Multi Choice Min Prophet

Yossi Azar
Blavatnik School of Computer Science, Tel Aviv University, Israel

Itamar Biran
Blavatnik School of Computer Science, Tel Aviv University, Israel

Amos Fiat
Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract

The prophet inequality is a fundamental problem in optimal stopping theory. Given n independent variables drawn from known distributions, a player observes values sequentially and must decide irrevocably whether to stop and accept the current value or continue. The goal is to select a single element while maximizing the ratio between the value chosen and that of the maximum value in the sequence. In this paper, we study the minimization counterpart, often termed the min prophet or cost prophet inequality. Unlike the maximization setting, where simple threshold algorithms achieve half of the prophet's value, the minimization setting is significantly harder, with an exponential lower bound even for i.i.d. variables.

We study a multi-choice relaxation in which the algorithm may select multiple variables and gets to choose the best amongst them (the minimum amongst those selected). Our goal is to minimize the expected number of selections while achieving a constant competitive ratio. For adversarial order, we show that a constant competitive ratio requires a nearly linear number of choices in expectation, ergo, $\Omega(n/\ln n)$. In contrast, we show that for the prophet secretary model (random order) one can attain constant competitiveness while requiring only an exponentially smaller expected number of choices i.e. $O(\ln n)$. We give a refined analysis and define M to be the ratio of the minimum expected value of any single variable to the expected minimum value of all variables (the prophet's value) and present an algorithm that achieves a constant competitive ratio with $O(\min\{\ln\ln M, \ln n\})$ choices in expectation for the prophet secretary. We show that this is tight up to low order log factors even for the special case of the i.i.d. model. Specifically, the lower bound on the expected number of choices for any constant competitive algorithm is $\Omega(\min\{\ln\ln M/\ln\ln\ln M, \ln n/\ln\ln n\})$. We also show that if we insist on a deterministic bound on the number of choices then every constant competitive algorithm requires n choices. This holds even in the i.i.d. setting and shows that to achieve a constant competitive algorithm there is an exponential gap between the lower bound on the deterministic number of choices and the upper bound on the expected number of choices.

Finally, we consider a variant where both the algorithm and the adversary choose r values and pay their sum, this is the minimization multi unit version. We extend our techniques to the multi-unit variant for i.i.d. variables, achieving a constant competitive ratio with a small expected number of choices.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Prophet inequalities, Min Cost, Multiple choice, Multi Unit

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.55

Related Version *Full Version*: https://arxiv.org/abs/2607.05085

1 Introduction

Consider the following problem, there are n rewards $X_1, \ldots, X_n$ drawn independently from known distributions $F_1, \ldots, F_n$. The player observes the values one by one and may stop at any point, collecting the last observed reward. The goal is to maximize the payoff. This problem is well known and is called the prophet inequality [24, 25]. The optimum offline

34th Annual European Symposium on Algorithms (ESA 2026).
Editors: Philip Bille, Seth Pettie, and Sabine Storandt; Article No. 55; pp. 55:1–55:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

solution (prophet's expected payoff) is $\mathbb{E}[\max_j X_j]$. For general variables, the best online algorithms achieve at least half of the optimum prophet's value [24, 25, 29]. For random order (also called prophet secretary) the bound is better (between 0.688 [10] and 0.723 [17]). For i.i.d. variables, the optimal online algorithm achieves 0.745 [20, 22, 11] . This result has been extended in many directions, including multi-choice prophet inequalities [6], multi-unit prophet settings [3], and resource augmentation [8].

In this work, we study the minimization counterpart, sometimes called the cost prophet inequality [26]. Here the X_i's represent costs, and the player aims to minimize the selected value. For intuition, one may think of a traveler deciding when to buy a flight ticket: prices arrive sequentially, and upon seeing one the traveler must decide immediately whether to purchase it or continue waiting. We study the three main variants for prophet inequalities: adversarial order, prophet secretary (random order arrival), and i.i.d. variables.

One of the key differences between the minimization and maximization problems is that in minimization the algorithm must choose a variable, otherwise it incurs a cost of ∞. Unlike the reward version, the minimization problem behaves very differently: while for certain i.i.d. distributions one can obtain a constant competitive ratio for example distributions with monotone hazard rate [26], for general distributions it has been shown that even under i.i.d. assumptions only an exponential competitive ratio is achievable [15]. This striking gap motivates our investigation of what guaranties can be obtained without such distributional assumptions.

We consider the multi-choice prophet inequality, in which the algorithm can choose multiple items. We compare the expected cost of the algorithm on the minimum of its choices (which is the last choice[1]) to the prophet's cost. The case of a single choice corresponds to the classic cost prophet inequality where n corresponds to the offline version.

Our goal in this paper is to study how many choices are needed to achieve a constant competitive ratio for the min prophet problem, defined as the expected cost of the algorithm (the expected minimum of the choices made) over the expected cost of the optimal algorithm. We say that algorithm ALG is c-competitive if

$$\mathbb{E}(\mathrm{cost}(\mathrm{ALG})) \leq c \cdot \mathbb{E}(\mathrm{cost}(\mathrm{OPT})) = c \cdot \mathbb{E}[\min_j X_j].$$

Since the choices made by the algorithm are adaptive, we view the number of choices as a random variable, and we are interested in its expectation.

To state our results, we use the parameter M which captures the power of the minimum of n variables compared with a single variable. Specifically, this parameter is

$$M = \min_j \mathrm{E}\left[X_j\right] / \mathrm{E}\left[\min_j X_j\right].$$

Intuitively, M measures how much better the global minimum is, in expectation, over a single draw. For many light-tailed distributions M grows only polynomially with n, while for heavy-tailed distributions M can be much larger.

First, we show that for adversarial order, a large number of choices are required. Specifically, any constant competitive algorithm requires $\Omega(\frac{n}{\ln n})$ choices in expectation. Surprisingly, for the random order (prophet secretary model), one can get an exponential improvement. Our main result is that a constant competitive ratio is possible with $O(\min\{\ln \ln M, \ln n\})$

[1] Without loss of generality, the algorithm's cost can be taken as the value of its last accepted choice, since selecting a larger value than already chosen never improves the outcome.

choices in expectation, and this is the best possible (up to low order factors). Interestingly, the lower bound works even for i.i.d. variables (in which case the order is irrelevant). Moreover, we show that if we insist on a deterministic upper bound on the number of choices, then, to achieve a constant competitive algorithm n choices are needed, even for i.i.d. variables. The most technically challenging problem is the lower bound on the expected number of choices.

1.1 Our results

We obtain several new results for the min prophet inequality under the multi-choice variant and the multi-unit variant. Recall that n is the number of variables and M is the ratio between the minimum of the expectation and the expectation of the minimum.

Multi-choice setting

- For adversarial order, we establish that any constant competitive algorithm requires $\Omega(\frac{n}{\ln n})$ choices in expectation. See Section 3.
- For the prophet secretary model (variables that arrive at a random order), we present a simple threshold algorithm (Section 4) that achieves a constant competitive ratio using only $O(\min\{\ln\ln M, \ln n\})$ choices in expectation. To complement this result, we show that even in the i.i.d. setting, any constant-competitive algorithm requires at least

  $$\Omega\big(\min\left\{\tfrac{\ln\ln M}{\ln\ln\ln M}, \tfrac{\ln n}{\ln\ln n}\right\}\big) = \widetilde{\Omega}\left(\min\{\ln\ln M, \ln n\}\right)$$

  choices in expectation, establishing that our result is tight up to a low order factor. See Section 5.
- We show that "choices in expectation" above is required even for i.i.d. variables. We prove that no algorithm that deterministically makes less than n choices can be constant-competitive (see Section 6). In combination with the upper bound above, this shows an exponential gap between requiring a deterministic and an expected number of choices.

These results illustrate the fundamental gap between maximization and minimization prophet inequalities. In particular, while the classical reward setting admits constant-competitive algorithms without multiple choices, the cost setting provably requires multiple choices, and any non-trivial result is possible only under the relaxation that the number of choices holds in expectation. Moreover, in the maximization version, the competitive ratios for adversarial and random order settings differ by a constant factor. Conversely, to achieve any constant competitive ratio for the minimization problem, the two settings (adversarial vs. random order) are very different, and there is an exponential gap between the number of choices required in the two settings.

Multi-unit setting

We generalize the multi choice problem to the *multi unit min prophet problem*. In this setting, both the online algorithm and the adversary have to collect r items and may replace previously chosen items with other choices. One may think of this setting as having a series of tasks that need to be done, and one must do r of these tasks. The cost associated with the r items eventually chosen is the sum of their values.

The algorithm may make multiple choices but always keeps the r smallest cost items amongst them. As above, we are interested in the number of choices made in expectation and the competitive ratio attained. The multi unit problem reduces to the multi choice problem for $r = 1$.

We design a threshold-based algorithm, closely related to that of the multi-choice setting, that achieves a constant competitive ratio for i.i.d. variables while using only a small expected number of choices: If $r \le \ln M$, the algorithm achieves a constant competitive ratio with

$$O\big(\min\big\{r\ln(\tfrac{\ln M}{r}),\ r\ln(\tfrac{n}{r})\big\}\big)$$

choices in expectation. If $r \ge \ln M$, the algorithm achieves a constant competitive ratio with only $O(r)$ choices in expectation. See Appendix C. Moreover, for any $1 \le r \le n-1$, we show that if the algorithm is restricted to making r selections in total, i.e., no extra choices, then no constant competitive ratio is possible (even for i.i.d. inputs). See Appendix D.

