Ordinal Secretaries with Advice

Hasti Nourmohammadi[1], **Ying Cao**[2], **Bo Sun**[3], **Xiaoqi Tan**[1]
[1]University of Alberta, Alberta Machine Intelligence Institute
[2]Hong Kong University of Science and Technology
[3]University of Ottawa, Vector Institute
{hnourmoh, xiaoqi.tan}@ualberta.ca, ycaoan@connect.ust.hk, bo.sun@uottawa.ca

Abstract

We study the ordinal secretary problem, where a sequence of candidates arrives in uniformly random order, and the goal is to select the best candidate using only pairwise comparisons. We consider a learning-augmented setting that incorporates potentially erroneous predictions about the best candidate's position. Our goal is to design online algorithms that balance robustness against poor predictions while having high performance when predictions are accurate. Using an optimization-based framework, we develop deterministic and randomized algorithms that extend classical strategies and explicitly model the trade-off between consistency and robustness. Also, we show the flexibility of our approach by applying it to multiple secretary problem variants, including multiple-choice and rehiring.

1 Introduction

We study the *secretary problem*, a classical model of online decision-making under uncertainty, where the goal is to select the best candidate from a sequence of N candidates arriving in uniformly random order. The decision-maker observes candidates sequentially and must decide immediately and irrevocably whether to accept or reject each one. As a fundamental model in online decision-making and optimal stopping theory, the secretary problem has broad applications in online algorithms, multi-agent systems, and mechanism design, and has consequently inspired a rich body of work spanning theoretical computer science, artificial intelligence, economics, and operations research.

Depending on the decision-maker's knowledge of candidate valuations, the secretary problem is typically studied in either the *cardinal* or *ordinal* setting. In the *cardinal secretary* problem, the decision-maker observes a scalar value for each candidate and aims to maximize the total value obtained from the selection (Kesselheim et al. 2013). For example, in online selection or hiring scenarios (Jiang et al. 2021), a candidate's value may represent their qualification, and the goal is to choose the candidate with the highest qualification. In the ordinal setting, the decision-maker does not observe numerical values and instead only receives ordinal information, such as how the current candidate ranks relative to those previously seen. In the absence of value information, a natural objective is to maximize the probability of selecting the top-ranked candidate. This variant is commonly referred to as the *ordinal secretary* problem (Gilbert and Mosteller 2006; Gravin, Sun, and Tang 2023; Soto, Turkieltaub, and Verdugo 2021; Bérczi et al. 2025; Hoefer and Kodric 2017). Surprisingly, both versions of the secretary problem admit an optimal algorithm based on the well-known $\frac{1}{e}$ rule, which achieves the best possible competitive ratio of $\frac{1}{e}$. This $\frac{1}{e}$ strategy, commonly referred to as the *wait-and-accept* rule, rejects the first $\frac{1}{e}$ fraction of candidates and then selects the next candidate who exceeds all previously observed ones (Gilbert and Mosteller 2006; Lindley 1961).

While the secretary problem and its many variants have been extensively studied (Gilbert and Mosteller 2006; Ferguson 1989; Freeman 1983; Albers and Ladewig 2021; Stewart 1981), recent work has increasingly explored the integration of machine-learned predictions into online decision-making. This emerging research direction—often referred to as *algorithms with predictions* or *learning-augmented algorithms*—aims to strengthen classical approaches by leveraging side information to improve performance when predictions are accurate (a property known as *consistency*), while preserving *robustness* under arbitrarily poor predictions. Despite notable progress in this area (Fujii and Yoshida 2024; Antoniadis et al. 2023; Jiang et al. 2021; Braun and Sarkar 2024), most existing studies focus on the cardinal secretary setting with value-based predictions, leaving the ordinal secretary problem with predictions largely underexplored. This gap motivates our central question: *How does a prediction about the best candidate's position benefit an online algorithm in the ordinal setting?*

In the absence of value observations, we argue that incorporating *position-based predictions* (i.e., predictions about the position of the best candidate) is a natural and intuitive approach. However, to the best of our knowledge, the ordinal secretary problem with such predictions has not been previously explored. In this paper, we fill this gap and show that even coarse or noisy advice regarding the best candidate's position can substantially enhance algorithmic performance in the ordinal secretary problem and its extensions.

1.1 Our Contribution

We investigate a novel variant of the secretary problem, termed *ordinal secretaries with advice* (OSA), in a learning-augmented setting where predictions provide error-prone advice about the likely positions of strong candidates. Unlike prior studies that focus on predictions of adversarially chosen candidate values, our approach leverages position-based predictions aligned with the inherent randomness of the ordinal secretary problem and its variants. Our main contributions are summarized as follows.

We begin our study of OSA by providing a formal definition and showing how error-prone advice can be used to design a deterministic algorithm that achieves a Pareto-optimal tradeoff between consistency and robustness within the well-known class of wait-and-accept (WNA) algorithms. We then move beyond deterministic approaches and introduce a randomized algorithm that uses WNA as a subroutine. Specifically, we develop an optimization-based framework that models the performance of this meta-algorithm, enabling the systematic selection of its key parameters. This framework yields a randomized algorithm with provably tight guarantees for both consistency and robustness. We further extend the optimization-based framework to several variants of the ordinal secretary problem, obtaining state-of-the-art performance guarantees. These include the multiple-choice variant, which permits several selections, and a rehiring model, in which previously rejected candidates may become available again at the end of the process. Together, these results highlight the flexibility and generality of our approach across multiple extensions of the problem.

