Secretary Problem with Predictions and Ordering

Kang Yiming
Independent

Abstract

The classic secretary problem, a cornerstone of optimal stopping theory, assumes a random, immutable arrival order of candidates. While recent work has integrated machine-learned predictions to improve selection, the power to set the arrival order based on these predictions remains largely untapped. This paper introduces a novel framework for the secretary problem that leverages predictions for both valuation and strategic scheduling. We propose an algorithm that strategically controls the arrival time of the top-predicted candidate and dynamically adapts its hiring policy based on observed prediction accuracy. Our analysis shows that this approach achieves a worst-case competitive ratio of 0.229, surpassing the 0.215 bound of state-of-the-art algorithms that do not control ordering, bringing it closer to the upper bound of $1/e \approx 0.368$ while maintaining consistency guarantees. Furthermore, we demonstrate that our ordering framework can be adapted to improve fairness guarantees, doubling the success probability in a known fair algorithm from 1/16 to 1/8. Our results highlight that controlling the sequence is a powerful tool for building more robust and fair learning-augmented online algorithms.

1 INTRODUCTION

The secretary problem, renowned for its elegant solution and wide-ranging applications from hiring to resource allocation, fundamentally addresses the challenge of selecting the best option from a sequentially arriving set of candidates under uncertainty (Ferguson, 1989). The classic solution guarantees that the best candidate with probability 1/e is selected by observing a fraction of candidates and then selecting the first subsequent one that surpasses all previously seen. However, this assumes no prior knowledge and an immutable random arrival order.

In an era increasingly dominated by data-driven decision-making, the paradigm of learning-augmented algorithms has emerged, seeking to enhance classical algorithms with machine-learned predictions, first formalized in Lykouris and Vassilvitskii (2021). For the secretary problem, this has led to models that leverage predicted candidates' values to improve selection, often achieving performance that gracefully degrades with prediction error (Antoniadis et al., 2023). We quantify prediction error multiplicatively as $\epsilon := \max_{i \in \mathcal{N}} \left|\hat{v}_i / v_i - 1\right|$. Following the standard framework for learning-augmented design (Purohit et al., 2018; Lykouris and Vassilvitskii, 2021; Antoniadis et al., 2023), we target three desiderata:

- **Consistency:** For the multiplicative prediction error $\epsilon \geq 0$, the competitive ratio $\mathrm{CR}(\epsilon)$ approaches 1 as $\epsilon \to 0$ (near-optimality under perfect advice).
- **Robustness:** There exists a universal constant $c > 0$ such that $\mathrm{CR}(\epsilon) \geq c$ for all inputs, independent of the advice quality (i.e., even under arbitrarily bad predictions).
- **Smoothness:** The guarantee degrades gracefully with error; typically one obtains an interpolation bound of the form $\mathrm{CR}(\epsilon) \geq \max\{c, 1 - \alpha\,\epsilon\}$ for a problem-dependent constant $\alpha > 0$.

However, a critical modeling assumption remains largely unexamined: the arrival order of candidates is treated as random and exogenous. This contrasts with many real-world settings in which predictions are available upfront. For example, a hiring committee with preliminary scores (predictions) would rarely interview candidates in a uniformly random order; instead, it would strategically schedule the most promis-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

ing ones. Ignoring the ability to order based on predictions is a missed opportunity for substantial performance gains. We address this by introducing and analyzing the Secretary Problem with Predictions and Ordering. We argue that, when predictions are available, control over the interview schedule is a powerful algorithmic primitive. We formalize this control by allowing the algorithm to design the arrival distribution of the top-predicted candidate, and we develop an algorithm that leverages it to achieve improved robustness against adversarial prediction errors.

2 RELATED WORKS

2.1 Secretary problem

The secretary problem is a classic topic in optimal stopping theory. Dynkin (1963) formulated a simple yet optimal algorithm that identifies the best candidate with a success probability of at least 1/e. The problem's fundamental and elegant nature has led to extensive research in the field. In addition, its core principles have been adapted to more complex variations, such as the variants of k-secretary (Kleinberg, 2005), matroid-secretary(Babaioff et al., 2018), and knapsack-secretary (Babaioff et al., 2007).

2.2 Secretary problem with predictions

The integration of machine-learned predictions into the secretary problem aims to improve performance when predictions are of high quality while maintaining worst-case guarantees when prediction quality becomes arbitrarily low. While early work (Antoniadis et al., 2023) used a single prediction, recent work has focused on per-candidate predictions with multiplicative error models. An algorithm that focuses on the value maximization variant of the problem (Fujii and Yoshida, 2023) achieves a competitive ratio of at least $\max(0.215, 1 - O(\epsilon))$, where ϵ denotes multiplicative prediction error. Recent work (Balkanski et al., 2024) considers the fairness metric, defined as the probability of the algorithm selecting the best secretary, and achieves a competitive ratio of $\max(0.0625, 1 - O(\epsilon))$

2.3 Prophet secretary problem with order

The strategic use of ordering is a powerful tool in sequential selection. In the prophet secretary problem where candidates' values are drawn from a known distribution, a bound of 0.669 is known to hold for random arrivals (Correa et al., 2021), but this improves in settings with more structure. A model with full control over the arrival permutation was studied, achieving the competitive ratio of 0.725 (Peng and Tang, 2022). The question of whether selecting the order is provably better than a random order was recently settled (Bubna and Chiplunkar, 2023) They resolved this by establishing a separation between the two settings: their 0.7258-competitive algorithm for order selection beats the new 0.7254 hardness limit they proved for the random order setting. This body of research, however, differs from our model in its informational assumptions (known distributions vs. learned predictions) and the scope of control over the arrival sequence.

3 PRELIMINARIES AND PROBLEM FORMULATION

We formalize the problem of the secretary problem augmented with machine-learned predictions and the ability to order candidates.

Candidates and Values. We are given a set of n candidates, denoted by $\mathcal{N} = \{1, 2, \ldots, n\}$. Each candidate $i \in \mathcal{N}$ has a true, intrinsic value $v_i > 0$, which is unknown to the decision-maker beforehand. The best candidate is $i^* = \arg\max_{i \in \mathcal{N}} v_i$, with the optimal value $v_{i^*} = \max_{i \in \mathcal{N}} v_i$.

Predictions and Error Model. At the outset of the problem (time $t = 0$), we receive a vector of predicted values $\{\hat{v}_i\}_{i \in \mathcal{N}}$ from a machine learning model. We denote the candidate with the highest predicted value as $\hat{\imath} = \arg\max_{i \in \mathcal{N}} \hat{v}_i$.

Internal discrepancy threshold θ. We fix an algorithm parameter $\theta \in (0, 1)$ that controls when the algorithm distrusts a prediction and switches to a secretary-style fallback. Crucially, θ is *chosen by the algorithm*; it is not a bound on the true prediction error, and the adversary may produce arbitrarily large errors.

Given θ, we classify predictions as "bad" relative to this threshold:

$$M = \Big\{ i \in \mathcal{N} : \big|1 - \hat{v}_i / v_i\big| > \theta \Big\}.$$

The size of this set is $m = |M|$. This notation is used solely to analyze how the algorithm behaves once it decides a prediction is unreliable (i.e., when the observed discrepancy exceeds θ).

Arrival Process with Ordering. This is our key departure from the standard model. Candidates arrive sequentially over a continuous time interval $[0, 1]$. While most candidates arrive at random, the algorithm has the power to strategically schedule the top-predicted candidate, $\hat{\imath}$. The arrival times t_i are determined as follows:

- For all candidates $i \in \mathcal{N} \setminus \{\hat{\imath}\}$, the arrival time t_i is drawn independently and uniformly from $\mathcal{U}(0,1)$.
- The arrival time of the top-predicted candidate, $t_{\hat{\imath}}$, is drawn from a distribution T, which is designed by our algorithm as part of its strategy.

Upon a candidate's arrival at time t_i, their true value v_i is revealed. An immediate and irrevocable decision must be made to either hire or reject the candidate.

