Optimal Posted Pricing with Unknown Demand under Uniform Arrivals

Nick Gravin[1] Hao Li[2] Zhiqi Wang[1]
[1]ITCS, Shanghai University of Finance and Economics
[2]Wuhan College

Abstract.
We consider the classic problem of selling a single item to a stream of buyers with private values, which is a well-studied setting of single-choice i.i.d. prophet inequality. The optimal online policy and uniform pricing are crucially dependent on *demand* n, i.e., the number of buyers in the stream. In practice, such as online commerce applications, the demand is often hard to predict and estimate in advance. To address this unpredictability of the demand, we propose a natural online learning setting in which an adversary chooses n and the algorithm learns the demand by observing the arrival times of the buyers. We assume that customers in the stream arrive independently at random times drawn from a known distribution. This model is equivalent to uniform i.i.d. arrivals in the $[0, 1]$ time interval. We evaluate the online algorithm in the worst case over the demand n against two standard benchmarks: the offline prophet $\mathsf{OFF}(n)$ and the optimal online policy $\mathsf{ON}(n)$ with known demand n.

We analyze single-threshold strategies that use the first arrival time t_1 as a *one-point estimator* of demand and then commit to a uniform price. We first show that a simple rule is $1/2$-competitive with respect to the OFF benchmark and find a more sophisticated rule that achieves a $\ln 3/2 \approx 0.549$ competitive ratio. Finally, we develop fully polynomial-time approximation schemes (FPTAS) to compute near-optimal unknown-demand policies against both benchmarks. The core of our approach is a no-regret learning procedure that computes a Nash equilibrium in a zero-sum game between the algorithm and an adversary who chooses n. As an application, we implement our FPTAS and identify instances showing a separation in approximation ratio between our unknown-demand model and the classical known-demand i.i.d. prophet inequality.

1 Introduction

We consider the fundamental economic problem of selling a single item to buyers with private values to maximize the seller's revenue. The seminal Myerson's paper [29] tells us that the best the seller can do for I.I.D. buyers is to run a second-price auction with a common reserve price. However, running an auction is often infeasible, and in practice a commonly used approach is to sell an item via posted prices, i.e., let buyers arrive sequentially and offer each of them a take-it-or-leave-it price. The posted pricing mechanisms have been extensively studied in the fields of marketing [7, 10] and econ-cs [2, 5, 8, 19, 23, 34] and have been directly connected [9, 23] to the *prophet inequality* in optimal stopping theory.

Prophet Inequality. In the standard single-choice I.I.D. prophet inequality, n agents arrive one by one and reveal their non-negative real values $\mathbf{v} = (v_i)_{i\in[n]}$. These values are drawn independently from the same known distribution $v_i \sim F_v$, i.e., $\mathbf{v} \sim \mathbf{F}$ with $\mathbf{F} = \prod_{i=1}^n F_v$. It observes the values and has to make an irrevocable decision upon the arrival of each agent: either discard the agent and continue, or accept his value and stop. The goal is to maximize the expectation of the stopping value. The value of the optimal online algorithm $\mathsf{ON}(n)$ is compared with the expected offline optimum (also called the prophet) $\mathsf{OFF}(n)$, and the prophet inequality refers to the worst-case approximation ratio $\alpha = \inf_{n,F_v} \frac{\mathsf{ON}(n)}{\mathsf{OFF}(n)}$, which is known to be equal to $\alpha \approx 0.745$ [1, 15, 24]. Crucially, both ON and OFF algorithms *know the demand* (number of agents) n in advance, however, which is often hard to estimate in an online setting.

Unknown Demand. In most online commerce applications, the seller does not know the population of buyers and instead advertises the goods online to the stream of incoming visitors on a website. The seller can typically estimate website traffic as a function of time, but may not easily predict the conversion rate (the interest of people from the stream in the good) in a single-item advertising campaign. In other words, it may be unrealistic to have a known demand n, or even any prior Bayesian belief $n \sim F_n$ for a stream. It is natural to assume that initially the seller *knows nothing* about demand n, but *learns* it over time based on the conversion of traffic to prospective buyers. In mathematical terms, we can capture this learning process by modeling each buyer's arrival as an independent random variable that follows the common traffic distribution over time. The online algorithm observes the visit times of prospective buyers and can adjust the offered price as a function of time. These scenarios can be formally described as follows.

1.1 Our Setting

We consider a single-item prophet inequality with i.i.d. values and an *unknown number* of agents $n \in \mathbb{N}$. Each agent $i \in [n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$ arrives at a time $t_i \in [0, 1]$ uniformly at random. That is, the arrival times $\{t_i\}_{i\in[n]}$ are drawn i.i.d. from the uniform distribution[1] on $[0, 1]$. For convenience, we order agents according to their arrival and write their arrival times as a vector $\mathbf{t}(n) \stackrel{\text{def}}{=} (t_1, \ldots, t_n)$ with $t_1 \leq t_2 \leq \cdots \leq t_n$.

Upon arrival at time t_i, we observe the value v_i of the agent i. The demand n is *unknown* to the online algorithm and may be chosen by an adversary. Thus, at time t_i, the algorithm observes the pair (t_i, v_i) and must irrevocably decide whether to stop at i and receive the reward v_i, or reject and continue. If the algorithm never accepts any agent, its payoff is 0. The values $\mathbf{v}(n) \stackrel{\text{def}}{=} (v_i)_{i\in[n]}$ are drawn i.i.d. from the known prior distribution $\mathcal{F}$ on $\mathbb{R}$ and are mutually independent of arrival times $(t_i)_{i\in[n]}$.

We write $\mathsf{ALG}(\mathbf{v}, \mathbf{t})$ for the (possibly randomized) payoff of a given online algorithm ALG on the realization $(\mathbf{v}, \mathbf{t})$. All expectations are taken with respect to the randomness in $(\mathbf{v}, \mathbf{t})$ (and the internal randomness of ALG, if any). Since the demand n is unknown and may be chosen by an adversary, an "optimal" online policy that is allowed to depend directly on n is no longer well defined. Instead of searching for such a policy, we compare algorithms in our unknown-demand model against standard benchmarks from the classical single-choice i.i.d. prophet inequality with known demand.

Benchmarks. When the number of agents n is known, the standard upper bounds are as follows.

- The *offline prophet benchmark* $\mathsf{OFF}(n)$, which knows all the values and arrival times in advance and always selects the maximum value $\mathsf{OFF}(n) = \mathbb{E}_{\mathbf{v}(n)}[\max_{i\in[n]} v_i]$.
- The *online benchmark with known demand* $\mathsf{ON}(n)$, which is the value of the optimal online policy when both n and $\mathcal{F}$ are known. In this case, the information about arrival times $\mathbf{t}(n)$ can be ignored, and the optimal policy is a threshold-based strategy: there exist thresholds $(\tau_k)_{k\in[n]}$ such that the algorithm accepts the first agent i with $v_i \geq \tau_i$, where the thresholds are given by a simple dynamic program $\tau_n = 0$ and $\tau_k = \mathbb{E}_{v\sim\mathcal{F}}[\max\{\tau_{k+1}, v\}]$ for $k = 0, \ldots, n-1$.

Competitive ratios. We measure the performance of an online algorithm ALG that does not know n in advance in the worst case over all possible demands $n \in \mathbb{N}$. We consider two competitive ratios, one against the offline prophet benchmark and one against the best online benchmark with known demand:

$$\gamma_{\mathsf{OFF}}(\mathsf{ALG}) = \inf_{n\in\mathbb{N}} \frac{\mathbb{E}_{\mathbf{v}(n),\mathbf{t}(n)}[\mathsf{ALG}(\mathbf{v}(n), \mathbf{t}(n))]}{\mathsf{OFF}(n)},$$
$$\gamma_{\mathsf{ON}}(\mathsf{ALG}) = \inf_{n\in\mathbb{N}} \frac{\mathbb{E}_{\mathbf{v}(n),\mathbf{t}(n)}[\mathsf{ALG}(\mathbf{v}(n), \mathbf{t}(n))]}{\mathsf{ON}(n)}.$$

Here, both vectors $\mathbf{v}(n)$ and $\mathbf{t}(n)$ are composed of n random variables. When the algorithm ALG is clear from context, we simply write γ_{OFF} and γ_{ON}.

[1] Assuming uniformity of the arrival time is without loss of generality, as one can arbitrarily stretch and compress different intervals on the timeline.

