

# “Who is Next in Line?”

## On the Significance of Knowing the Arrival Order in Bayesian Online Settings\*

Tomer Ezra<sup>†</sup>    Michal Feldman<sup>‡</sup>    Nick Gravin<sup>§</sup>    Zhihao Gavin Tang<sup>‡</sup>

### Abstract

We introduce a new measure for the performance of online algorithms in Bayesian settings, where the input is drawn from a known prior, but the realizations are revealed one-by-one in an online fashion. Our new measure is called *order-competitive ratio*. It is defined as the worst case (over all distribution sequences) ratio between the performance of the best *order-unaware* and *order-aware* algorithms, and quantifies the loss that is incurred due to lack of knowledge of the arrival order. Despite the growing interest in the role of the arrival order on the performance of online algorithms, this loss has been overlooked thus far.

We study the order-competitive ratio in the paradigmatic *prophet inequality* problem, for the two common objective functions of (i) maximizing the expected value, and (ii) maximizing the probability of obtaining the largest value; and with respect to two families of algorithms, namely (i) adaptive algorithms, and (ii) single-threshold algorithms. We provide tight bounds for all four combinations, with respect to deterministic algorithms. Our analysis requires new ideas and departs from standard techniques. In particular, our adaptive algorithms inevitably go beyond single-threshold algorithms. The results with respect to the order-competitive ratio measure capture the intuition that adaptive algorithms are stronger than single-threshold ones, and may lead to a better algorithmic advice than the classical competitive ratio measure.

### 1 Introduction

As part of the growing literature on “beyond worst case analysis” (see [Rou21] for a textbook treatment), the area of online algorithms has seen a notable shift from adversarial analysis to average-case analysis. A prominent framework is *Bayesian online settings*, where the input is assumed to be drawn from a known prior, but the actual realizations are revealed in an online fashion. Many online problems have been studied within the Bayesian online framework, such as prophet inequality, stochastic matching, Steiner tree, and welfare and revenue maximization in combinatorial auctions [KS78, KS77, SC84, GW19, EFGT20, GGLS08, FGL15].

The most common performance measure for the analysis of online algorithms is the *competitive ratio*, defined as the ratio between the performance of the algorithm and the optimal offline solution. That is, the performance of an online algorithm is evaluated by its performance relative to the performance of a “prophet” who can see into the future. The competitive ratio measure has been applied directly to the Bayesian online setting, where the performance of both the algorithm and the optimal solution is taken in expectation over the input.

However, the prophet benchmark may be too strong for Bayesian online settings. In particular, it may provide an overly pessimistic performance prediction, and perhaps more concerning, it may fail to differentiate between different classes of algorithms, and may consequently provide misleading algorithmic advice. A natural way to

---

\*This work is supported by Science and Technology Innovation 2030 –“New Generation of Artificial Intelligence” Major Project No.(2018AAA0100903), Innovation Program of Shanghai Municipal Education Commission, Program for Innovative Research Team of Shanghai University of Finance and Economics (IRTSHUFE) and the Fundamental Research Funds for the Central Universities. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 866132), by the Israel Science Foundation (grant number 317/17), by an Amazon Research Award, and by the NSF-BSF (grant number 2020788). Tomer Ezra was partially supported by the ERC Advanced Grant 788893 AMDROMA “Algorithmic and Mechanism Design Research in Online Markets” and MIUR PRIN project ALGADIMAR “Algorithms, Games, and Digital Markets”. Zhihao Gavin Tang is supported by NSFC grant 61902233. Nick Gravin is supported by NSFC grant 62150610500.

<sup>†</sup>Sapienza University of Rome, Email: [ezra@diag.uniroma1.it](mailto:ezra@diag.uniroma1.it)

<sup>‡</sup>Blavatnik School of Computer Science, Tel Aviv University, and Microsoft Research Israel, Email: [mfeldman@tau.ac.il](mailto:mfeldman@tau.ac.il)

<sup>§</sup>ITCS, Shanghai University of Finance and Economics, Email: [{nikolai, tang.zhihao}@mail.shufe.edu.cn](mailto:{nikolai, tang.zhihao}@mail.shufe.edu.cn)

address this problem is to propose a more realistic benchmark. Indeed, there has been much interest recently in considering alternative, more realistic, benchmarks for this setting (see Section 1.2).

For concreteness, let us consider the well known *prophet inequality* problem — perhaps the most paradigmatic problem within Bayesian online settings.

**Prophet Inequality [KS77, KS78, SC84].** In a prophet inequality setting, there are  $n$  boxes, each box  $t = 1, \dots, n$  contains a value  $v_t$  drawn from a known probability distribution  $F_t$ . The boxes arrive online. Upon the arrival of box  $t$ , its value  $v_t$  is revealed, and the online algorithm needs to decide immediately and irrevocably whether to accept it, in which case the game ends with a reward of  $v_t$ , or to skip it, in which case the reward  $v_t$  is lost forever and the game proceeds to the next box. The prophet inequality problem captures many real-life scenarios where a decision maker inevitably makes decisions based on partial information regarding the future. It has become central in the study of market design due to its close connection to mechanism design and posted price mechanisms [HKS07].

In its standard and most well-studied variant, the objective is to maximize the expected accepted value (corresponding to social welfare maximization in the market scenario above). Hence, the competitive ratio is the worst-case ratio (over all possible distribution sequences) between the expected value accepted by the algorithm, and the expected maximum value.

The problem has been first studied under the assumption that the arrival order is known by the algorithm, where the optimal algorithm uses a sequence of  $n$  different thresholds, computed by a backward induction process. A celebrated result states that this *order-aware* algorithm achieves an expected value at least  $1/2$  of the prophet [KS77, KS78], and this result is tight with respect to the prophet benchmark; that is, no online algorithm can obtain a better competitive ratio.

Quite remarkably, it has been later shown that the same competitive ratio of  $1/2$  can be obtained by a single-threshold algorithm; namely, an algorithm that determines a single fixed threshold, and accepts the first value that exceeds it. Moreover, this algorithm is *order-unaware*, namely, in order to apply it, no information about the arrival order is needed [SC84, KW19]. Order-unawareness is a desirable property of online algorithms, as in many cases, no information about the arrival order is provided, and even if provided, order-unaware algorithms are robust against unexpected changes in the arrival order that may occur.

A natural question arises: what is the ratio between the performance of *order-unaware* and *order-aware* algorithms? This essentially suggests a natural and more realistic benchmark for Bayesian online problems — the best order-aware online algorithm. Such an algorithm still makes decisions online, but has an informational advantage over order-unaware algorithms, as it knows the arrival order in advance and can use this information to possibly make better decisions. This leads us to the definition of a new performance measure for Bayesian online settings — the *order-competitive ratio*.

**Order-competitive ratio.** We introduce the *order-competitive ratio*, defined as the worst-case ratio (over all distribution sequences) between the performance of the best *order-unaware* algorithm and the best *order-aware* algorithm)<sup>1</sup>. Thus, the order-competitive ratio **quantifies the loss that is incurred by Bayesian online algorithms due to unknown arrival order**.

Recent years have seen a growing interest in the effect of the arrival order on the performance of Bayesian online algorithms. Indeed, various arrival models have been studied, ranging from adversarial order [KS77, KS78, SC84], to random arrival order (a variant known as prophet secretary, as it combines the Bayesian assumption of prophet with the random order of secretary) [EHL17, ACK18, EHK18, CSZ21], all the way to “free order”, where the algorithm may dictate the arrival order [BGL<sup>+</sup>18, ASZ20, PT22]. It is quite surprising that despite this surge of interest, the loss incurred due to lack of information about the arrival order (or put differently, the power obtained by knowledge of the arrival order) has not been studied.

To see that knowledge of the arrival order may be useful, consider the following example.

**EXAMPLE 1.1.** (see Figure 1) Suppose there are three boxes. Two boxes have deterministic values  $\sqrt{2}$  and 1, respectively, and one box has value  $1/\epsilon$  with probability  $\epsilon$  (and 0 otherwise). Suppose further that the objective is to maximize the expected accepted value. Suppose the first observed value is  $\sqrt{2}$ , and an order-unaware algorithm ALG needs to decide immediately whether to accept it. If ALG accepts it, the next arriving value is the random one, followed by 1 (Figure 1 (a)); else, the order flips, and 1 arrives next, followed by the random box (Figure 1 (b)).

<sup>1</sup>Note that this new measure uses the same benchmark as the one in [PPSW21, NSS18], but unlike [PPSW21], our algorithms are order-unaware, thus enjoy the robustness of order-unaware algorithms.

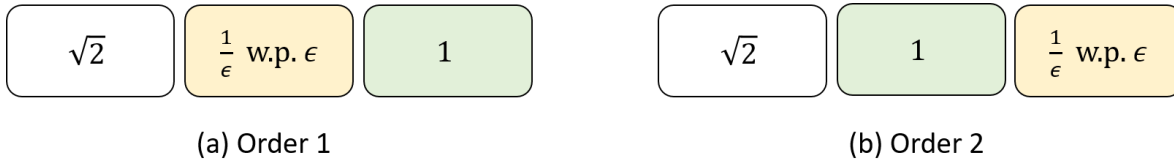


Figure 1: An example showing an upper bound of  $1/\sqrt{2}$  on the order-competitive ratio of deterministic algorithms.

Consider now the best online algorithm that knows the order. In the former case (where ALG accepted  $\sqrt{2}$ ), it rejects  $\sqrt{2}$ , and accepts the second value iff it is positive, gaining an expected value of  $\sim 2$ . The order-competitive ratio is then  $1/\sqrt{2}$ . In the latter case (where ALG rejected  $\sqrt{2}$ ), it accepts  $\sqrt{2}$ , while ALG gets an expected value of 1, leading again to an order-competitive ratio of  $1/\sqrt{2}$ .

Example 1.1 shows that the order-competitive ratio cannot be better than  $1/\sqrt{2} \approx 0.707$  (for deterministic algorithms). Clearly, it is at least  $1/2$ , as the order-competitive ratio is always weakly greater than the classic competitive ratio. It follows that the order-competitive ratio lies somewhere in the interval  $[\frac{1}{2}, \frac{1}{\sqrt{2}}]$ . As we show in this paper, the tight answer is the inverse of the golden ratio.

**Objective functions.** The prophet inequality problem has been studied with respect to two well-motivated objective functions, namely (i) maximizing the expected value (hereafter, max-expectation objective); and (ii) maximizing the probability to catch the maximum value (hereafter, max-probability objective).

The max-expectation objective has been the subject of our discussion thus far; the max-probability objective has been first studied for the special case of i.i.d. random variables [GM66], and more recently in its general form [EHL20], both with respect to the classical competitive ratio measure. This objective resembles the famous secretary problem [Fer89] (but with values drawn from known distributions, and under an adversarial order, where the secretary problem considers adversarial values arriving in a random order). The prophet, who can see into the boxes, can trivially accept the maximum value with probability 1, and so the competitive ratio of an algorithm is precisely its probability of catching the maximum value. Esfandiari et al. [EHL20] devised an order-unaware single-threshold algorithm that gives a tight (up to lower-order terms) competitive ratio of  $1/e$ .