1.2 Related Work

The paradigm of *algorithms with predictions* enhances classical online algorithms by incorporating side information, often in the form of machine-learned predictions. This approach bridges the gap between worst-case guarantees and empirical performance, delivering strong results when predictions are accurate while maintaining robustness under adversarial errors (Purohit, Svitkina, and Kumar 2018; Bamas, Maggiori, and Svensson 2020). Among its many applications, caching has been a primary focus (Lykouris and Vassilvitskii 2021; Rohatgi 2019; Wei 2020), alongside extensive work on ski rental variants (Bhattacharya and Das 2022; Shin, Lee, and An 2023; Wang, Li, and Wang 2020). Non-clairvoyant job scheduling has also received considerable attention (Purohit, Svitkina, and Kumar 2018; Lindermayr and Megow 2022; Im et al. 2023; Benomar and Perchet 2024). In covering problems, (Bamas, Maggiori, and Svensson 2020) introduced a prediction-aware primal-dual method, later extended to non-linear settings by (Thang 2021). The framework has further been applied to knapsack problems (Gehnen, Lotze, and Rossmanith 2024; Im et al. 2021; Boyar, Favrholdt, and Larsen 2022; Zeynali et al. 2020), demonstrating its versatility and broad applicability. For a broader overview, we refer readers to (Roughgarden 2021; Mitzenmacher and Vassilvitskii 2021) and the community-maintained repository.[1]

As a fundamental model of decision-making under uncertainty, the secretary problem has recently attracted interest in this setting. In what follows, we review key developments on secretary problems with predictions in the two main settings: the *cardinal* and the *ordinal* settings.

Cardinal secretaries with predictions. As mentioned earlier, most existing studies on secretary with predictions have focused on the cardinal setting, where the algorithm receives predictions about candidate values as part of its input. Various approaches have been proposed, differing in the form and accuracy of these predictions. One line of work considers error-prone predictions of the maximum candidate value, as in (Antoniadis et al. 2023), which introduced a flexible framework that adjusts the classical competitive ratio from $\frac{1}{e}$ to $\frac{1}{ce}$ for any $c \geq 1$ by varying the length of the initial waiting phase. (Jiang et al. 2021) proposed a time-dependent threshold algorithm that incorporates an interval prediction containing the maximum value, refining the classical secretary rule. (Braun and Sarkar 2024) examined predictions of the additive gap between the highest and k-th highest values, while (Fujii and Yoshida 2024) considered predictions for all candidate values and developed an algorithm that adaptively integrates prediction-based selection with traditional waiting-based thresholds. It has been shown that perfect consistency (achieving a competitive ratio of 1 when predictions are correct) cannot be attained simultaneously with worst-case robustness of $\frac{1}{e}$ in certain settings (Choo and Ling 2024; Fujii and Yoshida 2024). (Dütting et al. 2021) further provided a unifying framework by introducing binary signals (i.e., "Yes" or "No") to indicate whether each candidate is optimal, thereby capturing a broad class of prediction models under a single formalism. More recently, (Balkanski, Ma, and Maggiori 2024) examined the cardinal setting and introduced algorithms that ensure a guaranteed level of fairness in selecting the best candidate, even in the presence of biased predictions.

Ordinal secretaries with predictions. In contrast to the cardinal setting, the ordinal setting has received considerably less attention in the literature on algorithms with predictions. In this context, (Benomar and Perchet 2023) introduced a model in which algorithms interact with an omniscient oracle through binary queries. The algorithm is permitted to issue at most B such queries, each asking whether any unseen candidates are better than those observed so far. The formulation and timing of these queries are crucial to the algorithm's overall performance. The work most closely related to ours is that of (Dütting et al. 2021), which studied a position-based prediction model. In their framework, each candidate is accompanied by a binary signal indicating whether they are the best candidate, with the signal being correct with probability p. Although their model shares similarities with ours in the type of information conveyed, it is structurally different because it attaches signals directly to candidates. As a result, their analytical techniques do not readily extend to our setting.

2 D-WNA: A Deterministic WNA Algorithm for OSA

In this section, we formally define OSA and introduce the well-known *wait-and-accept* (WNA) rule for the secretary problem. We then present a deterministic algorithm for OSA

[1] https://algorithms-with-predictions.github.io/

and establish its Pareto-optimality in the class of WNA rules.

2.1 Problem Statement of OSA

We study an ordinal secretary problem in which N candidates arrive sequentially in random order. In each round, the algorithm observes only the relative ranking of the current candidate compared to those previously encountered, hence the term "*ordinal*." The arrival order is drawn uniformly at random. Let p denote an error-prone prediction of the position of the best candidate, denoted by o. The objective is to maximize the probability of selecting the best candidate, i.e., the candidate at position o.

For an online algorithm, say ALG, we define its **consistency** η as

$$\eta = \mathbb{E}_{\sigma_p, p}\left[\frac{\mathbb{E}[\mathsf{ALG}(\sigma_p)]}{\mathsf{OPT}(\sigma_p)}\right],$$

and **robustness** γ as

$$\gamma = \mathbb{E}_{\sigma}\left[\frac{\mathbb{E}[\mathsf{ALG}(\sigma)]}{\mathsf{OPT}(\sigma)}\right],$$

where σ is a uniformly random input sequence, and σ_p denotes a random sequence in which the best candidate appears at position p. Here, OPT denotes the success probability of an omniscient algorithm that always selects the best candidate. Following standard convention, *consistency* measures an algorithm's expected performance relative to OPT when the input sequence aligns with the prediction. Specifically, η is computed by averaging over all possible predicted positions p and their corresponding permutations σ_p. In contrast, *robustness* captures the algorithm's average-case performance over all input sequences, regardless of any prediction. Since $\mathsf{OPT} = 1$ in our setting—representing the success probability of an oracle with complete information—both consistency and robustness simplify to:

$$\eta = \mathbb{E}_{\sigma_p, p}\left[\mathbb{E}[\mathsf{ALG}(\sigma_p)]\right], \quad \gamma = \mathbb{E}_{\sigma}\left[\mathbb{E}[\mathsf{ALG}(\sigma)]\right],$$

where the inner expectations, $\mathbb{E}[\mathsf{ALG}(\sigma_p)]$ and $\mathbb{E}[\mathsf{ALG}(\sigma)]$, are taken with respect to the internal randomness of the algorithm ALG.

2.2 WNA(x): The Wait-and-Accept Rule

A well-known class of strategies for the secretary problem involves first observing candidates up to a predetermined position and then selecting the next candidate who is better than all previously observed ones. We refer to this class of algorithms as the *wait-and-accept* rule, denoted by WNA (Gilbert and Mosteller 2006).