Objective. The goal is to design an algorithm that maximizes the competitive ratio, ϕ. This ratio is the expected value of the selected candidate, normalized by the value of the best candidate, minimized over all possible adversarial inputs (i.e., choices of values v_i and predictions $\hat{v}_i$).

$$\phi = \min_{\text{inputs } \{v_i, \hat{v}_i\}} \frac{\mathbb{E}[\text{value of selected candidate}]}{v_{i^*}}$$

A robust algorithm guarantees a high competitive ratio even in the worst case, where the predictions may be arbitrarily unhelpful.

4 ALGORITHM MOTIVATION

Our algorithm design is guided by the fundamental trade-off between *exploiting* potentially accurate machine-learned predictions and *hedging* against arbitrarily poor ones. To manage this trade-off, our algorithm operates in one of two states: a *Prediction Mode* and a *Secretary Mode*.

Prediction Mode. This is the default, optimistic state. Assuming the predictions are reliable, the best course of action is to select the candidate with the highest predicted value, $\hat{\imath}$. The primary goal of this mode is to hire $\hat{\imath}$ as soon as they arrive, capitalizing on the predictive information.

Secretary Mode. Predictions can be misleading. An adversary could provide predictions that identify a mediocre candidate as the best. If the algorithm observes evidence of such a mistake, it can no longer fully trust the predictive guidance. Upon detecting a significant prediction error, the algorithm transitions to a robust, prediction-agnostic *Secretary Mode*. This mode reverts to a classic Dynkin-style strategy.

The Power of Ordering: When to Interview $\hat{\imath}$? The crucial innovation of our work lies in how we leverage the power to order candidates, which introduces a new strategic dimension: *when should $\hat{\imath}$ be interviewed?* A fixed, deterministic choice for the arrival time of $\hat{\imath}$, denoted $t_{\hat{\imath}}$, is highly vulnerable. Consider the dilemma:

- **Scheduling $\hat{\imath}$ too early** is risky. If we hire a mediocre $\hat{\imath}$ at the beginning, we forfeit the opportunity to see potentially superior candidates who arrive later. The adversary can exploit this by making a poor candidate have the highest prediction.
- **Scheduling $\hat{\imath}$ too late** is also risky. A truly great candidate, i^*, might arrive before $\hat{\imath}$. If the algorithm is still in Prediction Mode (naively waiting for $\hat{\imath}$), it will incorrectly reject i^*, again leading to a suboptimal outcome.

A deterministic choice for $t_{\hat{\imath}}$ allows an adversary to craft an instance that perfectly exploits one of these two vulnerabilities. Our solution is to make the arrival time of $\hat{\imath}$ a random variable, with a distribution T that the algorithm strategically designs. We concentrate the probability at critical points $\tau, \tau+\delta, 1$ which allows us to analyze the different behavior effectively.

4.1 Why Only Place the Top-Predicted Candidate?

One may ask why we only control the arrival time for the top-predicted candidate $\hat{\imath}$, rather than reordering many or all candidates.

Two-candidate equivalence. When only the relative order of two candidates i and j matters, any reordering policy is summarized by a single number $p = \Pr(t_i < t_j)$. Controlling only t_i while drawing $t_j \sim \mathcal{U}(0,1)$ realizes any desired $p \in [0,1]$, since

$$\Pr(t_i < t_j) = 1 - \mathbb{E}[t_i].$$

Thus, for two candidates with order-only objectives, controlling one arrival is without loss of generality.

$n \geq 3$ candidates Our analysis will show that the worst case is achieved when $m = 1$, which is achievable with only 2 candidates. Hence, in our algorithm fixing the order for more candidates fail to improve the performance of the algorithm. This is largely due to the fact that in the worst case the predictions of the candidates can be arbitrarily bad and so the candidates after the first candidate are indistinguishable solely based off their predicted values.

5 ALGORITHM

Our algorithm, Ordered-Learned-Dynkin, uses a discrete probability distribution T for the arrival time

of $\hat{\imath}$:

$$t_{\hat{\imath}} \sim T = \begin{cases} \tau & \text{with probability } p_0 \\ \tau + \delta & \text{with probability } p_1 \\ 1 & \text{with probability } 1 - p_0 - p_1 \end{cases}$$

where τ is a time threshold analogous to the one in the classic secretary algorithm, and we set the slack δ to a vanishing sequence $\delta_s = s^{-2}$ (any s^{-k} with fixed $k \geq 2$ suffices) where asymptotics are taken as $s \to \infty$. This ensures threshold ties have probability $o(1)$ and leaves asymptotic guarantees unchanged. This three-point distribution allows us to balance different adversarial scenarios, as detailed in our analysis. Meanwhile, the arrival time of the rest of the candidates remain uniformly random. The algorithm not only tracks the top-predicted candidate, but also the second- and third-best predicted candidates, $\hat{\jmath}$ and $\hat{k}$ respectively, and uses these values to set a robust value threshold X if the prediction for a higher-ranked candidate proves to be erroneous.

6 ANALYSIS OF THE COMPETITIVE RATIO

We now prove the main result of our paper. The proof proceeds by providing a lower bound on the competitive ratio for several worst-case scenarios, defined by the relationship between the best candidate (i^*), the top-predicted candidate ($\hat{\imath}$), and the set of badly-predicted candidates (M). The final competitive ratio is the minimum of these bounds over all cases and all possible sizes of M.

Theorem 1. *Algorithm 1, with parameters* $\tau = 0.3887$, $p_0 = 0.5416$, $\theta = 0.627$, $p_1 = 0.08345$, *and* prob $= 0.5$, *achieves a competitive ratio of* $\max\{0.229, \frac{1-\epsilon}{1+\epsilon}\}$, *where* $\epsilon := \max_{i \in N} \left| \frac{\hat{v}(i)}{v(i)} - 1 \right|$.

In particular, when $\epsilon = 0$, the competitive ratio equals 1 (consistency). Moreover, the competitive ratio degrades smoothly as a function of ϵ.

Proof. The proof follows from the series of lemmas below, which analyze the algorithm's performance under disjoint adversarial settings. The competitive ratio ϕ is the minimum performance across these settings. □

Lemma 1 ($M = \emptyset$)**.** *The competitive ratio is* $\frac{1-\epsilon}{1+\epsilon}$ *when* $M = \emptyset$

Proof. In this case, since $\epsilon \leq \theta$, the mode of the algorithm is PREDICTION throughout the process, and therefore, the algorithm hires $\hat{\imath}$. From the definition of ϵ, we have $(1-\epsilon)v(i) \leq \hat{v}(i) \leq (1+\epsilon)v(i)$ for all

Algorithm 1 ORDERED-LEARNED-DYNKIN

Require: Time threshold $\tau \in (0,1)$, internal discrepancy threshold $\theta \in (0,1)$, threshold randomization parameter prob $\in [0,1]$, predictions $\hat{v} : \mathcal{N} \to \mathbb{R}$

Set arrival times:
$\hat{\imath} \leftarrow \arg\max_{k \in \mathcal{N}} \hat{v}_k$; $\hat{\jmath} \leftarrow \arg\max_{k \in \mathcal{N} \setminus \{\hat{\imath}\}} \hat{v}_k$; $\hat{k} \leftarrow \arg\max_{k \in \mathcal{N} \setminus \{\hat{\imath}, \hat{\jmath}\}} \hat{v}_k$.
Draw $t_{\hat{\imath}} \sim T$ where $T(\tau) = p_0$, $T(\tau + \delta) = p_1$, $T(1) = 1 - p_0 - p_1$; for each $k \neq \hat{\imath}$, draw $t_k \sim \mathcal{U}(0,1)$ independently.
Let π be the permutation that orders candidates by increasing t_i.
Initialize: mode $\leftarrow$ Prediction; internal threshold $X \leftarrow -\infty$; best-so-far value $B \leftarrow -\infty$.
for each candidate i in order π **do**
  Observe v_i and update $B \leftarrow \max\{B, v_i\}$.
  if $|1 - \hat{v}_i / v_i| > \theta$ **then**
    if $i = \hat{\imath}$ and mode = Prediction **then**
      $X \leftarrow (1 - \theta)\, \hat{v}_{\hat{\jmath}}$
    else if $i = \hat{\jmath}$ and X $> -\infty$ **then**
      Draw $u \sim \mathcal{U}(0,1)$.
      if $u \leq$ prob **then**
        $X \leftarrow (1 - \theta)\, \hat{v}_{\hat{k}}$
      else
        $X \leftarrow -\infty$
    else
      $X \leftarrow -\infty$
    mode $\leftarrow$ Secretary
  if mode = Prediction and $i = \hat{\imath}$ and $t_i \geq \tau$ **then**
    return i
  if mode = Secretary and $t_i > \tau$ and $v_i \geq B$ and $v_i \geq X$ **then**
    return i
return the last arriving candidate in π