1.2 Techniques

For the upper bound, we employ a threshold-based algorithm, calibrated so that the expected number of relevant samples (below the threshold) is logarithmic in M. We analyze the expected number of choices using record statistics, relying on Jensen's inequality to bound the updates to $O(\min\{\ln\ln M, \ln n\})$.

For our lower bounds, we construct hard instances using heavy-tailed distributions. For the adversarial order lower bound, we utilize n distinct heavy-tailed continuous distributions to demonstrate that an algorithm must switch its selection with high probability at nearly every step to avoid incurring a large cost. For the lower bound on the expected number of choices in the i.i.d. setting we divide the n variables into intervals of increasing size that depend on M and n, such that, in total, there are $\Omega(\min\{\ln\ln M/\ln\ln\ln M, \ln n/\ln\ln n\})$ intervals. We prove that any constant competitive algorithm must make a choice in each of these intervals with very high probability, thereby establishing the required lower bound. For the lower bound with deterministic number of choices, we construct a discrete distribution with super-exponentially growing values. We show that for any algorithm restricted to $n-1$ choices, there exists a specific realization sequence where the algorithm fails to select the minimum, leading to an unbounded competitive ratio.

1.3 Organization

The remainder of the paper is organized as follows. Section 2 formalizes the multi-choice minimization prophet setting. Section 3 shows that for adversarial order any constant-competitive algorithm must perform $\Omega(n/\ln n)$ choices in expectation. Section 4 presents our constant-competitive threshold algorithm and analyzes both its performance and the expected number of choices it makes. Section 5 complements this with a nearly tight lower bound on the expected number of choices required by any constant-competitive algorithm. Section 6 turns to deterministic choices, proving that any algorithm restricted to at most $n-1$ choices cannot achieve a constant competitive ratio. Some proof are deferred to the Appendices A, B. Appendices C and D focus on the multi-unit version of the problem: Appendix C presents an algorithm that is almost identical to the threshold-based algorithm of Section 4, that achieves a constant competitive ratio while using only a small expected number of choices, and Appendix D establishes an exponential lower bound, showing that when the algorithm is allowed only r selections, no constant competitive ratio can be guaranteed.

1.4 Related work

Min prophet inequality

The first study of the min prophet problem (also called the cost prophet inequality) was by Assaf et al. [5], who considered the case of two choices under certain distributions and compared the algorithm's performance with both the one-choice version and the prophet's value. In contrast to the maximization setting, where constant competitive ratios are achievable, their results already suggested that the minimization variant is significantly harder. After a long gap, Esfandiari et al. [15] established that even in the i.i.d. setting the competitive ratio can be exponential, highlighting a stark difference from the reward prophet inequality, where a ratio of 2 is attainable even for non i.i.d. distributions. This result shifted the research focus towards identifying structural assumptions – such as distribution classes or fractional cost models, under which constant competitive guarantees may still be possible.

Livanos and Mehta [26] analyzed the optimal algorithm for i.i.d. variables and showed that constant guarantees can be obtained under additional assumptions on the distributions, specifically through properties of the hazard rate. More recently, Livanos [27] extended this line of work using extreme value theory. For distributions where the maxima and minima of a sample converge in distribution, they proved that for single-threshold algorithms the optimal competitive ratio in the i.i.d. setting is polylogarithmic in n, and that in the k-multi-unit minimization prophet inequality there exist constant-competitive single-threshold algorithms whenever $k \geq \log n$.

Another related variant is the prophet inequality with monomial cost functions, in which each random variable corresponds to a cost function and the player may purchase a fractional amount $x_i \in [0, 1]$ from each, subject to the total mass summing to 1. Qin et al. [28] analyzed this model and, under mild regularity conditions on the distributions, proved that one can obtain a competitive ratio approaching 1 asymptotically. In our work, all proofs hold for arbitrary distributions.

Max prophet inequality

The classical prophet inequality was introduced by Krengel and Sucheston [24, 25], who showed that a gambler observing n independent random variables can guarantee at least half of the expected maximum. The optimal asymptotic competitive ratio for the i.i.d. prophet inequality is $\alpha \approx 0.745$, obtained by combining the Hill–Kertz upper bound [20] with the matching threshold-algorithm guarantee of Correa et al. [11].

An intermediate variant is the *prophet secretary* problem, where variables arrive in uniformly random order. Esfandiari et al. [15] designed an adaptive-threshold algorithm achieving a $(1 - 1/e)$-competitive ratio. Azar et al. and later Chen et al. [7, 10] improved this bound to 0.688. Correa et al. [12] established that no algorithm can beat 0.732. Further progress has been made for multiple selections [4] and for broader feasibility constraints such as matroids [13, 2]. Yet another model, the free-order setting (where the algorithm can choose the order) was also studied in [1].

The most general problem is that of adversarial order. As for prophet secretary, many generalizations have been considered. The classical 1/2 guarantee extends to general downward-closed systems, including matroids [23]. In the special case where the player may select up to k values and is evaluated on their sum, Alaei [3] proved a $(1 - \sqrt{1/(k + 3)})$-competitive algorithm, later improved for small k [9] and shown to be tight for all k [21]. A related model evaluates the player on the maximum of k choices: Assaf and Samuel-Cahn [6] showed that in this setting the gambler can secure more than $\frac{k}{k+1}$ of the prophet's value. Ezra et al. [16] introduced the ℓ-out-of-k framework, which interpolates between these models by allowing up to k tentative selections while only the best ℓ are evaluated. In the special case $\ell = 1$, they improved the bound of Assaf and Samuel-Cahn by giving an algorithm that achieves $1 - \frac{3}{2}e^{-k/6}$ of the prophet's value. This guarantee was later improved in [19]; related stronger guarantees for oracle-augmented variants were obtained in [18]. Another relevant direction is the prophet inequality with *recourse* (or buyback), where the decision maker can discard previously selected elements to accept new ones, often subject to a cancellation cost. Recent work by Ekbatani et al. [14] formalized this setting, showing that the ability to cancel decisions allows for improved competitive ratios in settings where standard irrevocable algorithms are limited. Finally, another approach is *resource augmentation*, where the algorithm is granted multiple samples or additional copies of the instance. Brüstle et al. [8] proved that block-threshold algorithms require only $\Theta(\log\log(1/\varepsilon))$ extra copies to obtain a $(1-\varepsilon)$-approximation.

2 Preliminaries

In this section, we will formalize the problems considered herein. We introduce the parameter M, defined as

$$M = \min_j \mathrm{E}\left[X_j\right] / \mathrm{E}\left[\min_j X_j\right].$$

Note that for i.i.d. variables for all j the values $\mathrm{E}\left[X_j\right]$ are the same, so the minimum in the numerator becomes redundant.

We assume that $\min_j \mathrm{E}\left[X_j\right] > 0$, otherwise there exists X_i s.t. $\Pr[X_i = 0] = 1$ and in this case the algorithm is trivial. Unless stated otherwise, we assume that all random variables $X_1, \ldots, X_n$ are independent. Note that the algorithm must choose at least one variable otherwise the competitive ratio is infinity. In general the distributions $F_1, \ldots, F_n$ may be arbitrary. We consider adversarial order, random arrival (prophet secretary), and the i.i.d. model.

Multi Choice

We study a multi-choice variant of the prophet inequality. An online algorithm is given n random variables $X_1, X_2, \ldots, X_n$ drawn independently from distributions $F_1, F_2, \ldots, F_n$, arriving sequentially. The algorithm is allowed to make multiple selections. Its cost is defined as the minimum chosen value. Formally, if the algorithm selects indices $i_1, \ldots, i_k$, then its cost is

$$\mathrm{ALG} = \min\{X_{i_1}, \ldots, X_{i_k}\}.$$

Obviously, no algorithm will ever choose an item of higher cost than a current option, so the minimal cost item will always be the last chosen.

The benchmark is the *prophet*, who observes all n values in advance and pays

$$\mathrm{OPT} = \min_{1 \le i \le n} X_i.$$

We compare the expected cost of the algorithm to the expected cost of the prophet i.e.,

$$\sup_{F_1,\ldots,F_n} \frac{\mathbb{E}[\mathrm{ALG}]}{\mathbb{E}[\mathrm{OPT}]}.$$

Special cases of this model include:

- single choice: the classic prophet inequality for minimization.
- n choices: the offline optimum, where the algorithm is equivalent to the prophet.

Our goal is to determine how many choices are required (in expectation) for an algorithm to guarantee a constant competitive ratio. If we insist on a deterministic upper bound on the number of choices and seek a constant competitive ratio, then $n-1$ choices do not suffice, even in the i.i.d. setting.

Multi unit

In the *multi-unit minimization* variant, the algorithm must acquire *exactly* r items, where each selection represents purchasing one unit of the item. The total cost is the sum of the r smallest values among the items it has selected. Formally, if the algorithm makes k selections at indices $i_1, \ldots, i_k$ (with $k \geq r$), its cost is

$$\mathrm{ALG} = \sum_{j=1}^{r} X_{(j)}^{\mathrm{ALG}},$$

where $X_{(1)}^{\mathrm{ALG}} \leq X_{(2)}^{\mathrm{ALG}} \leq \cdots \leq X_{(k)}^{\mathrm{ALG}}$ are the values of the items chosen by the algorithm. That is, the algorithm's cost is determined by the smallest r values among its k selections.