Definition 1 (The Wait-and-Accept Algorithm: WNA(x)). *We define* WNA(x) *as a subclass of wait-and-accept algorithms that operate as follows: the algorithm observes the first x candidates without making any selections. Beginning from position $x+1$, it selects the first candidate who is better than all previously observed ones. If no such candidate is found by position $N-1$, the algorithm selects the final candidate at position N by default.*

Definition 2 (Probability of Selecting the Best: $\Phi_N(x)$). *Let $\Phi_N(x)$ denote the probability that* WNA(x) *selects the best candidate when there are N candidates in total. That is, $\Phi_N(x)$ is the probability that the algorithm successfully selects the candidate with the highest rank under the strategy* WNA(x).

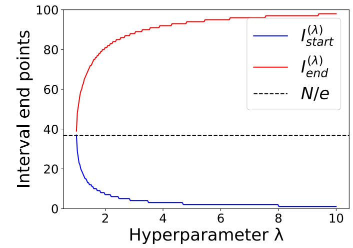


Figure 1: Illustration of the $I_{\text{start}}^{(\lambda)}$ and $I_{\text{end}}^{(\lambda)}$ as a function of λ when $N = 100$.

If we run WNA(r), the value of $\Phi_N(r)$ is $\frac{r}{N}\ln\frac{N}{r}$ asymptotically (Gilbert and Mosteller 2006). Maximizing $\Phi_N(x)$ without any predictions yields the classical solution $x = \frac{N}{e}$, achieving a $\frac{1}{e}$ probability of selecting the best candidate.

2.3 D-WNA: A Deterministic WNA Algorithm

Based on WNA(x) and the robustness parameter $\lambda \geq 1$, we propose a deterministic algorithm for OSA in Algorithm 1 below.

Algorithm 1: Deterministic WNA for OSA (D-WNA)

Inputs: Predicted position of the best candidate p; robustness parameter $\lambda \geq 1$.
Set $I_{\text{start}}^{(\lambda)}$ and $I_{\text{end}}^{(\lambda)}$ based on Eq. (1).
if $p < I_{\text{start}}^{(\lambda)}$ **then**
  Run WNA $(\lceil \frac{N}{e} \rceil)$.
else if $p \in I$ **then**
  Run WNA $(p-1)$.
else if $p > I_{\text{end}}^{(\lambda)}$ **then**
  Run WNA $(I_{\text{end}}^{(\lambda)} - 1)$.
end if

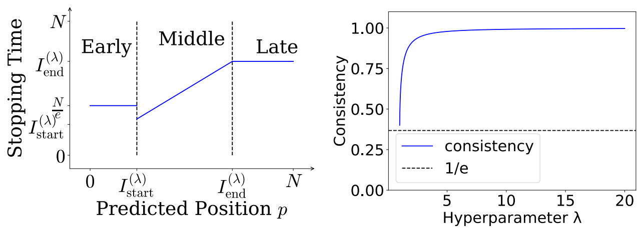
Algorithm 1 relies on two key parameters, $I_{\text{start}}^{(\lambda)}$ and $I_{\text{end}}^{(\lambda)}$ as functions of a given parameter of $\lambda \geq 1$, which are defined as follows:

$$I_{\text{start}}^{(\lambda)} = \left\lceil \exp\left(W_{-1}\left(\frac{-1}{\lambda e}\right)\right) \cdot N \right\rceil + 1, \tag{1a}$$

$$I_{\text{end}}^{(\lambda)} = \left\lfloor \exp\left(W_{0}\left(\frac{-1}{\lambda e}\right)\right) \cdot N \right\rfloor + 1. \tag{1b}$$

In the definition of $I_{\text{start}}^{(\lambda)}$ and $I_{\text{end}}^{(\lambda)}$, $W_{-1}(x)$ and $W_0(x)$ are the two solutions to $ye^y = x$, which are represented by the Lambert W function. Figure 1 illustrates the intuition behind how the increment of the parameter λ changes the values of $I_{\text{start}}^{(\lambda)}$ and $I_{\text{end}}^{(\lambda)}$.

Let I be defined as $I := [I_{\text{start}}^{(\lambda)}, I_{\text{end}}^{(\lambda)}] \subseteq [1, N]$. Algorithm 1 breaks the predictions to three different segments: (i) the



(a) Stopping Policy (b) Consistency of Algorithm 1

Figure 2: (a): Stopping time used by D-WNA as a function of the predicted position p. (b): Consistency of Algorithm 1 as a function of λ, with $N = 1000$.

predicted best candidate is early (i.e., $p \in [1, I^{(\lambda)}_{\text{start}})$), (ii) the predicted best candidate is somehow neither too early nor too late (i.e., $p \in I$), and finally, (iii) the predicted best candidate is late (i.e., $p \in (I^{(\lambda)}_{\text{end}}, N]$). When p falls outside I, it changes the duration of waiting in an asymmetric way for $p < I^{(\lambda)}_{\text{start}}$ and $p > I^{(\lambda)}_{\text{end}}$. Specifically, if $p < I^{(\lambda)}_{\text{start}}$, the algorithm does not use the prediction at all and maintains the waiting duration at the optimal waiting length of $\lfloor N/e \rfloor$ as established in the original setting of the secretary problem without predictions. When $p > I^{(\lambda)}_{\text{end}}$, the algorithm waits for the first $I^{(\lambda)}_{\text{end}}$ candidates and picks the next best so far.

Figure 2a visualizes how the stopping threshold evolves as a function of the predicted position of the best candidate, p. In the early segment (i.e., $p < I^{(\lambda)}_{\text{start}}$), the algorithm ignores the prediction and sticks to the classical stopping rule used when no prediction is available—namely, waiting for the first $\lfloor N/e \rfloor$ candidates before considering selections. The transition from the early to the middle segment is sharp: once p falls within the interval I, the stopping threshold begins to shift in response to the prediction to adapt its stopping point to align more closely with it. However, as p approaches the boundary of the late segment (i.e., $p = I^{(\lambda)}_{\text{end}}$), the threshold stabilizes and remains fixed thereafter. This design ensures robustness: the algorithm leverages informative predictions when available but avoids overcommitting to unreliable or extreme ones.

Here, we show the performance of the D-WNA algorithm in terms of its consistency and robustness.