Known Bounds. Given the relation between the ON and OFF benchmarks for each n, we have $1 \geq \gamma_{\mathsf{ON}} \geq \gamma_{\mathsf{OFF}}$ for any fixed algorithm ALG. Moreover, $\mathsf{ON}(n) \geq \mathsf{ALG}(n)$ for any demand n. This means that $\gamma_{\mathsf{OFF}} \leq 0.745$, since the worst-case ratio between $\mathsf{ON}(n)$ and $\mathsf{OFF}(n)$ approaches $\alpha \approx 0.745$ as $n \to \infty$. This raises a natural question: "Is the bound $\alpha \approx 0.745$ tight even when the demand is unknown, or is the problem with the unknown demand more difficult?" A similar question in the case of known demand n about a separation between the I.I.D. prophet inequality with *unknown value distribution* and the secretary problem has a negative answer [14]. Specifically, [14] showed that if the algorithm knows demand n, but does not know the value distribution in the I.I.D. Prophet Inequality, then it cannot do better than the simple $1/e$ competitive algorithm for the secretary problem. In fact, the same $1/e$-competitive algorithm for the version of the problem with uniform arrival times (see, e.g. [6] or [20]) yields a $1/e$ lower bound on $\gamma_{\mathsf{ON}} \geq \gamma_{\mathsf{OFF}} \geq 1/e$.

Necessity of knowing the traffic distribution over time. We emphasize that ALG must have some prior information about the distribution of each agent's arrival time. Indeed, without knowing this distribution, one cannot meaningfully predict future demand n at any point in time against the worst-case adversary and thus cannot achieve a constant competitive ratio. The following Theorem 1.1 formalizes this intuition by showing that no online algorithm can guarantee a constant competitive ratio with respect to $\mathsf{OFF}(n)$ simultaneously over all i.i.d. distributions of arrival times and demand values n. The proof of Theorem 1.1 is given in the appendix.

Theorem 1.1. *For every constant $c > 0$ and any online algorithm ALG there exists an instance (value distribution $\mathcal{F}$, demand n, and distribution $\mathcal{T}$ of arrival time) such that $\mathbb{E}[\mathsf{ALG}(\mathbf{t}, \mathbf{v})] \leq c \cdot \mathbb{E}[\mathsf{OFF}(\mathbf{t}, \mathbf{v})]$, where agents arrive at i.i.d. times $\mathbf{t} \sim \mathcal{T}^n$ and have i.i.d. values $\mathbf{v} \sim \mathcal{F}^n$.*

1.2 Related Work

Hajiaghayi et al. [23] and Chawla et al. [9] have noted a direct connection of posted pricing to prophet inequality. Since then, different variants and combinatorial extensions of the prophet inequality have become an active research area in their own right [11, 17, 18, 27, 31] (see [28] for a survey). The single-choice I.I.D. prophet inequality was first considered by Hill and Kertz [24]. The optimal online algorithm is $\alpha \approx 0.745$-competitive [1, 15, 24] against the prophet, and a single-threshold online algorithm is $1 - \frac{1}{e}$-competitive and this bound is tight [18].

In recent years, many works have considered prophet inequalities in the learning framework, where prior value distributions F_v are not known but given with only a few independent and identically distributed (I.I.D.) samples. For the classic single choice (non I.I.D.) prophet inequality [32] shows that 1 sample per distribution is enough to get the tight $1/2$ approximation. For the I.I.D. case [14] first proved that $O(n^2)$ samples are sufficient to obtain the best possible competitive ratio of $0.745 - \varepsilon$. This sample complexity bound was later improved to $O(n/\varepsilon^6)$ by [32], to $O(n/\varepsilon^2)$ by [22], and by [11] to $O(n/\varepsilon)$. With exactly n samples, [14] showed how to achieve $\frac{e}{e-1}$ approximation, and later better approximations were obtained in [12, 13, 26]. For the non I.I.D. case with I.I.D. arrivals or the free-order arrivals, i.e., where the algorithm chooses the arrival order, [16] shows that $O(n/\varepsilon^6)$ samples are enough to match the respective approximation guaranties in the full-information case. Finally, [21] and [11] consider the regime with a low number of samples (less than one sample per distribution). [21] showed that the approximation guaranty smoothly degrades from $1/2$ to 0 respectively for the non I.I.D. case, and [11] showed a similar smooth interpolation between 0.745 and $1/e$ for the I.I.D. case. The latter bound of $1/e$ corresponds to the $1/e$ guaranty in the secretary problem for the case of a completely unknown prior F_v.

All works mentioned above assume that the demand n is known in advance. The uncertainty about demand n in the prophet inequality is much less understood. Allaart [4] considers the model with a known Poisson distribution $n \sim F_n$. More recently, Alijani et al. [3] considered the I.I.D. prophet inequality with demand n drawn from a known MHR distribution $n \sim F_n$. They showed that a single-threshold strategy gives a $2 - \frac{1}{\mu}$-approximation for F_n with mean μ. Importantly, both papers assume prior Bayesian knowledge about the demand n. Our work introduces a natural online learning framework for the uncertainty of demand n and thus complements the literature on learning the value distribution $\mathcal{F}$ to learning the demand n.

On the other hand, a closely related setting of the secretary problem has been studied under the assumption that the number of candidates n is not known in advance. In particular, [30] described how to reduce the problem to optimal stopping on a Markov chain for any given distribution $n \sim F_n$. Stewart [33] considers the model with unknown n, but where the arrival times of the candidates are independently distributed exponential random variables. Hill and Krengel [25] studied the min-max optimal policy for the secretary problem, where only an upper bound $n \leq N$ is known, and showed that it is $\Theta(\frac{1}{\log N})$ competitive. Recently, [35] proposed three variants of the secretary problem with unknown n: (i) known distribution $n \sim F_n$, (ii) worst-case n with a given upper bound $n \leq N$; (iii) worst-case n with a given expectation $\mathbb{E}[n] = \mu$. They obtain a list of approximation results for the class of single-threshold strategies.

1.3 Our Results

We investigate how well a seller can perform in the single-item I.I.D. prophet setting when the demand n is unknown and must be learned from the buyers' arrival pattern. Our approach is to benchmark performance by establishing upper and lower bounds on the competitive ratio relative to the offline prophet OFF. We also study this problem under a computational lens by searching for the min-max optimal online algorithm against ON and OFF benchmarks. Our contributions are as follows.

Single-threshold algorithms. In Section 2, we first study simple algorithms that use a single threshold determined by the arrival time of the first agent. In particular, we show that using the first arrival time t_1 as a quantile threshold already yields a $1/2$-competitive algorithm, improving over the simple $1/e$ bound inherited from the secretary problem. We also find another single-threshold algorithm and, through a more involved analysis, show that it achieves a competitive ratio of 0.549. We empirically evaluated a broader family of linear threshold policies with one-point and two-point estimators that include the two single-threshold algorithms above. A grid search over the parameters demonstrates that these families can outperform the worst-case guarantees for the analytically designed single-threshold rules, achieving competitive ratios above 0.59 (one-point) and 0.61 (two-points) on demands up to $n \leq 100$ (Section 2.3, Figures 1–2).

FPTAS for the optimal online algorithm. Unlike the case of known demand n, it is no longer clear how to describe online algorithms with the optimal competitive ratio for unknown demand. The main challenge comes from the case when the demand n is small and cannot be easily estimated in a short period of time. In Section 3 and Section 4, we provide a Fully Polynomial-Time Approximation Schemes (FPTAS) for the optimal online algorithm against the online and offline benchmarks. Our algorithms rely on no-regret learning in a zero-sum game played between the algorithm and another player who adversarially chooses the demand n.

Separation from the classical I.I.D. prophet inequality. In Section 5, we establish a clear separation between the I.I.D. Prophet Inequality with the known demand and our setting with unknown demand. Specifically, we found a concrete instance with a competitive ratio below 0.739 by implementing our FPTAS in a simulated annealing search. This demonstrates that our FPTAS is not just a theoretical complexity result but an algorithm that can be efficiently implemented in practice.

2 Single Threshold Algorithms

In this section, we study the simple *single-threshold* strategies in our model. The key idea is to use the first arrival time t_1 as a signal about the (unknown) demand, and then commit to one threshold for the rest of the time horizon. Working in the quantile space, this threshold is specified by a quantile level $q(t_1) \in [0, 1]$: after t_1, the algorithm accepts the first agent with (randomized) quantile rank of at least

$q(t_1)$. In fact, in this section, we essentially ignore the shape of the value distribution F_v, since we only use quantile queries to F_v of the form "is $v_i \geq F_v^{-1}(q)$" for a specified quantile q.