**Adaptive vs. single-threshold algorithms.** The discussion above suggests that, when evaluated according to the competitive ratio measure, for both objective functions, single-threshold algorithms are optimal (providing tight  $1/2$  and  $1/e$  competitive ratios, respectively). Thus, the competitive ratio measure fails to differentiate between single-threshold and adaptive algorithms for Bayesian online settings. And yet, intuition suggests that adjusting one's decisions to the observed input may be quite useful. This intuition is nicely captured by the order-competitive ratio. Indeed, as we show in Section 1.1, the order-competitive ratio of adaptive algorithms is strictly better than the order-competitive ratio of single-threshold algorithms. In this respect, the classical competitive ratio may provide misleading algorithmic advice.

**Single-threshold algorithms.** Single-threshold algorithms are particularly appealing, due to their simplicity and robustness. They also correspond to a particularly appealing policy in a natural market scenario captured by the prophet inequality problem, described as follows: Consider a single-item sale, in a market with  $n$  buyers, each with value  $v_t$  drawn from a known distribution  $F_t$ . The agents arrive to the market in an online fashion, revealing their values upon arrival. The seller needs to determine a selling policy. In this scenario, a single-threshold algorithm corresponds to a fixed price that is posted from the outset, where the first buyer whose value exceeds the price buys the item.

Interestingly, the single-threshold algorithms that obtain the  $1/2$  and  $1/e$  competitive ratios for the two objectives, respectively, are order-unaware. However, single-threshold algorithms may be order-aware. In this paper we study the order-competitive ratio with respect to this class of algorithms as well. That is, we compare the performance of the best *order-unaware* single-threshold algorithm to that of the best *order-aware* single-threshold algorithm.

**1.1 Our Results** Our results are summarized in Table 1. We study the order-competitive ratio with respect to two objective functions, namely, (i) maximizing the expected value (left column), and (ii) maximizing the

		Objective function	
		max-expectation	max-probability
Algorithm type	Adaptive	$\frac{1}{\phi} \sim 0.618$	$\sim 0.806$
	Single-threshold	$\frac{1}{e} \sim 0.618$	$\sim 0.646$

Table 1: Our results: tight (with respect to deterministic algorithms) order-competitive ratios. Note that the competitive ratios in the two rows are defined with respect to different benchmarks (best adaptive and best single-threshold, respectively).

probability to obtain the maximum value (right column); and with respect to two families of algorithms, namely, (i) adaptive algorithms (top row), and (ii) single-threshold algorithms (bottom row). We provide tight bounds for all four combinations, with respect to deterministic algorithms.

For example, the top left cell in the table shows that the order-competitive ratio with respect to the max-expectation objective function and adaptive deterministic algorithms is  $1/\phi \approx 0.618$  (where  $\phi$  is the golden ratio). Namely, there exists a deterministic online *order-unaware* algorithm that obtains an expected value of at least  $1/\phi \approx 0.618$  of the best online *order-aware* algorithm. Moreover, this is tight (with respect to deterministic algorithms).

This result is a significant improvement over the competitive ratio of  $1/2$ , as measured against the standard prophet benchmark. As observed by [NSS18], this result cannot be obtained by a single-threshold algorithm. Indeed, our algorithm is more complex and is inevitably adaptive. The other results in the table should be interpreted analogously, with respect to the corresponding objective functions (max-expectation or max-probability) and families of algorithms (adaptive or single-threshold).

Surprisingly, the same tight bound of  $1/\phi \approx 0.618$  applies also for the family of single-threshold algorithms (bottom left cell), comparing the maximum expected value of the best deterministic online order-unaware single-threshold algorithm to that of the best online order-aware single-threshold algorithm. To the best of our understanding, the two results are unrelated. Indeed, despite obtaining the exact same ratio, they are derived using different analysis and techniques.

For the max-probability objective, we give a tight bound of  $\sim 0.806$  for adaptive algorithms (top right cell), and a tight bound of  $\sim 0.646$  for the family of single-threshold algorithms (bottom right cell). The former result is a huge improvement over the competitive ratio of  $1/e$  [EHL20], with respect to the standard prophet benchmark (which is trivially 1, as the prophet can always select the box with the maximum value). Furthermore, the upper bound of 0.646 on the order-competitive ratio for the single-threshold algorithms implies that the best adaptive algorithm with order-competitive ratio 0.806 cannot be single-threshold as opposed to the optimal algorithm of [EHL20] for the prophet benchmark.

Recall that for the standard competitive ratio measure, the max-expectation objective yields better results than the max-probability objective. Indeed, the competitive ratio with respect to max-expectation is tightly  $1/2$ , while it is tightly  $1/e$  with respect to max-probability. Interestingly, for the order-competitive ratio the order flips; namely, the max-probability objective exhibits better order-competitive ratios than the max-expectation objective.

**Our Techniques.** The algorithms for all of our settings use novel policies that depart from standard analysis of prophet inequalities. The derivation of these algorithms has been obtained by simultaneously considering policies that provide good bounds and analyzing their corresponding worst-case instances, and can be viewed as an application of the primal-dual approach. For simplicity of presentation, however, in our positive results we avoid explicit references to the worst-case instances, and focus instead on directly obtaining the essential inequalities. All worst-case instances turn out to consist of sequences of Bernoulli random variables with values  $\{0, a_i\}$ , with vanishingly small probabilities of  $a_i$  (in some cases we could simplify those instances to have only a constant number of boxes). On the other hand, each of the four settings we consider requires a unique set of ideas, as we briefly discuss below.

First, our unrestricted algorithms for both max-expectation and max-probability objectives utilize *adaptive thresholds*. Adaptive thresholds have been also used for multi-choice prophet inequality [KW12]. However, the analysis is fundamentally different since in our case the value of the benchmark is unknown, and may change depending on the arrival order. Part of the challenge is to gradually learn the benchmark and adjust the algorithm

accordingly. This is in contrast to the prophet benchmark which is known from the outset.

Our approach for the max-expectation objective includes two novel ideas. First, we combine two different static threshold policies. Namely, at each step of the algorithm we take the *higher of two* thresholds, which are variants of known thresholds for the classic prophet inequality. The first threshold is  $\tau_1 = \frac{1}{\phi} \mathbf{E}_{\bar{v}}[\max_i v_i]$ ; the second threshold is  $\tau_2$  such that  $\mathbf{E}_{\bar{v}}[(\max_i v_i - \phi\tau_2)^+] = \tau_2$ . While the latter threshold is less known, a variant of it has already appeared in [SC84]. The first threshold ( $\tau_1$ ) is an estimation of the expected reward under the best order (scaled by  $1/\phi$ ). In some cases, this threshold is too low. To this end, we consider also a second threshold ( $\tau_2$ ), which is a scaled estimation of the expected reward under the worst order. Second, since the benchmark is unknown, we use adaptive variants of the corresponding thresholds. Namely, at each stage we define the thresholds with respect to the remaining boxes only. Our algorithm essentially balances the two thresholds, in an adaptive way, to reflect the information we've gained about the (unknown) benchmark. We are unaware of prior work on prophet inequalities that uses the better of two strategies at every individual step of the algorithm.

For the max-probability objective, our analysis proceeds by comparing the behavior of order-aware and order-unaware algorithms box-by-box, similar to the case of max-expectation. On the other hand, in the worst-case instance there is a deterministic box that comes first and the algorithm needs to commit whether to accept it or not. If the order-unaware algorithm decides to select it, then the other boxes arrive in the optimal order (decreasing values). Otherwise, the boxes arrive in the worst order (increasing values). The hard instance is constructed by balancing these two cases.

In the case of single-threshold algorithms, each one of the order-aware and order-unaware algorithms can be described by a single real number (the corresponding threshold), leading to an analysis that involves 2 parameters. By fixing these two thresholds, we obtain a relatively simple class of worst-case Bernoulli instances that minimizes the performance of order-unaware algorithms while maximizing the performance of order-aware ones. (We do so separately for each one of our objective functions). This process essentially induces a max-min optimization problem on the threshold values. The most technically-involved part of our analysis is the solution of the induced max-min optimization problems.

## 1.2 Related Work

**Different arrival models.** Besides the adversarial order, random arrival order, and free order settings that we discussed above, another recent study related to the arrival order in prophet inequality settings has shown that for any arrival order  $\pi$ , the better of  $\pi$  and the reverse order of  $\pi$  achieves a competitive ratio of at least the inverse of the golden ratio, namely  $1/\phi \approx 0.618$  [ADK21]. To the best of our understanding, while our first main result obtains the exact same ratio, the two results are unrelated.

**Alternative benchmarks.** Considering alternative benchmarks to the “prophet” (i.e., optimal offline) benchmark has attracted a lot of interest recently [KSSW21, NSS18, PPSW21].

For example, [NSS18] quantify the loss due to single-threshold algorithms by the worst-case ratio between the best single-threshold algorithm and the best adaptive online algorithm (single-threshold or not), both under a known order, and show that the  $1/2$  ratio is tight.

As another example, [PPSW21] consider the problem of online matching in bipartite graphs. This problem is known to admit a  $1/2$ -competitive algorithm with respect to the prophet benchmark [FGL15]. But [PPSW21] propose to study the ratio between the optimal polynomial (order-aware) algorithm and the optimal computationally-unconstrained (order-aware) algorithm, and show that this ratio exceeds  $1/2$ . (Note that this question makes sense in the matching variant, where the optimal online algorithm for matching is computationally hard even for known order.) The ratio is then improved to  $0.526$  by [SW21] and to  $1 - 1/e$  by [BDL22].

**Beyond single-choice settings.** A related line of work, initiated by [Ken85, Ken87, Ker86], extends the single choice optimal stopping problem to multiple-choice settings. More recent work extended it to additional combinatorial settings, including matroids [KW19, AKW14], polymatroids [DK15]), matching [GW19, EFGT20], combinatorial auctions [FGL15, DFKL20, DKL20], and general downward closed feasibility constraints [Rub16].

**Limited information models.** Prophet inequality problems have been also studied under limited information about the underlying distributions, where the emphasis is on the sample complexity of the problem [AKW14, CDFS19, CCES20, EFN18, RWW19, KNR20, CCF<sup>+</sup>21, DLPLV21, CDF<sup>+</sup>22, KNR22].