Proposition 1. *For a given $\lambda \geq 1$, D-WNA is $(\frac{L^{(\lambda)}}{N} + \frac{1}{\lambda e})$-consistent and $\frac{1}{\lambda e}$-robust, where $L^{(\lambda)} = I^{(\lambda)}_{\text{end}} - I^{(\lambda)}_{\text{start}}$.*

Intuitively, as $\lambda \to \infty$ (i.e., full trust in the prediction), $L^{(\lambda)}$ expands to cover all N candidates (i.e., $L^{(\lambda)} \to N$), causing the consistency to approach 1. However, this comes at the cost of reduced robustness, since the algorithm effectively stops blindly based on the predicted value. Conversely, as $\lambda \to 1$ (i.e., no trust in the prediction), the interval length collapses to a single point at $\frac{N}{e}$ (i.e., $L^{(\lambda)} \to 0$), and both consistency and robustness converge to $\frac{1}{e}$, which is the optimal competitive ratio in the absence of predictions.

2.4 Pareto-Optimality of D-WNA

Now we show that Algorithm 1 is Pareto-optimal for OSA in the class of all possible WNA algorithms, namely, by fixing one metric of robustness or consistency the other could not get improved.

Proposition 2. D-WNA *achieves the Pareto-optimal trade-off between consistency and robustness over all deterministic algorithms in the class of* WNA *algorithms for* OSA.

Figure 2b compares the consistency of Algorithm 1 with the classical best competitive ratio for this problem. As λ increases (i.e., we place more trust in the prediction), the consistency approaches 1, whereas as λ decreases toward 1 (i.e., no trust in the prediction), it recovers the $\frac{1}{e}$ consistency.

3 R-WNA-ψ: A Randomized WNA Algorithm for OSA

In this section, we develop a randomized algorithm for OSA. We begin by introducing a meta-algorithm that leverages the WNA rule as a subroutine. We then demonstrate how to optimally design the associated parameters to balance consistency and robustness.

3.1 R-WNA-ψ: A Meta-Algorithm

Algorithm 1 selects the stopping point deterministically based on the prediction p. In contrast, the key idea behind the randomized algorithm is to sample the stopping point from a probability distribution that depends on p. This approach is formalized in Algorithm 2.

Algorithm 2: Randomized WNA for OSA (R-WNA-ψ)

1: **Inputs:** Prediction of the position of the best candidate p; probability density function $\psi_p(t) \to [0, 1]$.
2: With probability of $\psi_p(t)$, run WNA$(t-1)$.

In Algorithm 2, given a prediction p, a family of probability density functions $\psi_p(t)$ is used to determine the stopping behavior of the algorithm. Specifically, $\psi_p(t)$ denotes the probability density of stopping at time t, which corresponds to beginning the selection phase at candidate positioned at t (i.e., starting to consider best-so-far candidates from position t onward). In the discrete setting, this means stopping at candidate $t - 1$. Our objective is to determine the optimal design of $\psi_p(t)$ that balances consistency and robustness for each possible prediction value p.

3.2 Optimal Design of ψ_p Functions and the Consistency-Robustness Trade-off

From this point onward, we transition from the discrete to a continuous setting for ease of analysis and presentation. This transformation is without loss of generality, as the candidate indices $\{1, \ldots, N\}$ can be scaled to the unit interval $[0, 1]$. Correspondingly, the prediction p, the stopping point, and the associated functions $\psi_p(t)$ are redefined to operate over this continuous domain.

For notational simplicity, we continue to denote the scaled values $\exp\left(W_{-1}\left(-\frac{1}{\lambda e}\right)\right)$ and $\exp\left(W_0\left(-\frac{1}{\lambda e}\right)\right)$ by $I^{(\lambda)}_{\text{start}}$ and

$I_{\text{end}}^{(\lambda)}$, respectively. Accordingly, we use $I = [I_{\text{start}}^{(\lambda)}, I_{\text{end}}^{(\lambda)}]$ to represent the trusted prediction interval in the continuous setting. Furthermore, we use δ to denote the Dirac delta function, which is zero everywhere except for an infinite spike at a single point (e.g., $t = 0$).

Theorem 1. *For any $\lambda \geq 1$, define ψ_p as*

$$\psi_p(t) = \begin{cases} A^{(\lambda)}\delta(t-p) + B^{(\lambda)}\delta\left(t - \frac{1}{e}\right) & \text{if } p < I_{start}^{(\lambda)}, \\ \delta(t-p) & \text{if } p \in I, \\ \delta\big(t - I_{end}^{(\lambda)}\big) & \text{if } p > I_{end}^{(\lambda)}, \end{cases} \tag{2}$$

where $A^{(\lambda)} = \frac{\frac{1}{e}\left(1-\frac{1}{\lambda}\right)}{\frac{1}{e}-p\ln\frac{1}{p}}$ and $B^{(\lambda)} = 1 - \frac{\frac{1}{e}\left(1-\frac{1}{\lambda}\right)}{\frac{1}{e}-p\ln\frac{1}{p}}$. The R-WNA-*$\psi$ algorithm with ψ described in Eq.* (2) *is $\frac{1}{\lambda e}$-robust and $\eta(\lambda)$-consistent, where $\eta(\lambda)$ is given by*

$$\eta(\lambda) = \int_{t=0}^{I_{start}^{(\lambda)}} \frac{\frac{1}{e}(1-\frac{1}{\lambda})}{\frac{1}{e} - t\ln\frac{1}{t}} dt + L^{(\lambda)} + \frac{1}{\lambda e}. \tag{3}$$

Recall that here $L^{(\lambda)} = I_{end}^{(\lambda)} - I_{start}^{(\lambda)}$, defined in Eq. (1)*.*

Equation (2) provides the optimal design of the ψ_p functions for different prediction values. In this context, the Dirac delta function is used to represent probability mass concentrated at a single point within the continuous framework. This formulation is particularly useful for translating cumulative probability distributions $\psi_p(t)$ into discrete algorithmic decisions. For instance, when $p \in I$, our design specifies $\psi_p(t) = \delta(t-p)$, which, after appropriate scaling, corresponds to deterministically stopping at position $p-1$ in the discrete setting. In other words, the algorithm begins accepting candidates from position p, consistent with the intuition underlying wait-and-accept strategies.

In what follows, we derive the ψ_p functions in Eq. (2) using an optimization-based approach and present the proof of the resulting performance guarantees for Algorithm 2 under this optimal construction.