Quantile thresholds. Let F be the CDF of the value distribution $\mathcal{F}$. For $q \in [0,1]$, define the q-quantile threshold as $p(q) \overset{\text{def}}{=\!=} \inf\{x : \mathcal{F}(x) \geq q\}$. When $\mathcal{F}$ has atoms, the event $v = p(q)$ has positive probability, and a deterministic tie-breaking rule generally does *not* correspond to a clean "quantile $\geq q$" condition. To make the definition unambiguous, we allow randomized tie-breaking at the threshold: accept if $v > p(q)$; reject if $v < p(q)$; and if $v = p(q)$, accept with probability

$$\rho(q) \overset{\text{def}}{=\!=} \begin{cases} 1, & \mathcal{F}(p(q)) = \mathcal{F}(p(q)^-), \\ \dfrac{\mathcal{F}(p(q)) - q}{\mathcal{F}(p(q)) - \mathcal{F}(p(q)^-)}, & \text{otherwise}, \end{cases}$$

where $\mathcal{F}(x^-) \overset{\text{def}}{=\!=} \lim_{y \uparrow x} \mathcal{F}(y)$.

2.1 A $1/2$-competitive quantile threshold

Our first result shows that the natural choice $q(t_1) = 1 - t_1$ guarantees the competitive ratio of $1/2$.

Theorem 2.1 ($1/2$-competitive ratio)**.** *The single-threshold algorithm with quantile threshold $q(t_1) = 1 - t_1$ achieves competitive ratio $1/2$ in our model.*

Proof. We prove the desired $1/2$ guarantee in quantile space by showing that for every $q \in [0,1]$, the probability that the algorithm selects a value with (randomized) quantile of at least q is at least a $1/2$ fraction of the corresponding probability for the offline optimum.

Fix n and $q \in [0,1]$, the offline optimum attains a quantile at least q (i.e., $\mathbf{Pr}[\mathsf{OFF} \geq F_v(q)]$) if $\max_i U_i \geq q$ for i.i.d. $U_i \sim \mathrm{Uni}[0,1]$, which happens with probability $1 - q^n$. Let us calculate the same probability (i.e., $\mathbf{Pr}[\mathsf{ALG} \geq F_v(q)]$) for the ALG. Let $S \overset{\text{def}}{=\!=} 1 - t_1$ be the algorithm's (random) quantile threshold. Since t_1 is the minimum of n i.i.d. $\mathrm{Uni}[0,1]$ random variables, we have its CDF $\mathbf{Pr}[S \leq x] = x^n$ and its PDF $f_S(x) = nx^{n-1}$ for $x \in [0,1]$. Assuming that $S = x$ and that algorithm accepts an agent, the quantile probability of the accepted agent is distributed uniformly on $[x,1]$. Hence, the probability that the algorithm selects a quantile at least q equals

$$f(q) \overset{\text{def}}{=\!=} \int_0^q nx^{n-1} \cdot (1 - x^n) \cdot \frac{1-q}{1-x} \,\mathrm{d}x + \int_q^1 nx^{n-1} \cdot (1 - x^n) \,\mathrm{d}x,$$

where two integrals correspond to two cases when $S < q$ and $S \geq q$; PDF of S is nx^{n-1}; probability of ALG accepting an agent is $1 - x^n$ for $S = x$; probability of choosing quantile at least q for $U \sim \mathrm{Uni}[x,1]$ is $\frac{1-q}{1-x}$ in the first integral and 1 in the second one. Using that $\frac{1-x^n}{1-x} = \sum_{k=0}^{n-1} x^k$ in the first integral, we get

$$f(q) = (1-q)n \cdot \sum_{k=0}^{n-1} \frac{q^{n+k}}{n+k} + (1 - q^n) - \frac{1 - q^{2n}}{2}.$$

Now, it suffices to show that $2f(q) \geq 1 - q^n$ for all n and q, which is equivalent to proving

$$\forall q \in [0,1],\ \forall n, \qquad 2n(1-q) \cdot \sum_{k=0}^{n-1} \frac{q^{n+k}}{n+k} - q^n + q^{2n} \ \geq\ 0.$$

Since $n + k \leq 2n$, we have $\sum_{k=0}^{n-1} \frac{q^{n+k}}{n+k} \ \geq\ \frac{1}{2n} \sum_{k=0}^{n-1} q^{n+k} = \frac{q^n - q^{2n}}{2n(1-q)}$, which concludes the proof.

The bound $1/2$ is tight for this algorithm. Consider the instance with demand $n = 1$ and $F_v : v \equiv 1$. Our algorithm accepts the agent with probability $\frac{1}{2}$, while $\mathsf{OFF}(1) = 1$. $\square$

2.2 An improved $\frac{\ln 3}{2}$-competitive quantile threshold

We next introduce another relatively simple single-threshold algorithm that yields a better competitive ratio. The algorithm again uses the first arrival time t_1 to set a single quantile threshold of $q(t_1) \stackrel{\text{def}}{=\!=} \max\{1-2t_1, 0\}$.

Theorem 2.2 ($\ln 3/2$-competitive ratio)**.** *The single-threshold algorithm with the threshold* $q(t_1) = \max\{1-2t_1, 0\}$ *achieves competitive ratio of at least* $\ln 3/2 \approx 0.549$ *in our model.*

Although the rule itself is simple, the analysis is substantially more involved. The proof follows the same high-level approach as in Theorem 2.1, but requires a long case analysis on n. We defer the proof to the appendix.

2.3 One-point and two-point estimators: numerical evaluation

In addition to analytically proven guarantees, we numerically evaluate two simple parametric policy families of linear thresholds. First, similar to the previous two algorithms, we consider the family of *one-point estimators* $q(t_1) = \max\{1-\alpha t_1,\, 0\}$ given by a single parameter $\alpha > 0$. Second, we introduce *two-point estimator* $q_1(t_1) = \max\{1-\alpha t_1,\, 0\}$ and $q_2(t_2) = \max\{1-\beta t_2,\, 0\}$ given by two parameters $\alpha, \beta > 0$, where t_2 denotes the second earliest arrival time. This algorithm uses the first threshold q_1 only for the first value v_1 and the second threshold q_2 for all remaining agents starting from v_2.

Evaluation methodology. As in the analysis of single-threshold algorithms (Sections 2.1 and 2.2), for each fixed demand n and each quantile level $q \in [0,1]$, we compare

$$f_n(q) \stackrel{\text{def}}{=\!=} \mathbf{Pr}\left[Q_{\mathsf{ALG}} \geq q\right] \qquad \text{and} \qquad \mathbf{Pr}\left[Q_{\mathsf{OPT}} \geq q\right] = 1 - q^n,$$

where Q_{ALG} denotes the (randomized) quantile of the value selected by the algorithm and Q_{OPT} is the quantile selected by the prophet OFF. For each demand n, we define the ratio curve $R_n(q) \stackrel{\text{def}}{=\!=} \frac{f_n(q)}{1-q^n}$ which gives a lower bound on the competitive ratio at each quantile level q. Therefore, $\inf_{q\in[0,1]} R_n(q)$ is the competitive ratio for a given n. In our experiments, we numerically evaluate demands up to $n = 100$ and report the minimum over q and $n \leq 100$ for each policy. Specifically, $f_n(q)$ in both families can be written as an integral of a piecewise polynomial function (in q) induced by the joint density of random variables t_1 and t_2. We evaluate these integrals symbolically (reducing to explicit rational polynomials) and then minimize $R_n(q)$ over $q \in [0,1]$ by checking endpoints and stationary points, in the same spirit as the computer-assisted verification described in the appendix.

For the one-point estimator, we perform a grid search on α and evaluate $\min_{q\in[0,1]} R_n(q)$ for each $n = 1, \ldots, 100$. The best parameter we found is $\alpha \approx 1.57$, which yields a competitive ratio exceeding 0.59 in this range of demands. Figure 1 plots the ratio curves $R_n(q)$ as a function of the quantile level q (x-axis), with $R_n(q)$ on the y-axis. Each curve corresponds to one fixed demand $n \in \{1, \ldots, 20\}$. For a fixed n, the competitive ratio of the policy is $\inf_{q\in[0,1]} R_n(q)$, i.e., the minimum height of the corresponding curve.

For the two-point estimator, we perform a grid search on two parameters (α, β) and evaluate the ratio curves as above for $n = 1, \ldots, 100$. The best parameters we found are $\alpha \approx 1.24$ and $\beta \approx 0.62$, which result in a competitive ratio exceeding 0.61. This lower bound gets close to the known upper bound of $1 - 1/e \approx 0.632$ for the single-threshold prophet inequality [18] with known demand n. Figure 2 shows the corresponding ratio curves $R_n(q)$ (again plotted for $n = 1, \ldots, 20$).

3 FPTAS Algorithm Description

We now present another technical result. We describe how to compute (up to an arbitrarily small error) an *optimal* online policy in the unknown-demand model against either the online or the offline benchmark. Our FPTAS is summarized below in Algorithm 1.

