

The tight instance is where all candidates have a different color and all values are zero with probability 1, except for one, which is zero with probability $1-1/n$ and n with probability $1/n$. In this instance, $\mathbf{E}(\textsc{FairOpt}) = 1$. If we repeat the previous analysis, all inequalities are equality except for the last one, where we only lose an additive $O(1/n)$ term. □

References

Alaei S (2014) Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM J. Comput.* 43(2):930–972.

Alpern S, Baston V (2017) The secretary problem with a selection committee: Do conformist committees hire better secretaries? *Management Sci.* 63(4):1184–1197.

Arsenis M, Kleinberg R (2022) Individual fairness in prophet inequalities. *Proc. 23rd ACM Conf. Econom. Comput. (EC)* (ACM, New York), 245.

Azar P, Kleinberg R, Weinberg SM (2014) Prophet inequalities with limited information. *Proc. ACM-SIAM Sympos. Discrete Algorithms (SODA)* (SIAM, Philadelphia).

Babaioff M, Immorlica N, Kempe D, Kleinberg R (2018) Matroid secretary problems. *J. ACM* 65(6):35.

Becker GS (1971) *The Economics of Discrimination* (University of Chicago Press, Chicago).

Bhattacharjya D, Deleris LA (2014) The value of information in some variations of the stopping problem. *Decision Anal.* 11(3):189–203.

Buchbinder N, Jain K, Singh M (2014) Secretary problems via linear programming. *Math. Oper. Res.* 39(1):190–206.

Chow YS, Robbins HE, Siegmund D (1971) *Great Expectations: The Theory of Optimal Stopping* (Houghton Mifflin, Boston).

Correa J, Dütting P, Fischer F, Schewior K (2022) Prophet inequalities for independent and identically distributed random variables from an unknown distribution. *Math. Oper. Res.* 47(2): 1287–1309.

Correa J, Cristi A, Feuilloley L, Oosterwijk T, Tsigonias-Dimitriadis A (2021a) The secretary problem with independent sampling. *Proc. ACM-SIAM Sympos. Discrete Algorithms (SODA)* (SIAM, Philadelphia).

Correa J, Foncea P, Hoeksma R, Oosterwijk T, Vredeveld T (2021b) Posted price mechanisms and optimal threshold strategies for random arrivals. *Math. Oper. Res.* 46(4):1452–1478.

Drasgow F (1984) Scrutinizing psychological tests: Measurement equivalence and equivalent relations with external variables are the central issues. *Psych. Bull.* 95(1):134–135.

Dütting P, Kesselheim T, Lucier B (2020a) An $O(\log\log m)$ prophet inequality for subadditive combinatorial auctions. *Proc. IEEE Sympos. Foundations Comput. Sci. (FOCS)* (SIAM, Philadelphia), 306–317.

Dütting P, Feldman M, Kesselheim T, Lucier B (2020b) Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM J. Comput.* 49(3):540–582.

Dütting P, Lattanzi S, Paes Leme R, Vassilvitskii S (2021) Secretaries with advice. *Proc. 22nd ACM Conf. Econom. Comput. (EC)* (ACM, New York), 409–429.

Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. *Proc. Theory Cryptography Conf. (TCC)* (Springer, Berlin, Heidelberg), 265–284.

Ehsani S, Hajiaghayi M, Kesselheim T, Singla S (2018) Prophet secretary for combinatorial auctions and matroids. *Proc. ACM-SIAM Sympos. Discrete Algorithms (SODA)* (SIAM, Philadelphia), 700–714.

Ezra T, Feldman M, Gravin N, Tang ZG (2020) Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. *Proc. ACM Conf. Econom. Comput. (EC)* (ACM, New York), 769–787.

Feldman M, Tennenholtz M (2012) Interviewing secretaries in parallel. *Proc. ACM Conf. Electronic Commerce (EC)* (ACM, New York), 550–567.

Feldman M, Gravin N, Lucier B (2015a) Combinatorial auctions via posted prices. *Proc. ACM-SIAM Sympos. Discrete Algorithms (SODA)* (SIAM, Philadelphia), 123–135.

Feldman M, Svensson O, Zenklusen R (2015b) A simple $O(\log\log(\mathrm{rank}))$-competitive algorithm for the matroid secretary problem. *Proc. ACM-SIAM Sympos. Discrete Algorithms (SODA)* (SIAM, Philadelphia), 1189–1201.

