

The Secretary Problem with Predictions and a Chosen Order

Helia Karisani ✉ 

University of Massachusetts Amherst, MA, USA

Mohammadreza Daneshvaramoli ✉ 

University of Massachusetts Amherst, MA, USA

Hedyeh Beyhaghi ✉ 

University of Massachusetts Amherst, MA, USA

Mohammad Hajiesmaili ✉ 

University of Massachusetts Amherst, MA, USA

Cameron Musco ✉ 

University of Massachusetts Amherst, MA, USA

Abstract

We study a learning-augmented variant of the secretary problem, recently introduced by Fujii and Yoshida (2023). In this variant, the decision-maker has access to machine-learned predictions of candidate values in advance. The key challenge is to balance *consistency* and *robustness*: when the predictions are accurate, the algorithm should hire a near-best secretary; however, if they are inaccurate, the algorithm should still achieve a bounded competitive ratio.

We consider both the standard *Random Order Secretary Problem (ROSP)*, where candidates arrive in a uniform random order, and a more natural model in the learning-augmented setting, where the decision-maker can choose the arrival order based on the predicted candidate values. This model, which we call the *Chosen Order Secretary Problem (COSP)*, can capture scenarios such as an interview schedule that is set by the decision-maker.

We propose a novel algorithm that applies to both ROSP and COSP. Building on the approach of Fujii and Yoshida, our method switches from fully trusting predictions to a threshold-based rule when a large deviation of a prediction is observed. Importantly, unlike the algorithm of Fujii and Yoshida, our algorithm uses randomization as part of its decision logic. We show that if $\epsilon \in [0, 1]$ denotes the maximum multiplicative prediction error, then for ROSP our algorithm achieves competitive ratio $\max\{0.221, \frac{1-\epsilon}{1+\epsilon}\}$, improving on a previous bound of $\max\{0.215, \frac{1-\epsilon}{1+\epsilon}\}$ due to Fujii and Yoshida [36]. For COSP, our algorithm achieves $\max\{0.262, \frac{1-\epsilon}{1+\epsilon}\}$. This surpasses a 0.25 upper bound on the worst-case competitive ratio that applies to the approach of Fujii and Yoshida, and gets closer to the classical secretary benchmark of $1/e \approx 0.368$, which is an upper bound for any algorithm. Our result for COSP highlights the benefit of integrating predictions with arrival-order control in online decision-making.

2012 ACM Subject Classification Theory of computation → Online algorithms; Theory of computation → Design and analysis of algorithms

Keywords and phrases Secretary problem, learning-augmented algorithms, online algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.86

Related Version *Full Version*: <https://arxiv.org/abs/2601.07482>

Funding This research is supported by National Science Foundation grants 2045641, 2325956, 2512128, and 2533814.



© Helia Karisani, Mohammadreza Daneshvaramoli, Hedyeh Beyhaghi, Mohammad Hajiesmaili, and Cameron Musco;

licensed under Creative Commons License CC-BY 4.0

17th Innovations in Theoretical Computer Science Conference (ITCS 2026).

Editor: Shubhangi Saraf; Article No. 86; pp. 86:1–86:24



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The secretary problem, popularized by Gardner in his 1960 *Scientific American* column [37], is one of the most well-studied problems in online decision-making. In the classic setting, candidates arrive in a uniform random order. When a candidate arrives, the decision-maker observes a real-valued score for the candidate, and must make an immediate, irrevocable decision to either hire or reject the candidate. The classic objective is to maximize the probability of selecting the best candidate from the sequence. Alternatively, one can aim to maximize the *expected value* of the selected candidate – this is the variant we will focus on. The secretary problem and its many variants have been extensively studied [47, 3, 10, 18, 22], and applied in domains such as resource allocation [10] and mechanism design [49].

Despite this work, the underlying assumption of the secretary problem – that the decision-maker has no information about the values of candidates before they arrive – is often unrealistic. In practice, decision-makers often have estimates of candidate values before interviewing them. For example, in applications to hiring, these predictions may be made based on resumes, portfolios, or public reputation. In fact, machine learning methods that leverage such information to make interview and hiring decisions are widespread in industry [7, 71, 64, 65, 24, 59, 58, 68, 70, 59].

Given this, several recent works have studied variants of the Secretary Problem where the decision-maker has access to predictions, including estimates of the maximum value among the candidates [6] or probabilistic forecasts of whether stronger candidates will appear later [15]. In this work, we focus on a variant introduced by Fujii and Yoshida [36], where the decision-maker has prior access to a prediction \hat{v}_i of the value v_i of each candidate $i \in [n] := \{1, 2, \dots, n\}$. They design an algorithm in this model that waits to select the candidate with the highest predicted value, unless any observed value deviates too much from its prediction; in that case, the algorithm switches to Dynkin’s classical worst-case $1/e$ -competitive strategy [32].

Let the *competitive ratio* be defined as the worst-case ratio (over all possible assignments of candidate values) of the expected value of the candidate selected by an algorithm to the value of the true best candidate. Fujii and Yoshida [36] prove that their algorithm achieves a competitive ratio of $\max\left\{0.215, \frac{1-\epsilon}{1+\epsilon}\right\}$, where $\epsilon = \max_i \left|1 - \frac{\hat{v}_i}{v_i}\right|$ denotes the maximum multiplicative prediction error across all candidates. Their result demonstrates that if predictions are accurate (i.e., $\epsilon \rightarrow 0$), the algorithm approaches optimal performance (i.e., the algorithm is *consistent*); otherwise, it guarantees a worst-case constant-factor approximation (i.e., the algorithm is *robust*).

1.1 Our Contributions

We build on the approach of Fujii and Yoshida, making the following main contributions:

Chosen Order Secretary Problem (COSP) with Predictions. Fujii and Yoshida restrict their attention to the classic setting in which candidates arrive in a uniform random order. We call this setting the *Random Order Secretary Problem (ROSP)*. However, when predictions of candidate values are available up front, it is natural to use them to modify the arrival order of the candidates. For example, in applications to hiring, the interview order is typically set by the decision-maker and may be chosen based on the available predictions. To capture this setting, we introduce the *Chosen Order Secretary Problem (COSP)*, in which the decision-maker has full control of the candidate arrival sequence. COSP is analogous

to existing chosen order settings for prophet inequalities, in which a distribution over each candidate's value is known up front [43, 21, 62]. However, like the classic secretary problem, COSP is fundamentally worst-case, with no stochastic assumptions made on the inputs.

Improved Bounds for COSP. We design a simple algorithm for COSP inspired by the approach of Fujii and Yoshida [36]. Intuitively, one might imagine that interviewing candidates in e.g., decreasing order of their predicted values, would be optimal. However, it is easy to find simple worst-case instances that defeat these related strategies. Instead, our algorithm only uses the power of COSP in a limited way: letting $\hat{i} \in [n]$ be the candidate with the highest predicted value, we place \hat{i} at a fixed position in the arrival sequence, and randomize the order of all other candidates. We optimize the position of \hat{i} to achieve maximum competitive ratio.

After fixing the interview order, our algorithm proceeds similarly to that of Fujii and Yoshi. It initially operates in a prediction-trusting mode, waiting to hire \hat{i} , the candidate with the highest predicted value. If it encounters a candidate whose true value significantly deviates from its prediction (i.e., exceeds a specified error threshold), it transitions to Dynkin's classical threshold-based algorithm [32]. This algorithm ignores a fixed fraction of candidates, and then hires the first candidate (if any) it sees that is better than all the ignored candidates. Critically, we modify the approach of Fujii and Yoshida by not always hiring \hat{i} if it is meant to be hired in the Dynkin mode of the algorithm. We introduce two parameters, γ and δ , that determine the probability of hiring \hat{i} in different cases, and by tuning these parameters, are able to achieve a larger competitive ratio.

We show that our algorithm is robust and consistent, achieving a competitive ratio of at least

$$\max \left\{ 0.262, \frac{1 - \epsilon}{1 + \epsilon} \right\},$$

recalling that ϵ is the largest multiplicative prediction error. As in the algorithm of Fujii and Yoshida, our method does not require ϵ as input. When $\epsilon \rightarrow 0$, the competitive ratio approaches 1, and when ϵ is large, it remains bounded below by 0.262. Importantly, our COSP result surpasses the 0.25 upper bound that arises in the analysis of the Fujii–Yoshida algorithm in the random-order model.¹

Improved Bounds for ROSP. We show that our approach also applies to the classic ROSP setting, where instead of optimizing the arrival time of the best predicted candidate \hat{i} , we simply average over a uniform random choice of this time. In this case, by optimizing the parameters γ and δ , we can show that our algorithm achieves a competitive ratio of at least

$$\max \left\{ 0.221, \frac{1 - \epsilon}{1 + \epsilon} \right\},$$

improving on the previous bound of $\max \left\{ 0.215, \frac{1 - \epsilon}{1 + \epsilon} \right\}$ of Fujii and Yoshida [36]. As in the COSP setting, the algorithm does not require ϵ as input. Our worst-case competitive ratio gets closer to an upper bound of 0.33, which [25] proves for any consistent algorithm solving ROSP. However, a substantial gap between the best known upper and lower bounds remains.

¹ Fujii and Yoshida prove a 0.25 upper bound for the random order setting where the decision-maker only sees the order of candidate arrivals (not their exact random arrival times), and where the decision algorithm is deterministic. By leveraging arrival-order control and randomization, our algorithm exceeds this barrier. See Section Section 5 for discussion.

1.2 Technical Overview

We now overview the key ideas behind our improved algorithms for both COSP and ROSP. Throughout (see Section 2 for details) we assume a continuous arrival model, in which each candidate $i \in [n]$ arrives at some time $t_i \in [0, 1]$. In ROSP, t_i is chosen independently and uniformly at random. In COSP, t_i is set by the algorithm.