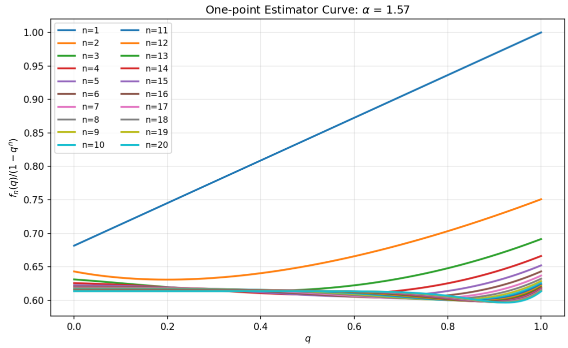

Figure 1: Ratio curves $R_n(q) = f_n(q)/(1-q^n)$ for the best linear single-threshold policy ($\alpha \approx 1.57$), for $n = 1, \dots, 20$.

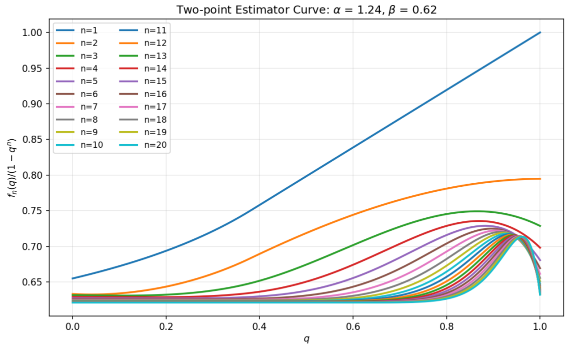

Figure 2: Ratio curves $R_n(q) = f_n(q)/(1-q^n)$ for the best linear two-threshold policy ($\alpha \approx 1.24$, $\beta \approx 0.62$), for $n = 1, \dots, 20$.

Algorithm 1. FPTAS for the online algorithm's strategy

Input: parameter $\varepsilon \in (0, \frac{1}{3})$, value distribution F_v; agents arriving online with time $t \in [0, 1]$.
Output: randomized algorithm Alg.

1. For $t \in [0, \varepsilon]$ let Alg skip all agents.
2. Let $n_\varepsilon \leftarrow$ number of arrived agents in $[0, \varepsilon]$.
3. Set $N = \varepsilon^{-4}$. *Upper threshold on n*
4. **if** $n_\varepsilon \geq \varepsilon N - (\varepsilon N)^{2/3}$ **then**
   5. $Alg \leftarrow \mathsf{ON}\left((n_\varepsilon - 2n_\varepsilon^{2/3}) \cdot \frac{1-\varepsilon}{\varepsilon}\right)$. *Opt. online*
6. **else**
   7. For $t \in [\varepsilon, 1]$, discretize time in $T = \varepsilon^{-9}$ steps.
   8. In $R = \varepsilon^{-2} \cdot \log(\frac{1}{\varepsilon})$ rounds, let the algorithm $\mathcal{A}(r)$ and the adversary $Adv(r)$ converge to equilibrium:
   9. **for** $r = 1$ to R **do**
      10. $Adv(r) \leftarrow \texttt{No-Regret}\left((\mathcal{A}(i))_{i=1}^{r-1}\right)$. *For N actions $n \in [N]$*
      11. $\mathcal{A}(r) \leftarrow \texttt{Best-Response}_{t \in [T]}(Adv(r))$.
   12. **end for**
   13. $Alg \leftarrow \texttt{Uni}\{\mathcal{A}[r], r \in [R]\}$.
14. **end if**

The FPTAS takes as input the target loss ε and the value distribution F_v and produces a randomized strategy Alg for the online algorithm with $1 - O(\varepsilon)$ approximation to the optimal competitive ratio. The algorithm proceeds as follows.

First, in lines 1–5 we ensure that Alg is doing very well against any large $n \geq N$. Specifically, the FPTAS uses the initial time interval $[0, \varepsilon]$ to get a conservative estimate on the number of remaining agents $n - n_\varepsilon$ and then uses the optimal online threshold policy ON for this estimate $(n_\varepsilon - 2n_\varepsilon^{2/3}) \cdot \frac{1-\varepsilon}{\varepsilon}$. The estimate is rather accurate when n is large: the expected number of remaining agents is $K = n_\varepsilon \cdot (1-\varepsilon)/\varepsilon$ and by a Chernoff bound it is unlikely to be smaller than $K - 2K^{2/3}$; similarly, $\mathbb{E}[n_\varepsilon] = \varepsilon \cdot n$ and it is very likely that $n_\varepsilon \geq \varepsilon N - (\varepsilon N)^{2/3}$ when $n \geq N$.

When n is small ($n < N$), we employ a no-regret learning procedure to find the Nash equilibrium of the zero-sum 2-player game between the adversary and the algorithm players (lines 7–13). To this end, we discretize (line 7) the remaining time $[\varepsilon, 1]$ into T equal-length time intervals and assume that all agents arrive in separate intervals[2]. This event $\mathcal{E}$ occurs with a high probability when $n < N$ for a sufficiently fine discretization $T = \omega(N^2)$. The discretization allows us to succinctly represent deterministic algorithms $\mathcal{A}(r)$ in each round $r \in [R]$. Indeed, conditioned on the good event $\mathcal{E}$, the optimal strategy (a value threshold) of Alg upon agent i's arrival at time $t \in [\varepsilon, 1]$ should only depend on the time interval's index

[2]If two agents ever arrive in the same discrete time interval, we simply assume that Alg gets 0 value.

and the number of agents that have arrived before i, but not on their specific arrival times. That is, any deterministic algorithm can be described by at most $N \times T$ thresholds. Furthermore, the adversary has only $n \in [N]$ polynomial in $1/\varepsilon$ number of pure strategies and thus employs a mixed strategy $F_n \in \Delta([N])$. Hence, the adversary (line 10) may use any of the no-regret learning algorithms (such as, e.g., HEDGE or Multiplicative-weights) to choose a mixed strategy $Adv(r) \in \Delta([N])$ against the history of the algorithm's player actions $(\mathcal{A}(r))_{r\in[R]}$. At the same time, the algorithm plays the best response to the respective strategy $Adv(r) \in \Delta([N])$ of the adversary in each round r of the no-regret learning process (lines 10-11). The final algorithm Alg picks one of the deterministic algorithms $\mathcal{A}(r)$ uniformly at random from the history $(\mathcal{A}(r))_{r\in[R]}$ of the algorithm's moves. According to the property of no-regret learning, the average value of the best-responses of the algorithm player against $(Adv(r))_{r\in[R]}$ after $R = \varepsilon^{-2} \cdot \log(\frac{1}{\varepsilon})$ rounds would not be much larger than the value of Alg against any pure strategy $n \in [N]$ of the adversary.

Best Response $\mathcal{A}(r)$: We describe now how the Best-Response (line 11) of the algorithm player works for a known (Bayesian) prior distribution $n \sim F_n$, where $F_n \in \Delta([N])$, against the online benchmark[3] $\mathsf{ON}(n)$. The algorithm needs to maximize

$$\mathbb{E}_{n\sim F_n}\left[\frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[\mathsf{ALG}(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)}\right] = \sum_{k=1}^{N} \frac{\mathbf{Pr}[n=k]}{\mathsf{ON}(k)} \mathbb{E}_{\mathbf{v},\mathbf{t}}\left[\mathsf{ALG}(\mathbf{v},\mathbf{t})\right].$$

Note that the optimal (Bayesian) algorithm is deterministic. We next explain how to calculate the optimum algorithm via a simple dynamic program.

By updating $\mathbf{Pr}[n = k] \leftarrow c \cdot \frac{\mathbf{Pr}[n=k]}{\mathsf{ON}(k)}$ the prior distribution F_n for an appropriate constant c, we simplify the algorithm's objective to $\mathbb{E}_{n\sim F_n}[\mathbb{E}_{\mathbf{v},\mathbf{t}}[\mathsf{ALG}(\mathbf{v},\mathbf{t})]]$. We compute $N \times T$ thresholds using a backward induction on time $t \in [T]$. Suppose that we have seen exactly $k \in [N]$ agents by time t, when the k-th agent has arrived at the t-th time interval (recall that we condition on the good event $\mathcal{E}$ that at most one agent arrives in each time interval). The value threshold for the k-th agent in this case should be equal to the expected utility $\mathtt{U}[k,t]$ that the algorithm player can get in the future after time t. The latter can be calculated in the Bayesian setting, where the algorithm player can update the posterior distribution of n upon observing k agents at time t for the given prior distribution $n \sim F_n$. In fact, we can write a simple DP formula for $\mathtt{U}[k,t]$ as follows.