## 2 Model and Preliminaries

Consider a setting with  $n$  boxes. Every box  $t$  contains some value  $v_t$  drawn from an underlying independent distribution  $F_t$ . The underlying distributions are known from the outset, but the values are revealed sequentially in an online fashion. For convenience of notations, we assume that the boxes arrive in an order  $1, 2, \dots, n$ . I.e., at stage  $t$ , we observe the realized value  $v_t = \theta_t$ , where  $v_t \sim F_t$ , and the identity of the arriving box. It will be clear from the context whether the order of arrival is assumed to be known. In any case, the identity of the arriving box is known. We denote by  $\mathbf{F} = \prod_t F_t$  the (product) distribution of the value profile  $\vec{v} = (v_1, \dots, v_n)$ . Upon the arrival of value  $v_t = \theta_t$ , the algorithm needs to decide, immediately and irrevocably, whether to accept the box.

We consider two different objectives: (i) maximizing the expected value of the accepted box, and (ii) maximizing the probability of catching the box with the largest value.

An online algorithm is said to be *order-aware* if it knows the arrival order of the boxes from the outset, and is said to be *order-unaware* if it doesn't.

Our goal is to measure the performance of order-unaware algorithms against that of the best online order-aware algorithm. Our order-unaware algorithm will be denoted by  $\text{ALG}$ , and the best order-aware algorithm by  $\text{OPT}$ .

Given an arrival order  $\pi$  and a value profile  $\vec{v}$ , we denote by  $\text{ALG}(\vec{v}, \pi)$  the value accepted by  $\text{ALG}$  under values  $\vec{v}$  and arrival order  $\pi$ , and by  $\text{OPT}(\vec{v}, \pi)$  the value accepted by  $\text{OPT}$  under  $\vec{v}, \pi$ .

We denote by  $\text{ALG}(\pi)$  the performance of  $\text{ALG}$ . For the first objective, it is the expected accepted value, i.e.,  $\text{ALG}(\pi) = \mathbf{E}_{\vec{v} \sim \mathbf{F}}[\text{ALG}(\vec{v}, \pi)]$ . For the second objective, it is the probability of catching the maximum value, i.e.,  $\text{ALG}(\pi) = \Pr_{\vec{v} \sim \mathbf{F}}[\text{ALG}(\vec{v}, \pi) = \max_i v_i]$ . When studying the objective of catching the maximum value, we assume that the maximum value is unique with probability 1 (as in the case where, e.g., the supports of the distributions are disjoint, or where the distributions are atomless).

The order-competitive ratio of an order-unaware algorithm  $\text{ALG}$  measures the loss in performance due to unknown order. It is defined as the worst-case ratio of the performance of  $\text{ALG}$  and the performance of  $\text{OPT}$ , over all arrival orders.

**DEFINITION 2.1.** *The order-competitive ratio of an order-unaware algorithm  $\text{ALG}$  is*

$$\Gamma(\text{ALG}) = \min_{\pi} \frac{\text{ALG}(\pi)}{\text{OPT}(\pi)}$$

**2.1 Single-Threshold Algorithms** We also consider an important restricted family of algorithms, namely single-threshold algorithms. A single-threshold algorithm  $\text{ALG}_{\tau}$ , parameterized by a single number<sup>2</sup>  $\tau \geq 0$ , stops at the first box  $i$  with  $v_i \geq \tau$ . We denote by  $\text{ALG}_{\tau}(\pi)$  the algorithm's expected value for the arrival order  $\pi$ . I.e.,  $\text{ALG}_{\tau}(\pi) = \mathbf{E}_{\vec{v} \sim \mathbf{F}}[\text{ALG}_{\tau}(\vec{v}, \pi)]$ .

When studying the objective of catching the maximum value, we assume that the distributions are atomless, i.e., every cumulative distribution function is continuous. As standard (see, e.g., [EHL20]), we extend the definition of single-threshold algorithms for discrete distributions by allowing the algorithm to randomize when the realized value  $v_i$  equals to the threshold  $\tau$ .

We note that the threshold that gives the best performance may depend on the arrival order  $\pi$  and consequently there is a gap between order-unaware and order-aware single-threshold algorithms. We define the order-competitive ratio for a single-threshold algorithm  $\text{ALG}_{\tau}$ , with respect to the class of single-threshold algorithms, as the worst-case (over all arrival orders  $\pi$ ) ratio between  $\text{ALG}_{\tau}(\pi)$  and  $\text{ALG}_{\tau'}(\pi)$ , for any single threshold  $\tau'$  (where the threshold  $\tau'$  may depend on  $\pi$ , but  $\tau$  is unaware of  $\pi$ ).

**DEFINITION 2.2.** *The order-competitive ratio of a single-threshold algorithm  $\text{ALG}_{\tau}$ , with respect to the class of single threshold algorithms, is*

$$\Gamma_{ST}(\text{ALG}_{\tau}) = \inf_{\pi, \tau'} \frac{\text{ALG}_{\tau}(\pi)}{\text{ALG}_{\tau'}(\pi)}.$$

Recall that if one compares the performance of a single-threshold algorithm to the best online algorithm, then a competitive ratio of  $1/2$  is tight, even if the arrival order is known [NSS18].

<sup>2</sup>More generally, the threshold  $\tau$  can be chosen at random. But we only consider deterministic single-threshold algorithms.

### 3 Maximizing the Expected Value

In this section we study the objective function of maximizing the expected value. Section 3.1 gives the order-competitive ratio with respect to adaptive algorithms, and Section 3.2 gives the order-competitive ratio with respect to single-threshold algorithms.

**3.1 Adaptive Algorithms** Our main result in this section is a deterministic order-unaware algorithm that obtains a tight order-competitive ratio of the inverse of the golden ratio (i.e.,  $\frac{1}{\phi} \approx 0.618$ ) with respect to the objective of maximizing the expected value.

**An order-unaware algorithm.** For the convenience of notations, we assume the boxes arrive in a specific order from  $1, 2, \dots, n$ . I.e., at stage  $t$ , we observe  $v_t = \theta_t$  where  $v_t \sim F_t$ . It will be clear from the description of our algorithm that it is order-unaware. We define the following series of random variables

$$\text{Prophet in the future: } y_t \stackrel{\text{def}}{=} \max_{s>t} v_s.$$

At each step  $t \in [n]$ , our algorithm will use the larger of the following two thresholds

$$(3.1) \quad \alpha_t \stackrel{\text{def}}{=} \frac{1}{\phi} \cdot \mathbf{E}[y_t] \quad \beta_t \stackrel{\text{def}}{=} x \text{ satisfying } \mathbf{E}[(y_t - \phi \cdot x)^+] = x,$$

where  $\phi = \frac{\sqrt{5}+1}{2}$  is the golden ratio. When  $t = n$ ,  $\alpha_t = \beta_t = 0$ . Note that the equation defining  $\beta_t$  has a unique solution, since its LHS is a strictly decreasing continuous function in  $x$  that starts from a non-negative number when  $x = 0$  and goes to 0 for  $x = \infty$ , and the RHS is a strictly increasing continuous function in  $x$  that starts from 0 when  $x = 0$  and goes to  $\infty$  when  $x = \infty$ . Our algorithm ALG stops at box  $t$  if and only if the realized value  $\theta_t$  of  $v_t$  exceeds the threshold  $\tau_t$ , defined as follows

$$(3.2) \quad \tau_t \stackrel{\text{def}}{=} \max(\alpha_t, \beta_t).$$

Note that ALG is order-unaware, since it does not need to know the arrival order of the remaining boxes in order to calculate  $\alpha_t, \beta_t$ . Denote the expected value obtained by this algorithm as ALG and the one obtained by the best order-aware algorithm as OPT.

Our main theorem in this section is the following.

**THEOREM 3.1.** *For every arrival order  $\pi$ ,  $\text{ALG}(\pi) \geq \frac{1}{\phi} \text{OPT}(\pi)$ , where  $\phi$  is the golden ratio.*

*Proof.* Fix an order  $\pi$ . Let  $\text{ALG}_t$  denote the expected value of ALG when run only on the boxes from  $t$  to  $n$ . We first establish the following useful bound on the performance of  $\text{ALG}_t$  relative to the thresholds  $\alpha_t, \beta_t$ .

**LEMMA 3.1.** *It holds that  $\text{ALG}_{t+1} \geq \beta_t \geq \frac{1}{\phi^2} \mathbf{E}[y_t] = \frac{1}{\phi} \alpha_t$  for any  $t \in [n-1]$ .*

*Proof.* According to the definition of  $\beta_t$ ,  $\beta_t = \mathbf{E}[(y_t - \phi \cdot \beta_t)^+] \geq \mathbf{E}[y_t] - \phi \cdot \beta_t$ . Hence,  $\beta_t \geq \frac{1}{\phi+1} \mathbf{E}[y_t] = \frac{1}{\phi^2} \mathbf{E}[y_t]$ , which concludes the proof of the second inequality.

We next prove the first inequality by induction on the number of remaining boxes. We use  $t = n$  as the base case of our induction, which is satisfied since  $\alpha_n = \beta_n = 0$ . Assume that  $\text{ALG}_{t+1} \geq \beta_t$ . We need to prove that  $\text{ALG}_t \geq \beta_{t-1}$ . First, observe that

$$(3.3) \quad \begin{aligned} \text{ALG}_t &= \mathbf{E}_{v_t} [\mathbb{1}(v_t \geq \tau_t) \cdot v_t + \mathbb{1}(v_t < \tau_t) \cdot \text{ALG}_{t+1}] \\ &\geq \mathbf{E}_{v_t} [\mathbb{1}(v_t \geq \tau_t) \cdot v_t + \mathbb{1}(v_t < \tau_t) \cdot \beta_t] \geq \beta_t, \end{aligned}$$

where the first inequality holds by the induction hypothesis, and to obtain the second inequality we observe that  $\tau_t \geq \beta_t$ . Consider the difference function  $f(x) \stackrel{\text{def}}{=} \mathbf{E}[(y_{t-1} - \phi \cdot x)^+] - x$  (see Equation (3.1)).  $f$  is strictly decreasing, with  $f(x) = 0$  for  $x = \beta_{t-1}$ , by definition of  $\beta_{t-1}$ . Therefore, to prove that  $\text{ALG}_t \geq \beta_{t-1}$  it is sufficient