$i \in N$. Therefore, the value obtained by the algorithm is bounded as follows.

$$v(\hat{\imath}) \geq \frac{1}{1+\epsilon}\hat{v}(\hat{\imath}) \geq \frac{1}{1+\epsilon}\hat{v}(i^*) \geq \frac{1-\epsilon}{1+\epsilon}v(i^*). \qquad (1)$$

Hence, the algorithm achieves the competitive ratio $\frac{1-\epsilon}{1+\epsilon}$. In this case, since $\epsilon \leq \theta = 0.627$, the competitive ratio is at least $\frac{1-\epsilon}{1+\epsilon} \geq \frac{1-0.627}{1+0.627} \geq 0.229$. □

We will now proceed to consider the case where $M \neq \emptyset$

Lemma 2 (Case 1 & 2: $\hat{\imath} = i^*$)**.** *The competitive ratio ϕ is bounded by:*

$$\phi \geq p_1 + (1 - p_0 - p_1)\tau$$

Proof. The analysis is identical whether $i^* \in M$ or $i^* \notin M$.

- If $T = \tau + \delta$, $\hat{\imath}$ will always be selected.
- If $T = 1$, $\hat{\imath}$ will be selected as long as the best element appearing before T also appears before τ. This occurs with a probability τ.

In total, the bound is as stated. □

Lemma 3 (Case 3: $\hat{\imath} \neq i^*$, $\hat{\imath} \in M$, $i^* \in M$)**.** *The total competitive ratio is:*

$$\begin{aligned}\phi \geq\ & p_0(1-\text{prob})\int_\tau^1 \frac{\tau}{t^*}\,\mathrm{d}t^* \\ & + (1-p_0-p_1)\int_\tau^1 \big(1-(1-t^*)^{m-2}\big)\frac{\tau}{t^*}\,\mathrm{d}t^* \\ & + (1-p_0-p_1)\int_\tau^1 (1-t^*)^{m-2}\,\mathrm{d}t^*\end{aligned}$$

Proof. Since $\hat{\imath} \in M$, the algorithm switches to the Secretary mode when observing $\hat{\imath}$. The algorithm hires i^* if $t^* \in [\tau, 1]$ and no candidate is hired before i^*. Hence, a sufficient condition for hiring i^* is that $t^* \in [\tau, 1]$ and the best candidate in $[0, t^*)$ appears before τ.

- **Case** $T = \tau$: Condition on $t^* \in [\tau, 1]$. A conservative guaranteed-success branch is when the line-12 randomization does *not* keep a restrictive internal threshold, which occurs with probability $(1 - \text{prob})$; in that branch we have $X = -\infty$ and recover the classical secretary condition. Therefore, conditioned on this branch, i^* is selected whenever the best element in $[0, t^*)$ arrives before τ, which happens with probability $\frac{\tau}{t^*}$. Integrating over t^* gives the contribution $p_0(1-\text{prob})\int_\tau^1 \frac{\tau}{t^*}\,\mathrm{d}t^*$.
- **Case** $T = 1$: There is a $1-(1-t^*)^{m-2}$ probability that an element in M appears before i^*, and in this case, a sufficient condition is that the best element before t^* appears before τ. In the other case, there is a probability of $(1-t^*)^{m-2}$ that i^* is the first element amongst all elements in M and hence will definitely be chosen when it arrives.

Weighting these by their probabilities (p_0 and $1-p_0-p_1$) yields the overall bound. □

Lemma 4 (Case 4: $\hat{\imath} \neq i^*$, $\hat{\imath} \in M$, $i^* \notin M$)**.** *The total competitive ratio for this case is the minimum of all cases below. For $m = 1$:*

$$\phi_{m=1} \geq p_0 \int_\tau^1 \left(\frac{\tau}{t^*} + \frac{t^*-\tau}{t^*}\frac{1-\theta}{1+\theta}\right)\mathrm{d}t^*$$

For $m = 2$, let $A = 1-(1-\tau)^{m-1}$, $B = (1-(1-t^*)^{m-1}) - A$, $C = 1-A-B$, where A is the probability that the first element in M arrives before τ, B is the probability that the first element in M arrives after τ but before t^*, and C be the probability that the first element in M arrives after t^*.

$$\begin{aligned}\phi_{m=2} \geq\ & p_0 \int_\tau^1 \Big(A\frac{\tau}{t^*} + B\frac{\tau}{t^*} \\ & \qquad + B\cdot\text{prob}\cdot\frac{t^*-\tau}{t^*}\frac{1-\theta}{1+\theta} + C\frac{\tau}{t^*}\Big)\mathrm{d}t^* \\ & + (1-p_0-p_1)\int_\tau^1 \big(1-(1-t^*)^{m-1}\big)\frac{\tau}{t^*}\,\mathrm{d}t^*\end{aligned}$$

For $m \geq 3$:

$$\begin{aligned}\phi_{m\geq 3} \geq\ & p_0 \int_\tau^1 \frac{\tau}{t^*}\mathrm{d}t^* \\ & + (1-p_0-p_1)\int_\tau^1 (1-(1-t^*)^{m-1})\frac{\tau}{t^*}\mathrm{d}t^*\end{aligned}$$

Proof. We consider values of m separately, and competitive ratio is the minimum across all values of m.

- **Case** $T = \tau$ **(prob** p_0**)**:
  - For $m = 1$: With the exception of $\hat{\imath}$, all predictions are close to the true value of each element. Hence, in secretary mode, either the algorithm picks the best element i^* with $\frac{\tau}{t^*}$ probability (when the best element before i^* appears before τ), or (if secretary fails) obtains a $\frac{1-\theta}{1+\theta}$-competitive element.
  - For $m = 2$: In addition to $\hat{\imath}$, another element has a true value that deviates significantly from its predicted value. If the other element is not $\hat{\jmath}$, then we are guaranteed to achieve a value that is as good as the case of $m = 1$,

since $i^* \notin M$, the true maximum value remains unchanged. Otherwise, in the case where $\hat{\jmath}$ is the other element in M, then $i^* \neq \hat{\jmath}$ and it is possible that $v_{i^*} < (1-\theta)\hat{v}(\hat{\jmath})$. The fallback $\frac{1-\theta}{1+\theta}$-competitive guarantee is available only when both $t_{\hat{\imath}} < t_{\hat{\jmath}} < t_{i^*}$ (event B) and the line-12 randomization succeeds, which contributes the factor prob.

  - For $m > 2$: For $m > 2$, we do not rely on the internal threshold to guarantee an element to be $\frac{1-\theta}{1+\theta}$ competitive.

- **Case** $T = 1$ **(prob** $1 - p_0 - p_1$**)**:
  - For $m = 1$: The competitive ratio is 0.
  - For $m > 1$: The bound is $(1-p_0-p_1)\int_\tau^1((1-(1-t^*)^{m-1})\frac{\tau}{t^*}dt^*$. The probability of the first element in M arriving before i^* is $(1-(1-t^*)^{m-1}$ and the probability that the best element arriving before i^* arrives before τ is $\frac{\tau}{t^*}$. Integrating over t^* gives the above integral.