The benchmark (the prophet) observes all values in advance and pays

$$\mathrm{OPT} = \sum_{j=1}^{r} X_{(j)},$$

where $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$ denote the order statistics of $\{X_1, \ldots, X_n\}$. Thus, the prophet always takes the r globally smallest items. The competitive ratio in this model is defined as

$$\mathrm{CR}_{\text{multi-unit}}(r) = \sup_{F_1,\ldots,F_n} \frac{\mathbb{E}[\mathrm{ALG}]}{\mathbb{E}[\mathrm{OPT}]}.$$

This model captures settings in which the decision-maker must procure multiple units (e.g., resources or tasks) and seeks to minimize the *total* cost, rather than a single value as in the multi-choice case.

As shown later, if the algorithm is forced to choose *exactly* r values, no algorithm can guarantee better than an *exponential* competitive ratio. This impossibility motivates relaxing the model so that the algorithm may make any number of selections and instead aims to *minimize the expected number of choices* required to achieve a constant competitive ratio.

In our algorithm, we assume that the distributions are continuous. This is not a restriction, as if the distribution contains atoms, then we use the standard technique of perturbing the atoms (or just make the algorithm randomized at that point of mass). Also, our lower bounds use distributions with atoms of probability mass, but the distributions can be perturbed slightly, giving an equivalent bound with continuous distributions.

3 Adversarial order: almost linear lower bound on the expected number of choices

We analyze the adversarial order and show that any constant-competitive algorithm for arbitrary independent variables must make at least an almost linear number of choices in n.

▶ **Theorem 1.** *Any algorithm for arbitrary independent (but not identical) variables that is constant competitive requires $\Omega\left(\frac{n}{\ln n}\right)$ choices in expectation.*

Proof intuition

The distributions are chosen so that, with high probability, the values decrease over time, meaning that each early variable offers an important opportunity to improve the algorithm's current choice. On the other hand, if the algorithm rejects one of these improving values, it is forced, with small but non-negligible probability, to keep a value much larger than the offline minimum. Although this bad event occurs with low probability, its contribution to the expectation is large enough that any algorithm that does not accept many of the early variables has an unbounded competitive ratio.

Proof. Define the following variables $X_1, .., X_n$ as follows. For $1 \leq k \leq n-1$

$$F_{X_k}(x) = \begin{cases} 1 - \dfrac{1}{x^{n^{2k-2n}}}, & 1 \leq x < e^{n^{2n}}, \\ 1, & x \geq e^{n^{2n}}, \end{cases}$$

and for X_n:

$$F_{X_n}(x) = \begin{cases} 1 - \dfrac{1}{x^{1-\sum_{k=1}^{n-1} n^{2k-2n}}}, & 1 \leq x < e^{n^{2n}}, \\ 1, & x \geq e^{n^{2n}}. \end{cases}$$

First, we claim the following

▷ Claim 2. The expectation of the minimum of n independent variables is $n^{2n} + 1$

Proof. See Appendix A.1. ◁

We define $A_k = e^{n^{2n-2k-1}}$ and prove the following claim:

▷ Claim 3. For every $1 \leq k \leq n-1$, Denote Y_k as the minimum value for first k variables. For every k, Y_k satisfies that

$$Y_k \geq A_k = e^{n^{2n-2k-1}}$$

with probability of at least $1 - \frac{2}{n}$ for large enough n.

Proof. See appendix A.2 ◁

Next, for each k, define $Z_k = \min\{X_{k+1}, \ldots, X_n\}$ which is the minimum of the remaining variables, and $C_k = \mathbb{E}[\min(A_{k-1}, Z_k)]$, which is a lower bound on the algorithm's value if we do not choose x_k.

▷ Claim 4. For every $1 \leq k \leq n-1$ it holds that:

$$C_k = \mathbb{E}[\min(A_{k-1}, Z_k)] \geq \frac{1}{3} \cdot n^{2n-2k-2} \cdot e^n.$$

Proof.

$$\begin{aligned}
C_k = \mathbb{E}\left[\min\left(A_{k-1}, Z_k\right)\right] &= 1 + \int_1^{A_{k-1}} \frac{1}{x^{1-\sum_{i=1}^{k} \frac{n^{2i}}{n^{2n}}}} dx \\
&= 1 + \frac{1}{\sum_{i=1}^{k} \frac{n^{2i}}{n^{2n}}} \cdot \left(A_{k-1}^{\sum_{i=1}^{k} \frac{n^{2i}}{n^{2n}}} - 1 \right) \\
&\geq \frac{1}{\sum_{i=1}^{k} \frac{n^{2i}}{n^{2n}}} \cdot \left(e^{\frac{n^{2n}}{n^{2k-1}} \sum_{i=1}^{k} \frac{n^{2i}}{n^{2n}}} - 1 \right) \\
&\geq \frac{n^{2n}}{\sum_{i=1}^{k} n^{2i}} \cdot \left(e^{\frac{n^{2k}}{n^{2k-1}}} - 1 \right) \\
&\geq \frac{n^{2n}}{2n^{2k+2}} \cdot \left(e^n - 1\right) \\
&\geq \frac{n^{2n}}{3n^{2k+2}} \cdot e^n = \frac{1}{3} \cdot n^{2n-2k-2} \cdot e^n \ ,
\end{aligned}$$

where the second inequality holds because we decreased the exponent, and the next inequality holds because n^{2k+2} dominates the sum. ◁

▷ Claim 5. Let $m = \frac{n}{4 \ln n}$. Any constant-competitive algorithm must make m choices among $\{X_1, \ldots, X_m\}$ with probability at least $\frac{1}{2}$.

Proof. Suppose there exists a constant competitive algorithm that, with probability of at least $\frac{1}{2}$, makes fewer than m choices among $\{X_1, \ldots, X_m\}$. For each $j \in \{1, \ldots, m\}$, let G_j be the event that the algorithm does *not* choose X_j. Let E be the event that the algorithm makes fewer than m choices among $\{X_1, \ldots, X_m\}$. By assumption, $\Pr(E) \geq \frac{1}{2}$. On the event E the algorithm must skip at least one index in $\{1, \ldots, m\}$, hence by using the union bound we get

$$\sum_{j=1}^{m} \Pr(G_j) \ \geq \ \Pr\left(\bigcup_{j=1}^{m} G_j \right) \geq \Pr(E) \geq \frac{1}{2},$$

and therefore there exists $k \in \{1, \ldots, m\}$ such that

$$\Pr(G_k) \ \geq \ \frac{1}{2m} = \frac{2 \ln n}{n}.$$

Hence, there exists a k such that the algorithm didn't choose a variable with a probability of at least $\frac{2 \ln n}{n}$. Denote this variable as X_k. When the algorithm skipped the value of X_k, its decision was based only on the first $k-1$ variables. Hence, we can assume it is holding a previously selected value – specifically, one of $\{X_1, \ldots, X_{k-1}\}$. We can assume using claim 3 that with probability of at least

$$\Pr[\min\{X_1, \ldots, X_{k-1}\} \geq A_{k-1}] = \Pr[Y_{k-1} \geq A_{k-1}] \geq 1 - \frac{2}{n},$$

that the values of the variables $X_1, \ldots, X_{k-1}$ are at least A_{k-1}; it follows that the algorithm's current value is at least A_{k-1}. Consequently, the expected performance of the algorithm is lower bounded by the minimum of its current value and the minimum of the remaining unseen variables. Since it rejected X_k, we can assume it continues with the same value when it observes X_{k+1}. Therefore, its expected cost is at least $C_k = \mathbb{E}[\min(A_{k-1}, Z_k)]$. Hence by claim 4 its expected cost is at least:

$$C_k \geq \frac{1}{3} \cdot n^{2n-2k-2} \cdot e^n \ .$$

Denote the event that $\min\{X_1, \ldots, X_{k-1}\} \geq A_{k-1}$ as S_k. We can bound $\Pr(S_k \cap G_k)$ by using the fact that

$$\Pr(S_k) = \Pr(S_k \cap G_k) + \Pr(S_k \cap G_k^c) \leq \Pr(S_k \cap G_k) + \Pr(G_k^c).$$

Hence,

$$\Pr(S_k \cap G_k) \geq \Pr(S_k) - \Pr(G_k^c) \geq \left(1 - \frac{2}{n}\right) - \left(1 - \frac{2\ln n}{n}\right) = \frac{2\ln n}{n} - \frac{2}{n} \geq \frac{1}{n}.$$

Where the last inequality holds for large enough n. Therefore, we can bound the expectation of such an algorithm by at least

$$\mathbb{E}[ALG] \geq \Pr(S_k \cap G_k) \cdot C_k \geq \frac{1}{n} \cdot \frac{1}{3} n^{2n-2k-2} \cdot e^n = \frac{1}{3} \cdot n^{2n-2k-3} \cdot e^n.$$