A central requirement for consistency in the secretary problem with predictions is that the algorithm must hire the top-predicted candidate \hat{i} if they arrive before any candidate with a large deviation from its prediction has been observed. Otherwise, if all predictions end up being accurate (i.e., $\epsilon \approx 0$), the algorithm may not hire a candidate with the highest value ($v^* \approx \hat{v}$), and thus will not obtain a $\frac{1-\epsilon}{1+\epsilon} \approx 1$ competitive ratio.

The algorithm of Fujii and Yoshida [36] builds on this principle. It begins in *prediction mode*, where it plans to hire \hat{i} upon arrival, ignoring all preceding candidates. However, if the algorithm encounters a candidate whose observed value deviates significantly from its prediction – a candidate we refer to as a “mistake” or “prediction error” – before observing \hat{i} , it abandons the prediction-guided strategy and switches to *secretary mode*, where it applies the classical threshold-based algorithm of Dynkin [32]. This algorithm provides a worst-case $1/e$ -competitive guarantee – the best possible in the worst-case.

The main technical challenge in the work of Fujii and Yoshida [36] lies in analyzing the worst-case competitive ratio of their algorithm in the large ϵ regime. Once the algorithm switches to Dynkin’s strategy, the resulting behavior depends intricately on the timing and identity of the candidate that triggered the switch, as well as the arrival time of the remaining candidates. To establish robustness guarantees, they perform a detailed case analysis that accounts for all possible positions of the top predicted and the true top candidate, as well as the arrival of all candidates with high prediction error. Ultimately, they show that in the large ϵ setting, their algorithm achieves a competitive ratio of 0.215. Note that a natural upper bound here is $1/e \approx 0.368$ – which would be obtained if one knew in advance that there would be a significant prediction error, and followed Dynkin’s rule from the beginning. Moreover, in the random-order model, any algorithm that is consistent – i.e., achieves a competitive ratio approaching 1 when predictions are accurate – cannot have robustness exceeding 0.33; this follows from the upper bound of [25] for ROSP. This bound applies to consistent algorithms operating under random arrival, but whether an analogous limitation holds for COSP is currently unknown.

Fixing the arrival of \hat{i} . Interestingly, in the analysis by Fujii and Yoshida, the arrival position of the top-predicted candidate \hat{i} has a significant impact on the competitive ratio. This observation motivates our focus on COSP, and on strategically fixing the arrival time of \hat{i} in particular. To illustrate this, suppose we fix \hat{i} ’s arrival time to $\beta \approx 0$ and keep all other arrival times uniformly random. The adversary can set all predictions correctly except for the true best candidate i^* , which is underestimated. With probability β , this candidate arrives before \hat{i} , triggers a switch to secretary mode, and is skipped. With probability $1 - \beta$, the algorithm hires \hat{i} before ever seeing i^* . In either case, the competitive ratio approaches zero. On the other hand, if $\beta \approx 1$, the adversary may overestimate \hat{i} , causing the algorithm to skip all earlier candidates in favor of \hat{i} , which results in success only with probability $1 - \beta$ – again close to zero. Therefore, setting $\beta = 0$ or $\beta = 1$ is provably suboptimal, and we instead optimize over β to find a value that balances these trade-offs and maximizes the competitive ratio. We note that placing a distribution over the arrival time β of \hat{i} , rather than fixing it deterministically, may further improve performance. However, we leave this

possibility for future work. We conjecture that controlling the arrival times of candidates other than \hat{i} (rather than simply randomizing them) cannot significantly improve our bounds – however, we again leave investigating this question for future work.

Tuning the probability of hiring \hat{i} . Unfortunately, simply optimizing the arrival time of \hat{i} and applying the algorithm of Fujii and Yoshida [36] directly can fail. Recall that Dynkin’s worst-case algorithm ignores all candidates before some threshold time τ , and then hires the first candidate that arrives after τ and is better than all preceding candidates. If we set $\beta < \tau$ and the top predicted candidate \hat{i} is in fact the true best candidate, but its prediction deviates significantly from its value, the algorithm switches to secretary mode upon its arrival and rejects \hat{i} since it arrives before τ . This can result in a near zero competitive ratio. Conversely, if we fix $\beta > \tau$, consider a scenario with no deviation between the predictions and the true values, except that \hat{i} is predicted to be the best candidate but is actually the second-best candidate (and by a wide margin). If the true best candidate i^* arrives before β , it will be ignored since the algorithm remains in prediction mode. If it arrives after β , the algorithm will have already selected \hat{i} as it would be the best candidate seen after τ and after switching to secretary mode. Again, we have a competitive ratio of near zero.

To handle the above issues, we set $\beta > \tau$, but modify the algorithm to not always hire \hat{i} when it is the first mistake and the best candidate observed so far. Instead, we introduce a parameter γ that specifies the probability of hiring \hat{i} in this case. This change ensures that the algorithm retains flexibility in handling situations where blindly trusting \hat{i} could lead to poor outcomes. Optimizing γ allows us to avoid deterministic failure while still preserving good performance in other cases.

We introduce a second parameter δ to handle the case where \hat{i} is not the first observed mistake. It does not make sense to use the same hiring probability γ in this setting, as the context is different: the algorithm has already seen one or more mistakes and switched to secretary mode before seeing \hat{i} . The likelihood that \hat{i} is truly optimal may be lower or higher than the case when it is the first mistake, so we allow a separate hiring probability δ . This additional degree of freedom enables us to optimize the algorithm’s behavior even further.

As in the work of Fujii and Yoshida, a key technical challenge is to analyze the competitive ratio against all possible adversarial inputs. This requires accounting for every configuration of the true best candidate i^* , the top-predicted candidate \hat{i} , and the locations of all large-deviation candidates. As in their analysis, this leads to a detailed case decomposition. Following their notation, let M be the set of large-deviation candidates, and let $m = |M|$. We further refine the structure of the high-deviation set M by comparing the true values of its members to the true value of the best-predicted candidate \hat{i} . Specifically, we define

$$m_2 = |\{i \in M : v_i < v_{\hat{i}}\}|, \quad m_1 = m - m_2,$$

so m_2 counts the number of “mistakes” whose true value is *smaller* than the true value of \hat{i} . Although both quantities are well defined, only m_2 plays a role in our competitive-ratio bounds. In addition, we define

$$k = |\{i \in [n] : v_i > v_{\hat{i}}\}|,$$

which is the number of candidates whose true value exceeds that of the best-predicted candidate \hat{i} . The parameter k captures how many truly better candidates the adversary places relative to \hat{i} , and therefore directly influences the worst-case behavior of the algorithm in prediction mode.

These parameters allow us to partition the analysis into fine-grained subcases, each capturing a distinct adversarial structure. We then optimize over β , γ , and δ to obtain the worst-case competitive ratio.

Despite the added generality, the algorithm retains a simple structure amenable to analysis. By carefully organizing the cases based on the parameters, we are able to compute the worst-case competitive ratio numerically. Through this analysis, we obtain a provable lower bound of $\max\left\{0.262, \frac{1-\epsilon}{1+\epsilon}\right\}$ for COSP, demonstrating the benefit of tuning β , γ , and δ .

Extension to RO SP. In the COSP setting, for each fixed value of β , we evaluate the competitive ratio across all adversarial configurations and then optimize the parameters γ and δ for that fixed β . We then optimize over β to obtain the final COSP bound.

In the RO SP setting, however, the arrival time of \hat{i} is itself a uniform random variable in $[0, 1]$, and cannot be chosen by the algorithm. Accordingly, the analysis must first take the expectation of the competitive ratio over $\beta \sim \text{Unif}[0, 1]$. We then optimize the parameters γ and δ with respect to this new competitive ratio objective. This yields a lower bound of

$$\max\left\{0.221, \frac{1-\epsilon}{1+\epsilon}\right\},$$

slightly improving on the prior bound of [36].

Why only choose arrival of \hat{i} and the decision at arrival time? One natural question is whether introducing additional parameters – for example, tuning the algorithm’s behavior when encountering the second- or third-highest predicted candidates – could further improve the robustness. Our analysis, as well as the structure of [36], suggests that this is unlikely: in all adversarial instances that determine the worst-case competitive ratio, the only critical decision point is when the algorithm reaches \hat{i} or first mistake (deviated prediction) while still following the classical secretary rule. Lower-predicted candidates never influence these worst-case configurations in a meaningful way unless they are mistakes (deviated predictions). By contrast, in extensions of the secretary problem where the goal is to select the top $k > 1$ individuals or where multiple high-valued candidates may be accepted, controlling the arrival times of several top-predicted candidates could indeed be beneficial. Developing such multi-candidate scheduling strategies is an interesting direction for future work.

1.3 Related works

Classical and Variant Secretary Problems. The classical secretary problem, popularized by Gardner [37] and others [32, 39, 54, 34, 55, 46, 27, 67, 28, 9], has inspired a broad and influential line of work in online algorithms, economics, and decision theory. Numerous extensions have been studied, including multiple-choice variants [49], matroid and knapsack constraints [11], online matching and graphic/transversal matroids [50], submodular objectives under cardinality and packing constraints [48], and structured feasibility systems such as packing integer programs [8]. We refer the reader to the survey by Gupta and Singla [42] for a broader overview of results in random-order models.