□

Lemma 5 (Case 5: $\hat{\imath} \neq i^*$, $\hat{\imath} \notin M$, $i^* \in M$)**.** *The overall bounds are:*

$$\begin{aligned}\phi \geq (1-p_0-p_1)\Bigg(&\int_\tau^1 (1-t^*)^{m-1}\mathrm{d}t^* \\ &+ \int_\tau^1 (1-(1-t^*)^{m-1})\frac{\tau}{t^*}\mathrm{d}t^*\Bigg) \\ &+ p_0\left(\int_\tau^1 (1-(1-\tau)^{m-1})\frac{\tau}{t^*}\mathrm{d}t^*\right)\end{aligned}$$

Proof.
- **Case** $T = 1$: i^* is selected if and only if i^* is the first element to appear in M with probability $(1-t^*)^{m-1}$. Otherwise, the maximum element before i^* must arrive before τ with probability $\frac{\tau}{t^*}$.

- **Case** $T = \tau$: In this case, an element in M has to arrive before T, with probabilty $1-(1-\tau)^{m-1}$, and the best element arriving before i^* arrives before τ with probability $\frac{\tau}{t^*}$

Weighting by probabilities gives the overall bound. □

Lemma 6 (Case 6: $\hat{\imath} \neq i^*$, $\hat{\imath} \notin M$, $i^* \notin M$)**.** *The overall bounds are:*

$$\begin{aligned}\phi \geq\ & p_0\left(\int_\tau^1 \left((1-(1-t)^m)\frac{\tau}{t} + (1-t)^m\frac{1-\theta}{1+\theta}\right)\mathrm{d}t\right) \\ &+ (1-p_0-p_1)\left(\int_\tau^1 (1-(1-\tau)^m)\frac{\tau}{t}\right. \\ &\left.+(1-\tau)^m\frac{1-\theta}{1+\theta}\right)\mathrm{d}t\end{aligned}$$

Proof.
- **Case** $T = 1$: at t_{i^*}, there is a $1-(1-t)^m$ probability that $t_m < t_{i^*}$ and hence i^* is accepted if the best element occurring before i^*, i_b , is such that $t_{i_b} < \tau$. Otherwise, having missed i^*, it will select $\hat{\imath}$ instead as it will be forced to select the last element.

- **Case** $T = \tau$:
  - Define t_m as the time of arrival of the first element in M. Case $t_m < T < t^*$: Consider probability of i^* being selected. We must have $t_m < T$ to switch into *Secretary* mode and $t_{i_b} < \tau$ so that i_b is not selected.
  - Case $T < t_m$: In this case, $\hat{\imath}$ will be selected in *Prediction* mode.

- **For** $T = \tau + \delta$: For this case, we are able to generate bounds based off similar analysis as above cases. However, as the above cases are already enough for Case 6 to be bounded below by other cases, we can skip the analysis for this sub-case.

The overall bound combines these scenarios. □

6.1 Results and Worst-Case Analysis

We found the parameters $(\tau, p_0, p_1, \text{prob}, \theta)$ by performing a numerical grid search to maximize the competitive ratio across all adversarial cases. A set of hyperparameters were found to be $\tau = 0.3887$, $p_0 = 0.5416$, $p_1 = 0.08345$, prob $= 0.5$, and $\theta = 0.627$.

A critical step in this analysis is identifying the worst-case choice of m (the number of badly-predicted candidates) for the adversary. To determine this, we numerically evaluated the competitive ratio bound $\phi_k(m)$ for each case k over a range of $m \in [1, 50]$ and computed the analytical limit as $m \to \infty$.

The results are presented in Figure 1. The plot provides a clear visual proof of the behavior of the bounds:

- **Cases 1 & 2**: The bound is constant with respect to m.
- **Case 3**: The bound is monotonically decreasing, approaching its minimum as $m \to \infty$.
- **Cases 4, 5, & 6**: The bounds are shown to be concave or monotonically increasing for $m > 1$. Their minimum value for any given set of parameters is found at a small value of m (typically $m = 1, 2$, or 3).

This empirical analysis demonstrates that the minimum for each case occurs at a boundary (i.e., small m or $m \to \infty$). A formal proof of these properties

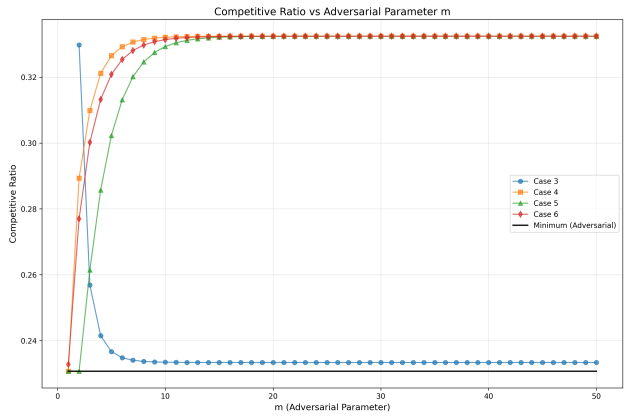

Figure 1: Competitive ratio lower bound $\phi_k(m)$ as a function of the number of badly-predicted candidates, m. The plot shows that the minimum for each case occurs at a boundary (small m or $m \to \infty$). The overall competitive ratio is determined by the lowest point across all curves.

via derivatives is possible and can be found in the appendix. By identifying the minimum value along each curve in Figure 1 and then taking the minimum over all cases, we establish the final worst-case competitive ratio. Our choice of parameters carefully balances multiple adversarial scenarios, with the final guarantee of 0.229 being simultaneously constrained by Case 1 and 2, Case 4 at $m = 2$, and Case 5 at $m = 1$. For case 3 where $i^* \neq \hat{\imath}, i^* \in M, \hat{\imath} \in M, m = |M| \geq 2$ so the value of $m = 1$ is not defined.

7 FAIRNESS

Another key emerging line of work in the Secretary problem is achieving the fairness goal, which is when the best candidate has a constant value of being accepted even when predictions are arbitrarily unfair / inaccurate. Beyond improving the competitive ratio of a Dynkin-style algorithm, we now demonstrate that our ordering framework is a general paradigm that can also enhance other objectives, like fairness. To show this, we adapt the ADDITIVE-PEGGING algorithm (Balkanski et al., 2024). For clarity, we present their algorithm pseudocode. The main idea behind this algorithm is balancing between the secretary condition $\mathcal{F}$, the condition that the current element is the best so far, and $\mathcal{C}$, the condition that the current element is the maximum so far. This is done by maintaining a set of "pegged" candidates whereby they are a "large enough" with respect to the predicted maximum value.

By modifying the arrival time, we improve the fairness significantly from 0.0625 to 0.125.

Algorithm 2 ADDITIVE-PEGGING

/ The algorithm stops when it accepts a candidate by executing* $\mathcal{A} \leftarrow i$*. */*

Initialization: $I^{\text{pegged}} \leftarrow \emptyset$
while candidate i arrives at time t_i **do**
  if $i \in I^{\text{pegged}}$ **then**
    if $|I^{\text{pegged}}| = 1$ **then**
      $\mathcal{A} \leftarrow i$
    else
      $I^{\text{pegged}} \leftarrow I^{\text{pegged}} \setminus \{i\}$
  $\mathcal{F} \leftarrow (v_i > \max_{j \prec i} v_j) \wedge (t_i > 1/2)$, $\mathcal{C} \leftarrow (i = \hat{\imath})$, $\varepsilon_{t_i} \leftarrow \max_{j:t_j \leq t_i} |\hat{v}_j - v_j|$
  if $\mathcal{C} \wedge \mathcal{F}$ **then**
    $\mathcal{A} \leftarrow i$
  else if $\mathcal{C} \wedge \bar{\mathcal{F}}$ **then**
    $I^{\text{pegged}} \leftarrow \{\, j : j \succ \hat{\imath},\ v_{\hat{\imath}} < \hat{v}_j + \varepsilon_{t_i} \,\}$ *(here* $\hat{\imath} = i$*)*
    if $I^{\text{pegged}} = \emptyset$ **then**
      $\mathcal{A} \leftarrow i$
  else if $\bar{\mathcal{C}} \wedge \mathcal{F}$ **then**
    if $v_i > \hat{v}_{\hat{\imath}} - \varepsilon_{t_i}$ **then**
      $\mathcal{A} \leftarrow i$
return $\mathcal{A}$