Observe that, by the backward induction, we would apply the threshold $\mathtt{U}[k+1,t+1]$, if a new agent arrives in the next $t+1$ time interval (assuming that exactly k agents have arrived so far at time t). Let "k at t" denote the event that exactly k agents arrived in the first t intervals (conditioned on $\mathcal{E}$). Let $\mathtt{Ar}[k,t] \stackrel{\text{def}}{=} \mathbf{Pr}[k+1 \text{ at } t+1 \;\big|\; k \text{ at } t, n \sim F_n]$ be the probability that a new agent will arrive in the next $(t+1)$-th interval given that exactly k agents have arrived so far (which can be easily computed for any given prior[4] F_n). Then we get the following DP formula for $\mathtt{U}[k,t]$

$$\begin{aligned}\mathtt{U}[k,t] = \mathtt{Ar}[k,t] \cdot \Big(&\mathbb{E}_{v\sim F_v}\left[v \cdot \mathbb{I}\left[v \geq \mathtt{U}[k+1,t+1]\right]\right] \\ &+ \mathtt{U}[k+1,t+1] \cdot \mathbf{Pr}_{v\sim F_v}\left[v < \mathtt{U}[k+1,t+1]\right]\Big) + (1 - \mathtt{Ar}[k,t]) \cdot \mathtt{U}[k,t+1].\end{aligned}$$

Indeed, there are two cases of what happens in the interval $t+1$: a single agent i arrives (with probability $\mathtt{Ar}[k,t]$), or no agents show up (with probability $1 - \mathtt{Ar}[k,t]$). In the first case, we accept i whenever her value is greater than the threshold $\mathtt{U}[k+1,t+1]$, and otherwise we obtain the expected utility of $\mathtt{U}[k+1,t+1]$ when skipping agent i. In the second case, the expected utility will be $\mathtt{U}[k,t+1]$.

[3]It is the same for OFF benchmark with simple modifications.

[4]We would need to slightly modify F_n conditioned on the event $\mathcal{E}$. I.e., we update $p_i = \mathbf{Pr}_{n\sim F_n}[n = i \mid \mathcal{E}]$ (larger n leads to a lower probability that $\mathcal{E}$ occurs).

4 FPTAS Algorithm Analysis

In this section, we shall prove that our Algorithm 1 is indeed an FPTAS.

Theorem 4.1. *The Algorithm 1 has running time* $\mathrm{poly}(1/\varepsilon)$ *and outputs a* $1 - O(\varepsilon)$ *approximation to the optimum.*

Proof. We first address the main technical challenge of showing that Alg has an approximation error of at most $O(\varepsilon)$. Then we will briefly discuss implementation aspects and the running time of Algorithm 1. Our analysis of the approximation part proceeds in three steps.

First Step. We reduce the action space of the adversary (who chooses the demand n) to the finite space $n \in [N]$ of actions, where $N = \mathrm{poly}(1/\varepsilon)$. This is the most straightforward step in our analysis. Formally, we prove that

$$\sup_{Alg} \min_{n\in[N]} \frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[Alg(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)} \geq \sup_{Alg} \inf_{n\in\mathbb{N}} \frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[Alg(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)} \geq \sup_{Alg} \min_{n\in[N]} \frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[Alg(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)} - O(\varepsilon).$$

That is, restricting the adversary does not change much the approximation guaranty. The first inequality above is trivial, since it holds for any given algorithm. The focus of this step is to prove the second inequality. Let A be the (nearly) optimal algorithm restricted to $n \in [N]$. Consider the following extension to an algorithm ALG that works for any $n \in \mathbb{N}$: if $n_\varepsilon \geq \theta \overset{\text{def}}{=\!=} \varepsilon N - (\varepsilon N)^{2/3}$, then let Alg try to run the optimal online algorithm for our conservative estimate $\eta(n_\varepsilon) \overset{\text{def}}{=\!=} (n_\varepsilon - 2n_\varepsilon^{2/3}) \cdot \frac{1-\varepsilon}{\varepsilon}$ of the remaining demand $n - n_\varepsilon$; otherwise run A on the remaining time $[\varepsilon, 1]$ as if the time interval was $[0, 1]$. That is, we have

$$ALG(\mathbf{t}, \mathbf{v}) \geq \begin{cases} \mathsf{ON}(\eta(n_\varepsilon))\mathbb{I}[n - n_\varepsilon \geq \eta(n_\varepsilon)] & \text{if } n_\varepsilon \geq \theta, \\ A(\mathbf{t}(n - n_\varepsilon), \mathbf{v}(n - n_\varepsilon)) & \text{if } n_\varepsilon < \theta. \end{cases}$$

Note that lines 1–5 in Algorithm 1 implement exactly such an extension. We show next that this ALG loses at most $O(\varepsilon)$ in γ_{OFF} or γ_{ON} compared to A on limited demands $n \in [N]$.

In the proof, we rely on the following standard Chernoff bound to characterize the concentration of the number of agents n_ε, which we observe in the initial time interval $[0, \varepsilon]$.

Proposition 4.2 (Chernoff bound)**.** *Let* $X_1, \ldots, X_n$ *be independent Bernoulli random variables, and* $X = X_1 + \ldots + X_n$ *with* $\mu = \mathbb{E}[X]$*. Then for any* $\tau \in [0, \mu)$

$$\mathbf{Pr}\left[|X - \mu| > \tau\right] \leq 2 \cdot e^{-\tau^2/3\mu}.$$

Denote the expected value of ALG and A for a specific demand n by $ALG(n)$ and $A(n)$. Then we have

$$ALG(n) \geq \mathbf{E}_{\mathbf{t}}\left[\mathbb{I}\Big[n_\varepsilon(\mathbf{t}) < \theta\Big] \cdot A(n - n_\varepsilon(\mathbf{t})) + \mathbb{I}\Big[n_\varepsilon(\mathbf{t}) \geq \theta\Big] \cdot \mathbb{I}\Big[n - n_\varepsilon \geq \eta(n_\varepsilon)\Big] \cdot \mathsf{ON}(\eta(n_\varepsilon))\right]. \tag{1}$$

We analyze the ratio $ALG(n)/\mathsf{ON}(n)$ (or similar ratio for the benchmark OFF) according to two cases for the demand n: (A) $n \leq N - 2N^{2/3}$ is small; (B) $n > N - 2N^{2/3}$ is large. In Case (A), we have the following claim, whose proof uses the Chernoff bound to argue that $n_\varepsilon < \theta$ with probability $1 - O(\varepsilon)$ and reduces $ALG(n)$ to $A(n - n_\varepsilon)$.

Claim 4.3. *For any small demand* $n \leq N - 2 \cdot N^{2/3}$,

$$\frac{ALG(n)}{\mathit{ON}(n)} \geq (1 - O(\varepsilon)) \min_{k\in[N]} \frac{A(k)}{\mathit{ON}(k)},$$
$$\frac{ALG(n)}{\mathit{OFF}(n)} \geq (1 - O(\varepsilon)) \min_{k\in[N]} \frac{A(k)}{\mathit{OFF}(k)}.$$

In Case (B), an analogous bound holds; in fact, when $n \geq N$ the Chernoff concentration on the high-probability event $\mathcal{E}_1 \stackrel{\text{def}}{=} \{|n_\varepsilon - \varepsilon n| \leq (\varepsilon n)^{2/3}\}$ yields the strictly stronger estimate

$$ALG(n) \geq (1 - O(\varepsilon)) \cdot \mathsf{ON}(n) \geq (1 - O(\varepsilon)) \cdot A(n). \tag{2}$$

The remaining regime $N - 2N^{2/3} < n < N$ is captured by the following claim.

Claim 4.4. *For any demand $N - 2 \cdot N^{2/3} < n < N$*

$$\frac{ALG(n)}{\mathit{ON}(n)} \geq (1 - O(\varepsilon)) \min_{k \in [N]} \frac{A(k)}{\mathit{ON}(k)},$$
$$\frac{ALG(n)}{\mathit{OFF}(n)} \geq (1 - O(\varepsilon)) \min_{k \in [N]} \frac{A(k)}{\mathit{OFF}(k)}.$$

Claims 4.3, 4.4 together with the bound (2) cover all possible demands n and conclude the proof of the first step. The proofs of both claims and of (2) are deferred to the appendix.

Second Step. We further restrict the action space of the algorithm by discretizing the time into T equal time intervals and by assuming that all agents arrive in separate intervals (we denote this event as $\mathcal{E}$). As in the birthday paradox, $\mathcal{E}$ occurs with high probability if $T = \omega(N^2)$. That is, assuming $\mathcal{E}$ only causes a small loss to the expected value of the algorithm. Formally, we change our objective γ_{ON} to $\mathbb{E}_{\mathbf{v},\mathbf{t}}[A(\mathbf{v}, \mathbf{t}) \cdot \mathbb{I}[\mathcal{E}]]/\mathsf{ON}(n)$ (and similarly change γ_{OFF}) and show that it has only a negligibly small effect.