Secretary Problems with Predictions. Recent work incorporates predictions into the secretary problem to enable improved performance. In the model that we adopt, each candidate comes with a predicted value \hat{v}_i estimating their true value v_i , as in [36, 25]. Other forms of predictive feedback include binary comparisons [15], probabilistic signals about

candidate quality [61, 31], single-shot predictions of the best candidate [6], and predictions of the value difference between the best and the k -th best candidate [19]. Some work has studied algorithms that ensure fairness by guaranteeing the best candidate is accepted with constant probability, even under biased predictions [12]. Additionally, Amanatidis et al. [4] study value maximization with predictions in an online budget-feasible mechanism design setting, using techniques related to the secretary problem.

Prophet Inequalities and Order Selection. Prophet inequalities are a classical model in optimal stopping theory that can be seen as a stochastic version of the secretary problem. A decision-maker observes a sequence of values drawn from known distributions and aims to compete with a benchmark that selects the maximum in hindsight. The classical result guarantees a $1/2$ competitive ratio [51, 69], which has inspired a rich body of work in theoretical computer science and economics; see surveys such as [44, 56, 26]. Numerous variants have been studied, including the adversarial order model [51], the random order (prophet secretary) model [33], and the order-selection (free-order) model [43]. This model is especially related to our Chosen Order Secretary Problem (COSP), in which the arrival order is determined by predicted candidate values. Relevant works in the order-selection model include [23, 16, 62, 21]. Intuitively, prophet inequalities capture similar scenarios in which the decision-maker has in advance information about candidate values; however, they operate under stochastic assumptions (values drawn from known distributions), whereas our setting is fully worst-case, with no distributional assumptions. Interestingly, until recently, it was unclear whether the order-selection and random-arrival models for prophet inequalities differed in their achievable competitive ratios. A separation has now been established for prophet inequalities [21], but in our setting, the relationship remains open: the best upper bound for the random-arrival variant (ROSP) currently exceeds the best known competitive ratio for order selection (COSP).

Learning-Augmented and Online Algorithms with Predictions. A growing literature studies learning-augmented online algorithms in which predictions guide online decisions without compromising worst-case robustness. Notable examples include caching [57, 66, 73], ski rental [63, 41, 74, 30], online scheduling [52], and metrical task systems [5]. Learning-augmented approaches have also been applied in online primal-dual methods [14], speed scaling [13], online bidding [1], and auctions [40]. We refer the reader to surveys such as [60] for comprehensive overviews. Moreover, learning-augmented approaches have also been applied to online purchasing-type problems with prediction, such as one-way trading, online conversion [2, 72, 53], and online knapsack problems with prediction [45, 38, 17, 29]. These problems often share structural similarities with secretary problems – e.g., online knapsack with unit-size items and unit capacity resembles the secretary problem under random arrival.

1.4 Outline of the Paper

This paper is organized as follows. In Section 2, we give preliminaries and formally define our problem settings and the notation. In Section 3, we study the Chosen Order Secretary Problem (COSP), presenting a randomized algorithm that achieves competitive ratio $\max\{0.262, \frac{1-\epsilon}{1+\epsilon}\}$. In Section 4, we study the Random Order Secretary Problem (ROSP), extending our algorithm for COSP to obtain a competitive ratio of $\max\{0.221, \frac{1-\epsilon}{1+\epsilon}\}$ in this setting. Finally, in Section 5, we examine the relationship between deterministic and randomized algorithms in the classic secretary problem with predictions, showing that under the standard value-maximization formulation, both models achieve the same competitive ratio.

Full Version

Due to the ITCS 20-page limit (excluding title page and references), several technical sections – most notably Section 4 and Section 5 and Appendix – are omitted from this conference version. The full version of the paper contains complete proofs, extended discussion, and all appendices. Throughout this paper, when referring to these results we point the reader to the full version.

2 Preliminaries

We let $N = \{1, \dots, n\}$ denote the set of candidates. Throughout, we adopt a continuous-time framework, where each candidate i is assigned an arrival time $t_i \in [0, 1]$. In ROSP, each t_i is drawn independently from $\text{Unif}(0, 1)$ [49, 20, 35]. This induces an arrival order that is a uniform random permutation. In COSP, the decision-maker selects the arrival time t_i of each candidate. Note that in ROSP, for any subinterval $A \subseteq [0, 1]$, the probability that candidate i arrives within A is equal to its length.

Because all arrival times are distinct with probability 1, the continuous-time model is equivalent to the related model where candidates are just assigned some arrival position in $[n]$ – in that model, the decision-maker could always draw a set of random arrival times to assign to the candidates based on their arrival order.

Each candidate $i \in [n]$ has a true value $v_i \in \mathbb{R}_{\geq 0}$, which remains unknown to the decision-maker until the candidate appears at time t_i . Upon observing each candidate, the decision-maker must irrevocably choose whether to select or reject them; once a candidate is selected, the process terminates. We evaluate performance under the value-maximization objective: maximizing the expected value of the selected candidate, where the expectation is taken over any randomness in the arrival order and any randomness used by the selection algorithm. As defined before, the performance measurement tool is competitive ratio, the worst-case ratio (over all possible assignments of candidate values) of the expected reward of the algorithm to the maximum true value, i.e., $\mathbb{E}[\text{ALG}]/v^*$, where $\mathbb{E}[\text{ALG}]$ denotes the expected reward achieved by the algorithm and $v^* = \max_{i \in [n]} v_i$ is the highest true value among all candidates, corresponding to candidate i^* arrived at time t^* .

In addition to the true values v_i , a prediction $\hat{v}_i \in \mathbb{R}_{\geq 0}$ is provided in advance for each candidate $i \in [n]$, representing an estimate of v_i . We let \hat{i} denote the candidate with the highest predicted value arriving at time \hat{t} . Let $\epsilon = \max_{i \in [n]} \left| 1 - \frac{\hat{v}_i}{v_i} \right|$ denote the largest multiplicative prediction error among all candidates. We will often make use of the following error guarantee on predictions:

$$(1 - \epsilon)v_i \leq \hat{v}_i \leq (1 + \epsilon)v_i \quad \text{for all } i \in [n]. \quad (1)$$

Without loss of generality, we assume uniqueness for both the predicted values and the true values; hence, there is a uniquely defined top candidate i^* and top predicted candidate \hat{i} . To handle cases with repeated values, we can slightly perturb the values to break ties. In particular, when the same true value v_i appears more than once in the sequence, we treat each subsequent occurrence as slightly larger than the previous one by adding an infinitesimal offset (e.g., $v \mapsto v + \eta$ with $\eta > 0$ arbitrarily small). This guarantees that all values are distinct, while ensuring that the modified values remain arbitrarily close to the original ones. The algorithm is then competitive with respect to this perturbed sequence, and since the perturbed values differ from the true values by at most an infinitesimal factor, the resulting competitiveness carries over to the original sequence. We can use a similar approach for the predicted values \hat{v}_i .

3 Chosen Order Secretary Problem (COSP) with Prediction

In this section, we analyze the competitive ratio of our algorithm. We begin by providing intuition for the algorithm, then define the model and set up the key parameters. We then analyze the competitive ratio under a fixed arrival time β of the top predicted candidate.

3.1 Algorithm Intuition

We describe our algorithm for the COSP setting, where the arrival order is partially controlled based on predictions. Specifically, the candidate with the highest predicted value – denoted by $\hat{i} = \arg \max_i \hat{v}_i$ – is scheduled to arrive at a fixed time $\beta \in [0, 1]$, i.e., we set $\hat{t} = \beta$, while all other candidates are assigned independent arrival times drawn uniformly from $[0, 1]$.

The algorithm operates in two modes: **Prediction mode** and **Secretary mode**. It begins in Prediction mode and switches to Secretary mode upon observing a candidate i such that the prediction error satisfies $\left|1 - \frac{\hat{v}_i}{v_i}\right| \geq \theta$, indicating that the predictions is unreliable (Line 5 of Algorithm 1). Here, θ specifies the maximum acceptable prediction error. The algorithm seeks to allow as many predictions as possible to be considered reliable, while still guaranteeing consistency whenever all errors are within θ .

In our algorithm, if a mode switch occurs, the algorithm transitions to Secretary mode at the time denoted in this paper as t_M . It then follows a variant of Dynkin’s rule, the well-known $1/e$ -competitive strategy that skips all candidates arriving before a threshold time τ and hires the first candidate thereafter whose value exceeds all previously observed ones.

Our algorithm skips all candidates arriving before a time threshold τ , and selects the first candidate thereafter with a value higher than all previously observed ones. Special care is taken if \hat{i} arrives in Secretary mode: it is hired with probability γ if it triggers the mode switch (i.e., arrives at time t_M), and with probability δ if it arrives after the mode switched. These randomized rules ensure robustness, avoiding over-reliance on potentially incorrect predictions while still preserving high performance when predictions are accurate. Moreover, when θ is large enough, the algorithm is less likely to switch to secretary mode and more likely to hire \hat{i} , relying on prediction and hiring the highest predicted value candidate \hat{i} .

Later, we show that with carefully chosen parameters, our algorithm for COSP setting achieves a competitive ratio of at least

$$\max \left\{ 0.262, \frac{1 - \epsilon}{1 + \epsilon} \right\}$$

across all instances with maximum multiplicative prediction error ϵ .

3.2 Competitive Ratio Guarantee

In this subsection, we formally analyze the performance of our algorithm by establishing a lower bound on its competitive ratio. Specifically, we divide the analysis into multiple cases and prove that the algorithm achieves a worst-case competitive ratio of at least $\max \left\{ 0.262, \frac{1 - \epsilon}{1 + \epsilon} \right\}$ for the Chosen Order Secretary Problem (COSP).

► **Theorem 1.** *Algorithm 1 with parameters $\theta = 0.58$, $\tau = 0.37$, $\beta = 0.64$, $\gamma = 0.27$, and $\delta = 0.46$ is at least $\max \left\{ 0.262, \frac{1 - \epsilon}{1 + \epsilon} \right\}$ -competitive for the Chosen Order Secretary Problem (COSP).*

86:10 Secretary Problem with Predictions and a Chosen Order

■ **Algorithm 1** Prediction-Aware Secretary Algorithm for COSP.