7.1 Distribution of $\hat{\imath}$

We set the arrival time of $\hat{\imath}$, T, as a random variable such that

$$T = \begin{cases} \frac{1}{2} - \delta & \text{with probability } \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

7.2 Analysis

Lemma 7 (Fairness of Ordered-Additive-Pegging)**.** *Let* $\mathcal{A}_{\text{OAP}}$ *denote the modified Additive-Pegging algorithm under the arrival distribution* T*, and define*

$$\text{Fair}(\mathcal{A}_{\text{OAP}}; v, \hat{v}) \ := \ \Pr\big[\mathcal{A}_{\text{OAP}}(v, \hat{v}) = i^*\big],$$

where $i^* = \arg\max_i v_i$ *is the index of the best candidate. Then, for every instance* $(v, \hat{v})$

$$\text{Fair}(\mathcal{A}_{\text{OAP}}; v, \hat{v}) \ \geq \ \tfrac{1}{8}.$$

Proof. For the analysis, we focus on 3 distinct cases as proposed in Additive-Pegging with the competitive ratio given by the minimum of these 3 cases. Specifically, concentrating all of T's probability masses after $t = \frac{1}{2}$ at $t = 1$ we are able to improve the constant from $\frac{1}{16}$ to $\frac{1}{8}$ for the first case, as stated below. We define v_i to be the actual value of the secretary, $\hat{v}_i$ to be the predicted value of the secretary, and ε_i to be the absolute difference between the actual and predicted values of a certain secretary. We define $\tilde{\imath}$ to be the index of the candidate with the highest true value except i^* and $\hat{\imath}$.

7.2.1 $i^* \neq \hat{i}$ and $v_{i^*} > \hat{v}_{\hat{i}} - \varepsilon_{i^*}$

This case was previously the bottleneck for the Additive-Pegging algorithm. In this case, E_2 is a sufficient condition for i^* to be accepted as both C and F are false until i^*, and the algorithm definitely accepts i^* at t_{i^*} as $\varepsilon_{t_{\tilde{i}}^*} \geq \varepsilon_{i^*}$ and the acceptance condition of $\bar{C} \wedge F$ is fulfilled.

Define $E_2 = \{t_{\tilde{i}} < 1/2 < t_{i^*} < t_{\hat{i}}\}$

$$\begin{aligned} P[E_2] &= P[t_{\tilde{i}} < 1/2] \cdot P[1/2 < t_{i^*} < t_{\hat{i}}] \\ &= P[t_{\tilde{i}} < 1/2] \cdot P[1/2 < t_{i^*} \wedge t_{\hat{i}} = 1] \\ &= (1/2) \cdot (1/2) \cdot (1/2) = 1/8 \end{aligned}$$

For completeness, We also present the proof of the other 2 cases which is similar what was presented in Additive-Pegging.

7.2.2 $i^* = \hat{i}$

$$P(t_{\tilde{i}} < \frac{1}{2} < t_{i^*}) = P(t_{\tilde{i}} < \frac{1}{2}) * P(\frac{1}{2} < t_{i^*}) = \frac{1}{4}$$

7.2.3 $i^* \neq \hat{i}$ and $v_{\hat{i}} < \hat{v}_{i^*} + \varepsilon_{\hat{i}}$

$$P(\{t_{\tilde{i}} < 1/2\} \wedge \{t_{\hat{i}} < 1/2\} \wedge \{1/2 < t_{i^*}\}) = (\frac{1}{2})^3 = \frac{1}{8}$$

By combining these 3 cases, we achieve a new fairness of $\frac{1}{8}$. The full proof can be found in the appendix. □

8 EXPERIMENTS

Experimental Setup. We simulate our algorithms in the same setup as Fujii and Yoshida as well as Balkanski et al. We present the description and pseudocode of the LEARNED-DYNKIN algorithm in the appendix. We follow the Almost-constant, Uniform test cases as proposed by them, and present the Deception as well as Robustness test cases to gain more insight about the algorithms' competitive ratio when faced with data that is harder.

In Almost-constant, Uniform and Deception, the maximum multiplicative error between the true and predicted values of candidates is characterised by ϵ, which varies across test groups. Setting $\epsilon = 0$ creates instances where predictions are exactly the true values, while increasing ϵ creates instances with predictions more prone to error.

Almost-constant models a situation where one candidate has a true value of $\frac{1}{1-\varepsilon}$ and the rest of the candidates have a value of 1. All predictions are set to 1.

In **Uniform**, we sample each v_i independently from the exponential distribution with parameter 1. The exponential distribution generates a large value with a small probability and consequently models a situation where one candidate is significantly better than the rest. All predicted values are generated by perturbing the actual value with the uniform distribution, i.e., $\hat{v}_i = \delta_i \cdot v_i$, where δ_i is sampled uniformly and independently from $[1-\varepsilon, 1+\varepsilon]$.

We also introduce a **Deception** instance type. This setup is motivated by scenarios where a candidate might exaggerate their qualifications to appear more valuable than they are. Specifically, one 'decoy' candidate with a moderate true value, v_{decoy}, has its prediction artificially inflated, $\hat{v}_{\text{decoy}} = (1+\varepsilon_{\text{decoy}})v_{\text{decoy}}$. In contrast, the truly best candidate's value, u_{best}, is predicted perfectly, $\hat{v}_{\text{best}} = v_{\text{best}}$. All other candidates have low background values. The deception occurs when the decoy's exaggerated prediction surpasses the true best's, i.e., $\hat{v}_{\text{decoy}} > \hat{v}_{\text{best}}$, testing whether an algorithm can be misled by a single, highly-inflated prediction.

Finally, we introduce an adversarial **Robustness** instance, designed specifically to probe the performance of the additive-pegging algorithm. This input features a candidate with a vastly superior true value but a severely underestimated prediction (e.g., $v_1 = 100000, \hat{v}_1 = 1$). Conversely, a much weaker candidate, whose true value is only a fraction of the best, is predicted perfectly and thus appears to be the most attractive option ($v_2 = \hat{v}_2 = 10$). Finally, $v_3 = 8, \hat{v}_3 = 3 \implies \varepsilon_3 = 5$ so that $v_3 > \hat{v}_2 - \varepsilon_3$. This guarantees that v_3 would be accepted if it were to appear before the other two and after $t = \frac{1}{2}$. This adversarial structure tests if the algorithm can overcome the misleading signal from the predictions and explore a seemingly low-value candidate to find the true optimum, a critical behavior for its fairness guarantee to hold.

Our code is written in Python 3.9.21 and we conduct experiments on an Intel(R) Xeon(R) CPU E5-2683 v4 CPU with 32 GB of RAM, on Linux. Common python libraries such as numpy and matplotlib was used. The total runtime is less than 5 minutes.

Our experiments evaluate the performance of our proposed algorithms against their state-of-the-art counterparts. The results are benchmarked across four distinct environments, measuring the competitive ratio. Overall, our proposed methods demonstrate significantly improved robustness in challenging scenarios.