Feldman M, Svensson O, Zenklusen R (2016) Online contention resolution schemes. *Proc. ACM-SIAM Sympos. Discrete Algorithms (SODA)* (SIAM, Philadelphia), 1014–1033.

Georgiou N, Kuchta M, Morayne M, Niemiec J (2008) On a universal best choice algorithm for partially ordered sets. *Random Structures Algorithms* 32(3):263–273.

Gilbert JP, Mosteller F (1966) Recognizing the maximum of a sequence. *J. Amer. Statist. Assoc.* 61(313):35–73.

Gravin N, Wang H (2019) Prophet inequality for bipartite matching: Merits of being simple and non adaptive. *Proc. ACM Conf. Econom. Comput. (EC)* (ACM, New York), 93–109.

Joseph M, Kearns M, Morgenstern JH, Roth A (2016) Fairness in learning: Classic and contextual bandits. *Proc. Conf. Neural Inform. Processing Systems (NIPS)* (Curran Associates Inc., Red Hook, NY), 325–333.

Kaplan H, Naori D, Raz D (2020) Competitive analysis with a sample and the secretary problem. *Proc. ACM-SIAM Sympos. Discrete Algorithms (SODA)* (SIAM, Philadelphia), 2082–2095.

Kearns M, Roth A (2019) *The Ethical Algorithm: The Science of Socially Aware Algorithm Design* (Oxford University Press, Oxford, UK).

Kesselheim T, Radke K, Tönnis A, Vöcking B (2013) An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. *Proc. Eur. Sympos. Algorithms (ESA)* (Springer, Berlin, Heidelberg), 589–600.

Khatibi A, Jacobson S (2018) Generalized sequential stochastic assignment problem. *Stochastic Systems* 8(4):293–306.

Kleinberg R, Weinberg SM (2012) Matroid prophet inequalities. *Proc. ACM Sympos. Theory Comput. Conf. (STOC)* (ACM, New York), 123–136.

Kumar R, Lattanzi S, Vassilvitskii S, Vattani A (2011) Hiring a secretary from a poset. *Proc. ACM Conf. Electronic Commerce (EC)* (ACM, New York), 39–48.

Lee E, Singla S (2018) Optimal online contention resolution schemes via ex-ante prophet inequalities. *Proc. Eur. Sympos. Algorithms (ESA)*, vol. 57 (Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Wadern, Germany), 1–14.

Lien RW, Iravani SMR, Smilowitz KR (2014) Sequential resource allocation for nonprofit operations. *Oper. Res.* 62(2):301–317.

Preater J (1999) The best-choice problem for partially ordered objects. *Oper. Res. Lett.* 25(4):187–190.

Rubinstein A, Wang JZ, Weinberg SM (2020) Optimal single-choice prophet inequalities from samples. *Proc. Innovations Theoret. Comput. Sci. (ITCS)* (Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Wadern, Germany), 60:1–60:10.

Salem J, Gupta S (2024) Secretary problems with biased evaluations using partial ordinal information. *Management Sci.* 70(8): 5337–5366.

Samuel-Cahn E (1984) Comparison of threshold stop rules and maximum for independent nonnegative random variables. *Ann. Probab.* 12(4):1213–1216.

Sowell T (2004) *Affirmative Action Around the World: An Empirical Study* (Yale University Press, New Haven, CT).

José Correa is a professor in the Department of Industrial Engineering at Universidad de Chile. His research interests are in game theory and mechanism design, mostly from an algorithmic viewpoint. Recently, he has worked on online allocation problems when users are forward looking.

Andrés Cristi is an assistant professor in the College of Management of Technology at the Ecole Polytechnique Fédérale de Lausanne. His work studies how incentives shape the behavior of agents participating in a market and how the underlying combinatorial structure affects the efficiency of the potential outcomes.

Paul Dütting is a research scientist at Google Research Zurich. His research interests lie at the intersection of economics and computation, and in recent years, he has focused on the study of pricing in online platforms as well as the efficient design of contracts.

Ashkan Norouzi-Fard is a research scientist at Google Research Zurich. He has contributed to theoretical computer science, particularly in clustering algorithms, submodular maximization, and dynamic facility location problems. Recently, he has worked on streaming algorithms, robust optimization, and correlation clustering.