Require: Predicted values \hat{v}_i for all $i \in [n]$; threshold parameter $\theta \in [0, 1]$, $\tau \in [0, 1]$; fixed arrival time β of the top predicted candidate; probabilities $\gamma, \delta \in [0, 1]$:

- 1: $\hat{i} \leftarrow \arg \max_{i \in [n]} \hat{v}_i$ ▷ Top predicted candidate
- 2: Assign $t_{\hat{i}} \leftarrow \beta$; assign $t_i \sim \text{Unif}(0, 1)$ for all $i \neq \hat{i}$
- 3: mode \leftarrow **Prediction**
- 4: **for** each candidate $i \in [n]$ in increasing order of arrival time t_i **do**
- 5: **if** $|1 - \hat{v}_i/v_i| > \theta$ **and** mode = **Prediction** **then**
- 6: mode \leftarrow **Secretary**; $t_M \leftarrow t_i$ ▷ First large deviation triggers switch
- 7: **end if**
- 8: **if** mode = **Prediction** **and** $i = \hat{i}$ **then**
- 9: **return** hire candidate i
- 10: **else if** mode = **Secretary** **and** $t_i > \tau$ **and** $v_i > \max\{v_j : t_j < t_i\}$ **then**
- 11: **if** $i = \hat{i}$ **and** $t_i = t_M$ **then**
- 12: Hire i with probability γ
- 13: **else if** $i = \hat{i}$ **then**
- 14: Hire i with probability δ
- 15: **else**
- 16: Hire candidate i
- 17: **end if**
- 18: **end if**
- 19: **end for**

Proof. Following the case-based analysis of Fujii and Yoshida [36], we divide the proof into seven cases, depending on the structure of the high-deviation set M , which contains all candidates whose predictions deviate from their actual values by more than the threshold θ used to switch from Prediction to Secretary mode in Line 5 of Algorithm 1. Formally, we define this set as:

$$M = \left\{ i \in [n] : \left| 1 - \frac{\hat{v}_i}{v_i} \right| > \theta \right\}.$$

Our cases will consider whether the candidate with the highest predicted value \hat{i} and/or the candidate with the highest true value i^* belong to M . Also, we define i_M as the first candidate's index to be seen that falls in M and we denote its arrival time as t_M . We also let $k = |\{i \in [n] \mid v_i > v_{\hat{i}}\}|$ denote the number of candidates with true value greater than \hat{i} , and let $m_2 = |\{i \in M \mid v_i < v_{\hat{i}}\}|$ denote the number of candidates in set M with true value less than \hat{i} .

Throughout the analysis, we discuss the regime where the fixed arrival time β of \hat{i} satisfies $\beta > \tau$, as this aligns with parameters obtained in our optimization over parameters $\tau, \beta, \gamma, \delta$ (Theorem 1). This focus simplifies the presentation, but both conditions $\beta > \tau$ and $\beta < \tau$ are fully analyzed in the next section to establish guarantees across all parameter choices. We denote the worst-case competitive ratio achieved in each case $j \in \{0, 1, \dots, 6\}$ by $C_j(\cdot)$ respect to any input set by the adversary, where $C_j(\cdot)$ is a function of parameters set by the adversary (i.e., m, k, v_i , and \hat{v}_i). The case ordering follows the structure introduced in [36].

We will formally verify that our algorithm achieves a competitive ratio of at least $\max\left\{0.262, \frac{1-\epsilon}{1+\epsilon}\right\}$ across all cases. For clarity of presentation, we defer the technical lemmas used in the analysis to the appendix, while the main text focuses on the case-by-case argument structure.

To organize the argument, we present each of the seven cases as a separate part, analyzing the conditions under which the algorithm hires a candidate and deriving a lower bound on the competitive ratio in each scenario.

Case 0: $|M| = 0$. In this case, all predictions fall within the acceptable error threshold (i.e., $\left|1 - \frac{\hat{v}_i}{v_i}\right| < \theta$), so the high-deviation set M is empty, and the algorithm remains in prediction mode throughout.

This implies that the maximum multiplicative prediction error satisfies $\epsilon \leq \theta$, and by the Multiplicative Error Guarantee (Equation (1)), all predicted values remain within the $(1 \pm \epsilon)$ bound.

Since the algorithm selects $\hat{i} = \arg \max_i \hat{v}_i$, it follows that

$$v_i \geq \frac{\hat{v}_i}{1 + \epsilon} \geq \frac{\hat{v}_{i^*}}{1 + \epsilon}, \quad (2)$$

and moreover,

$$v_i \geq \frac{\hat{v}_{i^*}}{1 + \epsilon} \geq \frac{(1 - \epsilon)v_{i^*}}{1 + \epsilon}. \quad (3)$$

Together, these inequalities imply a competitive ratio of at least $\frac{1-\epsilon}{1+\epsilon}$. Since $\epsilon \leq \theta = 0.58$, we have

$$\frac{1 - \epsilon}{1 + \epsilon} \geq \frac{1 - 0.58}{1 + 0.58} > 0.262. \quad (4)$$

Thus, in this case the algorithm achieves a competitive ratio of $\frac{1-\epsilon}{1+\epsilon}$, which is at least $\max\left\{0.262, \frac{1-\epsilon}{1+\epsilon}\right\}$, as claimed, since $\epsilon \leq \theta$.

Case 1: $\hat{i} = i^* \in M$. Here, the top predicted candidate \hat{i} is also the true optimum i^* , but it lies in the high-deviation set M . The algorithm switches to secretary mode upon encountering the first candidate in M , which may or may not be i^* itself.

We analyze two subcases, depending on whether i^* triggers the mode switch or not:

Case(1.1) $\beta = t_M$: That is, i^* is the first candidate in M , and it arrives at time β . The probability of this subcase occurring is equal to the probability that all other candidates in M arrive after β , which is $(1 - \beta)^{m-1}$. Also, the probability of successfully selecting i^* is γ since i^* is the first observed error. Thus, the contribution of subcase *Case(1.1)* to the competitive ratio of this case is $\gamma(1 - \beta)^{m-1}$: the event that i^* is the first observed member of M with probability $(1 - \beta)^{m-1}$, and conditioned on this event the algorithm selects it with probability γ (Algorithm 1, Line 11).

Case(1.2) $\beta \neq t_M$: Here, some other candidate in M arrived before \hat{i} , so we are already in Secretary mode by the time i^* arrives at time $\beta > \tau$. The probability of this subcase happening is equal to $1 - (1 - \beta)^{m-1}$ which is at least one candidate in M arrives before \hat{i} . Now, assuming we are in this subcase, in order to reach \hat{i} (to accept it), the algorithm must not accept anyone before \hat{i} . It is sufficient (but not necessary) that the best candidate in the interval $[0, \beta)$ came before τ , as then no one would be hired before it. The probability of this scenario is τ/β . Furthermore, the Algorithm accepts \hat{i} with probability δ according to Line 13 of Algorithm 1. It is worth mentioning that the arrival of the best candidate in $[0, \beta)$ before τ is independent of whether a candidate in M arrives before \hat{i} , since candidates in M arriving before β are also uniformly distributed in $[0, \beta)$.

Combining both subcases, the competitive ratio in Case 1 for any m is lower-bounded by:

$$C_1(m, \cdot) \geq \frac{\tau}{\beta} \cdot \delta \cdot (1 - (1 - \beta)^{m-1}) + \gamma \cdot (1 - \beta)^{m-1}. \quad (5)$$

Case 2: $\hat{i} = i^* \notin M$. Under this condition, the top predicted candidate \hat{i} is also the true optimum i^* , and its prediction is accurate enough that it does not belong to the high-deviation set M . The algorithm may or may not have entered secretary mode before observing i^* , depending on whether another candidate in M has been seen earlier.

To obtain a pessimistic lower bound in this case, we assume that $i^* \in M$, even though by the conditions of case 2 we know $i^* \notin M$. This assumption decreases the chance of hiring i^* . If the algorithm is already in secretary mode when it sees i^* , then whether or not $i^* \in M$ has no effect – the selection behavior and probabilities are the same (see Line 13 of Algorithm 1). However, if the algorithm is still in prediction mode when i^* arrives, then: (1) If $i^* \notin M$, the algorithm selects it deterministically. (2) If $i^* \in M$, the algorithm does not select it immediately and instead switches to secretary mode, where it will only be selected with probability $\gamma \leq 1$. Thus, the pessimistic assumption $i^* \in M$ reduces the chance of hiring i^* , and we conclude the result is always better than *Case 1*, presented above, with one extra candidate in M :

$$C_2(m, \cdot) \geq C_1(m + 1, \cdot). \quad (6)$$

Case 3: $\hat{i} \neq i^*$, $\hat{i} \in M$, $i^* \in M$. In this situation, both the top predicted candidate \hat{i} and the true optimum i^* belong to the high-deviation set M . We lower bound this case by Case 4, where $\hat{i} \neq i^*$, $\hat{i} \in M$, and $\hat{i} \notin M$. The only difference between the two cases is that here $i^* \in M$. We further analyze this case by considering two subcases depending on whether t^* is greater than or less than β .

Case(3.1): If $t^* > \beta$, the algorithm is already in secretary mode when i^* arrives because \hat{i} arrives at time β is in set M , so whether or not $i^* \in M$ does not affect its eligibility for hiring. The algorithm selects i^* if it is the best seen so far and no better candidate has already been selected in Line 16 of Algorithm 1.