In the **Uniform** setting, which represents a standard problem space, our algorithms show a clear advantage. As the multiplicative error bound ϵ increases, both ORDERED-ADDITIVE-PEGGING maintain a higher competitive ratio as compared to

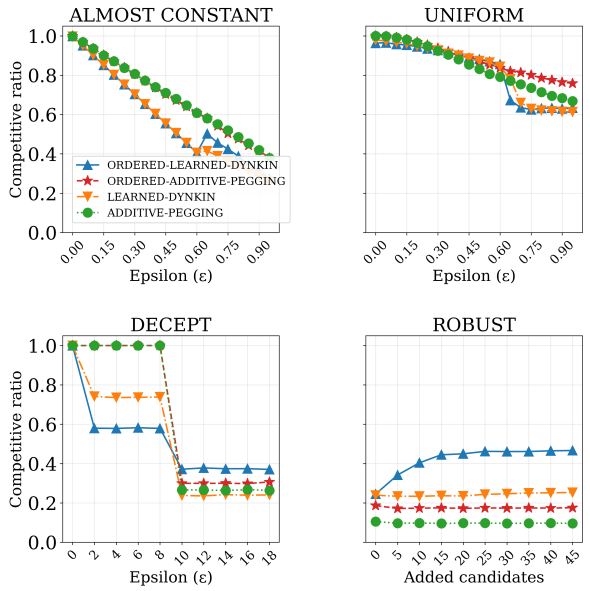

Figure 2: A comparison of algorithm performance across the four instance types.

ADDITIVE-PEGGING.

The **Decept** environment was designed as a stress test to probe for algorithm brittleness. The results are striking. While the state-of-the-art algorithms perform perfectly for low multiplicative error ϵ, they suffer a significant performance collapse at $\epsilon = 10$, where their competitive ratio plummets. This highlights a key contribution: our method avoids the critical failure mode exhibited by the current state-of-the-art, making it far more reliable for real-world problems.

The **Robust** experiment tests performance as low value candidates are added to the problem. The baseline algorithms show poor performance, with ADDITIVE-PEGGING achieving a competitive ratio of below 0.10.

References

Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. Secretary and online matching problems with machine learned advice. *Discrete Optimization*, 48:100778, 2023. ISSN 1572-5286. doi: https://doi.org/10.1016/j.disopt.2023.100778. URL `https://www.sciencedirect.com/science/article/pii/S1572528623000208`.

Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. In Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 16–28, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74208-1.

Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. Matroid secretary problems. *J. ACM*, 65(6), November 2018. ISSN 0004-5411. doi: 10.1145/3212512. URL `https://doi.org/10.1145/3212512`.

Eric Balkanski, Will Ma, and Andreas Maggiori. Fair secretaries with unfair predictions. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 3682–3716. Curran Associates, Inc., 2024.

Archit Bubna and Ashish Chiplunkar. Prophet inequality: Order selection beats random order. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC '23, page 302–336, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701047. doi: 10.1145/3580507.3597687. URL `https://doi.org/10.1145/3580507.3597687`.

Jose Correa, Raimundo Saona, and Bruno Ziliotto. Prophet secretary through blind strategies. *Mathematical Programming*, 190(1):483–521, 2021. doi: 10.1007/s10107-020-01544-8. URL `https://doi.org/10.1007/s10107-020-01544-8`.

Eugene B. Dynkin. The optimum choice of the instant for stopping a Markov process. *Soviet Mathematics - Doklady*, 4:627–629, 1963.

Thomas S Ferguson. Who solved the secretary problem? *Statistical science*, 4(3):282–289, 1989.

Kaito Fujii and Yuichi Yoshida. The secretary problem with predictions. *Mathematics of Operations Research*, 49, 08 2023. doi: 10.1287/moor.2022.0031.

Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, page 630–631, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 0898715857.

Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *J. ACM*, 68(4), July 2021. ISSN 0004-5411. doi: 10.1145/3447579. URL `https://doi.org/10.1145/3447579`.

Bo Peng and Zhihao Gavin Tang. Order Selection Prophet Inequality: From Threshold Optimization to Arrival Time Design . In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 171–178, Los Alamitos, CA, USA, November 2022. IEEE Computer Society. doi: 10.1109/FOCS54457.2022.00023. URL `https://doi.ieeecomputersociety.org/10.1109/FOCS54457.2022.00023`.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms with machine learning. In *Advances in Neural Information Processing Systems 31*, 2018.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Sections 1–7 define the setting, model, and algorithms.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] Sections 5–6 analyze the competitive ratio and fairness guarantees.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] Code and dependencies are included in the supplementary material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] The assumptions are stated in Sections 3 and 5.
   (b) Complete proofs of all theoretical results. [Yes] Main proofs are in the paper, with additional details in the appendix.
   (c) Clear explanations of any assumptions. [Yes] The paper explains the prediction model, ordering control, and error assumptions.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] Reproduction code and instructions are provided in the supplementary material.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] Section 7 describes the simulation setup and parameter choices.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [No] The paper reports competitive ratios over repeated simulations, but it does not include error bars.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] Section 7 specifies the CPU, RAM, OS, and runtime.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Secretary Problem with Predictions and Ordering: Supplementary Material

A PRESENTATION OF BASELINE ALGORITHMS

A.1 Learned-dynkin

Algorithm 3 Learned Dynkin

Require: Time $\tau = 0.313$, internal discrepancy threshold $\theta = 0.646$, predictions $\hat{v} : N \to \mathbb{R}$.

$\hat{i} \in \text{argmax}_{i \in N} \hat{v}(i)$.
mode $\leftarrow$ PREDICTION.
for each candidate $i \in N$ in random order **do**
  if $|1 - \frac{\hat{v}(i)}{v(i)}| > \theta$ **then**
    mode $\leftarrow$ SECRETARY.
  if mode = PREDICTION and $i = \hat{i}$ **then**
    Hire i.
  if mode = SECRETARY and $t_i > \tau$ and i is the best so far **then**
    Hire i.

This algorithm achieves a competitive ratio of $\max\{\frac{1-\theta}{1+\theta}, 0.215\}$ by following a similar proof technique.

B DETAILED PROOF OF FAIRNESS GUARANTEE

We analyze the modified Additive-Pegging algorithm where the arrival time of $\hat{\imath}$, $t_{\hat{\imath}}$, is drawn from the distribution T where $T = \frac{1}{2} - \delta$ with probability $1/2$ and $T = 1$ with probability $1/2$, for some small $\delta > 0$. The goal is to lower bound the probability of selecting the best candidate, i^*. We consider all cases. We denote by $\tilde{\imath}$ the index of the candidate with the highest true value except i^* and $\hat{\imath}$, i.e., $\tilde{\imath} = \text{argmax}_{i \neq i^*, \hat{\imath}} v_i$. Our modification concerns the arrival time of $\hat{\imath}$. Note every scenario falls in exactly one of these cases as proven in the paper. Note that we are using the same argument as the original paper, but due to our modification on arrival time, we are able to improve the probability.

- **Case 1:** $i^* = \hat{\imath}$
  We need to find a scenario where i^* is guaranteed to be selected. Consider the event $E_1 = \{t_{\tilde{\imath}} < 1/2\} \cap \{t_{i^*} > 1/2\}$. If $t_{\tilde{\imath}} < 1/2$, the threshold at time t_{i^*} will be based on predictions seen before $1/2$. Since $\hat{v}_{\tilde{\imath}}$ is the highest prediction seen so far, the threshold will be at most $\hat{v}_{\tilde{\imath}} - \varepsilon_{\tilde{\imath}}$. So, $\mathbb{P}(\text{select } i^*) \geq \mathbb{P}(t_{\tilde{\imath}} < 1/2 \text{ and } t_{i^*} = 1) = (1/2) \cdot (1/2) = 1/4$.