3.3 Proof of Theorem 1

We begin by presenting an overview of our proof. For each predicted position p, we formulate an optimization problem with the objective of maximizing the probability of selecting the best candidate, under the assumption that p is the predicted location of the optimal candidate. Simultaneously, we impose a robustness constraint to ensure competitive performance compared to $\texttt{CR}^*$, even when the prediction is inaccurate. Here, $\texttt{CR}^*$ denotes the best competitive ratio achievable in the absence of any prediction (Gilbert and Mosteller 2006). In the discrete setting, for a given stopping point x and correct prediction p, the probability of selecting the best candidate is $\frac{x}{p-1}$, while the robustness is approximately given by $\frac{x}{N}\ln\frac{N}{x}$ (Gilbert and Mosteller 2006). In Algorithm 2, the stopping point is chosen randomly according to a distribution $\psi_p(t)$, where $t \in [0,1]$ denotes the normalized stopping position. For a fixed $t \leq p$, the success probability is $\psi_p(t) \cdot \frac{t}{p}$, and the robustness is $\psi_p(t) \cdot t\ln\frac{1}{t}$. For $t > p$, the success probability is zero, as the algorithm would have already passed the best candidate. Therefore, the performance of Algorithm 2 can be modeled using $\psi_p(t)$, yielding the following optimization problem:

$$\max_{\psi_p(t)} \quad \int_{t=0}^{p} \psi_p(t) \cdot \frac{t}{p}\, dt \tag{4a}$$

$$\text{s.t.} \quad \int_{t=0}^{1} \psi_p(t) \cdot t\ln\frac{1}{t}\, dt \geq \frac{1}{\lambda e}, \tag{4b}$$

$$\int_{t=0}^{1} \psi_p(t)\, dt = 1, \tag{4c}$$

$$\psi_p(t) \geq 0, \quad \forall t \in [0,1]. \tag{4d}$$

The objective function (4a) represents the *consistency* of Algorithm 2, while the constraint (4b) enforce the *robustness* requirement, and the rests ensure that ψ_p is a valid probability distribution. To ensure robustness, we solve Problem (4) for each prediction value p, guaranteeing a robustness of at least $\frac{1}{\lambda e}$ through the imposed constraint. The remaining task is to determine the optimal design of the stopping distributions $\psi_p(t)$ for each p. We now justify the construction of $\psi_p(t)$ as defined in Eq. (2), by analyzing the solution to the above optimization problem in three prediction regimes:

- **Prediction is too early** (i.e., $p < I_{\text{start}}^{(\lambda)}$): In this case, the objective function and the robustness constraint imply that $\psi_p(t) = 0$ for all $t < p$, as selecting earlier provides no consistency gain. To maximize consistency, we concentrate as much probability mass as possible at $t = p$, subject to satisfying the robustness constraint. Since $t\ln\frac{1}{t}$ achieves its maximum at $t = \frac{1}{e}$, it is optimal to place the remaining mass at $t = \frac{1}{e}$. That is, the support of $\psi_p(t)$ consists of exactly two points: $t = p$ and $t = \frac{1}{e}$. Solving the equation

$$\psi(p) \cdot p\ln\frac{1}{p} + (1-\psi(p)) \cdot \frac{1}{e} \geq \frac{1}{\lambda e}$$

yields the necessary probability weights, where $\psi(p)$ denotes the weight at $t = p$. The normalization condition is satisfied using Dirac delta functions.

- **Prediction lies within the interval** (i.e., $p \in [I_{\text{start}}^{(\lambda)}, I_{\text{end}}^{(\lambda)}]$): Here, the objective $\int_0^p \psi_p(t) \cdot \frac{t}{p} dt$ is maximized when all probability mass is placed at $t = p$, since $\frac{t}{p}$ increases with t. Thus, the optimal solution is:

$$\psi_p(t) = \delta(t-p),$$

which corresponds to always choosing $p-1$ as the stopping point. This solution also satisfies the robustness constraint, as $p \in I$ was designed to ensure feasibility.

- **Prediction is too late** (i.e., $p > I_{\text{end}}^{(\lambda)}$): In this case, we set $\psi_p(t) = 0$ for all $t > p$, as selecting beyond p yields no consistency benefit and insufficient robustness contribution. Similarly, for $t < \frac{1}{e}$, the contribution to the robustness constraint is suboptimal. Thus, we restrict $\psi_p(t)$ to the interval $[\frac{1}{e}, p]$. We now show that the optimal solution is $\psi_p(t) = \delta(t - I_{\text{end}}^{(\lambda)})$. First, this function is feasible and makes the robustness constraint tight. Consider alternative distributions with support strictly to the left of $t = I_{\text{end}}^{(\lambda)}$

(i.e., $\psi_p(t) = 0$ for $t \in (I_{\text{end}}^{(\lambda)}, p]$). While the robustness constraint remains satisfied by construction of $I_{\text{end}}^{(\lambda)}$, the objective decreases since $\frac{t}{p}$ is increasing. On the other hand, distributions with all mass on $(I_{\text{end}}^{(\lambda)}, p]$ are infeasible, as $t \ln \frac{1}{t} < 1/\lambda e$ in this interval. Distributions spreading mass across both sides of $I_{\text{end}}^{(\lambda)}$ are also suboptimal: since $t \ln \frac{1}{t}$ has increasing negative slope in $[\frac{1}{e}, p]$, and $\frac{t}{p}$ increases linearly, shifting mass toward the right (i.e., $I_{\text{end}}^{(\lambda)}$) improves both robustness and consistency. Therefore, the optimal solution is:

$$\psi_p(t) = \delta(t - I_{\text{end}}^{(\lambda)}).$$

To compute consistency, assume the prediction p is correct, and recall that under the random arrival model, each p is equally likely with probability $1/N$. Let π_p^* denote the optimal objective value for each p from Problem (4). Then:

$$\begin{aligned}\eta &= \frac{1}{N}\sum_{p=1}^{N} \pi_p^* \approx \int_{t=0}^{1} \pi_t^* dt \\ &= \int_{t=0}^{I_{\text{start}}^{(\lambda)}} \pi_t^* dt + \int_{t=I_{\text{start}}^{(\lambda)}}^{I_{\text{end}}^{(\lambda)}} \pi_t^* dt + \int_{t=I_{\text{end}}^{(\lambda)}}^{1} \pi_t^* dt \\ &= \int_{t=0}^{I_{\text{start}}^{(\lambda)}} \frac{\frac{1}{e}\left(1-\frac{1}{\lambda}\right)}{\frac{1}{e} - t\ln\frac{1}{t}} dt + L^{(\lambda)} + \frac{1}{\lambda e}.\end{aligned}$$

The first term on the right-hand side of the above equation represents the consistency gain from randomization when the prediction is early (i.e., $p < I_{\text{start}}^{(\lambda)}$), a scenario in which the deterministic algorithm D-WNA fails to select the best candidate. The remaining terms match those from D-WNA, as the randomized strategy behaves identically in the trusted and late prediction regions. We thus complete the proof of Theorem 1.