Claim 4.5. *Let $T = \varepsilon^{-9}, N = \varepsilon^{-4}$. Then*

$$\sup_A \min_{n \in [N]} \frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[A(\mathbf{v}, \mathbf{t}) \cdot \mathbb{I}(\mathcal{E})]}{\mathit{ON}(n)} + O(\varepsilon) \geq \sup_A \min_{n \in [N]} \frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[A(\mathbf{v}, \mathbf{t})]}{\mathit{ON}(n)} \geq \sup_A \min_{n \in [N]} \frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[A(\mathbf{v}, \mathbf{t}) \cdot \mathbb{I}(\mathcal{E})]}{\mathit{ON}(n)}.$$

The proof, a standard birthday-paradox calculation, is deferred to the appendix.

Remark 4.6. The same inequalities as in Claim 4.5 hold for the OFF benchmark, as $\frac{\mathsf{ON}(n)}{\mathsf{OFF}(n)} \leq 1$.

We observe that in the event $\mathcal{E}$, the optimal (undiscretized) algorithm is indifferent to the precise arrival time of an agent in any given discrete interval. Indeed, if an agent arrives in the i-th interval, the distribution of all future arrival times under $\mathcal{E}$ is exactly the same for any exact time t in this i-th interval. Thus, we can restrict our attention only to discrete algorithms A with the objective $A(\mathbf{v}, \mathbf{t}) \cdot \mathbb{I}[\mathcal{E}(\mathbf{t})]$. Recall that the best algorithm A can be randomized. We think of such A as a distribution over a deterministic algorithm $\mathcal{A}(\mathbf{v}, \mathbf{t})$. For a discrete deterministic algorithm $\mathcal{A}$, we observe that at time t the algorithm does not need to use the full history of arrivals $\mathbf{t}_{\leq t}$ before t, but only needs the total number of agents that have arrived so far. Indeed, the distribution of future arrival times after time t is exactly the same for all arrival histories given the total number of arrivals by time t. Therefore, we can succinctly describe any deterministic algorithm $\mathcal{A}$ with $N \times T$ thresholds, as for each discrete time interval $t \in [T]$ we need to specify a threshold ($\mathcal{A}$ accepts agent i in the interval t given that exactly $k \in [N]$ agents arrived so far, if and only if $v_i \geq \tau(t, k)$). Let us denote the space of all such deterministic algorithms by $\mathfrak{D}$.

Third Step. We define the following zero-sum game between the adversary and the algorithm players: for the pure strategies of the algorithm $\mathcal{A}$ and the adversary's choice of demand $n \in [N]$, the value of the game is $\frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[\mathcal{A}(\mathbf{v},\mathbf{t}) \cdot \mathbb{I}[\mathcal{E}]]}{\mathsf{ON}(n)}$ (a similar game can be defined for the OFF benchmark). Note that this game admits a succinct representation (polynomial in $1/\varepsilon$). We also consider mixed strategies of the algorithm player (denoted as $\mathsf{ALG} \in \Delta(\mathfrak{D})$) and the adversary $F_n \in \Delta([N])$. Our Algorithm 1 finds a mixed Nash equilibrium of this game in lines 8–13 using one of the standard no-regret learning processes for the adversary and the best responses of the algorithm.

Formally, denote $\overline{A}$ as the average strategy of the algorithm player in R rounds. To simplify notations let $\mathcal{A}_r$ denote the strategy of algorithm player in the r-th round (line 11 of the algorithm), and $\mathcal{A}_r(n)$ denote the expected value of $\mathcal{A}_r$ when demand is n. Then

$$\begin{aligned}
&\max_{A \in \Delta(\mathfrak{D})} \min_{n \in [N]} \frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[A(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)} + O\left(\sqrt{\frac{\log(N)}{R}}\right) \\
\geq &\min_{n \in [N]} \frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[\overline{A}(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)} + O\left(\sqrt{\frac{\log(N)}{R}}\right) \\
= &\frac{1}{R}\sum_{r=1}^{R} \mathbb{E}_{n \sim Adv(r)}\left[\frac{\mathcal{A}_r(n)}{\mathsf{ON}(n)}\right] \\
\geq &\max_{A \in \Delta(\mathfrak{D})} \min_{F_n \in \Delta([N])} \mathbb{E}_{n \sim F_n}\left[\frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[A(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)}\right]
\end{aligned}$$

where the equality is due to the no-regret learning property of the Hedge algorithm; the last inequality holds, as the algorithm plays the best response to $Adv(r)$ in each round:

$$\begin{aligned}
\mathbb{E}_{n \sim Adv(r)}\left[\frac{\mathcal{A}_r(n)}{\mathsf{ON}(n)}\right] &\geq \max_{A} \min_{F_n} \mathbb{E}_{F_n}\left[\frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[A(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)}\right] \\
&= \min_{F_n} \max_{A} \mathbb{E}_{F_n}\left[\frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[A(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)}\right].
\end{aligned}$$

By taking $R = \frac{1}{\varepsilon^2} \cdot \log(\frac{1}{\varepsilon})$, we have the error term $O\left(\sqrt{\frac{\log(N)}{R}}\right) = O(\varepsilon)$. Thus the output of Algorithm 1, $\overline{A}$, is an approximate equilibrium with error $O(\varepsilon)$ after R rounds.

Running Time. The majority of the algorithm's runtime is devoted to the no-regret learning process. In each round, once the adversary's mixed strategy is determined, the algorithm player computes her $N \times T$ optimal thresholds using dynamic programming, as shown in line 11. This step takes $O(NT)$ time. Following that, the adversary updates the expected utility for each of the N actions based on these thresholds for the next round, which requires $O(N^2T)$ time. Consequently, the total running time is $O(N^2TR) = O(\varepsilon^{-19} \cdot \log(\frac{1}{\varepsilon}))$. It is important to note that we prioritized clarity of presentation over runtime optimization. Although running time could be significantly reduced, achieving this would require a slightly more complex algorithm and more involved analysis. □

5 Separation from I.I.D. Prophet Inequality

The purpose of this section is not just to provide a theoretical result, but also to demonstrate the practical relevance of our FPTAS. We first describe how we set up a simulated annealing algorithm to find a counterexample (the value distribution F_v) and then present the result of our search. Note that to construct a counterexample, one needs to find a value distribution F_v and a distribution F_n for the number of agents. However, we only search over the value distribution F_v and find the optimal F_n based on our Algorithm 1. For given F_v and F_n, one can easily find the optimal Bayesian online algorithm and its competitive ratio to the offline benchmark $\mathsf{OFF}(n)$. As it turned out, the competitive ratio is strictly below $\alpha \approx 0.745$.

In our search, we sought a discrete random value distribution with support size 5 and a total number of agents n ranging from 1 to 10. For every given value and number distribution, we discretize the time interval $[0, 1]$ into 10^5 points. We first assume that there is no collision (this happens with a probability greater than 99.95%) and use DP to solve the optimal algorithm under this assumption, as well as its competitive ratio. Then we add a small correction term to account for the rare collisions, giving our final result.

We then run the simulated annealing algorithm with initial temperature 5×10^{-4}, final temperature 10^{-6}, and cooling rate 0.95. We obtained the following F_v and F_n distributions after 122 iterations of the simulated annealing search. The following tables list the supports of the discrete random variables and the weight assigned to each possible value.

Value	8	13	19	25	6867
Weight	9138	5158	3229	2175	12

Number	1	2	3	4	5	6	7	8	9	10
Weight	4041	2198	1820	1504	1291	1019	769	576	397	437

With this instance of the value F_v and demand F_n distributions, we obtained the following result.

Theorem 5.1. *There is an instance of F_v of the I.I.D. prophet inequality with unknown demand, such that the competitive ratio $\gamma_{\mathsf{OFF}} < 0.739$ to the prophet.*

This establishes a separation from the I.I.D. prophet inequality. A figure in the appendix illustrates the convergence of the Hedge algorithm for this example: with the value distribution fixed and the initial number distribution set to uniform on integers 1–10, the competitive ratio drops below α after only 342 iterations and converges to its final value by the 10^3-th iteration.

6 Open Questions and Future Work

We proposed a natural online learning model to describe the demand uncertainty for selling a single item in online commerce applications. We have obtained non-trivial upper and lower bounds on the competitive ratio of the demand-unaware online algorithm to two benchmarks that know the demand. We also found an efficient way to compute the min-max optimal strategy for any given value distribution F_v.