to prove that  $f(\text{ALG}_t) \leq 0$ . We have

$$\begin{aligned}
 f(\text{ALG}_t) &= \mathbf{E}_{y_{t-1}} \left[ (y_{t-1} - \phi \cdot \text{ALG}_t)^+ \right] - \text{ALG}_t \leq \mathbf{E}_{y_{t-1}} \left[ (y_{t-1} - \phi \cdot \beta_t)^+ \right] - \text{ALG}_t \\
 &= \mathbf{E}_{y_t, v_t} \left[ (\max(y_t, v_t) - \phi \cdot \beta_t)^+ \right] - \text{ALG}_t \\
 &\leq \mathbf{E}_{y_t, v_t} \left[ (\max(y_t, \tau_t) - \phi \cdot \beta_t)^+ + (v_t - \tau_t)^+ \right] - \text{ALG}_t \\
 &\leq \mathbf{E}_{y_t} \left[ (\max(y_t, \tau_t) - \phi \cdot \beta_t)^+ \right] + \mathbf{E}_{v_t} [(v_t - \tau_t)^+] - \mathbf{E}_{v_t} [\mathbb{1}(v_t \geq \tau_t) \cdot v_t + \mathbb{1}(v_t < \tau_t) \cdot \beta_t] \\
 &= \mathbf{E}_{y_t} \left[ (\max(y_t, \tau_t) - \phi \cdot \beta_t)^+ \right] - \mathbf{E}_{v_t} [\mathbb{1}(v_t \geq \tau_t) \cdot \tau_t + \mathbb{1}(v_t < \tau_t) \cdot \beta_t] \\
 (3.4) \quad &\leq \mathbf{E}_{y_t} \left[ (\max(y_t, \tau_t) - \phi \cdot \beta_t)^+ \right] - \beta_t,
 \end{aligned}$$

where the first inequality follows since  $\text{ALG}_t \geq \beta_t$  by Equation (3.3); the second inequality holds since  $(\max(a, b) - c)^+ \leq (\max(a, d) - c)^+ + (b - d)^+$  for any  $a, b, c, d \in \mathbb{R}$ ; to get the third inequality, we use the first part of Equation (3.3); the last inequality follows not only in expectation over  $v_t$  but for any fixed value  $\theta_t$  of  $v_t$ , as  $\tau_t = \max(\alpha_t, \beta_t) \geq \beta_t$ . Furthermore,

$$\begin{aligned}
 f(\text{ALG}_t) &\stackrel{(3.4)}{\leq} \mathbf{E}_{y_t} \left[ (\max(y_t, \tau_t) - \phi \cdot \beta_t)^+ \right] - \beta_t = \mathbf{E}_{y_t} \left[ \max \left\{ (y_t - \phi \cdot \beta_t)^+, (\tau_t - \phi \cdot \beta_t)^+ \right\} \right] - \beta_t \\
 &\leq \mathbf{E}_{y_t} \left[ (y_t - \phi \cdot \beta_t)^+ + (\tau_t - \phi \cdot \beta_t)^+ \right] - \beta_t = \left( \mathbf{E}_{y_t} \left[ (y_t - \phi \cdot \beta_t)^+ \right] - \beta_t \right) + (\tau_t - \phi \cdot \beta_t)^+ = 0,
 \end{aligned}$$

where the first inequality is precisely (3.4); the second inequality follows by observing that  $\max(a, b) \leq a + b$  for any  $a, b \in \mathbb{R}$ ; the last equality follows by observing that both terms equal 0. The first term  $(\mathbf{E}_{y_t} [(y_t - \phi \cdot \beta_t)^+] - \beta_t)$  equals 0 by the definition of  $\beta_t$ , and the second term  $(\tau_t - \phi \cdot \beta_t)^+$  equals 0 by recalling that  $\tau_t = \max(\alpha_t, \beta_t)$  and by the fact that  $\beta_t \geq \frac{1}{\phi} \alpha_t$  proved above. Thus  $f(\text{ALG}_t) \leq 0$  and  $\text{ALG}_t \geq \beta_{t-1}$ , which concludes the proof of the induction step.  $\square$

We are now ready to prove Theorem 3.1. We prove the statement of the theorem by induction on the total number of boxes  $n$ . For the base case ( $n = 1$ ),  $\alpha_1 = \beta_1 = 0$  and  $\text{ALG}_n = \text{OPT}_n$ . Suppose that the statement of the theorem holds for any  $n - 1$  boxes. Let  $\text{OPT}_t$  denote the expected value of the optimal order-aware algorithm on the boxes  $t, \dots, n$ . We shall prove the induction step that  $\text{ALG}_1 = \text{ALG} \geq \frac{1}{\phi} \text{OPT} = \frac{1}{\phi} \text{OPT}_1$  for the case of  $n$  boxes. By the induction hypothesis we have  $\text{ALG}_2 \geq \frac{1}{\phi} \text{OPT}_2$ . We consider four cases based on the realized value  $\theta_1$  of the first box. We denote by  $\text{ALG}(\theta_1)$  and  $\text{OPT}(\theta_1)$  the respective expected values of our algorithm and the optimal order-aware algorithm, given that the value in the first box is  $v_1 = \theta_1$ . We show that  $\text{ALG}(\theta_1) \geq \frac{1}{\phi} \text{OPT}(\theta_1)$  for any  $\theta_1$ .

**Case 1** Both ALG and OPT stop and take value  $\theta_1$ . Then,  $\text{ALG}_1(\theta_1) = \text{OPT}_1(\theta_1) = \theta_1$ .

**Case 2** ALG takes value  $\theta_1$  but OPT doesn't. Then,  $\text{ALG}(\theta_1) \geq \alpha_1 = \frac{1}{\phi} \mathbf{E}[y_1] \geq \frac{1}{\phi} \text{OPT}_2 = \frac{1}{\phi} \text{OPT}_1(\theta_1)$ , where the second inequality is since  $\text{OPT}_2$  cannot do better than the prophet on boxes  $t \in \{2, \dots, n\}$ .

**Case 3** OPT takes  $\theta_1$ , but ALG doesn't. It holds that  $\text{ALG}_1(\theta_1) = \text{ALG}_2 \geq \max(\beta_1, \frac{1}{\phi} \alpha_1) \geq \frac{1}{\phi} \max(\beta_1, \alpha_1) \geq \frac{1}{\phi} \text{OPT}_1$ , where the first inequality follows by Lemma 3.1, and the last inequality holds since ALG rejected  $\theta_1 < \tau_1$ , whereas OPT selected it (thus  $\text{OPT}(\theta_1) = \theta_1 < \tau_1 = \max(\beta_1, \alpha_1)$ ).

**Case 4** Neither ALG nor OPT takes  $\theta_1$ . Then,  $\text{ALG}(\theta_1) = \text{ALG}_2$  and  $\text{OPT}(\theta_1) = \text{OPT}_2$ , and the claim holds by the induction hypothesis.

Therefore,  $\text{ALG} = \mathbf{E}_{v_1}[\text{ALG}(v_1)] \geq \mathbf{E}_{v_1}[\frac{\text{OPT}(v_1)}{\phi}] = \frac{1}{\phi} \text{OPT}$ . This concludes the proof.  $\square$

We next show that the above bound is tight, namely that no order-unaware deterministic algorithm may achieve an order-competitive ratio better than the golden ratio  $\frac{1}{\phi} = \frac{2}{\sqrt{5}+1}$ .



**THEOREM 3.2.** *For the objective of maximizing the expected value, no deterministic order-unaware algorithm achieves a better order-competitive ratio than  $\frac{1}{\phi}$  in the worst case.*

*Proof.* Consider an instance that consists of a set of boxes with deterministic values  $\phi, \phi - \varepsilon, \phi - 2\varepsilon, \dots, 1$ , and a single random box *HV* (we call it a high variance box) with value  $1/\varepsilon$  realized with probability  $\varepsilon$  and value 0 realized otherwise. Let *ALG* be any given order-unaware deterministic algorithm. Let  $\pi$  be an arrival order where the deterministic boxes arrive first, in decreasing order:  $\phi, \phi - \varepsilon, \phi - 2\varepsilon, \dots, 1$ , followed by the *HV* box.

**Case 1** *ALG* accepts some deterministic value  $x > 1$ . Consider now another arrival order  $\pi_x$  which is the same as  $\pi$  up to the deterministic  $x$  box, but with the *HV* box arriving immediately after it, and followed by the remaining deterministic boxes  $x - \varepsilon, \dots, 1$  in any order. Then, *ALG* achieves value  $x$  (we slightly abuse notations, and denote it  $\text{ALG}(\pi_x) = x$ ), whereas *OPT* for  $\pi_x$  achieves  $\text{OPT}(\pi_x) = \varepsilon \cdot 1/\varepsilon + (1 - \varepsilon)(x - \varepsilon)$ , by waiting for the *HV* box and taking it when its realized value is  $1/\varepsilon$  (otherwise *OPT* takes  $x - \varepsilon$ ). As  $\varepsilon$  goes to 0,  $\text{ALG}/\text{OPT}$  goes to  $\frac{x}{1+x} \leq \frac{\phi}{1+\phi} = \frac{1}{\phi}$ , where the inequality follows since  $x \leq \phi$ .

**Case 2** *ALG* waits for the last deterministic item or the *HV* box. Then,  $\text{ALG}(\pi) = 1$ , whereas *OPT*( $\pi$ ) selects the first item (with value  $\phi$ ), leading to an order-competitive ratio of  $\frac{1}{\phi}$ .

□

**3.2 Single-Threshold Algorithms** In this section we study the order-competitive ratio with respect to single-threshold algorithms. Interestingly, the exact same bound of the inverse of the golden ratio holds also with respect to single-threshold algorithms. That is, we provide a single-threshold deterministic order-unaware algorithm whose ratio with respect to the optimal single-threshold order-aware algorithm is at least the inverse of the golden ratio (i.e.,  $\frac{1}{\phi} \approx 0.618$ ), and this is tight.

**An order-unaware algorithm.** Let  $\tau$  be the (unique) value satisfying

$$(3.5) \quad \mathbf{E} [v_{\max} \cdot \mathbb{1}(v_{\max} \geq \tau)] \geq \phi \cdot \tau \geq \mathbf{E} [v_{\max} \cdot \mathbb{1}(v_{\max} > \tau)],$$

where  $v_{\max} = \max_i v_i$  is the random variable equal to the maximum value of  $v_1, \dots, v_n$ .

**THEOREM 3.3.** *For every arrival order  $\pi$ , and for every threshold  $\tau'$ , the performance of the single-threshold algorithm  $\text{ALG}_{\tau}$  is at least  $\frac{1}{\phi}$ -competitive against the single-threshold algorithm  $\text{ALG}_{\tau'}$ . I.e., for every  $\pi, \tau'$ , it holds that*

$$\text{ALG}_{\tau}(\pi) \geq \frac{1}{\phi} \text{ALG}_{\tau'}(\pi).$$

*Proof.* Fixing the arrival order  $\pi$ , and the threshold  $\tau'$ , we first show that for the threshold  $\tau'' = \min(\tau', \text{ALG}_{\tau'}(\pi))$  it holds that  $\text{ALG}_{\tau''}(\pi) \geq \text{ALG}_{\tau'}(\pi)$ , i.e.,  $\tau''$  is at least as good as  $\tau'$ .

**LEMMA 3.2.**  $\text{ALG}_{\tau''}(\pi) \geq \text{ALG}_{\tau'}(\pi)$ .

*Proof.* If  $\tau' \leq \text{ALG}_{\tau'}(\pi)$ , then  $\tau'' = \tau'$  and  $\text{ALG}_{\tau''}(\pi) = \text{ALG}_{\tau'}(\pi)$ , in which case we are done.