Case(3.2): If $t^* < \beta$, then for i^* to be selected, the algorithm must already be in secretary mode at or before time t^* . When $i^* \in M$, it can itself trigger this mode switch. However, if $i^* \notin M$, it can only be selected if another candidate in M appears earlier. Hence, in all scenarios, $i^* \notin M$ is selected with probability no greater than when $i^* \in M$.

Therefore, Case 3 is no harder than Case 4 with one fewer member in M (because of removing i^* from M and thus decreasing its size by 1), and we conclude that for a fixed m , the worst-case competitive ratios of the two cases satisfy the following relation:

$$C_3(m, \cdot) \geq C_4(m - 1, \cdot).$$

Case 4: $\hat{i} \neq i^*$, $\hat{i} \in M$, $i^* \notin M$. Under this condition, the top predicted candidate \hat{i} belongs to the high-deviation set M , while the true optimum $i^* \notin M$ has a reliable prediction. Therefore, the algorithm never selects a candidate in prediction mode and switches to secretary mode upon encountering the first candidate $i \in M$. We analyze two main subcases based on the arrival time t^* of i^* . The first subcase occurs when $t^* \in (\tau, \beta)$, and the second when $t^* \in (\beta, 1]$. We further divide the second subcase into two parts: (i) the best candidate in $[0, t^*)$ is not \hat{i} , and (ii) the best candidate in $[0, t^*)$ is \hat{i} . Combining the analyses of all these subcases yields the overall competitive ratio for Case 4.

Case(4.1) When $t^* \in (\tau, \beta)$: In this subcase, the algorithm can only consider i^* if it has already switched to secretary mode by time t^* . This requires that at least one candidate from M to have arrived before t^* , which occurs with probability $1 - (1 - t^*)^{m-1}$, also, it is sufficient (but not necessary) that the best candidate in $[0, t^*)$ arrives before τ as then

no one would be hired before i^* , which happens with probability τ/t^* . Similar to Case 1, these two probabilities are independent since any candidate in M which comes before t^* also uniformly arrives between $[0, t^*)$. We obtain the bound:

$$C_{4.1}(m, \cdot) \geq \int_{\tau}^{\beta} (1 - (1 - t^*)^{m-1}) \cdot \frac{\tau}{t^*} dt^*. \quad (7)$$

(see full version for details) To calculate this integral, we get:

$$C_{4.1}(m, \cdot) \geq \tau \sum_{i=1}^{m-1} \binom{m-1}{i} (-1)^{i+1} \cdot \frac{\beta^i - \tau^i}{i}. \quad (8)$$

Case(4.2) When $t^* \in (\beta, 1]$: In this subcase, the algorithm has already seen $\hat{i} \in M$, and we examine whether i^* can still be observed and selected under different conditions. We further divide this case into two subcases:

Case(4.2.1) The best candidate in $[0, t^*)$ is not \hat{i} : This occurs with probability $1 - (1 - t^*)^k$, where k is the number of candidates whose true value exceeds that of \hat{i} . For i^* to be selected, it is sufficient that the best candidate in $[0, t^*)$ arrives before τ , which happens with probability τ/t^* . Similar to previous cases, these two events are independent, since knowing that some candidate better than \hat{i} arrives before t^* does not affect the fact that all arrivals before t^* are uniformly distributed over $[0, t^*)$. Thus, we obtain the following equation:

$$C_{4.2.1}(m, \cdot) \geq \int_{\beta}^1 (1 - (1 - t^*)^k) \cdot \frac{\tau}{t^*} dt^*. \quad (9)$$

(see full version for details) to calculate the integral, we have:

$$C_{4.2.1}(m) \geq \tau \sum_{i=1}^k \binom{k}{i} (-1)^{i+1} \cdot \frac{1 - \beta^i}{i}. \quad (10)$$

Case(4.2.2) The best candidate in $[0, t^*)$ is \hat{i} : In this scenario, which occurs with probability $(1 - t^*)^k$, the algorithm must reach time t^* without hiring \hat{i} in order for i^* to be selected. This leads to two possible outcomes:

Case(4.2.2.1) A candidate in M other than \hat{i} triggers the switch to secretary mode before β : This occurs with probability $1 - (1 - \beta)^{m_2}$, where m_2 is the number of candidates in M whose values are less than that of \hat{i} . We use m_2 here because all candidates in M with values greater than \hat{i} must arrive after t^* , given that \hat{i} is assumed to be the best candidate in $[0, t^*)$. It is further required that the best candidate in $[0, \beta)$ arrives before τ , which happens with probability τ/β . In this case, we do not select any candidate before seeing \hat{i} , and when \hat{i} arrives, it must also not be hired, which occurs with probability $1 - \delta$ (Line 13 of Algorithm 1). Since the event of having a candidate arrive before β is independent of the fact that arrivals in $[0, \beta)$ are uniformly distributed, any best candidate in $[0, \beta)$ arrives uniformly at random. Considering statement *Case(4.2.2)* and multiplying these independent probabilities, we have:

$$C_{4.2.2.1}(m) \geq \int_{\beta}^1 (1 - t^*)^k (1 - (1 - \beta)^{m_2}) (1 - \delta) \frac{\tau}{\beta} dt^*. \quad (11)$$

Case(4.2.2.2) Candidate \hat{i} itself triggers the switch to secretary mode: This event occurs with probability $(1 - \beta)^{m_2}$, and is not hired with probability $1 - \gamma$. Therefore, incorporating the factor $(1 - t^*)^k$ from (4.2.2), we have:

$$C_{4.2.2.2}(m) \geq \int_{\beta}^1 (1 - t^*)^k (1 - \beta)^{m_2} (1 - \gamma) dt^*. \quad (12)$$

86:14 Secretary Problem with Predictions and a Chosen Order

Combining Equations (11) and (12) and (see full version for details) to solve the integrals, we have:

$$\begin{aligned} C_{4.2.2}(m) &\geq \int_{\beta}^1 (1-t^*)^k \left[(1-(1-\beta)^{m_2})(1-\delta)\frac{\tau}{\beta} + (1-\beta)^{m_2}(1-\gamma) \right] dt^* \\ &= \frac{(1-\beta)^{k+1}}{k+1} \cdot \left[(1-(1-\beta)^{m_2})(1-\delta)\frac{\tau}{\beta} + (1-\beta)^{m_2}(1-\gamma) \right]. \end{aligned} \quad (13)$$

Combining Equations (8), (10), and (13) from the subcases, we obtain the total lower bound for this case:

$$\begin{aligned} C_4(m) &\geq \tau \sum_{i=1}^{m-1} \binom{m-1}{i} (-1)^{i+1} \cdot \frac{\beta^i - \tau^i}{i} \\ &\quad + \tau \sum_{i=1}^k \binom{k}{i} (-1)^{i+1} \cdot \frac{1-\beta^i}{i} \\ &\quad + \frac{(1-\beta)^{k+1}}{k+1} \cdot \left[(1-(1-\beta)^{m_2})(1-\delta)\frac{\tau}{\beta} + (1-\beta)^{m_2}(1-\gamma) \right] \end{aligned} \quad (14)$$

Case 5: $\hat{i} \neq i^*$, $\hat{i} \notin M$, $i^* \in M$. In this case, the top predicted candidate \hat{i} is accurate and not in the high-deviation set M , while the true optimum $i^* \in M$. The algorithm switches to secretary mode upon observing the first candidate from M . We analyze the outcome based on the arrival time t^* of i^* , and the timing of the best candidate before t^* .

Case(5.1) $t^* \in [\tau, \beta]$: In this subcase, the competitive ratio depends on whether i^* is the first candidate in M , i.e., whether it triggers the switch to secretary mode. This subcase is further divided into two subcases:

Case(5.1.1) $i^* = t_M$: Under this condition, no other candidate in M arrives before t^* , and i^* is the first observed candidate in M . In this case, the algorithm switches to secretary mode at t^* , and immediately selects i^* . This happens with probability $(1-t^*)^{m-1}$, since all other $m-1$ candidates in M must arrive after t^* .

Case(5.1.2) $i^* \neq t_M$: In this scenario, some other candidate from M must arrive before t^* , causing the algorithm to switch modes before seeing i^* . In order for i^* to be selected, it must arrive after the switch, and the best candidate in $[0, t^*)$ must have arrived before τ , which occurs with probability τ/t^* . Also, the probability that at least one candidate in M arrives before t^* is $1 - (1-t^*)^{m-1}$. Hence, the contribution of this case is $(1 - (1-t^*)^{m-1}) \cdot \frac{\tau}{t^*}$. Combining both subcases *Case(5.1.1)* and *Case(5.1.2)* and (see full version for details) to solve the integral, we have:

$$\begin{aligned} C_{5.1}(m, \cdot) &\geq \int_{\tau}^{\beta} \left[(1 - (1-t^*)^{m-1}) \cdot \frac{\tau}{t^*} + (1-t^*)^{m-1} \right] dt^* \\ &= \tau \sum_{i=1}^{m-1} \binom{m-1}{i} (-1)^{i+1} \cdot \frac{\beta^i - \tau^i}{i} + \frac{(1-\tau)^m - (1-\beta)^m}{m}. \end{aligned} \quad (15)$$

Case(5.2) $t^* \in [\beta, 1]$: Here the optimal candidate arrives after the predicted top candidate. We distinguish two subcases based on whether the best candidate in $[0, t^*)$ is \hat{i} or not.