Since, $\mathbb{E}[OPT] = 1 + n^{2n}$ then the competitive ratio of the algorithm is

$$\frac{\mathbb{E}[ALG]}{\mathbb{E}[OPT]} \geq \frac{\frac{1}{3} \cdot n^{2n-2k-3} \cdot e^n}{2 \cdot n^{2n}} = \frac{1}{6} \cdot \frac{e^n}{n^{2k+3}} \geq \frac{1}{6} \frac{e^n}{n^{\frac{n}{2\ln n}+3}} = \frac{1}{6} \cdot \frac{e^n}{e^{n/2+3\ln n}} \geq e^{n/3}.$$

Note that for large enough n we have $e^{n/3} \geq n^3$. Hence, if an algorithm skips the variable X_k with probability of more than $\frac{2\ln n}{n}$ then its competitive ratio is at least $e^{n/3}$ which is not constant competitive. Hence, any algorithm must choose all of $\{X_1, \ldots, X_m\}$ with probability of at least $\frac{1}{2}$. ◁

To complete the proof, we note that in total each constant competitive algorithm for large enough n makes at least

$$\mathbb{E}[Choices] \geq \frac{1}{2} \cdot \frac{n}{4\ln n} = \frac{n}{8\ln n}$$

choices in expectation, which is $\Omega(\frac{n}{\ln n})$ as required. ◀

4 Constant competitive algorithm with a small expected number of choices

In this section, we present our constant-competitive algorithm and analyze both its competitive ratio and the expected number of choices it makes. The algorithm is threshold based. It establishes a threshold and only accepts values below it or the variable that has the minimum expectation. To describe the algorithm we assume that the variables have a continuous distribution. We now present the algorithm: denote the first variable with the lowest expectation as X_{i_0} and its index as i_0.

Algorithm 1 Constant competitive multi-choice algorithm.

---

1. Find a threshold t (e.g., by binary search) such that $\sum_i p_i = \min\{\ln M + 1, n\}$ where $p_i = \Pr(x_i \leq t)$.
2. Accept the first item with a value at most t, *or* X_{i_0}
3. Thereafter, switch only if a newly observed value is strictly smaller than the value currently held and less than t.

We consider the random order model (the prophet secretary problem).

▶ **Theorem 6.** *For arbitrary variables, in random order, the above algorithm is constant-competitive and makes* $O(\min\{\ln\ln M, \ln n\})$ *choices in expectation.*

We next analyze the algorithm's performance. The analysis proceeds in two parts: we first bound its competitive ratio and then the expected number of choices it makes, where the expectation is taken over the randomness of the input. For ease of exposition we assume $\mathbb{E}[min\{X_1, ...X_n\}] = 1$ which implies that $\mathbb{E}[X_{i_0}] = M$.

4.1 Competitive ratio and number of choices analysis

We analyze the algorithm separately for the competitive ratio and for the expected number of choices.

4.1.1 Competitive ratio analysis

First, observe that if $\ln M + 1 \geq n$ then $p_i = 1$ for every i. Hence, the algorithm always achieves the minimum and is 1-competitive. If $\ln M + 1 < n$ then the only situation in which the algorithm may fail to achieve the minimum is when all samples except for the sample X_{i_0} are above t. Since $\ln M + 1 < n$ then $\sum_{i=1}^n p_i = \ln M + 1$ which means that $\sum_{i=1}^n p_i - p_{i_0} \geq \ln M$, and thus the probability that all variables except X_{i_0} are above t is:

$$\Pr\Big(\forall i \in [n] \setminus \{i_0\} : X_i > t\Big) = \prod_{i \neq i_0} (1 - p_i) \;\leq\; \left(1 - \tfrac{\ln M}{n-1}\right)^{n-1} \;\leq\; e^{-\ln M} = \tfrac{1}{M},$$

where the first inequality is the mean inequality and the second one is $(1-\alpha)^n \leq e^{-n\alpha}$. In this case, the algorithm's expected cost is M since it always chooses the variable X_{i_0}.

If there is a sample below t, then the algorithm achieves the minimum and is therefore 1-competitive. Combining these cases, the expected performance of the algorithm is

$$\begin{aligned} \mathbb{E}[\text{ALG}] \quad &\leq \quad 1 \cdot E[\min\{X_1, ..., X_n\}] | \min\{X_1, ..., X_n\} \leq t) \;+\; \tfrac{1}{M} \cdot E[X_{i_0}] \\ &\leq \quad E[\min\{X_1, ..., X_n\}] + \frac{1}{M} \cdot M \;=\; 2. \end{aligned}$$

Hence, the competitive ratio of the algorithm is at most 2.

4.1.2 Number of Choices Analysis

Denote Z_1 = number of samples below t, Z_2 = number of choices overall. Then we have $\mathbb{E}[Z_1] = \min\{\ln M + 1, n\}$. As the number of expected new minima in a random permutation of length n is $\sum_{i=1}^n 1/i = H_n \leq \ln(n+1) + 1$. We get that $\mathbb{E}[Z_2|Z_1] \leq \ln(1 + Z_1) + 2$ since, there is 1 choice that is being made regardless of the number of samples. Then by the law of total expectation we have that

$$\mathbb{E}[Z_2] = \mathbb{E}[\mathbb{E}[Z_2|Z_1]] \leq \mathbb{E}[\ln(1 + Z_1) + 2] \;.$$

By Jensen's inequality, since ln is a concave function, we get that $\mathbb{E}[\ln(Z_1 + 1) + 2] \leq \ln(\mathbb{E}[Z_1 + 1]) + 2$, and hence

$$\mathbb{E}[Z_2] \leq \mathbb{E}[\ln(Z_1 + 1) + 2] \leq \ln(\mathbb{E}[Z_1 + 1]) + 2 = O\left(\min\{\ln\ln M, \ln n\}\right).$$

As required.

5 Lower bound on the expected number of choices

We next establish a nearly tight lower bound, showing that our upper bound on the expected number of choices is optimal up to lower-order terms. The lower bound is actually proved even for the i.i.d. and hence obviously holds for the prophet secretary. The proof relies on constructing a hard i.i.d. instance that forces any algorithm to make sufficiently many selections (even in expectation) to remain constant competitive.

▶ **Theorem 7.** *Any constant competitive algorithm for arbitrary i.i.d. variables requires* $\Omega\left(\min\{\frac{\ln\ n}{\ln\ln n}, \frac{\ln\ln M}{\ln\ln\ln M}\}\right)$ *choices in expectation.*

Proof. We will show a lower bound of $\Omega\left(\frac{\ln\ln\beta}{\ln\ln\ln\beta}\right)$ on the number of choices for any parameter $\beta \leq e^n$ where $M = \Theta\left(\frac{\beta}{\ln\beta}\right)$. This yields the required bound for $M = O\left(\frac{e^n}{n}\right)$. For $M = \Omega\left(\frac{e^n}{n}\right)$ we use $\beta = e^n$ and get a lower bound of $\Omega\left(\frac{\ln n}{\ln\ln n}\right)$.

Proof overview. The proof proceeds in four steps.

- We construct a heavy-tailed i.i.d. distribution and show that $M = \Theta(\beta/\ln\beta)$.
- We partition the sequence into intervals whose sizes grow doubly logarithmically.
- We show that skipping any such interval significantly increases the algorithm's expected cost.
- We conclude that any constant-competitive algorithm must make at least one choice in each interval with very high probability.

First, we define a random variable X that has the following cumulative distribution function (CDF):

$$F(x) = \begin{cases} 1 - \dfrac{1}{x^{1/n}}, & 1 \leq x < \beta, \\ 1, & x \geq \beta, \end{cases}$$

where $9 \leq \beta \leq e^n$. The probability density function (PDF) is

$$f(x) = \frac{d}{dx}F(x) = \frac{1}{n}x^{-1-\frac{1}{n}}.$$

There is also a point mass at $x = \beta$:

$$P(X = \beta) = 1 - \lim_{x\to(\beta)^-} F(x) = \frac{1}{\beta^{1/n}}.$$

▷ Claim 8. The expectation of a single variable X is between $\frac{\beta}{4}$ and 2β.

Proof.

$$\mathbb{E}[X] = \int_1^\beta x\, f(x)\, dx + \beta \cdot \frac{1}{\beta^{1/n}} = \frac{1}{n}\int_1^\beta x^{-\frac{1}{n}}\, dx + \beta^{1-\frac{1}{n}}.$$

We compute the integral:

$$\int_1^\beta x^{-\frac{1}{n}}\, dx = \left[\frac{x^{1-\frac{1}{n}}}{1-\frac{1}{n}}\right]_1^\beta = \frac{\beta^{1-\frac{1}{n}} - 1}{1-\frac{1}{n}}.$$

Hence, the expectation becomes:

$$\mathbb{E}[X] = \frac{1}{n} \cdot \frac{\beta^{1-\frac{1}{n}} - 1}{1 - \frac{1}{n}} + \beta^{1-\frac{1}{n}} = \frac{1}{n} \cdot \frac{\beta^{1-\frac{1}{n}} - 1}{\frac{n-1}{n}} + \beta^{1-\frac{1}{n}} = \frac{\beta^{1-\frac{1}{n}} - 1}{n-1} + \beta^{1-\frac{1}{n}} = \frac{n\beta^{1-\frac{1}{n}} - 1}{n-1}.$$

Hence,

$$\mathbb{E}[X] = \frac{n\beta^{1-\frac{1}{n}} - 1}{n-1} \leq 2\beta,$$

and

$$\mathbb{E}[X] = \frac{n\beta^{1-\frac{1}{n}} - 1}{n-1} \geq \frac{\beta}{e} - 1 \geq \frac{\beta}{4},$$

where $9 \leq \beta \leq e^n$, ergo $\frac{\beta}{4} \leq \mathbb{E}[X] \leq 2\beta$ as required. $\triangleleft$

$\triangleright$ Claim 9. The expectation of the minimum of n independent variables is $\ln \beta + 1$.