Assume  $\tau' > \text{ALG}_{\tau'}(\pi)$ . Let  $\text{ALG}_{\tau'}^t(\pi) = \mathbf{E}[\text{ALG}_{\tau'}(\pi) \mid v_1, \dots, v_{t-1} < \tau']$  denote the expected value obtained by running the threshold  $\tau'$  given that boxes  $1, \dots, t - 1$  were below  $\tau'$ . We first establish the following useful bound on the performance of  $\text{ALG}_{\tau'}^t(\pi)$ .

**CLAIM 3.1.** *If  $\tau' > \text{ALG}_{\tau'}(\pi)$ , then for every  $t = 1, \dots, n$ , it holds that  $\text{ALG}_{\tau'}(\pi) \geq \text{ALG}_{\tau'}^t(\pi)$ .*

*Proof.* It holds that

$$\begin{aligned} \text{ALG}_{\tau'}(\pi) &= \mathbf{E} [\text{ALG}_{\tau'} \mid \exists i < t, v_i \geq \tau'] \cdot \mathbf{Pr} [\exists i < t, v_i \geq \tau'] + (1 - \mathbf{Pr} [\exists i < t, v_i \geq \tau']) \cdot \text{ALG}_{\tau'}^t(\pi) \\ &\geq \tau' \cdot \mathbf{Pr} [\exists i < t, v_i \geq \tau'] + (1 - \mathbf{Pr} [\exists i < t, v_i \geq \tau']) \cdot \text{ALG}_{\tau'}^t(\pi) \\ &\geq \text{ALG}_{\tau'}(\pi) \cdot \mathbf{Pr} [\exists i < t, v_i \geq \tau'] + (1 - \mathbf{Pr} [\exists i < t, v_i \geq \tau']) \cdot \text{ALG}_{\tau'}^t(\pi), \end{aligned}$$

where the first inequality follows by  $\mathbf{E}[\text{ALG}_{\tau'} \mid \exists i < t, v_i \geq \tau'] \geq \tau'$ , since *ALG* <sub>$\tau'$</sub>  must stop before  $t$  and accept value  $v \geq \tau'$  under the condition  $\exists i < t, v_i \geq \tau'$ . The second inequality follows by the assumption that  $\tau' > \text{ALG}_{\tau'}(\pi)$ . Since  $\tau' > \text{ALG}_{\tau'}(\pi)$ , we get that the probability that we stop by time  $t$  is less than 1, thus, by rearranging and dividing by  $1 - \mathbf{Pr}[\exists i < t, v_i \geq \tau']$  which is strictly positive, we get  $\text{ALG}_{\tau'}(\pi) \geq \text{ALG}_{\tau'}^t(\pi)$ , as desired. □

Next we show that  $\text{ALG}_{\tau''}(\pi) \geq \text{ALG}_{\tau'}(\pi)$ . Consider a fixed valuation profile  $\vec{v}$  and the following two situations.

First, if  $\text{ALG}_{\tau''}$  and  $\text{ALG}_{\tau'}$  accept the same box in  $\vec{v}$ , then clearly  $\text{ALG}_{\tau''}$  and  $\text{ALG}_{\tau'}$  obtain the same value.

Second, let  $t$  be the first index where  $\text{ALG}_{\tau''}$  and  $\text{ALG}_{\tau'}$  diverge. Since  $\tau'' = \text{ALG}_{\tau'}(\pi) < \tau'$ , it means that  $\text{ALG}_{\tau''}$  accepts box  $t$ , while  $\text{ALG}_{\tau'}$  doesn't. Let  $\text{ALG}_{\tau''}(v_1, \dots, v_t)$  and  $\text{ALG}_{\tau'}(v_1, \dots, v_t)$  denote the expected value of  $\text{ALG}_{\tau''}$  and  $\text{ALG}_{\tau'}$ , respectively, with fixed values  $v_1, \dots, v_t$ , and where expectation is taken over  $v_{t+1}, \dots, v_n$ . Since  $\text{ALG}_{\tau''}$  stops at box  $t$ , it holds that  $\text{ALG}_{\tau''}(v_1, \dots, v_t) \geq \tau''$ . Since  $\text{ALG}_{\tau'}$  doesn't stop at box  $t$ , it holds that  $\text{ALG}_{\tau'}(v_1, \dots, v_t) = \text{ALG}_{\tau'}^{t+1}(\pi) \leq \text{ALG}_{\tau'}(\pi) = \tau''$ , where the inequality follows by Claim 3.1. It follows that  $\text{ALG}_{\tau''}(\pi) \geq \text{ALG}_{\tau'}(\pi)$ , as desired.  $\square$

We now distinguish between two cases.

**Case 1:**  $\tau'' \geq \tau$ . We assume that  $\tau'' > \tau$  (when  $\tau'' = \tau$  we are done). Then

$$\text{ALG}_{\tau''}(\pi) \leq \mathbf{E} [v_{\max} \cdot \mathbb{1}(v_{\max} \geq \tau'')] \leq \mathbf{E} [v_{\max} \cdot \mathbb{1}(v_{\max} > \tau)] \leq \phi \cdot \tau.$$

Thus, it remains to show that  $\text{ALG}_{\tau}(\pi) \geq \tau$ . We next use that  $\text{ALG}_{\tau''}(\pi) \geq \text{ALG}_{\tau'}(\pi) \geq \tau''$ .

$$\begin{aligned} \tau'' &\leq \text{ALG}_{\tau''}(\pi) \\ &= \sum_{t=1}^n \Pr [v_1, \dots, v_{t-1} < \tau''] \cdot \mathbf{E} [(v_t - \tau'')^+] + (1 - \Pr [v_1, \dots, v_n < \tau'']) \cdot \tau'' \\ (3.6) \quad &= \sum_{t=1}^n \prod_{i=1}^{t-1} \Pr [v_i < \tau''] \cdot \mathbf{E} [(v_t - \tau'')^+] + \left(1 - \prod_{t=1}^n \Pr [v_t < \tau'']\right) \cdot \tau''. \end{aligned}$$

Rearranging, we get

$$\sum_{t=1}^n \prod_{i=1}^{t-1} \Pr [v_i < \tau''] \cdot \mathbf{E} [(v_t - \tau'')^+] \geq \tau'' \cdot \prod_{t=1}^n \Pr [v_t < \tau''],$$

which is equivalent to

$$\sum_{t=1}^n \mathbf{E} [(v_t - \tau'')^+] \prod_{i=t}^n \frac{1}{\Pr [v_i < \tau'']} \geq \tau''.$$

Since  $\tau < \tau''$  (by Case 1), the last inequality must hold also when replacing  $\tau''$  by  $\tau$ . Indeed, the LHS increases while the RHS decreases. That is:

$$(3.7) \quad \sum_{t=1}^n \mathbf{E} [(v_t - \tau)^+] \prod_{i=t}^n \frac{1}{\Pr [v_i < \tau]} \geq \tau.$$

Similar to the derivation of (3.7) from (3.6) with  $\tau''$  replaced by the threshold  $\tau$  and  $\text{ALG}_{\tau''}(\pi)$  by  $\text{ALG}_{\tau}(\pi)$ , inequality (3.7) implies that  $\text{ALG}_{\tau}(\pi) \geq \tau$ , which concludes the proof of the first case.

**Case 2:**  $\tau'' < \tau$ . Then

$$(3.8) \quad \text{ALG}_{\tau''}(\pi) \leq \text{ALG}_{\tau}(\pi) + \Pr [v_{\max} < \tau] \cdot \tau,$$

since for any valuation profile  $\vec{v}$  we have either  $\text{ALG}_{\tau}(\pi) \geq \text{ALG}_{\tau''}(\pi)$  whenever  $\text{ALG}_{\tau}$  stops (i.e.,  $v_{\max} \geq \tau$ ), or  $\text{ALG}_{\tau''}(\pi) < \tau$  whenever  $v_{\max} < \tau$  (i.e.,  $\text{ALG}_{\tau}$  does not stop). Furthermore,

$$\begin{aligned} \text{ALG}_{\tau}(\pi) &\geq \Pr [v_{\max} \geq \tau] \cdot \tau + \Pr [v_{\max} < \tau] \cdot \mathbf{E} [(v_{\max} - \tau)^+] \\ &\geq \Pr [v_{\max} < \tau] \cdot (\mathbf{E} [(v_{\max} - \tau)^+] + \Pr [v_{\max} \geq \tau] \cdot \tau) \\ (3.9) \quad &\geq \Pr [v_{\max} < \tau] \cdot \phi \cdot \tau, \end{aligned}$$

where the first inequality is by the standard decomposition of the expected value of a threshold algorithm  $\text{ALG}_{\tau}$  into revenue and surplus, and the last inequality is equivalent to Equation (3.5). Combining Equations (3.8) and (3.9), it follows that

$$\frac{\text{ALG}_{\tau''}(\pi)}{\text{ALG}_{\tau}(\pi)} \leq 1 + \frac{1}{\phi} = \phi.$$

$\square$

REMARK 3.1. We note that the performance guarantee of Theorem 3.3 applies to single threshold algorithms that select the first value that is strictly greater than  $T$  (rather than at least  $T$ ). This is since such an algorithm with threshold  $T$  can be interpreted as  $\lim_{T' \rightarrow T^+} \text{ALG}_{T'}$ .

We next show that the above bound is tight, namely that no single-threshold order-unaware deterministic algorithm may achieve an order-competitive ratio (with respect to the optimal single-threshold algorithm) better than the inverse of the golden ratio  $\frac{1}{\phi} = \frac{2}{\sqrt{5}+1}$ .

THEOREM 3.4. For the objective of maximizing the expected value, no deterministic single-threshold order-unaware algorithm achieves a better order-competitive ratio (with respect to the optimal single-threshold algorithm) than  $\frac{1}{\phi}$  in the worst case.

*Proof.* Consider an instance that consists of two boxes. One box with a deterministic value 1, the other with value  $\phi/\varepsilon$  with probability  $\varepsilon$  (and 0 otherwise). If the threshold is at most 1, then the deterministic box arrives first, and we get  $\text{ALG} = 1$  and  $\text{OPT} = \phi$ . If the threshold is greater than 1, then the deterministic box arrives last, and we get  $\text{ALG} = \phi$ , while  $\text{OPT} = \phi + 1$ . In any case the ratio is  $1/\phi$ .  $\square$

#### 4 Maximizing the Probability of Catching the Maximum Value

In this section we study the objective function of maximizing the probability to catch the maximum value. Section 4.1 gives the order-competitive ratio with respect to adaptive algorithms, and Section 4.2 gives the order-competitive ratio with respect to single-threshold algorithms.