- **Case 2:** $i^* \neq \hat{\imath}$ **and** $v_{\hat{\imath}} < \hat{v}_{i^*} + \varepsilon_{\hat{\imath}}$ **(Prediction for $\hat{\imath}$ is not much larger than for i^*)**

  If $v_{\hat{\imath}} < \hat{v}_{i^*} + \varepsilon_{\hat{\imath}}$, then we define event $E_1 = \{t_{\tilde{\imath}} < 1/2\} \wedge \{t_{\hat{\imath}} < 1/2\} \wedge \{1/2 < t_{i^*}\}$ which is composed by 3 independent events and it happens with probability $\mathbb{P}[E_1] = (1/2)^3 = 1/8$. E_1 implies that $t_{\hat{\imath}} < t_{i^*} \Rightarrow \varepsilon_{t_{i^*}} \geq \varepsilon_{\hat{\imath}}$, thus we can deduce that $v_{i^*} > v_{\hat{\imath}} \geq \hat{v}_{\hat{\imath}} - \varepsilon_{\hat{\imath}} \geq \hat{v}_{\hat{\imath}} - \varepsilon_{t_{i^*}}$. Consequently, if until time t_{i^*} all candidates are rejected, E_1 implies that $\bar{C} \wedge F \wedge \{v_{i^*} > \hat{v}_{\hat{\imath}} - \varepsilon_{i^*}\}$ is true at time t_{i^*} and candidate i^* is hired. To argue that no candidate is accepted before time t_{i^*}, note that F is false at all times before t_{i^*} and at time $t_{\hat{\imath}}$ (when literal C is true) the set $\{j \succ \hat{\imath} : v_{\hat{\imath}} < \hat{v}_j + \varepsilon_{t_j}\} \supseteq \{j \succ \hat{\imath} : v_{\hat{\imath}} < \hat{v}_j + \varepsilon_{\hat{\imath}}\}$ contains i^*.

- **Case 3:** $i^* \neq \hat{\imath}$ **and** $v_{i^*} > \hat{v}_{\hat{\imath}} - \varepsilon_{i^*}$ **(Adversarial case from original paper)**
  Note that until time t_{i^*} no candidate is accepted since C and F are both false at all times. Between times

0 and 1/2 only $\hat{\imath}$ could have been accepted but its arrival time is after t_{i^*}, and between times 1/2 and t_{i^*} no candidate has a true value larger than $v_{\bar{\imath}}$. Finally, note that at time t_{i^*} we have $\varepsilon_{t_{i^*}} \geq \varepsilon_{i^*}$ and consequently $\bar{C} \wedge F \wedge \{v_{i^*} > \hat{v}_{\hat{\imath}} - \varepsilon_{t_{i^*}}\}$ is true and i^* gets accepted. $\mathbb{P}(t_{\bar{\imath}} < 1/2) = 1/2$. $\mathbb{P}(t_{\hat{\imath}} = 1) = 1/2$. $\mathbb{P}(1/2 < t_{i^*} < t_{\hat{\imath}} | t_{\hat{\imath}} = 1)$ is the probability that a uniform random variable on $[0, 1]$ falls in $[1/2, 1]$ The total probability is $\mathbb{P}(E_3) = (1/2) \cdot (1/2) \cdot (1/2) = 1/8$.

In all cases, we have found a scenario leading to the selection of i^* with a probability of at least 1/8. Thus, the fairness guarantee is 1/8 □

C PROOF OF LOWER BOUNDS

Case 3 Fix prob $\in [0, 1]$. Since $\hat{\imath}, i^* \in M$, we have $m = |M| \geq 2$. For integers $m \geq 2$ define the case-3 lower-bound function

$$\begin{aligned}\phi_m := p_0(1 - \text{prob})\,\tau \int_\tau^1 \frac{dt}{t} \\ + (1 - p_0 - p_1)\,\tau \int_\tau^1 \frac{1 - (1-t)^{m-2}}{t}\,dt \\ + (1 - p_0 - p_1) \int_\tau^1 (1-t)^{m-2}\,dt.\end{aligned}$$

Then $(\phi_m)_{m \geq 2}$ is (strictly) decreasing in m when $1 - p_0 - p_1 > 0$, and

$$\begin{aligned}\inf_{m \geq 2} \phi_m &= \lim_{m \to \infty} \phi_m \\ &= (1 - p_1 - p_0\,\text{prob})\,\tau\,\log\bigl(1/\tau\bigr).\end{aligned}$$

Proof. Only the last two terms depend on m. Using $(1 - (1-t)^{m-1}) - (1 - (1-t)^{m-2}) = t(1-t)^{m-2}$ and $(1-t)^{m-1} - (1-t)^{m-2} = -t(1-t)^{m-2}$, the forward difference $\Delta_m := \phi_{m+1} - \phi_m$ satisfies

$$\begin{aligned}\Delta_m &:= \phi_{m+1} - \phi_m \\ &= (1 - p_0 - p_1) \int_\tau^1 (1-t)^{m-2}(\tau - t)\,dt \\ &\leq 0.\end{aligned}$$

with strict inequality whenever $1 - p_0 - p_1 > 0$ and $\tau < 1$. Thus $(\phi_m)_{m \geq 2}$ is nonincreasing. For the limit, note that for $t \in [\tau, 1]$ we have $(1 - (1-t)^{m-2}) \uparrow 1$ and $(1-t)^{m-2} \downarrow 0$ as $m \to \infty$. By monotone/dominated convergence,

$$\begin{aligned}\lim_{m \to \infty} \phi_m &= p_0(1 - \text{prob})\,\tau \int_\tau^1 \frac{dt}{t} + (1 - p_0 - p_1)\,\tau \int_\tau^1 \frac{dt}{t} \\ &= (1 - p_1 - p_0\,\text{prob})\,\tau\,\log\bigl(1/\tau\bigr).\end{aligned}$$

This value is therefore a valid lower bound for Case 3 for all $m \geq 2$. □

Case 4 For $m = 1, 2$, we use the explicit lower bounds in Lemma 4. For integers $m \geq 3$ define

$$\begin{aligned}\phi_m := p_0\,\tau \int_\tau^1 \frac{dt}{t} \\ +\ (1 - p_0 - p_1)\,\tau \int_\tau^1 \frac{1 - (1-t)^{\,m-1}}{t}\,dt.\end{aligned}$$

Then $(\phi_m)_{m \geq 3}$ is (weakly) increasing in m, and is strictly increasing if $1 - p_0 - p_1 > 0$.

Proof. Only the second integral depends on m, so it suffices to show that the map

$$m \longmapsto \tau \int_{\tau}^{1} \frac{1-(1-t)^{m-1}}{t}\, dt$$

is increasing for integer $m \geq 3$. For consecutive integers m and $m+1$ we compute

$$\begin{aligned}
\phi_{m+1} - \phi_m &= (1-p_0-p_1) \int_{\tau}^{1} \big(1-(1-t)^{m}\big) \frac{\tau}{t}\, dt \\
&\quad - (1-p_0-p_1) \int_{\tau}^{1} \big(1-(1-t)^{m-1}\big) \frac{\tau}{t}\, dt \\
&= (1-p_0-p_1) \int_{\tau}^{1} \big((1-t)^{m-1} - (1-t)^{m}\big) \frac{\tau}{t}\, dt \\
&= (1-p_0-p_1) \int_{\tau}^{1} (1-t)^{m-1}\, \tau\, dt \\
&= (1-p_0-p_1)\, \tau \int_{\tau}^{1} (1-t)^{m-1}\, dt.
\end{aligned}$$

For $t \in [\tau, 1]$ and $m \geq 1$ we have $(1-t)^{m-1} \geq 0$, hence the last integral is nonnegative. Because $\tau > 0$ and $1-p_0-p_1 \geq 0$, it follows that $\phi_{m+1} - \phi_m \geq 0$. Moreover, if $1-p_0-p_1 > 0$ and $\tau < 1$, then

$$\int_{\tau}^{1} (1-t)^{m-1}\, dt \;=\; \frac{(1-\tau)^{m}}{m} > 0,$$

so $\phi_{m+1} > \phi_m$ in this case. Therefore $(\phi_m)_{m \geq 3}$ is (weakly) increasing in m, with strict increase when $1-p_0-p_1 > 0$. □

Case 5 Fix $\tau \in (0,1)$ and $p_0, p_1 \in [0,1]$ with $p_0 + p_1 \leq 1$. For integers $m \geq 1$ define

$$\begin{aligned}
\phi_m := (1-p_0-p_1) \Bigg[&\int_{\tau}^{1} (1-t)^{m-1}\, dt \\
&+ \tau \int_{\tau}^{1} \frac{1-(1-t)^{m-1}}{t}\, dt \Bigg] \\
&+ p_0\, \tau \int_{\tau}^{1} \frac{1-(1-\tau)^{m-1}}{t}\, dt.
\end{aligned}$$

Then $(\phi_m)_{m \geq 1}$ is monotonically increasing in m provided

$$p_0\, \tau^2 \log\!\big(1/\tau\big) \;\geq\; \frac{1-p_0-p_1}{2}\, (1-\tau)^2. \tag{$*$}$$

In particular, with $\tau = 0.3887$, $p_0 = 0.5416$, $p_1 = 0.08345$ the condition $(*)$ holds, hence $(\phi_m)_{m \geq 1}$ is (strictly) increasing for these parameters.