From the above two claims it follows that $M = \frac{\mathbb{E}[X_i]}{\mathbb{E}[\min\{X_1,\ldots,X_n\}]} = \Theta\left(\frac{\beta}{\ln \beta}\right)$.

Next define for $1 \leq k \leq \frac{\ln \ln \beta}{2 \ln \ln \ln \beta} - 1$ the prefix size $n_k = \frac{n(\ln \ln \beta)^{2k}}{\ln \beta}$.

We now prove the following:

$\triangleright$ Claim 10. The minimum value encountered Y_k in the first n_k variables for $1 \leq k \leq \frac{\ln \ln \beta}{2 \ln \ln \ln \beta} - 1$ satisfies $Y_k \geq e^{\frac{\ln \beta}{(\ln \ln \beta)^{2k+1}}}$ with probability of more than $1 - \frac{1}{\ln \ln \beta}$.

Proof. Denote $A_k = e^{\frac{\ln \beta}{(\ln \ln \beta)^{2k+1}}}$. The CDF of Y_k is

$$F_{Y_k}(x) = 1 - \frac{1}{x^{\frac{n(\ln \ln \beta)^{2k}}{n \ln \beta}}} = 1 - \frac{1}{x^{\frac{(\ln \ln \beta)^{2k}}{\ln \beta}}} .$$

So the probability that Y_k is bigger than A_k is $\Pr[A_k \leq Y_k] = 1 - F_{Y_k}(A_k)$. Bounding $F_{Y_k}(A_k)$:

$$F_{Y_k}(A_k) = 1 - \frac{1}{A_k^{\frac{(\ln \ln \beta)^{2k}}{\ln \beta}}} = 1 - \frac{1}{e^{\frac{\ln \beta}{(\ln \ln \beta)^{2k+1}} \cdot \frac{(\ln \ln \beta)^{2k}}{\ln \beta}}} = 1 - e^{-\frac{1}{\ln \ln \beta}} \leq 1 - 1 + \frac{1}{\ln \ln \beta} = \frac{1}{\ln \ln \beta} .$$

where the inequality implied by $e^{-a} \geq 1 - a$. Hence, $\Pr[A_k \leq Y_k] = 1 - F_{Y_k}(A_k) \geq 1 - \frac{1}{\ln \ln \beta}$. $\triangleleft$

Next, for all $1 \leq k \leq \frac{\ln \ln \beta}{2 \ln \ln \ln \beta} - 1$, define $Z_k = \min\{X_{n_k+1}, \ldots, X_n\}$, which is the expected minimum of the remaining variables, and $C_k = \mathbb{E}[\min(A_{k-1}, Z_k)]$ which is a lower bound on the value of the algorithm if it skips all variables in indices between n_{k-1} to n_k.

$\triangleright$ Claim 11. For every $1 \leq k \leq T$, we have $C_k \geq \frac{\ln^2 \beta}{2(\ln \ln \beta)^{2k}}$.

Proof.

$$
\begin{aligned}
C_k &= \int_0^1 1\,dx \;+\; \int_1^{e^{\frac{\ln\beta}{(\ln\ln\beta)^{2k-1}}}} \frac{1}{x^{\frac{n-\frac{n(\ln\ln\beta)^{2k}}{\ln\beta}}{n}}}\,dx \\
&= \int_0^1 1\,dx \;+\; \int_1^{e^{\frac{\ln\beta}{(\ln\ln\beta)^{2k-1}}}} \frac{1}{x^{1-\frac{(\ln\ln\beta)^{2k}}{\ln\beta}}}\,dx \\
&= 1 \;+\; \int_1^{e^{\frac{\ln\beta}{(\ln\ln\beta)^{2k-1}}}} x^{-1+\frac{(\ln\ln\beta)^{2k}}{\ln\beta}}\,dx \\
&= 1 \;+\; \left[\frac{x^{\frac{(\ln\ln\beta)^{2k}}{\ln\beta}}}{\frac{(\ln\ln\beta)^{2k}}{\ln\beta}}\right]_{x=1}^{x=e^{\frac{\ln\beta}{(\ln\ln\beta)^{2k-1}}}} \\
&= 1 \;+\; \frac{\ln\beta}{(\ln\ln\beta)^{2k}}\left(e^{\frac{\ln\beta}{(\ln\ln\beta)^{2k-1}}\cdot\frac{(\ln\ln\beta)^{2k}}{\ln\beta}} - 1\right) \\
&= 1 + \frac{\ln\beta}{(\ln\ln\beta)^{2k}}\left(e^{\ln\ln\beta} - 1\right).
\end{aligned}
$$

Hence,

$$
C_k = 1 + \frac{\ln\beta}{(\ln\ln\beta)^{2k}}\left(e^{\ln\ln\beta} - 1\right) = 1 + \frac{\ln\beta}{(\ln\ln\beta)^{2k}}\left(\ln\beta - 1\right) \geq \frac{\ln^2\beta}{2(\ln\ln\beta)^{2k}},
$$

for all $\beta \geq 5$. ◁

Now we are ready to complete our proof (with high probability):

▷ Claim 12. Let $T = \frac{\ln\ln\beta}{2\ln\ln\ln\beta} - 1$. Any constant competitive algorithm must make at least T choices with probability higher than $1 - \frac{1}{\ln\ln\ln\beta}$ in the first n_T variables.

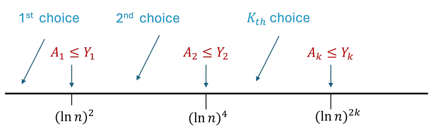

Figure 1 intervals for $\beta = e^n$.

Proof. Assume by contradiction that there exists an algorithm which makes fewer than T choices in the first n_T variables with probability of more than $\frac{1}{\ln\ln\ln\beta}$ and that its competitive ratio is constant. By looking at the disjoint intervals $[1, n_1], [n_1 + 1, n_2], ..., [n_{T-1} + 1, n_T]$.

Let E be the event that the algorithm makes fewer than T choices in the first n_T variables. On event E it must skip at least one of the T intervals, so by averaging, there exists an interval k with

$$
\Pr(\text{skip interval } k) \;\geq\; \Pr(E)\cdot\frac{1}{T} \;\geq\; \frac{1}{\ln\ln\ln\beta}\cdot\frac{2\ln\ln\ln\beta}{\ln\ln\beta} \;=\; \frac{2}{\ln\ln\beta}.
$$

Denote the index of the interval it skipped on with the highest probability as k. Hence, the probability that the algorithm skipped on it is bigger than $\frac{2}{\ln\ln\beta}$. When the algorithm skipped a value in interval k, its decision was based only on the first n_k variables. Hence, assuming $A_{k-1} \leq Y_{k-1}$, all values revealed among the first n_{k-1} variables are at least A_{k-1}; therefore any value chosen by the algorithm before interval k is at least A_{k-1}, and assuming the algorithm makes no choice in interval k, its expectation in this case is at least $\mathbb{E}[\min(A_{k-1}, Z_k)] = C_k$ and by Claim 11 we get that:

$$\mathbb{E}[\min(A_{k-1}, Z_k)] = C_k \geq \frac{\ln^2 \beta}{2(\ln\ln\beta)^{2k}}.$$

Denote the event that the algorithm skipped interval k as G and the event that $Y_{k-1} \geq A_{k-1}$ as S_k. We can bound $\Pr(S_k \cap G)$ by using the fact that

$$\Pr(S_k) = \Pr(S_k \cap G) + \Pr(S_k \cap G^c) \leq \Pr(S_k \cap G) + \Pr(G^c).$$

Hence,

$$\Pr(S_k \cap G) \geq \Pr(S_k) - \Pr(G^c) \geq 1 - \frac{1}{\ln\ln\beta} - \left(1 - \frac{2}{\ln\ln\beta}\right) = \frac{1}{\ln\ln\beta}.$$

Therefore, we can bound the expectation of the algorithm by:

$$\mathbb{E}[ALG] \geq \Pr(S_k \cap G) \cdot C_k \geq \frac{1}{\ln\ln\beta} \cdot \frac{\ln^2\beta}{2(\ln\ln\beta)^{2k}} = \frac{\ln^2\beta}{2(\ln\ln\beta)^{2k+1}}.$$