**4.1 Adaptive Algorithms** Our main result in this section is a deterministic order-unaware algorithm that obtains an order-competitive ratio of  $\ln \frac{1}{\lambda} \approx 0.806$ , where  $\lambda \approx 0.4464$  is the unique solution to  $\frac{x}{1-x} = \ln \frac{1}{x}$ , with respect to the objective of maximizing the probability to catch the maximum value.

To prove this result, we consider a slightly more general game: let there be an extra number  $\theta$  given in advance, and our objective is to maximize the probability of catching the box with the largest value that exceeds  $\theta$ . If all boxes have values less than  $\theta$ , no algorithm wins.

From now on, we shall work on this variant of the problem. Observe that the original problem is a special case where  $\theta = 0$ .

**An order-unaware algorithm.** At round  $t \in [n]$ , let  $v_s = \theta_s$  be the realized values for each  $s \leq t$ . Let  $\theta_0 = \theta$ . We accept the current box  $v_t = \theta_t$  if it satisfies the following condition:

$$\theta_t = \max_{0 \leq s \leq t} \theta_s \quad \text{and} \quad \Pr_{v_{t+1}, \dots, v_n} \left[ \max_{t+1 \leq s \leq n} v_s < \theta_t \right] \geq \lambda,$$

where  $\lambda$  is the unique solution to  $\frac{x}{1-x} = \ln \frac{1}{x}$ .

Note that our algorithm is order-unaware since calculating the probability  $\Pr[\max_{t+1 \leq s \leq n} v_s < \theta_t]$  requires no information on the order of remaining boxes. Indeed, it is the probability that all remaining boxes have value less than  $\theta_t$ . We use  $\text{ALG}(\pi)$  to denote the probability of catching the maximum value that exceeds  $\theta$  by our order-unaware algorithm and  $\text{OPT}(\pi)$  to denote the winning probability of the best order-aware algorithm when the actual arrival order is  $\pi$ .

THEOREM 4.1. For every arrival order  $\pi$ , the probability of catching the maximum value of the algorithm satisfies  $\text{ALG}(\pi) \geq \ln \frac{1}{\lambda} \cdot \text{OPT}(\pi)$ .

*Proof.* For simplicity, we omit  $\pi$  and write  $\text{ALG}$  and  $\text{OPT}$  instead of  $\text{ALG}(\pi)$  and  $\text{OPT}(\pi)$ , respectively. We prove the statement by induction on the number of boxes. The base case when  $n = 1$  is trivial, since both our algorithm and the optimal algorithm would accept the first box with value  $v_1$  if and only if  $v_1 > \theta_0$ . Suppose the statement is correct for  $n - 1$  boxes and consider the case for  $n$  boxes. We shall prove that for any realized value of  $v_1$  the winning probability of our algorithm is at least  $\ln \frac{1}{\lambda}$  times the winning probability of the optimal algorithm.

Consider the four cases depending on the behavior of our algorithm and the optimal algorithm on the realization of the first box  $v_1$ .

**Case 1** both  $\text{ALG}$  and  $\text{OPT}$  accept. In this case,  $\text{ALG} = \text{OPT}$ .

**Case 2** both ALG and OPT reject. In this case, we update the current maximum to  $\theta_1 = \max(\theta_0, v_1)$  and apply the induction hypothesis to conclude the proof. We remark that this is the place where we need the generalized version of the problem.

**Case 3** ALG accepts and OPT rejects. Then we have

$$\text{ALG} = \Pr \left[ \max_{s \geq 2} v_s < \theta_1 \right] \geq \lambda \quad \text{and} \quad \text{OPT} \leq \Pr \left[ \max_{s \geq 2} v_s \geq \theta_1 \right] \leq 1 - \lambda,$$

due to the second condition of our algorithm. Therefore,  $\text{ALG} \geq \frac{\lambda}{1-\lambda} \cdot \text{OPT}$ .

**Case 4** ALG rejects and OPT accepts. In this case, we must have  $\theta_1 \geq \theta_0$  and  $\Pr[\max_{s \geq 2} v_s < \theta_1] < \lambda$ . Let  $2 \leq t \leq n$  be the index such that

$$(4.10) \quad \Pr \left[ \max_{s > t} v_s < \theta_1 \right] \geq \lambda \quad \text{and} \quad \Pr \left[ \max_{s \geq t} v_s < \theta_1 \right] < \lambda.$$

We define  $p_s = \Pr[v_s \geq \theta_1]$  for all  $s \geq 2$ . Since the optimal algorithm takes the first box with realized value of  $\theta_1$ , its winning probability is

$$(4.11) \quad \text{OPT} = \Pr \left[ \max_{s \geq 2} v_s < \theta_1 \right] = \prod_{2 \leq s \leq n} (1 - p_s).$$

Next, we analyze our algorithm by studying the following events  $A_s$  for  $t \leq s \leq n$ :

$$A_s \stackrel{\text{def}}{=} \{v_s \geq \theta_1 \text{ and } v_k < \theta_1, \forall k \neq s\}.$$

CLAIM 4.1. *Our algorithm wins if  $A_s$  happens for any  $t \leq s \leq n$ .*

*Proof.* It suffices to show that our algorithm accepts box  $s$ . Indeed, ALG does not stop before box  $s$  since all other boxes have values smaller than  $\theta_1$ , violating the first condition of our algorithm. The second stopping condition of ALG is satisfied for box  $s$  as

$$\Pr \left[ \max_{k > s} v_k < \theta_s \right] \geq \Pr \left[ \max_{k > s} v_k < \theta_1 \right] \geq \Pr \left[ \max_{k > t} v_k < \theta_1 \right] \geq \lambda,$$

where the last inequality is by the first inequality of (4.10).  $\square$

Finally, we conclude the proof of the case:

$$\begin{aligned} \text{ALG} &\geq \Pr \left[ \bigcup_{t \leq s \leq n} A_s \right] = \sum_{t \leq s \leq n} \left( p_s \cdot \prod_{\substack{2 \leq k \leq n \\ k \neq s}} (1 - p_k) \right) = \prod_{2 \leq k \leq n} (1 - p_k) \cdot \left( \sum_{t \leq s \leq n} \frac{p_s}{1 - p_s} \right) \\ &\geq \prod_{2 \leq k \leq n} (1 - p_k) \cdot \left( \sum_{t \leq s \leq n} \ln \frac{1}{1 - p_s} \right) = \prod_{2 \leq k \leq n} (1 - p_k) \cdot \ln \left( \frac{1}{\prod_{t \leq s \leq n} (1 - p_s)} \right) \geq \text{OPT} \cdot \ln \frac{1}{\lambda}. \end{aligned}$$

The first inequality follows by Claim 4.1. The second inequality holds since  $\frac{x}{1-x} \geq \ln \frac{1}{1-x}$  for any  $x \in [0, 1]$ . The last inequality follows from Equations (4.10) and (4.11).

This finishes the proof of the inductive step.  $\square$

We next show that the above result is tight, namely that no order-unaware deterministic algorithm may achieve an order-competitive ratio better than  $\ln \frac{1}{\lambda} \approx 0.806$ , where  $\lambda$  is the unique solution to  $\frac{x}{1-x} = \ln \frac{1}{x}$ .

**THEOREM 4.2.** *For the objective of catching the maximum value, no deterministic order-unaware algorithm achieves a better order-competitive ratio than  $\ln \frac{1}{\lambda} \approx 0.806$ .*

*Proof.* Consider an instance with  $n + 1$  boxes. One of the boxes has a deterministic value of  $1/2$ . Among the remaining  $n$  boxes, the  $i$ -th box (for  $i = 1, \dots, n$ ) has value  $v_i = i$  with probability  $\varepsilon$ , and value 0 otherwise, where  $n \rightarrow \infty$ , and  $\varepsilon$  is a small number that satisfies  $(1 - \varepsilon)^n = \lambda$ . Let the deterministic box come first.

- Suppose the algorithm accepts the deterministic box. Then the algorithm wins when all the remaining boxes have value 0, i.e.  $\text{ALG} = (1 - \varepsilon)^n = \lambda$ . Then, the remaining boxes arrive in decreasing order, i.e.,  $n, n - 1, \dots, 1$ . The optimal algorithm that knows the arrival order would reject the deterministic box and accept the first box that has non-zero value. This algorithm wins when at least one of the randomized boxes has non-zero value, i.e.  $\text{OPT} = 1 - (1 - \varepsilon)^n = 1 - \lambda$ . This gives us an order-competitive ratio of  $\frac{\lambda}{1 - \lambda}$ .
- Suppose the algorithm rejects the deterministic box. Then, the remaining boxes arrive in increasing order, i.e.,  $1, 2, \dots, n$ . We study the best online algorithm that knows the order afterwards. It is straightforward to check that 1) the optimal algorithm only accepts non-zero boxes; 2) if the optimal algorithm accepts the  $i$ -th box when  $v_i \neq 0$ , it should also accept the  $j$ -th box for all  $j \geq i$  when  $v_j \neq 0$ . I.e., such algorithm can be described by a single parameter  $s \in [n]$  and it would simply accept the first non-zero box after the  $s$ -th box. Its winning probability is

$$\begin{aligned} \text{ALG} &= \Pr[\text{exactly one randomized box of } \{s, s + 1, \dots, n\} \text{ has non-zero value}] \\ &= (n - s + 1) \cdot \varepsilon \cdot (1 - \varepsilon)^{n-s} \leq n \cdot \varepsilon \cdot (1 - \varepsilon)^{n-1}. \end{aligned}$$

The last expression approaches  $\lambda \ln \frac{1}{\lambda}$  as  $n \rightarrow \infty$ . On the other hand, the optimal order-aware algorithm OPT would simply accept the deterministic box and win with probability  $\lambda$  (if all remaining boxes have value 0). Again, this leads to an order-competitive ratio of  $\frac{\lambda \ln \frac{1}{\lambda}}{\lambda} = \ln \frac{1}{\lambda}$ .

□

**4.2 Single-Threshold Algorithms** In this section, we provide a single-threshold deterministic order-unaware algorithm whose ratio with respect to the optimal single-threshold order-aware algorithm (with respect to the objective of maximizing the probability to catch the maximum value) is at least  $\mu = \frac{\ln(1/\rho)}{\ln(1/\rho)+1} \approx 0.646$ , where  $\rho \in [0, 1]$  ( $\approx 0.1609$ ) is the solution to the following equation.

$$(4.12) \quad \min_{\rho' \in (\rho, 1)} \left( \frac{\rho}{\rho'} + \frac{\rho}{1 - \rho'} \cdot \ln \frac{\rho'}{\rho} \right) = \frac{\ln(1/\rho)}{\ln(1/\rho) + 1}$$

Moreover, the ratio is tight. Since the LHS (resp. the RHS) of Equation (4.12) is strictly increasing (resp., decreasing) in  $\rho$ , it admits a unique solution.