Case(5.2.1) The best candidate in $[0, t^*)$ is not \hat{i} : In this subcase the algorithm may still consider i^* under certain favorable conditions. First, let us analyze the probability that \hat{i} is not the best candidate in $[0, t^*)$. This occurs with probability $1 - (1-t^*)^k$, where k is the number of candidates with true value exceeding that of \hat{i} . Now, conditioned on this event, there must exist some candidate in M arriving before β . Knowing this, the probability that at least one candidate in M arrives before β is at least $1 - (1-\beta)^{m-1}$. This is because:

The presence of a better candidate than \hat{i} before t^* can only increase the likelihood that a candidate from M also appears before β , or at worst, leaves it unchanged.

Therefore, even under this conditioning, we can safely lower-bound the probability of having some candidate in M before β by $1 - (1 - \beta)^{m-1}$.

Finally, in order to observe i^* , the best candidate in $[0, t^*)$ must arrive before τ , which occurs independently with probability τ/t^* . Thus, the combined lower bound is (which can be rewritten (see full version for details)):

$$\begin{aligned} C_{5.2.1}(m, \cdot) &\geq \int_{\beta}^1 (1 - (1 - t^*)^k) \cdot \frac{\tau}{t^*} \cdot (1 - (1 - \beta)^{m-1}) dt^* \\ &= \tau \sum_{i=1}^k \binom{k}{i} (-1)^{i+1} \cdot \frac{1 - \beta^i}{i} \cdot (1 - (1 - \beta)^{m-1}). \end{aligned} \quad (16)$$

Case(5.2.2): The best candidate in $[0, t^*)$ is \hat{i} : In this subcase, the algorithm must avoid selecting \hat{i} during secretary mode in order to eventually observe and select i^* . This happens with the following conditions are met:

1. The probability that \hat{i} is the best candidate in $[0, t^*)$ is $(1 - t^*)^k$, where k is the number of candidates with true value greater than $v_{\hat{i}}$.
2. For the algorithm to enter secretary mode before seeing \hat{i} , at least one candidate in M from the subset of M with value below $v_{\hat{i}}$ must arrive before β . This happens with probability $1 - (1 - \beta)^{m_2}$, where m_2 is the number of such candidates.
3. To avoid selecting \hat{i} , the algorithm must skip it, which happens with probability $1 - \delta$.
4. Finally, the best candidate before β must have arrived before τ , so that the algorithm does not terminate before observing \hat{i} . This occurs with probability τ/β .

Combining all these probabilities using the contribution from this setting is lower-bounded by:

$$\begin{aligned} C_{5.2.2}(m) &\geq \int_{\beta}^1 (1 - t^*)^k \cdot \left[(1 - (1 - \beta)^{m_2}) (1 - \delta) \frac{\tau}{\beta} \right] dt^* \\ &= \frac{(1 - \beta)^{k+1}}{k + 1} \cdot \left[(1 - (1 - \beta)^{m_2}) (1 - \delta) \frac{\tau}{\beta} \right]. \end{aligned} \quad (17)$$

Summing the contributions from cases *Case(5.1)* and *Case(5.2)* in Equations (15), (16), and (17), the total competitive ratio for *Case 5* is lower-bounded by:

$$\begin{aligned} C_5(m, \cdot) &\geq \tau \sum_{i=1}^{m-1} \binom{m-1}{i} (-1)^{i+1} \cdot \frac{\beta^i - \tau^i}{i} + \frac{(1 - \tau)^m - (1 - \beta)^m}{m} \\ &\quad + \tau \sum_{i=1}^k \binom{k}{i} (-1)^{i+1} \cdot \frac{1 - \beta^i}{i} \cdot (1 - (1 - \beta)^{m-1}) \\ &\quad + \frac{(1 - \beta)^{k+1}}{k + 1} \cdot \left[(1 - (1 - \beta)^{m_2}) (1 - \delta) \frac{\tau}{\beta} \right]. \end{aligned} \quad (18)$$

Case 6: $\hat{i} \neq i^*$, $\hat{i} \notin M$, $i^* \notin M$. In this case, neither the top-predicted candidate \hat{i} nor the true optimal candidate i^* belongs to the high-deviation set M . Consequently, the algorithm may either remain entirely in prediction mode or switch to secretary mode, depending on whether any element of M arrives before \hat{i} . The overall analysis thus combines contributions from *Case 4* (when $t^* \in [\tau, \beta]$), *Case 5* (when $t^* \in [\beta, 1]$), together with an additional term accounting for successful selection in prediction mode.

86:16 Secretary Problem with Predictions and a Chosen Order

Case(6.1): If the algorithm stays in prediction mode and hires \hat{i} , this occurs when all members of M arrive after β , which happens with probability $(1-\beta)^m$. Since $\hat{i} \notin M$, its predicted value is within a multiplicative factor ϵ of its true value. Thus, by Equation (1), the competitive ratio is at least $\frac{1-\epsilon}{1+\epsilon}$. Therefore, the contribution to the competitive ratio from this event is:

$$C_{6.1}(m, \cdot) = (1-\beta)^m \cdot \frac{1-\epsilon}{1+\epsilon}. \quad (19)$$

Case(6.2): If the algorithm switches to secretary mode before encountering \hat{i} , then the selection of i^* depends on its arrival time t^* .

Case(6.2.1): $t^* \in [\tau, \beta]$: Here, the algorithm is already in secretary mode, and the same logic as Equation (8) from *case 4* applies:

$$C_{6.2.1}(m, \cdot) \geq \tau \sum_{i=1}^m \binom{m}{i} (-1)^{i+1} \cdot \frac{\beta^i - \tau^i}{i}. \quad (20)$$

Case(6.2.2) $t^* \in [\beta, 1]$: Here, we follow the same analysis as in Equations (16) and (17) from *Case 5*, where m denotes the number of candidates in M , and m_2 denotes the number of candidates in M whose true values are less than v_i , respectively. We obtain the following two contributions:

$$C_{6.2.2}^{(1)}(m, \cdot) \geq \tau \sum_{i=1}^k \binom{k}{i} (-1)^{i+1} \cdot \frac{1-\beta^i}{i} \cdot (1-(1-\beta)^m), \quad (21)$$

$$C_{6.2.2}^{(2)}(m, \cdot) \geq \frac{(1-\beta)^{k+1}}{k+1} \cdot \left[(1-(1-\beta)^{m_2}) (1-\delta) \frac{\tau}{\beta} \right]. \quad (22)$$

Summing the contributions from *Case(6.1)*, *Case(6.2.1)*, and *Case(6.2.2)*, we obtain the overall lower bound for Case 6:

$$\begin{aligned} C_6(m, \cdot) &\geq (1-\beta)^m \cdot \frac{1-\epsilon}{1+\epsilon} \\ &\quad + \tau \sum_{i=1}^m \binom{m}{i} (-1)^{i+1} \cdot \frac{\beta^i - \tau^i}{i} \\ &\quad + \tau \sum_{i=1}^k \binom{k}{i} (-1)^{i+1} \cdot \frac{1-\beta^i}{i} \cdot (1-(1-\beta)^m) \\ &\quad + \frac{(1-\beta)^{k+1}}{k+1} \cdot \left[(1-(1-\beta)^{m_2}) (1-\delta) \frac{\tau}{\beta} \right]. \end{aligned} \quad (23)$$

Completing the Proof via Case Enumeration. Recall that for each of the structural cases analyzed above, we derived an explicit competitive-ratio expression $C_j(m, k, m_2)$, parameterized by the integers m , k , and m_2 that describe the adversarial instance. The algorithm's overall guarantee on a given instance is

$$C(m, k, m_2) = \min_{j \in \{0, \dots, 6\}} C_j(m, k, m_2),$$

and our goal is to show that

$$C(m, k, m_2) \geq \max \left\{ 0.262, \frac{1-\epsilon}{1+\epsilon} \right\} \quad \text{for all } (m, k, m_2).$$

A direct minimization of $C(m, k, m_2)$ over all integer triples (m, k, m_2) is not tractable: the closed-form expressions contain alternating sums and exponentially small terms such as $(1 - \beta)^m$ and $(1 - t^*)^k$, which makes it difficult to analyze these formulas simultaneously for all integers. However, these expressions simplify dramatically once m , k , or m_2 becomes sufficiently large. For the remaining small values of (m, k, m_2) , the minimum of the case formulas can be computed exactly using a finite grid, following an approach similar to that of [36].

This motivates a structured enumeration strategy: we partition the parameter space into eight “regimes” depending on whether each variable is *small* (at most 20) or *large* (greater than 20). Regime 1 corresponds to all three parameters being small, while Regimes 2–8 cover all remaining combinations in which at least one parameter is large.

For small values $(m, m_2, k \leq 20)$, we can evaluate each $C_j(m, k, m_2)$ exactly using its closed form. For large values, terms such as $(1 - \beta)^m$ become negligible (e.g., for $m > 20$ and $\beta = 0.64$, we have $(1 - \beta)^m < 10^{-4}$), and can be conservatively replaced by zero. These replacements only *decrease* the computed competitive ratio, so they yield valid symbolic lower bounds. With this approach, the verification reduces to a finite enumeration over the eight regimes, as summarized in Algorithm 2.

■ **Algorithm 2** Lower Bound Proof via Case Analysis.

Require: Formulas C_1, C_4, C_5, C_6 parameterized by (m, k, m_2)

Require: Thresholds $T_m = 20, T_k = 20$

Require: Target lower bound $B \leftarrow \max\{0.262, \frac{1-\epsilon}{1+\epsilon}\}$

1: **Regime 1: All parameters small**

2: **for** $m = 0$ to T_m **do**

3: **for** $k = 0$ to T_k **do**

4: **for** $m_2 = \max(0, m - k)$ to m **do**

5: $v \leftarrow \min(C_1(m, k, m_2), C_4(m, k, m_2), C_5(m, k, m_2), C_6(m, k, m_2))$

6: **Assert:** $v \geq B$

7: **end for**

8: **end for**

9: **end for**

10: **Regimes 2–8: At least one parameter large**

11: **for all** combinations of large/small status for (m, k, m_2) except Regime 1 **do**

12: Substitute the symbolic lower bounds appropriate for that regime

13: $L \leftarrow \min(C_1, C_4, C_5, C_6)$ under this regime

14: **Assert:** $L \geq B$

15: **end for**

16: **Conclusion:** All admissible (m, k, m_2) satisfy $C(m, k, m_2) \geq B$

The enumeration confirms that for all instances the competitive ratio is at least $\max\{0.262, \frac{1-\epsilon}{1+\epsilon}\}$, when using the parameters $\theta = 0.58, \tau = 0.37, \beta = 0.64, \gamma = 0.27, \delta = 0.46$.