Hence, the competitive ratio of such an algorithm is

$$\begin{aligned}\frac{\mathbb{E}[ALG]}{\mathbb{E}[OPT]} &\geq \frac{\frac{\ln^2\beta}{2(\ln\ln\beta)^{2k+1}}}{\ln\beta + 1} \geq \frac{\ln\beta}{4(\ln\ln\beta)^{2k+1}} \geq \frac{\ln\beta}{4(\ln\ln\beta)^{\frac{\ln\ln\beta}{\ln\ln\ln\beta}-1}} \\ &= \frac{\ln\ln\beta}{4} \cdot \frac{\ln\beta}{(\ln\ln\beta)^{\frac{\ln\ln\beta}{\ln\ln\ln\beta}}} = \frac{\ln\ln\beta}{4} \cdot \frac{\ln\beta}{\ln\beta} = \frac{\ln\ln\beta}{4}\end{aligned}$$

which means the algorithm is not constant competitive. Therefore, for any algorithm to be constant competitive it must make at least T choices with probability of more than $1 - \frac{1}{\ln\ln\ln\beta}$ in the first n_T variables. ◁

To complete the proof of Theorem 7 (in expectation) we use Claim 12 (high probability) to get that any constant competitive algorithm must make at least

$$\mathbb{E}[choices] \geq \left(1 - \frac{1}{\ln\ln\ln\beta}\right) \cdot \left(\frac{\ln\ln\beta}{2\ln\ln\ln\beta} - 1\right) \geq \frac{\ln\ln\beta}{4\ln\ln\ln\beta}$$

choices in expectation. As mentioned at the beginning of the proof this yields the required bound of $\Omega\left(\frac{\ln\ln M}{\ln\ln\ln M}\right)$ for $M = O\left(\frac{e^n}{n}\right)$ given the relationship $M = \Theta\left(\frac{\beta}{\ln\beta}\right)$. For $M = \Omega\left(\frac{e^n}{n}\right)$ we use $\beta = e^n$ and get a lower bound of $\Omega\left(\frac{\ln n}{\ln\ln n}\right)$. ◀

6 A lower bound on the worst number of choices

In the previous sections we provided a constant competitive algorithm with at most $O(\min\{\ln\ln M, \ln n\})$ choices in expectation. The natural question is to try to replace expectation with a deterministic upper bound on the number of choices. Unfortunately we show that this is impossible as a function of n. Specifically, we show that limiting the algorithm to at most $n-1$ choices prevents any constant competitive ratio, even for i.i.d. inputs. The proof constructs a discrete hard instance illustrating this limitation.

▶ **Theorem 13.** *Any algorithm for arbitrary i.i.d. variables that is always limited to $n-1$ choices cannot be constant competitive for large enough n.*

Proof Sketch. We consider only deterministic algorithms. Since the distribution is known in advance, any randomized algorithm can be expressed as a probability distribution over deterministic ones; hence, any lower bound for deterministic algorithms immediately extends to randomized algorithms.

We construct a highly skewed, discrete probability distribution over $n+1$ super-exponentially growing values, scaled by $t = n^{n!}$. The distribution is carefully calibrated so that the prophet, observing all n variables, achieves an expected minimum of $O(\ln n)$. The discrete values of the distribution are $v_1, \ldots, v_{n+1}$, where $v_1 > v_2 > \cdots > v_{n+1}$.

We then analyze the performance of any deterministic algorithm restricted to $n-1$ choices. The algorithm faces a fundamental dilemma in case the realization is σ_k which is the values $v_2, v_3, \ldots, v_{k+1}$ followed by values which are strictly larger then v_{k+1}. If it skips the k value, the sequence might be σ_k with certain probability. On the other hand, if it plays conservatively and accepts each arriving smaller value to avoid this risk, then it strictly exhausts its $n-1$ budget just before the final n-th variable, failing (with some probability) to capture the true global minimum. By partitioning all possible deterministic algorithms into classes based on the exact step k at which they first choose to skip upon encountering the prefix $v_2, v_3, \ldots, v_{k+1}$, we prove that every algorithm class incurs an expected cost of at least $\Omega(n)$, resulting in an unbounded competitive ratio. The complete formal analysis and probability bounds are deferred to B. ◀

7 Conclusion

In this paper, we study the minimization prophet inequality under a multi-choice relaxation, where the algorithm is allowed to select multiple variables and pays the minimum among them. We show that for adversarial order an almost linear number of choices in expectation is required to achieve a constant-competitive ratio. However, for prophet secretary settings we prove that allowing $O(\min\{\ln\ln M, \ln n\})$ choices *in expectation* is sufficient to achieve a constant competitive ratio (recall that M is the ratio of the minimum expected value of any single variable to the expected minimum value of all variables). We further establish that this bound is tight up to doubly or triply log factors. We also demonstrate that insisting on deterministic upper bound even for i.i.d. inputs, implies that n choices are required to achieve a constant competitive ratio. Finally, we extend our algorithm to the multi-unit variant, showing that for i.i.d. inputs, a constant competitive ratio is achievable with only a small expected number of choices. Our work leaves open the possibility that the impossibility results stem specifically from the magnitude of M, raising two open problems:

1. **Expected number of choices for adversarial order:** Our lower bound of $\Omega(\frac{n}{\ln n})$ expected choices for non-i.i.d. distributions uses a construction where $\ln\ln M \geq n$. Does there exist an algorithm for adversarial order that achieves a constant competitive ratio with $O(\min\{\ln\ln M, \frac{n}{\ln n}\})$ choices in expectation?
2. **deterministic number of choices** Our lower bound showing that $n-1$ choices are insufficient relies on a distribution where $\ln\ln M \geq n$. Does there exist an algorithm for i.i.d. variables that achieves a constant competitive ratio with a deterministic upper bound of $O(\min\{\ln\ln M, n\})$ on the number choices?

References

1 Melika Abolhassani, Soheil Ehsani, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Robert D. Kleinberg, and Brendan Lucier. Beating 1-1/e for ordered prophets. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 61–71. ACM, 2017. doi:10.1145/3055399.3055479.

2 Marek Adamczyk and Michal Wlodarczyk. Multi-dimensional mechanism design via random order contention resolution schemes. *SIGecom Exch.*, 17(2):46–53, 2019. doi:10.1145/3381329.3381334.

3 Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM J. Comput.*, 43(2):930–972, 2014. doi:10.1137/120878422.

4 Nick Arnosti and Will Ma. Tight guarantees for static threshold policies in the prophet secretary problem. *Oper. Res.*, 71(5):1777–1788, 2023. doi:10.1287/OPRE.2022.2419.

5 David Assaf, Larry Goldstein, and Ester Samuel-Cahn. Two-choice optimal stopping. *Advances in Applied Probability*, 36(4):1116–1147, 2004. doi:10.1239/aap/1103662960.

6 David Assaf and Ester Samuel-Cahn. Simple ratio prophet inequalities for a mortal with multiple choices. *Journal of Applied Probability*, 37(4):1084–1091, 2000. doi:10.1239/jap/1014843085.

7 Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Prophet secretary: Surpassing the 1-1/e barrier. In Éva Tardos, Edith Elkind, and Rakesh Vohra, editors, *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 303–318. ACM, 2018. doi:10.1145/3219166.3219182.

8 Johannes Brustle, José Correa, Paul Duetting, and Victor Verdugo. The competition complexity of dynamic pricing. *Math. Oper. Res.*, 49(3):1986–2008, 2024. doi:10.1287/MOOR.2022.0230.

9 Shuchi Chawla, Nikhil R. Devanur, and Thodoris Lykouris. Static pricing for multi-unit prophet inequalities. *Oper. Res.*, 72(4):1388–1399, 2024. doi:10.1287/OPRE.2023.0031.

10 Ziyun Chen, Zhiyi Huang, Dongchen Li, and Zhihao Gavin Tang. Prophet secretary and matching: the significance of the largest item. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 1371–1401. SIAM, 2025. doi:10.1137/1.9781611978322.42.

11 José Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Posted price mechanisms for a random stream of customers. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, pages 169–186, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3033274.3085137.

12 José Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Posted price mechanisms and optimal threshold strategies for random arrivals. *Math. Oper. Res.*, 46(4):1452–1478, 2021. doi:10.1287/MOOR.2020.1105.

13 Soheil Ehsani, MohammadTaghi Hajiaghayi, Thomas Kesselheim, and Sahil Singla. Prophet secretary for combinatorial auctions and matroids. *SIAM J. Comput.*, 53(6):1641–1662, 2024. doi:10.1137/19M1264047.

14 Farbod Ekbatani, Rad Niazadeh, Pranav Nuti, and Jan Vondrák. Prophet inequalities with cancellation costs. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1247–1258. ACM, 2024. doi:10.1145/3618260.3649786.

15 Hossein Esfandiari, MohammadTaghi Hajiaghayi, Vahid Liaghat, and Morteza Monemizadeh. Prophet secretary. *SIAM J. Discret. Math.*, 31(3):1685–1701, 2017. doi:10.1137/15M1029394.

16 Tomer Ezra, Michal Feldman, and Ilan Nehama. Prophets and secretaries with overbooking. In Éva Tardos, Edith Elkind, and Rakesh Vohra, editors, *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 319–320. ACM, 2018. doi:10.1145/3219166.3219211.

17 Giordano Giambartolomei, Frederik Mallmann-Trenn, and Raimundo Saona. Prophet inequalities: Separating random order from order selection. *CoRR*, abs/2304.04024, 2023. doi:10.48550/arXiv.2304.04024.

18 Sariel Har-Peled, Elfarouk Harb, and Vasilis Livanos. Oracle-augmented prophet inequalities. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, Tallinn, Estonia, July 8-12, 2024*, LIPIcs, pages 81:1–81:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.81.