Proof. Only the exponents in the integrands depend on m, so consider the forward difference

$$\Delta_m := \phi_{m+1} - \phi_m.$$

Using

$$\begin{aligned}
(1-t)^{m} - (1-t)^{m-1} &= -t(1-t)^{m-1}, \\
\big(1-(1-t)^{m}\big) - \big(1-(1-t)^{m-1}\big) &= t(1-t)^{m-1}.
\end{aligned}$$

and

$$\big(1-(1-\tau)^{m}\big) - \big(1-(1-\tau)^{m-1}\big) = \tau(1-\tau)^{m-1}.$$

we obtain

$$\Delta_m = (1-p_0-p_1)\int_\tau^1 (\tau-t)(1-t)^{m-1}\,dt + p_0\tau^2(1-\tau)^{m-1}\int_\tau^1 \frac{dt}{t}.$$

Both t–integrals are explicit:

$$\int_\tau^1 (\tau-t)(1-t)^{m-1}\,dt = -\frac{(1-\tau)^{m+1}}{m(m+1)},$$
$$\int_\tau^1 \frac{dt}{t} = \log(1/\tau).$$

Therefore

$$\Delta_m = (1-\tau)^{m-1}\Big(-(1-p_0-p_1)\frac{(1-\tau)^2}{m(m+1)}\Big) + (1-\tau)^{m-1}\,p_0\,\tau^2\log(1/\tau). \tag{$\dagger$}$$

Since $(1-\tau)^{m-1}>0$, the sign of Δ_m is the sign of the bracket. The function $m\mapsto \frac{1}{m(m+1)}$ is decreasing, so the worst case (most negative bracket) occurs at $m=1$. Thus a sufficient (and sharp) condition ensuring $\Delta_m \geq 0$ for all $m\geq 1$ is exactly $(*)$. Under $(*)$ we get $\Delta_m \geq 0$ for every $m\geq 1$, proving that $(\phi_m)_{m\geq 1}$ is monotonically increasing. □

Numerical verification (given parameters). For $\tau = 0.3887$, $p_0 = 0.5416$, $p_1 = 0.08345$,

$$p_0\,\tau^2\log(1/\tau) \approx 0.0773,$$
$$\frac{1-p_0-p_1}{2}(1-\tau)^2 \approx 0.0701.$$

Hence $(*)$ holds, so by $(\dagger)$ we have $\Delta_m > 0$ for all $m\geq 1$; in particular $\phi_{m+1} > \phi_m$ for all $m\geq 1$ with these parameters.

Case 6 Fix $\tau\in(0,1)$, $p_0,p_1\in[0,1]$ with $p_0+p_1\leq 1$, and set

$$c := \frac{1-\theta}{1+\theta}.$$

For integers $m\geq 1$ define

$$\phi_m := p_0\,\tau\int_\tau^1 \frac{1-(1-t)^m}{t}\,dt + p_0\int_\tau^1 (1-t)^m\,c\,dt + (1-p_0-p_1)\,\tau\int_\tau^1 \frac{1-(1-\tau)^m}{t}\,dt + (1-p_0-p_1)\int_\tau^1 (1-\tau)^m\,c\,dt.$$

Then $(\phi_m)_{m\geq 1}$ is strictly increasing for the parameter values

$$\tau = 0.3887, \qquad p_0 = 0.5416,$$
$$p_1 = 0.08345, \qquad \theta = 0.627.$$

$$\text{so } c = \tfrac{1-\theta}{1+\theta} \approx 0.229.$$

Proof. Compute the forward difference $\Delta_m := \phi_{m+1} - \phi_m$. Only the exponents m in the integrands change, hence

$$\Delta_m = p_0 \int_\tau^1 (1-t)^m \big(\tau - c\,t\big)\, dt \tag{2}$$

$$+ (1 - p_0 - p_1)(1-\tau)^m \tau \int_\tau^1 \Big(\frac{\tau}{t} - c\Big)\, dt. \tag{3}$$

For the first integral, regrouping gives

$$\big((1-t)^m - (1-t)^{m+1}\big)\frac{\tau}{t} + \big((1-t)^{m+1} - (1-t)^m\big)c$$
$$= (1-t)^m\big(\tau - c\,t\big).$$

For the second, use $(1-\tau)^{m+1} - (1-\tau)^m = -(1-\tau)^m\tau$ to obtain

$$\big((1-\tau)^m - (1-\tau)^{m+1}\big)\frac{\tau}{t} = (1-\tau)^m\tau\,\frac{\tau}{t},$$
$$\big((1-\tau)^{m+1} - (1-\tau)^m\big)c = -(1-\tau)^m\tau\,c.$$

Adding these expressions yields the integrand in (3). Therefore (3) holds. Both integrals in (3) admit closed forms. For the first, substitute $u = 1 - t$:

$$\int_\tau^1 (1-t)^m(\tau - c\,t)\, dt = \int_0^{1-\tau} u^m\big(\tau - c(1-u)\big)\, du$$
$$= (\tau - c)\frac{(1-\tau)^{m+1}}{m+1} \tag{4}$$
$$+ c\,\frac{(1-\tau)^{m+2}}{m+2}. \tag{5}$$

For the second,

$$\int_\tau^1 \Big(\frac{\tau}{t} - c\Big)\, dt = \tau\int_\tau^1 \frac{dt}{t} - c\int_\tau^1 dt$$
$$= \tau\log\big(1/\tau\big) \tag{6}$$
$$- c(1-\tau). \tag{7}$$

Combining (5) and (7) in (3) yields

$$\begin{aligned}\Delta_m &= p_0(\tau - c)\frac{(1-\tau)^{m+1}}{m+1} \\ &\quad + p_0\,c\,\frac{(1-\tau)^{m+2}}{m+2} \\ &\quad + (1 - p_0 - p_1)(1-\tau)^m\tau\Big[\tau\log\big(1/\tau\big) - c(1-\tau)\Big] \\ &= (1-\tau)^m\Bigg\{p_0(\tau - c)\frac{1-\tau}{m+1} + p_0\,c\,\frac{(1-\tau)^2}{m+2} \\ &\qquad + (1 - p_0 - p_1)\tau\Big[\tau\log\big(1/\tau\big) - c(1-\tau)\Big]\Bigg\}.\end{aligned} \tag{8}$$

Since $(1-\tau)^m > 0$ for all $m \geq 1$, the sign of Δ_m is the sign of the bracket in (8).

Numerical positivity for the given parameters. With $\tau = 0.3887$, $\theta = 0.627$ we have $c = \frac{1-\theta}{1+\theta} \approx 0.229$ and hence

$$\tau - c \approx 0.3887 - 0.2293 = 0.1594 > 0,$$
$$\tau\log(1/\tau) - c(1-\tau) \approx 0.3673 - 0.1401 = 0.2272 > 0.$$

All coefficients $p_0 = 0.5416$ and $1 - p_0 - p_1 = 0.37495$ are nonnegative. Since every summand inside the braces of (8) is thus strictly positive for every $m \geq 1$, we conclude $\Delta_m > 0$ for all $m \geq 1$.

Conclusion. $\phi_{m+1} - \phi_m = \Delta_m > 0$ for all $m \geq 1$ with the stated parameters. Hence $(\phi_m)_{m \geq 1}$ is strictly increasing. □