Before concluding this section, we plot the consistency–robustness curve of Algorithm 2 in Figure 3, constructed using the optimal stopping distributions $\psi_p(t)$ from Eq. (2), across various values of λ. Figure 3 shows this trade-off curve alongside that of the deterministic baseline (Algorithm 1), illustrating how the randomized algorithm improves the trade-off—especially in scenarios where early predictions would otherwise be disregarded.

The two curves intersect at $\lambda = 1$, where both algorithms achieve the worst-case competitive ratio of $1/e$, matching the classical secretary bound without advice. For $\lambda > 1$, the randomized algorithm better exploits predictions, improving the trade-off compared to the deterministic counterpart.

4 Extensions: Multi-Choice and Rehiring

In this section, we demonstrate that the position-based advice model extends naturally to several generalizations of the OSA setting, including the multiple-choice variant and the secretary with rehiring problem.

4.1 Multi-Choice Secretary

Consider the multiple-choice secretary problem, where the algorithm is allowed up to m selections and the objective is

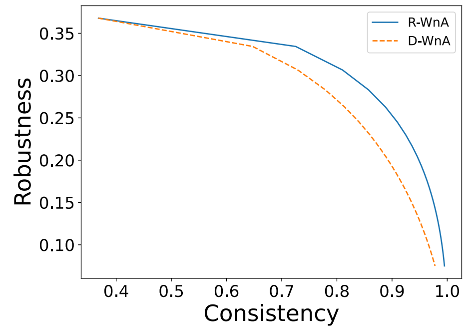


Figure 3: Illustration of the consistency-robustness curve of Algorithm 2 with the design of ψ_p in Eq. (2) compared to Algorithm 1.

to maximize the probability that the best candidate is among the chosen ones. When $m = 1$, this reduces to the classical secretary problem. We extend the OSA setting to this m-choice variant and focus on a strategy that waits until the t-th candidate and then selects the next m candidates who are best-so-far. More precisely, after the stopping point of t each candidate is checked for being best-so-far or not and if it is we pick it till we have picked at most m candidates or all N candidates are observed. We continue to refer to this class of strategies as WNA(t).

It has been shown that the optimal competitive ratio in the m-choice secretary problem using the WNA class of algorithms is $\texttt{CR}^* = \Phi_N(e^{-a^*} \cdot N) = e^{-a^*}\sum_{i=1}^{m}\frac{a^{*i}}{i!}$ where $a^* = (m!)^{\frac{1}{m}}$ (Gilbert and Mosteller 2006).

Analogous to the single-choice case, we introduce two key notations related to the stopping distribution. Let $I_{\text{start}}^{(\lambda)}$ and $I_{\text{end}}^{(\lambda)}$ be the two solutions to

$$x\sum_{i=1}^{m}\frac{\left(\ln\frac{1}{x}\right)^i}{i!} = \frac{1}{\lambda}\cdot\texttt{CR}^*,$$

where $I_{\text{start}}^{(\lambda)} < I_{\text{end}}^{(\lambda)}$. These endpoints define the trusted interval $I = [I_{\text{start}}^{(\lambda)}, I_{\text{end}}^{(\lambda)}]$, which specifies the range of prediction values considered informative. We also define $L^{(\lambda)} = I_{\text{end}}^{(\lambda)} - I_{\text{start}}^{(\lambda)}$ as the length of this interval.

The following theorem presents the design of the randomized algorithm for m-choice OSA, extending the same intuition used in the single-choice case (i.e., $m = 1$).

Theorem 2. *For any $\lambda \geq 1$, define ψ_p as*

$$\psi_p(t) = \begin{cases} A^{(\lambda)}\delta(t-p) + B^{(\lambda)}\delta\left(t - e^{-a^*}\right) & \textit{if } p < I_{start}^{(\lambda)}, \\ \delta(t-p) & \textit{if } p \in I, \\ \delta\left(t - I_{end}^{(\lambda)}\right) & \textit{if } p > I_{end}^{(\lambda)}, \end{cases}$$

where $A^{(\lambda)} = \frac{CR^*\cdot\left(1-\frac{1}{\lambda}\right)}{CR^* - p\sum_{i=1}^{m}\frac{(\ln\frac{1}{p})^i}{i!}}$ *and* $B^{(\lambda)} = 1 - \frac{CR^*\cdot\left(1-\frac{1}{\lambda}\right)}{CR^* - p\sum_{i=1}^{m}\frac{(\ln\frac{1}{p})^i}{i!}}$. *The* R-WNA-$\psi$ *algorithm is* $\frac{1}{\lambda}CR^*$*-robust*

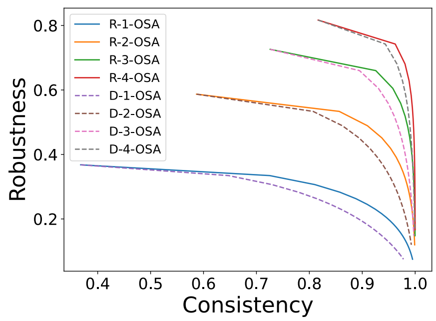


Figure 4: Consistency-robustness trade-off curves for both randomized and deterministic variants of R-WNA-ψ, shown for $1 \leq m \leq 4$. The randomized variant, labeled R-m-OSA, corresponds to the design of ψ given in Theorem 2, while the deterministic variant, labeled D-m-OSA, uses the same design of the ψ function for $I^{(\lambda)}_{\text{start}} \leq p$, and rolls back to the original setting with no prediction when $p < I^{(\lambda)}_{\text{start}}$.

and $\eta(\lambda)$-consistent, where $\eta(\lambda)$ is given by

$$\eta(\lambda) = \int_{t=0}^{I^{(\lambda)}_{start}} \frac{CR^* \cdot (1-\frac{1}{\lambda})}{CR^* - t\sum_{i=1}^{m} \frac{(\ln \frac{1}{t})^i}{i!}} dt + L^{(\lambda)} + \frac{1}{\lambda} CR^*.$$

Figure 4 illustrates the consistency–robustness trade-off curves for different values of m in Algorithm 2, using the stopping distribution design of ψ_p from Theorem 2. As expected, the trade-off improves as m increases; that is, the probability of selecting the best candidate becomes higher with more available choices. Moreover, for any fixed value of m, the randomized algorithm outperforms its deterministic counterpart by offering a non-zero probability of selecting the best candidate even when it arrives early in the sequence.