Several directions remain open. First, for our specific setting, a conceptual question is to identify the worst-case value distribution(s). Formally, this amounts to understanding the minimax quantities

$$\gamma_{\mathsf{OFF}} = \sup_{\mathsf{ALG}} \mathbb{E}_{n\sim F_n}\left[\frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[\mathsf{ALG}(\mathbf{v},\mathbf{t})]}{\mathsf{OFF}(n)}\right], \gamma_{\mathsf{ON}} = \sup_{\mathsf{ALG}} \mathbb{E}_{n\sim F_n}\left[\frac{\mathbb{E}_{\mathbf{v},\mathbf{t}}[\mathsf{ALG}(\mathbf{v},\mathbf{t})]}{\mathsf{ON}(n)}\right].$$

Algorithmically, it would be interesting to design provably near-optimal policies within structured families (e.g., multi-threshold rules) and to better understand the gap between the best such families and the optimal policy computed by our FPTAS. In particular, motivated by the empirical performance observed in Section 2.3, an appealing open problem is to give a *theoretical* analysis of these simple parametric families: prove (or refute) that the best linear single-threshold and linear two-threshold rules are optimal within their respective families, and rigorously characterize their worst-case competitive ratios. Finally, it is natural to consider learning-augmented variants in which the seller has access to imperfect predictions of n and seeks policies that are simultaneously consistent and robust.

Second, it would be interesting to extend our model to a multi-unit setting and non-I.I.D. buyers. It also makes sense to consider optimization under samples for the value distribution in our framework. Another interesting direction is to study supply uncertainty similar to the model in [3] under the online learning framework.

References

[1] Abolhassani, M., Ehsani, S., Esfandiari, H., HajiAghayi, M., Kleinberg, R., Lucier, B.: Beating 1-1/e for ordered prophets. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing. p. 61–71. STOC 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3055399.3055479, https://doi.org/10.1145/3055399.3055479

[2] Adamczyk, M., Borodin, A., Ferraioli, D., Keijzer, B.D., Leonardi, S.: Sequential posted-price mechanisms with correlated valuations. ACM Trans. Econ. Comput. **5**(4) (dec 2017). https://doi.org/10.1145/3157085, `https://doi.org/10.1145/3157085`

[3] Alijani, R., Banerjee, S., Gollapudi, S., Munagala, K., Wang, K.: Predict and match: Prophet inequalities with uncertain supply. Proceedings of the ACM on Measurement and Analysis of Computing Systems **4**(1), 1–23 (2020)

[4] Allaart, P.C.: Prophet inequalities for iid random variables with random arrival times. Sequential Analysis **26**(4), 403–413 (2007)

[5] Blumrosen, L., Holenstein, T.: Posted prices vs. negotiations: an asymptotic analysis. In: Proceedings of the 9th ACM Conference on Electronic Commerce. p. 49. EC '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1386790.1386801, `https://doi.org/10.1145/1386790.1386801`

[6] Bruss, F.T.: A unified approach to a class of best choice problems with an unknown number of options. The Annals of Probability pp. 882–889 (1984)

[7] Bult, J.R., Wansbeek, T.: Optimal selection for direct mail. Marketing Science **14**(4), 378–394 (nov 1995). https://doi.org/10.1287/mksc.14.4.378, `https://doi.org/10.1287/mksc.14.4.378`

[8] Chawla, S., Hartline, J.D., Kleinberg, R.: Algorithmic pricing via virtual valuations. In: Proceedings of the 8th ACM Conference on Electronic Commerce. p. 243–251. EC '07, Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1250910.1250946, `https://doi.org/10.1145/1250910.1250946`

[9] Chawla, S., Hartline, J.D., Malec, D.L., Sivan, B.: Multi-parameter mechanism design and sequential posted pricing. In: Proceedings of the Forty-Second ACM Symposium on Theory of Computing. p. 311–320. STOC '10, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1806689.1806733, `https://doi.org/10.1145/1806689.1806733`

[10] Cominetti, R., Correa, J.R., Rothvoß, T., Martín, J.S.: Optimal selection of customers for a last-minute offer. Oper. Res. **58**(4-Part-1), 878–888 (jul 2010). https://doi.org/10.1287/opre.1090.0787, `https://doi.org/10.1287/opre.1090.0787`

[11] Correa, J., Cristi, A., Epstein, B., Soto, J.A.: Sample-driven optimal stopping: From the secretary problem to the i.i.d. prophet inequality. Math. Oper. Res. **49**(1), 441–475 (apr 2023). https://doi.org/10.1287/moor.2023.1363, `https://doi.org/10.1287/moor.2023.1363`

[12] Correa, J., Dütting, P., Fischer, F.A., Schewior, K., Ziliotto, B.: Unknown I.I.D. prophets: Better bounds, streaming algorithms, and a new impossibility (extended abstract). In: Lee, J.R. (ed.) 12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference. LIPIcs, vol. 185, pp. 86:1–86:1. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPICS.ITCS.2021.86, `https://doi.org/10.4230/LIPIcs.ITCS.2021.86`

[13] Correa, J.R., Cristi, A., Epstein, B., Soto, J.A.: The two-sided game of googol and sample-based prophet inequalities. In: Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms. p. 2066–2081. SODA '20, Society for Industrial and Applied Mathematics, USA (2020)

[14] Correa, J.R., Dütting, P., Fischer, F.A., Schewior, K.: Prophet inequalities for I.I.D. random variables from an unknown distribution. In: EC. pp. 3–17. ACM (2019)

[15] Correa, J.R., Foncea, P., Hoeksma, R., Oosterwijk, T., Vredeveld, T.: Posted price mechanisms and optimal threshold strategies for random arrivals. Math. Oper. Res. **46**(4), 1452–1478 (2021)

[16] Cristi, A., Ziliotto, B.: Prophet inequalities require only a constant number of samples. In: Proceedings of the 56th Annual ACM Symposium on Theory of Computing. p. 491–502. STOC 2024, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3618260.3649773, `https://doi.org/10.1145/3618260.3649773`

[17] Dütting, P., Feldman, M., Kesselheim, T., Lucier, B.: Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. SIAM J. Comput. **49**(3), 540–582 (jan 2020). https://doi.org/10.1137/20M1323850, `https://doi.org/10.1137/20M1323850`

[18] Ehsani, S., Hajiaghayi, M., Kesselheim, T., Singla, S.: Prophet secretary for combinatorial auctions and matroids. In: Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms. pp. 700–714. SIAM (2018)

[19] Feldman, M., Gravin, N., Lucier, B.: Combinatorial auctions via posted prices. In: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms. p. 123–135. SODA '15, Society for Industrial and Applied Mathematics, USA (2015)

[20] Gnedin, A.: The best choice problem with random arrivals: How to beat the 1/e-strategy. Stochastic Processes and their Applications **145**, 226–240 (2022). https://doi.org/https://doi.org/10.1016/j.spa.2021.12.008

[21] Gravin, N., Li, H., Tang, Z.G.: Optimal prophet inequality with less than one sample. In: Hansen, K.A., Liu, T.X., Malekian, A. (eds.) Web and Internet Economics - 18th International Conference, WINE 2022, Troy, NY, USA, December 12-15, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13778, pp. 115–131. Springer (2022). https://doi.org/10.1007/978-3-031-22832-2_7, `https://doi.org/10.1007/978-3-031-22832-2\_7`

[22] Guo, C., Huang, Z., Tang, Z.G., Zhang, X.: Generalizing complex hypotheses on product distributions: Auctions, prophet inequalities, and pandora's problem. In: COLT. Proceedings of Machine Learning Research, vol. 134, pp. 2248–2288. PMLR (2021)

[23] Hajiaghayi, M.T., Kleinberg, R., Sandholm, T.: Automated online mechanism design and prophet inequalities. In: Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1. p. 58–65. AAAI'07, AAAI Press (2007)

[24] Hill, T.P., Kertz, R.P.: Comparisons of stop rule and supremum expectations of i.i.d. random variables. The Annals of Probability **10**(2), 336–345 (1982), `http://www.jstor.org/stable/2243434`

[25] Hill, T.P., Krengel, U.: Minimax-optimal stop rules and distributions in secretary problems. Annals of Probability **19**, 342–353 (1991)

[26] Kaplan, H., Naori, D., Raz, D.: Competitive analysis with a sample and the secretary problem. In: Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms. p. 2082–2095. SODA '20, Society for Industrial and Applied Mathematics, USA (2020)

[27] Kleinberg, R., Weinberg, S.M.: Matroid prophet inequalities. In: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing. p. 123–136. STOC '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2213977.2213991, `https://doi.org/10.1145/2213977.2213991`

[28] Lucier, B.: An economic view of prophet inequalities. SIGecom Exch. **16**(1), 24–47 (sep 2017). https://doi.org/10.1145/3144722.3144725, `https://doi.org/10.1145/3144722.3144725`