19 Elfarouk Harb. New prophet inequalities via poissonization and sharding. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 1222–1269. SIAM, 2025. doi:10.1137/1.9781611978322.37.

20 Theodore P. Hill and Robert P. Kertz. Stop rules and supremum expectations of i.i.d. random variables. *Annals of Probability*, 10(2):336–345, 1982. doi:10.1214/aop/1176993873.

21 Jiashuo Jiang, Will Ma, and Jiawei Zhang. Tight guarantees for multiunit prophet inequalities and online stochastic knapsack. *Oper. Res.*, 73(3):1703–1721, 2025. doi:10.1287/OPRE.2022.0309.

22 Robert P Kertz. Stop rule and supremum expectations of i.i.d. random variables: A complete comparison by conjugate duality. *Journal of Multivariate Analysis*, 19(1):88–112, 1986. doi:10.1016/0047-259X(86)90095-3.

23 Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games Econ. Behav.*, 113:97–115, 2019. doi:10.1016/J.GEB.2014.11.002.

24 Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bulletin of the American Mathematical Society*, 83(4):745–747, July 1977. doi:10.1090/S0002-9904-1977-14375-0.

25 Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. In *Probability on Banach Spaces*, volume 4, pages 197–266. Dekker, 1978.

26 Vasilis Livanos and Ruta Mehta. Minimization is harder in the prophet world. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 424–461. SIAM, 2024. doi:10.1137/1.9781611977912.17.

27 Vasilis Livanos and Ruta Mehta. Minimization I.I.D. prophet inequality via extreme value theory: A unified approach. In Itai Ashlagi and Aaron Roth, editors, *Proceedings of the 26th ACM Conference on Economics and Computation, EC 2025, Stanford University, Stanford, CA, USA, July 7-10, 2025*, pages 1157–1179. ACM, 2025. doi:10.1145/3736252.3742682.

28 Junjie Qin, Shai Vardi, and Adam Wierman. Minimization fractional prophet inequalities for sequential procurement. *Math. Oper. Res.*, 49(2):928–947, 2024. doi:10.1287/MOOR.2021.0173.

29 Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *The Annals of Probability*, 12(4):1213–1216, 1984. URL: http://www.jstor.org/stable/2243359.

A Missing proofs from section 3

A.1 Proof of Claim 2

Proof. Let

$$Y = \min\{X_1, X_2, \ldots, X_n\}.$$

Now we will calculate the CDF of Y:

$$\begin{aligned}\Pr[Y > x] &= \Pr[\min\{X_1, \ldots, X_n\} > x] = \prod_{k=1}^{n} \Pr[X_k > x] \\ &= \left(\prod_{k=1}^{n-1} \frac{1}{x^{n^{2k-2n}}}\right) \cdot \frac{1}{x^{1-\sum_{k=1}^{n-1} n^{2k-2n}}} = \frac{1}{x} \ .\end{aligned}$$

Hence,

$$F_Y(x) = \begin{cases} 1 - \dfrac{1}{x}, & 1 \le x < e^{n^{2n}}, \\ 1, & x \ge e^{n^{2n}}. \end{cases} \ .$$

Thus, the expectation of Y is given by

$$\mathbb{E}[Y] = 1 + \int_1^{e^{n^{2n}}} 1 - F_Y(x)\, dx = 1 + \int_1^{e^{n^{2n}}} \frac{1}{x} = 1 + \ln(e^{n^{2n}}) - \ln(1) = 1 + n^{2n}.$$

Hence,

$$\mathbb{E}\Big[\min\{X_1, \ldots, X_n\}\Big] = 1 + n^{2n}. \qquad \triangleleft$$

A.2 Proof of Claim 3

Proof. We define the CDF of Y_k as F_{Y_k}.

$$\begin{aligned}\Pr[Y_k > x] &= \Pr[\min\{X_1, \ldots, X_k\} > x] = \prod_{i=1}^{k} \Pr[X_i > x] \\ &= \left(\prod_{i=1}^{k} \frac{1}{x^{n^{2i-2n}}}\right) = \frac{1}{x^{\sum_{i=1}^{k} n^{2i-2n}}} = \frac{1}{x^{\frac{\sum_{i=1}^{k} n^{2i}}{n^{2n}}}}.\end{aligned}$$

Hence, $F_{Y_k} = 1 - x^{-\frac{\sum_{i=1}^{k} n^{2i}}{n^{2n}}}$. By the definition of the CDF

$$\Pr\big[A_k \le Y_k\big] = 1 - F_{Y_k}(A_k) \ .$$

First we bound $F_{Y_k}(A_k)$

$$\begin{aligned}F_{Y_k}(A_k) &= 1 - A_k^{-\frac{\sum_{i=1}^{k} n^{2i}}{n^{2n}}} = 1 - e^{-n^{2n-2k-1} \cdot \frac{\sum_{i=1}^{k} n^{2i}}{n^{2n}}} = 1 - e^{-\frac{\sum_{i=1}^{k} n^{2i}}{n^{2k+1}}} \\ &\le 1 - e^{-\frac{2n^{2k}}{n^{2k+1}}} = 1 - e^{-\frac{2}{n}} \le \frac{2}{n},\end{aligned}$$

where the first inequality holds for $n \ge 2$ since n^{2k} dominates the sum, and the last inequality holds since $e^{-a} \ge 1 - a$. Hence, we get that for every k : $\Pr[Y_k \ge A_k] = 1 - F_{Y_k}(A_k) \ge 1 - \frac{2}{n}$. $\triangleleft$

B Formal proof of Section 6

Proof. Define the following probability distribution over a random variable X.

Let

$t = n^{n!}$.

Then X takes values as follows:

$$\Pr[X = v_k] = \begin{cases} \frac{1}{t}, & \text{for } k = 1, \\ \frac{n-1}{t}, & \text{for } k = 2, \\ \frac{n^{(k-1)!} - n^{(k-2)!}}{t}, & \text{for } 3 \leq k \leq n, \\ 1 - \frac{n^{(n-1)!}}{t}, & \text{for } k = n+1 \end{cases}$$

where each value v_k is defined by:

$$v_k = \begin{cases} t^n, & \text{for } k = 1, \\ \frac{t^n}{k \cdot \left(n^{(k-1)!}\right)^n}, & \text{for } 2 \leq k \leq n, \\ 0, & \text{for } k = n+1. \end{cases}$$

Specifically $v_1 = t^n$, $v_2 = \frac{t^n}{2 \cdot (n)^n}$, $v_3 = \frac{t^n}{3 \cdot (n^2)^n}$, and so on up to v_n.

This defines a discrete probability distribution with $n+1$ outcomes. We will show that any algorithm with $n-1$ choices doesn't achieve a constant competitive ratio.

Expectation of the Minimum. Let $Y = \min\{x_1, \ldots, x_n\}$.

$$\mathbb{E}[Y] = \sum_{k=1}^{n+1} \Pr[Y = v_k] \cdot v_k \leq \sum_{k=1}^{n+1} \Pr[Y \geq v_k] \cdot v_k = \sum_{k=1}^{n} \Pr[Y \geq v_k] \cdot v_k \ .$$

By definition for $k \geq 2$

$$\Pr[Y \geq v_k] = \frac{\left(1 + (n-1) + \sum_{j=2}^{k-1} \left(n^{j!} - n^{(j-1)!}\right)\right)^n}{t^n} = \frac{\left(n^{(k-1)!}\right)^n}{t^n}.$$

This means that for $k \geq 2$:

$$\Pr[Y \geq v_k] \cdot v_k = \frac{\left(n^{(k-1)!}\right)^n}{t^n} \cdot \frac{t^n}{k \cdot \left(n^{(k-1)!}\right)^n} = \frac{1}{k} \ .$$

Moreover, for $k = 1$ we get that

$$\Pr[Y \geq v_1] = \frac{1}{t^n} \ .$$

Hence:

$$\Pr[Y \geq v_1] \cdot v_1 = \frac{1}{t^n} \cdot t^n = 1 \ .$$

Ergo, in summary, we get that:

$$\mathbb{E}[Y] \leq \sum_{k=1}^{n} \Pr[Y \geq v_k] \cdot v_k = \sum_{k=1}^{n} \frac{1}{k} \leq 2 \ln(n) \ .$$

Expectation of cost of any algorithm with $n-1$ choices. We can analyze any algorithm by looking only at what it does on the following series:

$$x_1 = v_2, x_2 = v_3, ..., x_{n-1} = v_n$$

or a prefix of it. Any algorithm when it sees the variable $x_k = v_{k+1}$ must decide if to skip the variable x_k and risk the option that all the next variables will be bigger than x_k. Otherwise, for every $1 \leq k \leq n-1$ take the variable x_k and risk the option that the last variable will be v_{n+1}. Hence, we analyze only the following specific cases and see how each one contributes to the expectation of the algorithm:

$$\begin{aligned}
\sigma_1 &= v_2, \{v_1\}^{n-1}, \\
\sigma_2 &= v_2, v_3, \{v_1, v_2\}^{n-2}, \\
\sigma_k &= v_2, v_3, ..., v_k, v_{k+1}, \{v_1, ..., v_k\}^{n-k}, \\
\sigma_{n-1} &= v_2, v_3, ..., v_n, \{v_1, .., v_n, v_{n+1}\}^1,
\end{aligned}$$

where $\{v_1, \ldots, v_k\}^{n-k}$ denotes all sequences where the last $n-k$ variables are independently drawn from this set.