Symbolic Lower Bounds for Large Parameters. We now list the symbolic simplifications for large m, k , and m_2 used in Regimes 2–8 (regimes to 8 cases based on m, k , and m_2 being small or large, i.e. regime 1 is all of them being small) of Algorithm 2. We have to calculate other regimes cause formulas are dependent to m, m_2 and k and we should bound them. For example, assuming a large $m > 20$ in the Equation (5), we can ignore the right-hand side and left-hand side $(1 - \beta)^{m-1}$ is less than 0.0001, so the left-hand side will be bounded by $(\tau/\beta) \times \delta \times 0.9999$ same idea works for Equation (7) and other similar equations. In general,

86:18 Secretary Problem with Predictions and a Chosen Order

in the following formulas, we rounded down all small numbers to 0. and All $1 - \text{small number}$ to 0.9999. As we can show, all these small numbers are less than 0.0001. Similarly, for other large $m > 20$ in the exponent, the corresponding terms approach zero, allowing us to simplify the equations as follows:

For large $m > 20$:

$$C_1(\cdot) \geq 0.9999 \cdot \frac{\tau}{\beta} \cdot \delta \quad (24)$$

$$C_{4.1}(\cdot) \geq 0.9999 \cdot \tau \cdot \ln\left(\frac{\beta}{\tau}\right) \quad (25)$$

$$C_{5.1}(\cdot) \geq 0.9999 \cdot \tau \cdot \ln\left(\frac{\beta}{\tau}\right) \quad (26)$$

$$C_{6.1}(\cdot) \geq 0 \quad (27)$$

$$C_{6.2.1}(\cdot) \geq 0.9999 \cdot \tau \cdot \ln\left(\frac{\beta}{\tau}\right) \quad (28)$$

The next parameter to discuss is k . Similar to m , when we raise fixed numbers to the power of $k > 20$, the result will be close to zero.

For large $k > 20$:

$$C_{4.2.1}(m, \cdot) \geq 0.9999 \cdot \tau \cdot \ln\left(\frac{1}{\beta}\right) \quad (29)$$

$$C_{4.2.2}(m, \cdot) \geq 0 \quad (30)$$

$$C_{5.2.1}(m, \cdot) \geq 0.9999 \cdot \tau \cdot \ln\left(\frac{1}{\beta}\right) \cdot (1 - (1 - \beta)^{m-1}) \quad (31)$$

$$C_{5.2.2}(m, \cdot) \geq 0 \quad (32)$$

$$C_{6.2.2}^{(1)}(m, \cdot) \geq 0.9999 \cdot \tau \cdot \ln\left(\frac{1}{\beta}\right) \cdot (1 - (1 - \beta)^m) \quad (33)$$

$$C_{6.2.2}^{(2)}(m, \cdot) \geq 0 \quad (34)$$

The last parameter is m_2 , which is bounded by m . However, when m is large, this parameter can also become large. By rewriting the equation for large m_2 , in the same way as in the previous two parts, we obtain:

For large m_2 :

$$C_{4.2.2}(\cdot) \geq \frac{(1 - \beta)^{k+1}}{k + 1} \cdot \left[0.9999 \cdot (1 - \delta) \cdot \frac{\tau}{\beta} \right] \quad (35)$$

$$C_{5.2.2}(\cdot) \geq \frac{(1 - \beta)^{k+1}}{k + 1} \cdot \left[0.9999 \cdot (1 - \delta) \cdot \frac{\tau}{\beta} \right] \quad (36)$$

$$C_{6.2.2}^{(2)}(\cdot) \geq \frac{(1 - \beta)^{k+1}}{k + 1} \cdot \left[0.9999 \cdot (1 - \delta) \cdot \frac{\tau}{\beta} \right] \quad (37)$$

We should also consider the case where both m and k are large, and determine which formula to examine. It is worth mentioning that we do not need to calculate the case where both m_2 and k are large separately, since m_2 being large implies that m is large, which simplifies all equations sufficiently to compute.

For large m and k :

$$C_{5.2.1}(\cdot) \geq 0.9999^2 \cdot \tau \cdot \ln\left(\frac{1}{\beta}\right) \quad (38)$$

$$C_{6.2.1}(\cdot) \geq 0.9999^2 \cdot \tau \cdot \ln\left(\frac{1}{\beta}\right) \quad (39)$$

Using all the above observations in Algorithm 2 establishes that the chosen parameters guarantee the competitive ratio, and thus the theorem holds. ◀

4 Random Order Secretary Problem (ROSP) with Prediction

We now analyze Algorithm 1 under a random-order arrival model, where each candidate – including the top predicted candidate \hat{i} – receives an independent arrival time $\hat{t} \sim \text{Unif}[0, 1]$, as in the classical secretary problem. This replaces the fixed arrival position β used in the COSP analysis of Section 3 and requires averaging all β -dependent expressions over $[0, 1]$. The algorithm itself is unchanged: it begins in prediction mode and switches to secretary mode upon encountering the first candidate whose prediction error exceeds θ .

As in the COSP analysis, the competitive ratio is obtained by dividing into seven cases depending on the relative positions of \hat{i} , the true optimum i^* , and members of the mistake set M . For each case, we reuse the corresponding COSP expression and replace every occurrence of β by the uniform variable \hat{t} , integrating the resulting bound over the appropriate range. When $\hat{t} < \tau$, the algorithm behaves as if the predicted-best candidate arrives during the sampling phase; when $\hat{t} \geq \tau$, we recover the COSP expressions averaged over $\hat{t} \in [\tau, 1]$. The seven resulting bounds are explicit, closed-form functions of the instance parameters (m, k, m_2) and of $(\theta, \tau, \gamma, \delta)$, and the supporting algebraic identities appear in the full version.

► **Theorem 2.** *With parameters $\theta = 0.63$, $\tau = 0.33$, $\gamma = 0.34$, and $\delta = 0.66$, and with $\hat{t} \sim \text{Unif}[0, 1]$, Algorithm 1 achieves competitive ratio*

$$\max\left\{0.221, \frac{1 - \epsilon}{1 + \epsilon}\right\}$$

against instances with multiplicative prediction error ϵ .

Proof Sketch. For each of the seven structural cases (as in Section 3), we integrate the COSP lower bound over the arrival time \hat{t} . The $\hat{t} < \tau$ part yields new expressions for early arrival of \hat{i} ; the $\hat{t} \geq \tau$ part is the COSP bound averaged over $[\tau, 1]$. For small values of (m, k, m_2) , we evaluate these expressions exactly; for large parameters, exponential terms such as $(1 - \hat{t})^m$ vanish and the same symbolic bounding strategy as in the COSP analysis applies. Combining these evaluations shows that the optimized parameters above guarantee a minimum ratio of 0.221 in the random-order setting, while prediction correctness yields the $(1 - \epsilon)/(1 + \epsilon)$ bound. Full technical details and all case analyses appear in the full version. ◀

5 Equivalence of Deterministic and Randomized Algorithms in the Classic Setting

In the random-arrival secretary model (ROSP), the arrival times themselves provide a source of randomness. As a consequence, any randomized algorithm can use this internal randomness, and therefore randomized and deterministic algorithms have equal power in this setting.

► **Theorem 3.** *In the secretary problem with predictions under the random-arrival model (ROSP), every randomized algorithm can be simulated by a deterministic algorithm with no loss in competitive ratio.*

Full details and a formal construction appear in the full version.

6 Concluding Remarks

In this paper, we studied the secretary problem with predictions under both random-order (ROSP) and chosen-order (COSP) arrivals. We presented a randomized algorithm that follows predictions when they are accurate, but switches to Dynkin’s classical rule when large deviation from prediction occur, achieving both consistency under accurate predictions and robust guarantees in the worst case. Our analysis shows that the algorithm attains competitive ratios of $\max\{0.221, (1 - \epsilon)/(1 + \epsilon)\}$ for ROSP and $\max\{0.262, (1 - \epsilon)/(1 + \epsilon)\}$ for COSP, improving the best-known bounds and going beyond previous barriers in these settings. Recall that ϵ is the largest multiplicative error in any prediction. We also proved that randomized and deterministic algorithms are equally powerful in the random-order model, thereby resolving a prior separation claim.

Future research could focus on improving the tightness of the bounds for both models, for instance by approaching the 0.33 upper limit known for ROSP or closing the gap between our 0.262 lower bound and the best upper bounds for COSP. Another line of work could be to extend these ideas to other variants such as the k -secretary problem or matroid and knapsack constraints. Since decision-makers often have more flexibility in real applications, exploring distributional scheduling of predicted candidates or calibrating algorithmic parameters from data may yield stronger guarantees in practice.