4.2 Secretary with Rehiring

This variant, studied in (Buchbinder, Jain, and Singh 2014), introduces the possibility of *rehiring*, where the decision maker is allowed to select the best candidate with probability q if no selection was made during the process. In other words, with probability q, the best candidate remains available after all candidates have been observed, allowing it to be chosen if no prior selection was made. It has been shown that the competitive ratio in this setting is $\texttt{CR}^* = e^{-(1-q)}$ (Buchbinder, Jain, and Singh 2014). We now demonstrate how this extension can be incorporated into our optimization framework.

To state the theorem precisely, we begin by defining the interval and parameters used, following the same structure as in the previous subsection. Let $\lambda \geq 1$ be the robustness parameter and p be the prediction of the best candidate's position. To identify the interval in which the algorithm partially trusts predictions, we define the function $g(x) = x \ln \frac{1}{x} + qx$, where $q \geq 0$ represents the probability of being able to rehire the best candidate if no selection is made during the process. Let x_1 and x_2 be the two solutions to the equation

$$g(x) = \frac{1}{\lambda} \cdot \texttt{CR}^*.$$

We define $I^{(\lambda)}_{\text{start}} = x_1$, $I^{(\lambda)}_{\text{end}} = \min(x_2, 1)$, and $L^{(\lambda)} = I^{(\lambda)}_{\text{end}} - I^{(\lambda)}_{\text{start}}$. The interval $I = [I^{(\lambda)}_{\text{start}}, I^{(\lambda)}_{\text{end}}]$ identifies the prediction range in which the algorithm adjusts its behavior to trade off between consistency and robustness.

We now present the main result in the following theorem.

Theorem 3. *For any $\lambda \geq 1$, define ψ_p as*

$$\psi_p(t) = \begin{cases} A^{(\lambda)}\delta(t-p) + \\ \quad B^{(\lambda)}\delta\left(t - e^{-(1-q)}\right) & \textit{if } p < I^{(\lambda)}_{start}, \\ \delta(t-p) & \textit{if } p \in I, \\ \delta\left(t - I^{(\lambda)}_{end}\right) & \textit{if } p > I^{(\lambda)}_{end}, \end{cases}$$

where $A^{(\lambda)} = \frac{CR^* \cdot (1-\frac{1}{\lambda})}{CR^* - p(\ln \frac{1}{p} + q)}$ and $B^{(\lambda)} = 1 - \frac{CR^* \cdot (1-\frac{1}{\lambda})}{CR^* - p(\ln \frac{1}{p} + q)}$. The R-WNA-$\psi$ algorithm is $\frac{1}{\lambda}CR^*$-robust and $\eta(\lambda)$-consistent, where $\eta(\lambda)$ is given by

$$\eta(\lambda) = \int_{t=0}^{I^{(\lambda)}_{start}} \left(q + (1-q) \frac{CR^* \cdot (1-\frac{1}{\lambda})}{CR^* - t \cdot (\ln \frac{1}{t} + q)} \right) dt + L^{(\lambda)} + \mathbb{1}_{\{\frac{e^{-(1-q)}}{q} > \lambda\}} \cdot \frac{1}{\lambda} CR^*.$$

Theorem 3 illustrates how the algorithm dynamically adjusts its strategy based on the strength of the rehiring mechanism. In the early stages, it balances random early selections with the potential to recover through rehiring. During the middle interval, it relies primarily on prediction-driven thresholding. When rehiring is weak, the algorithm incorporates a cautious late-stage decision phase to safeguard performance. However, as rehiring becomes stronger, this late phase vanishes, enabling the algorithm to delay decisions and depend on rehiring as a safety net. In the limiting cases, the absence of rehiring reduces the problem to the classical secretary setting, whereas near-certain rehiring supports a passive wait-and-rehire strategy that obviates the need for active stopping decisions.

5 Conclusions and Future Work

We initiated the study of the ordinal secretary problem with position-based predictions, a learning-augmented setting that incorporates predictions about the position of the best candidate rather than their value. Building on the classical wait-and-accept rule, our algorithms achieve a state-of-the-art balance between consistency and robustness, and extend naturally to variants such as the multiple-choice and rehiring secretary problems. Our findings also reveal several promising directions for future research. First, although our approach attains tight guarantees within the class of wait-and-accept algorithms, it remains unclear whether this trade-off is optimal when considering more general algorithmic families. Second, incorporating richer prediction signals (e.g., confidence scores) may enable more adaptive algorithms and support improved risk- or uncertainty-aware decision-making. Finally, our framework may extend beyond the current setting to problems such as online bipartite matching and prophet inequalities, where optimal stopping rules similarly play a central role.

Acknowledgments

Xiaoqi Tan acknowledges support from Alberta Machine Intelligence Institute (Amii), Alberta Major Innovation Fund, and NSERC Discovery Grant RGPIN-2022-03646.

References

Albers, S.; and Ladewig, L. 2021. New results for the k-secretary problem. *Theoretical Computer Science*, 863: 102–119.

Antoniadis, A.; Gouleakis, T.; Kleer, P.; and Kolev, P. 2023. Secretary and online matching problems with machine learned advice. *Discrete Optimization*, 48: 100778.

Balkanski, E.; Ma, W.; and Maggiori, A. 2024. Fair secretaries with unfair predictions. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24. Red Hook, NY, USA: Curran Associates Inc. ISBN 9798331314385.