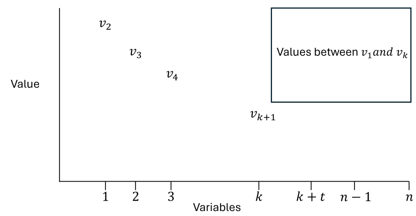

Figure 2 σ_k.

Denote ALG_k as the class of algorithms that skip x_k when it encounters the series $x_1 = v_2, x_2 = v_3, ...x_k = v_{k+1}$. And denote ALG_n the class of algorithms that don't skip any variable in the series $x_1 = v_2, x_2 = v_3, ...x_{n-1} = v_n$. Consider the specific prefix $x_1 = v_2, \ldots, x_{n-1} = v_n$. Since the algorithm is deterministic, its behavior on this prefix is fixed: at each step i it either takes x_i or skips it. If there exists an index $k \in \{1, \ldots, n-1\}$ such that the algorithm skips $x_k = v_{k+1}$ on this prefix, let k be the smallest such index; then the algorithm belongs to the class ALG_k. Otherwise, the algorithm takes all of $x_1, \ldots, x_{n-1}$, so it belongs to the class ALG_n. Thus, every deterministic algorithm belongs to at least one of the classes ALG_k, $1 \leq k \leq n$. It therefore suffices to lower bound the expected cost of every algorithm in each class ALG_k. For the class ALG_n The only problematic series may be σ_{n-1}

$$\Pr[\sigma = \sigma_{n-1}] = \Pr[x_1 = v_2, x_2 = v_3, .., x_{n-1} = v_n] = \frac{(n-1) \cdot \prod_{j=3}^{n} \left(n^{(j-1)!} - n^{(j-2)!}\right)}{t^{n-1}} .$$

So the contribution to the expectation of ALG_n is:

$$\begin{aligned}
\Pr[\sigma = \sigma_{n-1}] \cdot v_n &= \frac{(n-1) \cdot \prod_{j=3}^{n} \left(n^{(j-1)!} - n^{(j-2)!}\right)}{t^{n-1}} \cdot \frac{t^n}{n \cdot \left(n^{(n-1)!}\right)^n} \\
&= \frac{(n-1) \cdot \prod_{j=2}^{n-1} \left(n^{(j)!} - n^{(j-1)!}\right)}{n} \cdot \frac{t}{n^{n!}} \geq \frac{n^{(n-1)!}}{4} \geq n .
\end{aligned}$$

where the first inequality uses

$$\frac{n-1}{n} \geq \frac{1}{2} \text{ and } \left(n^{(n-1)!} - n^{(n-2)!}\right) \geq \frac{1}{2} \cdot n^{(n-1)!},$$

assuming $n \geq 3$.

Hence, for every algorithm $A \in ALG_n$ we get that $\mathbb{E}[A] \geq n$. For the class ALG_k the problematic series is σ_k since it doesn't achieve the minimum value on it. The probability that this series happens is:

$$\begin{aligned}
\Pr[\sigma = \sigma_k] &= \left(\frac{(n-1) \cdot \prod_{j=3}^{k+1}\left(n^{(j-1)!} - n^{(j-2)!}\right)}{t^k}\right) \cdot \left(\frac{1 + (n-1) + \sum_{j=3}^{k}\left(n^{(j-1)!} - n^{(j-2)!}\right)}{t}\right)^{n-k} \\
&= \left(\frac{(n-1) \cdot \prod_{j=3}^{k+1}\left(n^{(j-1)!} - n^{(j-2)!}\right)}{t^k}\right) \cdot \left(\frac{n^{(k-1)!}}{t}\right)^{n-k} \\
&\geq (n-1) \cdot \left(\prod_{j=2}^{k} \frac{1}{2} \cdot n^{(j)!}\right) \cdot \frac{\left(n^{(k-1)!}\right)^{n-k}}{t^n} \geq \frac{\left(\prod_{j=1}^{k} n^{(j)!}\right)}{2^k} \cdot \frac{\left(n^{(k-1)!}\right)^{n-k}}{t^n} \; .
\end{aligned}$$

Note that the inequalities hold since $n^{(k-1)!} - n^{(k-2)!} \geq \frac{1}{2} \cdot n^{(k-1)!}$ for $k \geq 3$ and that $(n-1) \geq \frac{1}{2} \cdot n$ for $n \geq 2$. Hence,

$$\begin{aligned}
\Pr[\sigma = \sigma_k] \cdot v_k \quad &\geq \quad \frac{\prod_{j=1}^{k} n^{(j)!}}{2^k} \cdot \frac{\left(n^{(k-1)!}\right)^{n-k}}{t^n} \cdot \frac{t^n}{k \cdot \left(n^{(k-1)!}\right)^n} \\
&= \quad \frac{\prod_{j=1}^{k} n^{(j)!}}{k \cdot 2^k \cdot n^{(k)!}} = \frac{n^{(k)!} \prod_{j=1}^{k-1} n^{(j)!}}{k \cdot 2^k \cdot n^{(k)!}} \\
&= \quad \frac{\prod_{j=1}^{k-1} n^{(j)!}}{k \cdot 2^k} \geq \frac{n}{8} \; .
\end{aligned}$$

The last inequality holds for $k \geq 2$. Thus, for every $A \in ALG_k$ we get that $\mathbb{E}[A] \geq \frac{n}{8}$ for sufficiently large n and for $k \geq 2$. For algorithm $A \in ALG_1$ the only problematic series we need to consider is σ_1.

$$\mathbb{E}[A] \geq \Pr[\sigma = \sigma_1] \cdot v_1 = \frac{n-1}{t} \cdot \left(\frac{1}{t}\right)^{n-1} \cdot t^n = n - 1 \; .$$

Combining the above cases, we have shown that for every $k \in \{1, \ldots, n\}$, every deterministic algorithm A in the class ALG_k satisfies $\mathbb{E}[A] \geq n/8$ for all sufficiently large n. Since every deterministic algorithm belongs to ALG_k for some $k \in \{1, \ldots, n\}$, it follows that every deterministic algorithm A with at most $n-1$ choices satisfies $\mathbb{E}[A] \geq n/8$. Thus, the competitive ratio for any algorithm with $n-1$ choices is at least:

$$\frac{\mathbb{E}[A]}{\mathbb{E}[Y]} \geq \frac{n}{16 \ln(n)}$$

which is unbounded as required, so no constant competitive ratio is possible. ◀

C The multi-unit algorithm

In this section we present an algorithm almost identical to that of Section 4, which achieves a constant competitive ratio in the multi-unit minimization problem while using only a small expected number of choices. We analyze both its competitive ratio and its expected number of choices.

We define the following algorithm:

Algorithm 2 Multi-unit algorithm.

1. Find a threshold t (e.g., by binary search) such that
$$\sum_i p_i = \min\{\max\{4\ln M + r,\, 5r\},\, n\},$$
where $p_i = \Pr(X_i \le t)$.
2. Wait until the first r items with values at most t appear. If only $\ell < r$ such items appear, take the remaining $r - \ell$ items from the last arrivals.
3. Thereafter, switch only if a newly observed value is strictly smaller than the highest value currently held among the r items.

▶ **Theorem 14.** *For i.i.d. variables and $r \le \ln M$, the algorithm above is constant competitive and uses*
$$O\big(\min\{r\ln\tfrac{\ln M}{r},\, r\ln\tfrac{n}{r}\}\big)$$
choices in expectation.

▶ **Theorem 15.** *For i.i.d. variables and $r \ge \ln M$, the algorithm above is constant competitive and uses $O(r)$ choices in expectation.*

D Lower bound for the multi-unit

We now consider the case where the algorithm selects exactly r items and its performance is measured by the sum of their values. We show that if the algorithm makes exactly r choices, then the competitive ratio in this case grows exponentially with n.

We consider the minimization variant with exactly r selections. Let $X_1, \ldots, X_n$ be i.i.d. where each variable gets 3 possible values. Specifically,
$$\Pr[X_i = 1] = 1 - \tfrac{1}{3^n} - \tfrac{1}{9^n}, \qquad \Pr[X_i = A] = \tfrac{1}{3^n}, \qquad \Pr[X_i = B] = \tfrac{1}{9^n},$$
where
$$A = 3^{\,n^2 - n(r-1)} = 3^{\,n^2 - nr + n}, \qquad B = 9^{\,n^2 - n(r-1)} = 3^{\,2n^2 - 2nr + 2n}.$$
The prophet benchmark is the sum of the r *smallest* order statistics:
$$\mathbb{E}[OPT(r)] = \mathbb{E}\Big[\sum_{j=1}^{r} X_{(j)}\Big].$$

▶ **Theorem 16.** *For every $1 \le r \le n-1$, any online algorithm that must select exactly r items satisfies*
$$\frac{\mathbb{E}[ALG(r)]}{\mathbb{E}[OPT(r)]} \ge \frac{1}{n^3}\left(\frac{3}{2}\right)^n .$$
In particular, the competitive ratio is exponential in n.