**An order-unaware algorithm.** Let  $\tau$  be the (unique) threshold value satisfying

$$(4.13) \quad \Pr \left[ \max_i v_i < \tau \right] = \rho.$$

The existence of  $\tau$  is guaranteed by the atomless assumption. We remark that this is the only use of the assumption throughout the proof.

**THEOREM 4.3.** *For every arrival order  $\pi$ , and for every threshold  $\tau'$ , the performance of the single-threshold algorithm  $\text{ALG}_\tau$  is at least  $\mu$ -competitive against the single-threshold algorithm  $\text{ALG}_{\tau'}$ . I.e., for every  $\pi, \tau'$ , it holds that*

$$\text{ALG}_\tau(\pi) \geq \mu \cdot \text{ALG}_{\tau'}(\pi).$$

*Proof.* We fix the arrival order  $\pi$  and the threshold  $\tau'$ . We omit  $\pi$  and write  $\text{ALG}_\tau$  and  $\text{ALG}_{\tau'}$  instead of  $\text{ALG}_\tau(\pi)$  and  $\text{ALG}_{\tau'}(\pi)$ , respectively, to simplify notations. Let  $\rho' = \Pr[\max_i v_i < \tau']$ . We consider the two following cases.

**Case 1:**  $\tau' \geq \tau$ . We assume that  $\tau' > \tau$  (when  $\tau' = \tau$  the statement is obviously true). Then

$$(4.14) \quad 1 - \rho' = \Pr \left[ \max_{1 \leq i \leq n} v_i \geq \tau' \right] \geq \text{ALG}_{\tau'} = \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau' \right] \cdot \Pr \left[ v_i \geq \tau', \max_{j > i} v_j < v_i \right].$$

We write  $\text{ALG}_{\tau}$  in a similar way to  $\text{ALG}_{\tau'}$ .

$$\begin{aligned} \text{ALG}_{\tau} &= \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau \right] \cdot \Pr \left[ v_i \geq \tau, \max_{j > i} v_j < v_i \right] = A + B, \quad \text{where} \\ A &\stackrel{\text{def}}{=} \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau \right] \cdot \Pr \left[ v_i \geq \tau', \max_{j > i} v_j < v_i \right] \\ B &\stackrel{\text{def}}{=} \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau \right] \cdot \Pr \left[ v_i \in [\tau, \tau'), \max_{j > i} v_j < v_i \right] \end{aligned}$$

Now, we have  $\Pr[\max_{j < i} v_j < \tau] \geq \Pr[\max_{j < i} v_j < \tau'] \cdot \frac{\Pr[\max_{j < i} v_j < \tau]}{\Pr[\max_{j < i} v_j < \tau']}$  in the  $A$  term, as  $\Pr[\max_{j < i} v_j < \tau'] = \Pr[\max_{j < i} v_j < \tau'] \cdot \Pr[\max_{j \geq i} v_j < \tau'] \geq \Pr[\max_{j < i} v_j < \tau'] \cdot \Pr[\max_{j \geq i} v_j < \tau]$ . Thus,

$$A \geq \frac{\Pr[\max_{j < i} v_j < \tau]}{\Pr[\max_{j < i} v_j < \tau']} \cdot \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau' \right] \cdot \Pr \left[ v_i \geq \tau', \max_{j > i} v_j < v_i \right] = \frac{\rho}{\rho'} \text{ALG}_{\tau'}.$$

In the  $B$  term,  $\Pr[v_i \in [\tau, \tau'), \max_{j > i} v_j < v_i] \geq \Pr[v_i \in [\tau, \tau')] \cdot \Pr[\max_{j > i} v_j < \tau]$ . Thus,

$$\begin{aligned} B &\geq \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau \right] \cdot \Pr[v_i \in [\tau, \tau')] \cdot \Pr \left[ \max_{j > i} v_j < \tau \right] \\ &= \Pr \left[ \max_{j < i} v_j < \tau \right] \cdot \sum_{1 \leq i \leq n} \frac{\Pr[v_i < \tau'] - \Pr[v_i < \tau]}{\Pr[v_i < \tau]} \geq \rho \cdot \sum_i \ln \frac{\Pr[v_i < \tau']}{\Pr[v_i < \tau]} = \rho \cdot \ln \frac{\rho'}{\rho}, \end{aligned}$$

where in the second inequality we use the fact that  $x \geq \ln(1+x)$  and the definition of  $\rho$ . By combining these bounds on  $A$  and  $B$  together we get

$$\text{ALG}_{\tau} \geq \frac{\rho}{\rho'} \cdot \text{ALG}_{\tau'} + \rho \cdot \ln \left( \frac{\rho'}{\rho} \right) \geq \text{ALG}_{\tau'} \cdot \left( \frac{\rho}{\rho'} + \frac{\rho}{1 - \rho'} \cdot \ln \left( \frac{\rho'}{\rho} \right) \right) \geq \mu \cdot \text{ALG}_{\tau'},$$

where the second inequality follows by (4.14).

**Case 2:**  $\tau' < \tau$ . We write a similar decomposition for  $\text{ALG}_{\tau'}$  as we did for  $\text{ALG}_{\tau}$  in case 1.

$$\begin{aligned} \text{ALG}_{\tau'} &= \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau' \right] \cdot \Pr \left[ v_i \geq \tau, \max_{j > i} v_j < v_i \right] + \\ &\quad \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau' \right] \cdot \Pr \left[ v_i \in [\tau', \tau), \max_{j > i} v_j < v_i \right] \\ &\leq \sum_{1 \leq i \leq n} \Pr \left[ \max_{j < i} v_j < \tau \right] \cdot \Pr \left[ v_i \geq \tau, \max_{j > i} v_j < v_i \right] + \Pr \left[ \max_i v_i < \tau \right] = \text{ALG}_{\tau} + \rho \end{aligned}$$

We also have the following lower bound on  $\text{ALG}_{\tau}$ .

$$\text{ALG}_{\tau} \geq \sum_{1 \leq i \leq n} \Pr \left[ v_i \geq \tau, \max_{j \neq i} v_j < \tau \right] = \rho \sum_i \frac{\Pr[v_i \geq \tau]}{\Pr[v_i < \tau]} \geq \rho \sum_i \ln \left( \frac{1}{\Pr[v_i < \tau]} \right) = \rho \cdot \ln \frac{1}{\rho},$$

where the second inequality follows by the fact that  $\frac{x}{1-x} > \ln(\frac{1}{1-x})$  for  $x \in [0, 1)$ . Therefore,

$$\begin{aligned} \text{ALG}_\tau &\geq \text{ALG}_{\tau'} \cdot \frac{\text{ALG}_\tau}{\text{ALG}_\tau + \rho} = \text{ALG}_{\tau'} \cdot \left(1 - \frac{\rho}{\text{ALG}_\tau + \rho}\right) \\ &\geq \text{ALG}_{\tau'} \cdot \left(1 - \frac{\rho}{\rho \ln(1/\rho) + \rho}\right) = \text{ALG}_{\tau'} \cdot \frac{\ln(1/\rho)}{\ln(1/\rho) + 1} = \mu \cdot \text{ALG}_{\tau'} \end{aligned}$$

□

We next show that the guarantee of Theorem 4.3 is tight.

**THEOREM 4.4.** *For the max-probability objective, no single-threshold order-unaware algorithm achieves a better order-competitive ratio (with respect to the optimal single-threshold algorithm) than  $\mu \approx 0.646$  in the worst case.*

*Proof.* Consider an instance that consists of  $n \rightarrow \infty$  boxes. For  $i = 1, \dots, n$ , the value of box  $i$  is  $i$  with probability  $\varepsilon$ , and 0 otherwise<sup>3</sup>, for  $\varepsilon \rightarrow 0$  such that  $\varepsilon \gg \frac{1}{n}$ . Consider an arbitrary single-threshold algorithm, and let  $T$  be its threshold. Without loss of generality, we assume that  $T$  is an integer and the single-threshold algorithm always accepts the box  $T$  when  $v_T = T$ , since  $\Pr[v_T = T] = \varepsilon \rightarrow 0$ . We distinguish between two cases, depending on the value of  $T$ .

**Case 1:** The threshold  $T$  satisfies  $(1 - \varepsilon)^{n-T} \leq \rho$ , where  $\rho$  is the solution to Equation (4.12). Then, let  $\rho' = (1 - \varepsilon)^{n-T}$ , let  $\rho^* = \arg \min_{\rho'' \in (\rho', 1)} \left(\frac{\rho'}{\rho''} + \frac{\rho'}{1-\rho''} \cdot \ln \frac{\rho''}{\rho'}\right)$ , and let  $T^* = \max\{T' \in [n] \mid (1 - \varepsilon)^{n-T'} \leq \rho^*\}$ . Since  $\rho^* \geq \rho'$ , it holds that  $T^* \geq T$ .

Consider an arrival order  $\pi$ , where

$$\pi_i = \begin{cases} \text{Box } i & \text{if } 1 \leq i < T^* \\ \text{Box } n + T^* - i & \text{if } i \geq T^* \end{cases}$$

Let  $X_1$  (respectively,  $X_2, X_3$ ) denote the random variable indicating the number of non-zero realized values below  $T$  (respectively, between  $T$  and  $T^*$ , and above  $T^*$ ). Thus,

$$\begin{aligned} \text{ALG}_T(\pi) &= \Pr[X_2 = 1 \wedge X_3 = 0] + \Pr[X_2 = 0 \wedge X_3 > 0] \\ &\approx (T^* - T) \cdot \varepsilon \cdot (1 - \varepsilon)^{T^*-T} \cdot (1 - \varepsilon)^{n-T^*} + (1 - \varepsilon)^{T^*-T} \cdot (1 - (1 - \varepsilon)^{n-T^*}) \\ &= (T^* - T) \cdot \varepsilon \cdot (1 - \varepsilon)^{n-T} + (1 - \varepsilon)^{T^*-T} - (1 - \varepsilon)^{n-T} \\ (4.15) \quad &\approx \varepsilon \cdot \rho' \cdot \log_{1-\varepsilon} \frac{\rho'}{\rho^*} + \frac{\rho'}{\rho^*} - \rho' \approx \rho' \ln \frac{\rho^*}{\rho'} + \frac{\rho'}{\rho^*} - \rho'. \end{aligned}$$

On the other hand,

$$(4.16) \quad \text{ALG}_{T^*}(\pi) = \Pr[X_3 > 0] \approx 1 - (1 - \varepsilon)^{n-T^*} \approx 1 - \rho^*.$$

Combining Equations (4.15) and (4.16), we get that

$$\frac{\text{ALG}_T(\pi)}{\text{ALG}_{T^*}(\pi)} \approx \frac{\rho' \ln \frac{\rho^*}{\rho'} + \frac{\rho'}{\rho^*} - \rho'}{1 - \rho^*} = \frac{\rho'}{\rho^*} + \frac{\rho'}{1 - \rho^*} \ln \frac{\rho^*}{\rho'} \leq \mu,$$

where the last inequality is by the definitions of  $\mu, \rho^*$ , since  $\rho' \leq \rho$ , and since the LHS of Equation (4.12) is an increasing function.