References

- 1 Gagan Aggarwal, Anupam Gupta, Xizhi Tan, and Mingfei Zhao. Multi-platform autobidding with and without predictions, 2025. doi:10.48550/arXiv.2502.19317.
- 2 Sabah al Binali. The competitive analysis of risk taking with applications to online trading. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1997.
- 3 Susanne Albers and Leon Ladewig. New results for the k -secretary problem. In *Proceedings of the International Symposium on Algorithms and Computation (ISAAC 2019)*, 2019.
- 4 Georgios Amanatidis, Evangelos Markakis, Christodoulos Santorinaios, Guido Schäfer, Panagiotis Tsamopoulos, and Artem Tsikiridis. Online budget-feasible mechanism design with predictions. *arXiv:2505.24624*, 2025.
- 5 Antonios Antoniadis, Christian Coester, Marek Eliás, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.
- 6 Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. Secretary and online matching problems with machine learned advice. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.
- 7 Antonios Foivos Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. Secretary and online matching problems with machine learned advice. *arXiv:2006.01026*, 2020.
- 8 C. J. Argue, Anupam Gupta, Marco Molinaro, and Sahil Singla. Robust secretary and prophet algorithms for packing integer programs. *arXiv:2112.12920*, 2021.
- 9 Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Prophet secretary: Surpassing the $1 - 1/e$ barrier. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, 2018.

- 10 Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. In *Proceedings of the 10th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2007.
- 11 Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.
- 12 Eric Balkanski, Will Ma, and Andreas Maggiori. Fair secretaries with unfair predictions. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.
- 13 Étienne Bamas, Andreas Maggiori, Lars Rohwedder, and Ola Svensson. Learning augmented energy minimization via speed scaling. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.
- 14 Étienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.
- 15 Ziyad Benomar and Vianney Perchet. Advice querying under budget constraint for online algorithms. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.
- 16 Hedyeh Beyhaghi, Negin Golrezaei, Renato Paes Leme, Martin Pál, and Balasubramanian Sivan. Improved revenue bounds for posted-price and second-price mechanisms. *Operations Research*, 2021.
- 17 Joan Boyar, Lene M. Favrholdt, and Kim S. Larsen. Online unit profit knapsack with predictions. *Algorithmica*, 2024. doi:10.1007/S00453-024-01239-Y.
- 18 Domagoj Bradac, Anupam Gupta, Sahil Singla, and Goran Zuzic. Robust algorithms for the secretary problem. In *Proceedings of the 11th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2020.
- 19 Alexander Braun and Sherry Sarkar. The secretary problem with predicted additive gap. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.
- 20 F. Thomas Bruss. A unified approach to a class of best choice problems with an unknown number of options. *The Annals of Probability*, 1984.
- 21 Archit Bubna and Ashish Chiplunkar. Prophet inequality: Order selection beats random order. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, 2023.
- 22 Niv Buchbinder, Kamal Jain, and Mohit Singh. Secretary problems via linear programming. *Mathematics of Operations Research*, 2014.
- 23 Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, 2010.
- 24 Zhisheng Chen. Ethics and discrimination in ai-enabled recruitment practices. *Humanities and Social Sciences Communications*, 2023.
- 25 Davin Choo and Chun Kai Ling. A short note about the learning-augmented secretary problem, 2024. doi:10.48550/arXiv.2410.06583.
- 26 Jose Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Recent developments in prophet inequalities. *SIGecom Exchanges*, 2019.
- 27 José R. Correa, Andrés Cristi, Laurent Feuilloley, Tim Oosterwijk, and Alexandros Tsigonias-Dimitriadis. The secretary problem with independent sampling. In *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2021.
- 28 José R. Correa, Raimundo Saona, and Bruno Ziliotto. Prophet secretary through blind strategies. *Mathematical Programming*, 2021.
- 29 Mohammadreza Daneshvaramoli, Helia Karisani, Adam Lechowicz, Bo Sun, Cameron N. Musco, and Mohammad Hajiesmaili. Near-optimal consistency–robustness trade-offs for learning-augmented online knapsack problems. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

- 30 Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, Ali Vakilian, and Nikos Zarifis. Learning online algorithms with distributional advice. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.
- 31 Paul Dütting, Silvio Lattanzi, Renato Paes Leme, and Sergei Vassilvitskii. Secretaries with advice. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, 2021.
- 32 E. B. Dynkin. Optimal choice of the stopping moment of a markov process. *Doklady Akademii Nauk SSSR*, 1963.
- 33 Hossein Esfandiari, MohammadTaghi Hajiaghayi, Vahid Liaghat, and Morteza Monemizadeh. Prophet secretary. In *Proceedings of the 23rd European Symposium on Algorithms (ESA)*, 2015.
- 34 Hossein Esfandiari, MohammadTaghi HajiAghayi, Brendan Lucier, and Michael Mitzenmacher. Prophets, secretaries, and maximizing the probability of choosing the best. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.
- 35 Moran Feldman, Joseph Naor, and Roy Schwartz. Improved competitive ratios for submodular secretary problems. In *Proceedings of the 14th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2011.
- 36 Kaito Fujii and Yuichi Yoshida. The secretary problem with predictions. *Mathematics of Operations Research*, 2023.
- 37 Martin Gardner. Mathematical games. *Scientific American*, 1960.
- 38 Matthias Gehnen, Henri Lotze, and Peter Rossmanith. Online simple knapsack with bounded predictions. In *Proceedings of the 41st International Symposium on Theoretical Aspects of Computer Science (STACS)*, 2024.
- 39 John P. Gilbert and Frederick Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 1966.
- 40 Vasilis Gkatzelis, Daniel Schoepflin, and Xizhi Tan. Clock auctions augmented with unreliable advice. *arXiv:2408.06483*, 2024.
- 41 Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.
- 42 Anupam Gupta and Sahil Singla. Random-order models. In *Beyond the Worst-Case Analysis of Algorithms*, 2021.
- 43 T. P. Hill. Prophet inequalities and order selection in optimal stopping problems. *Proceedings of the American Mathematical Society*, 1983.
- 44 Theodore P. Hill and Robert P. Kertz. A survey of prophet inequalities in optimal stopping theory. *Contemporary Mathematics*, 1992.
- 45 Sungjin Im, Ravi Kumar, Mahshid Montazer Qaem, and Manish Purohit. Online knapsack with frequency predictions. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021.
- 46 Haim Kaplan, David Naori, and Danny Raz. Competitive analysis with a sample and the secretary problem. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2020.
- 47 Anna R. Karlin and Eric Lei. On a competitive secretary problem. In *Proceedings of the AAAI Conference on Artificial (AAAI)*, 2015.
- 48 Thomas Kesselheim and Andreas Tönnis. Submodular secretary problems: Cardinality, matching, and linear constraints. In *Proceedings of the 20th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2017.
- 49 Robert D. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

- 50 Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP)*, 2009.
- 51 Ulrich Krengel and Louis Sucheston. On semimarts, amarts, and processes with finite value. In *Probability on Banach Spaces*, 1978.
- 52 Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Online scheduling via learned weights. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2020.
- 53 Adam Lechowicz, Nicolas Christianson, Bo Sun, Noman Bashir, Mohammad Hajiesmaili, Adam Wierman, and Prashant Shenoy. Online conversion with switching costs: Robust and learning-augmented algorithms. *SIGMETRICS Performance Evaluation Review*, 2024.
- 54 D. V. Lindley. Dynamic programming and decision theory. *Journal of the Royal Statistical Society: Series C*, 1961.
- 55 Xujun Liu, Olgica Milenkovic, and George V. Moustakides. Query-based selection of optimal candidates under the mallows model. *arXiv:2101.07250*, 2021.
- 56 Brendan Lucier. An economic view of prophet inequalities. *SIGecom Exchanges*, 2017.
- 57 Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.
- 58 Ali A. Mahmoud, Tahani AL Shawabkeh, Walid A. Salameh, and Ibrahim Al Amro. Performance predicting in hiring and appraisals using machine learning. In *International Conference on Information and Communication Systems (ICICS 2019)*, 2019.
- 59 Bakhtiyor Meraliyev, Balzhan Alibekova, and Islana Bekturganova. Machine learning as an effective tool for human resource management in the recruiting process. In *International Conference on Electronics, Computer and Computation (ICECCO 2023)*, 2023.
- 60 Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 2022.
- 61 G. V. Moustakides, X. Liu, and O. Milenkovic. Optimal stopping methodology for the secretary problem with random queries – addendum. *Journal of Applied Probability*, 2023.
- 62 Bo Peng and Zhihao Gavin Tang. Order selection prophet inequality: From threshold optimization to arrival time design. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2022.
- 63 Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.
- 64 Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT 2020)*, 2020.
- 65 Aaron Rieke and Miranda Bogen. Help wanted: An examination of hiring algorithms, equity, and bias, 2018.
- 66 Dhruv Rohatgi. Near-optimal bounds for online caching with machine learned advice. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2020.
- 67 Jad Salem and Swati Gupta. Secretary problems with biased evaluations. *Management Science*, 2023.
- 68 Dahlia Sam, M. Ganesan, S. Ilavarasan, and T. John Victor. Hiring and recruitment process using machine learning. In *International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF 2023)*, 2023.
- 69 Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for nonnegative iid variables. *Annals of Probability*, 1984.
- 70 Neha Sharma, Rigzen Bhutia, Vandana Sardar, Abraham P. George, and Farhan Ahmed. Novel hiring process using machine learning and nlp. In *IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT 2021)*, 2021.
- 71 Yuhang Shen and Xiaoyu Zhang. The impact of artificial intelligence on employment: The role of virtual agglomeration. *Humanities and Social Sciences Communications*, 2024.

86:24 Secretary Problem with Predictions and a Chosen Order

- 72 Bo Sun, Russell Lee, Mohammad Hajiesmaili, Adam Wierman, and Danny H. K. Tsang. Pareto-optimal learning-augmented algorithms for online conversion problems. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021.
- 73 Alexander Wei. Better and simpler learning-augmented online caching. In *Proceedings of the 23rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2020.
- 74 Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.