Bamas, E.; Maggiori, A.; and Svensson, O. 2020. The primal-dual method for learning augmented algorithms. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Benomar, Z.; and Perchet, V. 2023. Advice Querying under Budget Constraint for Online Algorithms. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Benomar, Z.; and Perchet, V. 2024. Non-clairvoyant Scheduling with Partial Predictions. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 3506–3538. PMLR.

Bérczi, K.; Livanos, V.; Soto, J. A.; and Verdugo, V. 2025. Matroid secretary via labeling schemes. In *International Conference on Integer Programming and Combinatorial Optimization*, 128–141. Springer.

Bhattacharya, A.; and Das, R. 2022. Machine Learning Advised Ski Rental Problem with a Discount. In Mutzel, P.; Rahman, M. S.; and Slamin, eds., *WALCOM: Algorithms and Computation*, 213–224. Cham: Springer International Publishing. ISBN 978-3-030-96731-4.

Boyar, J.; Favrholdt, L. M.; and Larsen, K. S. 2022. Online Unit Profit Knapsack with Untrusted Predictions. In Czumaj, A.; and Xin, Q., eds., *18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022)*, volume 227 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 20:1–20:17. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-236-5.

Braun, A.; and Sarkar, S. 2024. The secretary problem with predicted additive gap. *arXiv preprint arXiv:2409.20460*.

Buchbinder, N.; Jain, K.; and Singh, M. 2014. Secretary Problems via Linear Programming. *Math. Oper. Res.*, 39(1): 190–206.

Choo, D.; and Ling, C. K. 2024. A short note about the learning-augmented secretary problem. *arXiv preprint arXiv:2410.06583*.

Dütting, P.; Lattanzi, S.; Paes Leme, R.; and Vassilvitskii, S. 2021. Secretaries with Advice. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, 409–429. New York, NY, USA: Association for Computing Machinery. ISBN 9781450385541.

Ferguson, T. S. 1989. Who Solved the Secretary Problem? *Statistical Science*, 4(3): 282 – 289.

Freeman, P. R. 1983. The Secretary Problem and Its Extensions: A Review. *International Statistical Review / Revue Internationale de Statistique*, 51(2): 189–206.

Fujii, K.; and Yoshida, Y. 2024. The Secretary Problem with Predictions. *Mathematics of Operations Research*, 49(2): 1241–1262.

Gehnen, M.; Lotze, H.; and Rossmanith, P. 2024. Online Simple Knapsack with Bounded Predictions. In Beyersdorff, O.; Kanté, M. M.; Kupferman, O.; and Lokshtanov, D., eds., *41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)*, volume 289 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 37:1–37:20. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-311-9.

Gilbert, J. P.; and Mosteller, F. 2006. *Recognizing the Maximum of a Sequence*, 355–398. New York, NY: Springer New York. ISBN 978-0-387-44956-2.

Gravin, N.; Sun, E.; and Tang, Z. G. 2023. Online Ordinal Problems: Optimality of Comparison-based Algorithms and their Cardinal Complexity. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, 1863–1876. IEEE.

Hoefer, M.; and Kodric, B. 2017. Combinatorial Secretary Problems with Ordinal Information. *CoRR*, abs/1702.01290.

Im, S.; Kumar, R.; Montazer Qaem, M.; and Purohit, M. 2021. Online Knapsack with Frequency Predictions. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 2733–2743. Curran Associates, Inc.

Im, S.; Kumar, R.; Qaem, M. M.; and Purohit, M. 2023. Non-clairvoyant Scheduling with Predictions. *ACM Trans. Parallel Comput.*, 10(4).

Jiang, Z.; Lu, P.; Tang, Z. G.; and Zhang, Y. 2021. Online Selection Problems against Constrained Adversary. In *International Conference on Machine Learning*.

Kesselheim, T.; Radke, K.; Tönnis, A.; and Vöcking, B. 2013. An Optimal Online Algorithm for Weighted Bipartite Matching and Extensions to Combinatorial Auctions. In Bodlaender, H. L.; and Italiano, G. F., eds., *Algorithms – ESA 2013*, 589–600. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-40450-4.

Lindermayr, A.; and Megow, N. 2022. Permutation Predictions for Non-Clairvoyant Scheduling. In *Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '22, 357–368. New York, NY, USA: Association for Computing Machinery. ISBN 9781450391467.

Lindley, D. V. 1961. Dynamic programming and decision theory. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 10(1): 39–51.

Lykouris, T.; and Vassilvitskii, S. 2021. Competitive Caching with Machine Learned Advice. *J. ACM*, 68(4).

Mitzenmacher, M.; and Vassilvitskii, S. 2021. *Algorithms with Predictions*, 646–662. Cambridge University Press.

Purohit, M.; Svitkina, Z.; and Kumar, R. 2018. Improving Online Algorithms via ML Predictions. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Rohatgi, D. 2019. *Near-Optimal Bounds for Online Caching with Machine Learned Advice*, 1834–1845.

Roughgarden, T. 2021. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press.

Shin, Y.; Lee, C.; and An, H.-C. 2023. On Optimal Consistency-Robustness Trade-Off for Learning-Augmented Multi-Option Ski Rental. *ArXiv*, abs/2312.02547.

Soto, J. A.; Turkieltaub, A.; and Verdugo, V. 2021. Strong Algorithms for the Ordinal Matroid Secretary Problem. *Math. Oper. Res.*, 46(2): 642–673.

Stewart, T. J. 1981. The Secretary Problem with an Unknown Number of Options.

Thang, N. K. 2021. Online Primal-Dual Algorithms with Predictions for Non-Linear Packing/Covering Problems.

Wang, S.; Li, J.; and Wang, S. 2020. Online Algorithms for Multi-shop Ski Rental with Machine Learned Advice. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 8150–8160. Curran Associates, Inc.

Wei, A. 2020. Better and Simpler Learning-Augmented Online Caching. In Byrka, J.; and Meka, R., eds., *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 60:1–60:17. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-164-1.

Zeynali, A.; Sun, B.; Hajiesmaili, M. H.; and Wierman, A. 2020. Data-driven Competitive Algorithms for Online Knapsack and Set Cover. *ArXiv*, abs/2012.05361.