<sup>3</sup>This instance can be transformed into an instance with atomless distributions by adding a small noise of less than 1 to all values. The same proof applies to the noisy instance.

**Case 2:** The threshold  $T$  satisfies  $(1 - \varepsilon)^{n-T} > \rho$ . Then, let  $T^* = 1$ . Consider an arrival order  $\pi$ , where

$$\pi_i = \begin{cases} \text{Box } T + i & \text{if } 0 \leq i \leq n - T \\ \text{Box } n - i & \text{if } i > n - T. \end{cases}$$

Let  $Y_1$  (respectively,  $Y_2$ ) denote the random variable indicating the number of non-zero realized values of at least  $T$  (respectively, below  $T$ ). Thus,

$$\begin{aligned} \text{ALG}_T(\pi) &= \Pr[Y_1 = 1] \approx (n - T) \cdot \varepsilon \cdot (1 - \varepsilon)^{n-T} \\ (4.17) \quad &= \ln_{1-\varepsilon} \rho' \cdot \varepsilon \cdot \rho' \approx \frac{\rho'}{\ln(1/\rho')}. \end{aligned}$$

On the other hand,

$$\begin{aligned} \text{ALG}_{T^*}(\pi) &= \Pr[Y_1 = 1] + \Pr[Y_1 = 0 \wedge Y_2 > 0] \\ (4.18) \quad &\approx \frac{\rho'}{\ln(1/\rho')} + (1 - \varepsilon)^{n-T} = \frac{\rho'}{\ln(1/\rho')} + \rho'. \end{aligned}$$

Combining Equations (4.17) and (4.18), we get that

$$\frac{\text{ALG}_T(\pi)}{\text{ALG}_{T^*}(\pi)} \approx \frac{\frac{\rho'}{\ln(1/\rho')}}{\frac{\rho'}{\ln(1/\rho')} + \rho'} = \frac{\ln(1/\rho')}{\ln(1/\rho') + 1} \leq \mu,$$

where the last inequality is by the definition of  $\mu$ , since  $\rho' > \rho$ , and since the RHS of Equation (4.12) is a decreasing function.  $\square$

## 5 Open Problems

Our model and results suggest natural problems for future research.

1. Our bounds are tight with respect to deterministic algorithms. Can randomized algorithms provide better ratios?
2. Study the order-competitive ratio in combinatorial settings, where multiple elements can be accepted, subject to feasibility constraints. A clear candidate is matroid feasibility constraints, for which the competitive ratio of  $1/2$  with respect to the prophet benchmark carries over [KW12]. Our bounds for the expected value objective carry over to simple matroid settings, such as partition matroids (where a single element is chosen from each part). What is the order-competitive ratio for general matroids?
3. More generally, we believe that the order-competitive ratio is a meaningful measure, which captures the significance of knowing the arrival order in Bayesian online settings. It would be interesting to apply it to other Bayesian settings.
4. The algorithm proposed by [PPSW21] for online bipartite matching is order-aware. Can the same result be obtained by an order-unaware algorithm?

## References

- [ACK18] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Prophet secretary: Surpassing the  $1-1/e$  barrier. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 303–318, 2018.
- [ADK21] Makis Arsenis, Odysseas Drosis, and Robert Kleinberg. Constrained-order prophet inequalities. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2034–2046. SIAM, 2021.
- [AKW14] Pablo D Azar, Robert Kleinberg, and S Matthew Weinberg. Prophet inequalities with limited information. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1358–1377. SIAM, 2014.



- [ASZ20] Shipra Agrawal, Jay Sethuraman, and Xingyu Zhang. On optimal ordering in the optimal stopping problem. In Péter Biró, Jason D. Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, pages 187–188. ACM, 2020.
- [BDL22] Mark Braverman, Mahsa Derakhshan, and Antonio Molina Lovett. Max-weight online stochastic matching: Improved approximations against the online benchmark, 2022.
- [BGL<sup>+</sup>18] Hedyeh Beyhaghi, Negin Golrezaei, Renato Paes Leme, Martin Pal, and Balasubramanian Sivan. Improved approximations for free-order prophets and second-price auctions. *arXiv preprint arXiv:1807.03435*, 2018.
- [CCES20] José R Correa, Andrés Cristi, Boris Epstein, and José A Soto. The two-sided game of googol and sample-based prophet inequalities. In *SODA*, pages 2066–2081. SIAM, 2020.
- [CCF<sup>+</sup>21] José Correa, Andrés Cristi, Laurent Feuilloley, Tim Oosterwijk, and Alexandros Tsigonias-Dimitriadis. The secretary problem with independent sampling. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2047–2058. SIAM, 2021.
- [CDF<sup>+</sup>22] Constantine Caramanis, Paul Dütting, Matthew Faw, Federico Fusco, Philip Lazos, Stefano Leonardi, Orestis Papadigenopoulos, Emmanouil Pountourakis, and Rebecca Reiffenhäuser. Single-sample prophet inequalities via greedy-ordered selection. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1298–1325. SIAM, 2022.
- [CDFS19] José Correa, Paul Dütting, Felix Fischer, and Kevin Schewior. Prophet inequalities for iid random variables from an unknown distribution. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC*, pages 3–17. ACM, 2019.
- [CSZ21] José R. Correa, Raimundo Saona, and Bruno Ziliotto. Prophet secretary through blind strategies. *Math. Program.*, 190(1):483–521, 2021.
- [DFKL20] Paul Dütting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM J. Comput.*, 49(3):540–582, 2020.
- [DK15] Paul Dütting and Robert Kleinberg. Polymatroid prophet inequalities. In *Algorithms-ESA 2015*, pages 437–449. Springer, 2015.
- [DKL20] Paul Dütting, Thomas Kesselheim, and Brendan Lucier. An  $o(\log \log m)$  prophet inequality for subadditive combinatorial auctions. *ACM SIGecom Exchanges*, 18(2):32–37, 2020.
- [DLPLV21] Paul Dütting, Silvio Lattanzi, Renato Paes Leme, and Sergei Vassilvitskii. Secretaries with advice. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 409–429, 2021.
- [EFGT20] Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. In *EC*, pages 769–787. ACM, 2020.
- [EFN18] Tomer Ezra, Michal Feldman, and Ilan Nehama. Prophets and secretaries with overbooking. In *Proceedings of the 2018 ACM Conference on Economics and Computation, EC*, pages 319–320. ACM, 2018.
- [EHKS18] Soheil Ehsani, MohammadTaghi Hajiaghayi, Thomas Kesselheim, and Sahil Singla. Prophet secretary for combinatorial auctions and matroids. In *Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms*, pages 700–714. SIAM, 2018.
- [EHLM17] Hossein Esfandiari, MohammadTaghi Hajiaghayi, Vahid Liaghat, and Morteza Monemizadeh. Prophet secretary. *SIAM Journal on Discrete Mathematics*, 31(3):1685–1701, 2017.
- [EHLM20] Hossein Esfandiari, MohammadTaghi Hajiaghayi, Brendan Lucier, and Michael Mitzenmacher. Prophets, secretaries, and maximizing the probability of choosing the best. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 3717–3727. PMLR, 2020.
- [Fer89] Thomas S Ferguson. Who solved the secretary problem? *Statistical science*, 4(3):282–289, 1989.
- [FGL15] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 123–135. SIAM, 2015.
- [GGLS08] Naveen Garg, Anupam Gupta, Stefano Leonardi, and Piotr Sankowski. Stochastic analyses for online combinatorial optimization problems. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 942–951. SIAM, 2008.
- [GM66] John P. Gilbert and Frederick Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61(313):35–73, 1966.
- [GW19] Nikolai Gravin and Hongao Wang. Prophet inequality for bipartite matching: Merits of being simple and non adaptive. In *EC*, pages 93–109. ACM, 2019.
- [HKS07] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI*, volume 7, pages 58–65, 2007.
- [Ken85] Douglas P Kennedy. Optimal stopping of independent random variables and maximizing prophets. *The Annals of Probability*, pages 566–571, 1985.

- [Ken87] Douglas P Kennedy. Prophet-type inequalities for multi-choice optimal stopping. *Stochastic Processes and their applications*, 24(1):77–88, 1987.
- [Ker86] Robert P Kertz. Comparison of optimal value and constrained maxima expectations for independent random variables. *Advances in applied probability*, 18(2):311–340, 1986.
- [KNR20] Haim Kaplan, David Naori, and Danny Raz. Competitive analysis with a sample and the secretary problem. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2082–2095. SIAM, 2020.
- [KNR22] Haim Kaplan, David Naori, and Danny Raz. Online weighted matching with a sample. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1247–1272. SIAM, 2022.
- [KS77] Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bulletin of the American Mathematical Society*, 83(4):745–747, 1977.
- [KS78] Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. *Probability on Banach spaces*, 4:197–266, 1978.
- [KSSW21] Kristen Kessel, Amin Saberi, Ali Shameli, and David Wajc. The stationary prophet inequality problem. *arXiv preprint arXiv:2107.10516*, 2021.
- [KW12] Robert Kleinberg and Seth Matthew Weinberg. Matroid prophet inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 123–136, 2012.
- [KW19] Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games and Economic Behavior*, 113:97–115, 2019.
- [NSS18] Rad Niazadeh, Amin Saberi, and Ali Shameli. Prophet inequalities vs. approximating optimum online. In George Christodoulou and Tobias Harks, editors, *Web and Internet Economics - 14th International Conference, WINE 2018, Oxford, UK, December 15-17, 2018, Proceedings*, volume 11316 of *Lecture Notes in Computer Science*, pages 356–374. Springer, 2018.
- [PPSW21] Christos H. Papadimitriou, Tristan Pollner, Amin Saberi, and David Wajc. Online stochastic max-weight bipartite matching: Beyond prophet inequalities. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 763–764. ACM, 2021.
- [PT22] Bo Peng and Zhihao Gavin Tang. Order selection prophet inequality: From threshold optimization to arrival time design. *to appear in FOCS*, 2022.
- [Rou21] Tim Roughgarden. *Beyond the worst-case analysis of algorithms*. Cambridge University Press, 2021.
- [Rub16] Aviad Rubinfeld. Beyond matroids: Secretary problem and prophet inequality with general constraints. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 324–332, 2016.
- [RWW19] Aviad Rubinfeld, Jack Z Wang, and S Matthew Weinberg. Optimal single-choice prophet inequalities from samples. *arXiv preprint arXiv:1911.07945*, 2019.
- [SC84] Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *the Annals of Probability*, pages 1213–1216, 1984.
- [SW21] Amin Saberi and David Wajc. The greedy algorithm is not optimal for on-line edge coloring. In *ICALP*, volume 198 of *LIPICs*, pages 109:1–109:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.