[29] Myerson, R.B.: Optimal auction design. Mathematics of operations research **6**(1), 58–73 (1981)

[30] Presman, E.L., Sonin, I.M.: The best choice problem for a random number of objects. Theory of Probability & Its Applications **17**(4), 657–668 (1973). https://doi.org/10.1137/1117078, https://doi.org/10.1137/1117078

[31] Rubinstein, A., Singla, S.: Combinatorial prophet inequalities. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms. p. 1671–1687. SODA '17, Society for Industrial and Applied Mathematics, USA (2017)

[32] Rubinstein, A., Wang, J.Z., Weinberg, S.M.: Optimal single-choice prophet inequalities from samples. In: ITCS. LIPIcs, vol. 151, pp. 60:1–60:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)

[33] Stewart, T.J.: The secretary problem with an unknown number of options. Operations Research **29**(1), 130–145 (1981). https://doi.org/10.1287/opre.29.1.130, https://doi.org/10.1287/opre.29.1.130

[34] Yan, Q.: Mechanism design via correlation gap. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms. p. 710–719. SODA '11, Society for Industrial and Applied Mathematics, USA (2011)

[35] Zhang, J., Jaillet, P.: Secretary problems with random number of candidates: How prior distributional information helps (2023), https://arxiv.org/abs/2310.07884

A Proof of Theorem 1.1

In this appendix we provide a complete proof of Theorem 1.1. We construct a concrete distribution over instances that depends only on c (and not on the algorithm), and show that every deterministic online algorithm performs poorly in expectation relative to the offline benchmark under this distribution. By Yao's minimax principle, this immediately implies the desired hardness guarantee for arbitrary randomized online algorithms as well.

Proof. Fix an arbitrary online algorithm ALG (possibly randomized) and a constant $c > 0$. Pick a sufficiently small $\varepsilon \in (0, \frac{1}{4})$ and a sufficiently large $m \in \mathbb{N}$ (both to be chosen later as functions of c). Let $n_k \overset{\text{def}}{=\!=} \varepsilon^{-2k}$ and $v_k \overset{\text{def}}{=\!=} \varepsilon^{-k}$ for $k \in \{0, 1, \ldots, m\}$.

For each agent, we draw a value $v = v_k$ with probability $n_k^{-1} = \varepsilon^{2k}$ for every $k \in \{1, \ldots, m\}$, and $v = v_0 = 1$ otherwise. Let $\mathcal{F}$ denote this i.i.d. value distribution. For the demand distribution, we first draw a level K such that $\mathbf{Pr}[K = k] = \varepsilon^k$ for $k \in \{1, \ldots, m\}$, and $\mathbf{Pr}[K = 0]$ is the remaining probability mass. Conditioned on $K = k$, we draw the demand as $n \sim \mathrm{Poisson}(n_k)$. For the arrival time distribution, partition $[0, 1]$ into $m + 1$ equal-length disjoint intervals $I_0, \ldots, I_m$. Conditioned on $K = k$, each agent independently chooses a segment $j \in \{0, \ldots, k\}$ with probability $p_{j|k}$ where

$$p_{0|k} \overset{\text{def}}{=\!=} \frac{n_0}{n_k}, \qquad p_{j|k} \overset{\text{def}}{=\!=} \frac{n_j - n_{j-1}}{n_k} \quad \text{for } j \in \{1, \ldots, k\},$$

and then arrives uniformly at random within I_j. The online algorithm observes arrival times and values as agents arrive, but does not know K. For $j \in \{0, \ldots, m\}$, let N_j denote the number of agents whose chosen segment is j.

Lemma A.1 (Poisson splitting and indistinguishability)**.** *Fix $k \in \{0, \ldots, m\}$ and condition on $K = k$. Then $(N_0, \ldots, N_k)$ are mutually independent and*

$$N_j \sim \mathrm{Poisson}(n_k p_{j|k}) = \mathrm{Poisson}(n_j - n_{j-1}) \qquad \textit{for all } j \leq k.$$

Moreover, for any fixed segment index $j \leq m$, the joint distribution of all observations revealed up to the end of segment I_j is identical for every level $k \geq j$.

Proof. Condition on $K = k$. Since $n \sim \mathrm{Poisson}(n_k)$ and each of the n agents is independently marked with $J \in \{0, \ldots, k\}$ according to $(p_{j|k})_{j \leq k}$, the standard Poisson splitting property implies that the marked counts $N_j \overset{\text{def}}{=\!=} |\{i : J_i = j\}|$ are independent Poisson random variables with means $n_k p_{j|k}$. By our choice of probabilities, for $j \geq 1$ we have $n_k p_{j|k} = n_j - n_{j-1}$, and for $j = 0$ we have $n_k p_{0|k} = n_0$, proving the first claim.

For the indistinguishability statement, fix $j \leq m$ and consider any $k \geq j$. Since the segments $I_0, \ldots, I_m$ are disjoint time intervals, from each observed arrival time the algorithm can infer the unique segment index in which it falls. Hence, the algorithm's observations up to the end of segment I_j can be equivalently described by: (i) the vector of segment counts $(N_0, \ldots, N_j)$, and (ii) conditional on these counts, the corresponding arrival times within each segment (and their associated values).

Under level k, by the first part (applied with k in place of k) we have that $(N_0, \ldots, N_j)$ are independent and

$$N_r \sim \mathrm{Poisson}(n_r - n_{r-1}) \qquad \text{for every } r \leq j.$$

Crucially, this *joint* distribution of $(N_0, \ldots, N_j)$ does not depend on k. In particular, observing the entire early count vector (or just the cumulative number of arrivals up to the end of I_j, namely $\sum_{r=0}^{j} N_r \sim \mathrm{Poisson}(n_j)$) cannot reveal any information.

Conditional on the counts $(N_0, \ldots, N_j)$, the construction draws arrival times i.i.d. uniformly within each segment and independently across segments, and draws the values i.i.d. from $\mathcal{F}$, independent of arrival times. Therefore, the complete joint distribution of everything observed up to the end of segment I_j is identical for every level $k \geq j$. □

Lemma A.2 (Poisson thinning). *If $N \sim \mathrm{Poisson}(\lambda)$ and, conditional on N, we keep each of the N items independently with probability p, then the number of kept items X satisfies $X \sim \mathrm{Poisson}(\lambda p)$.*

Proof. For $s \in [0, 1]$, we have

$$\mathbb{E}[s^X] \;=\; \mathbb{E}\big[\mathbb{E}[s^X \mid N]\big] \;=\; \mathbb{E}[(1-p+ps)^N] \;=\; \exp(\lambda(1-p+ps-1)) \;=\; \exp(\lambda p(s-1)),$$

which is the probability generating function of $\mathrm{Poisson}(\lambda p)$. $\square$

Step 1 (Lower bound the offline benchmark). For any realization $(\mathbf{t}, \mathbf{v})$ with n agents, the offline benchmark equals $\mathsf{OFF} = \max_{i\in[n]} v_i$ and hence depends only on $\mathbf{v}$. Fix $k \in \{1, \ldots, m\}$ and condition on $K = k$. Let W_k be the number of agents with value exactly v_k. Since $\mathbf{Pr}[v = v_k] = n_k^{-1}$ and $n \sim \mathrm{Poisson}(n_k)$, Lemma A.2 gives $W_k \sim \mathrm{Poisson}(n_k \cdot n_k^{-1}) = \mathrm{Poisson}(1)$, hence $\mathbf{Pr}[W_k \geq 1] = 1 - 1/e$. Therefore,

$$\mathbb{E}[\mathsf{OFF} \mid K = k] \;\geq\; (1-1/e)\, v_k \;=\; (1-1/e)\, \varepsilon^{-k}.$$

Averaging over $\mathbf{Pr}[K = k] = \varepsilon^k$ and summing over $k = 1, \ldots, m$ gives

$$\mathbb{E}[\mathsf{OFF}] \;\geq\; (1-1/e) \sum_{k=1}^{m} \varepsilon^k \cdot \varepsilon^{-k} \;=\; (1-1/e)\, m. \tag{3}$$

Step 2 (Upper bound ALG via gifts + upgrades). We upper bound ALG by analyzing a more favorable comparison process. In this comparison process, we (i) gift the algorithm additional value for free after the game ends, and (ii) allow a free upgrade within each time segment. Since both modifications can only increase the algorithm's payoff on every realization, any upper bound proved for the comparison process also upper bounds the original ALG.

In segment $j \in \{0, 1, \ldots, m-1\}$, any arriving value at least v_{j+1} is considered a surprise for that segment, and we give *all* such surprise values to the algorithm for free after the game ends. After removing all surprises from segment j, every remaining value in that segment is at most v_j. We further allow a free upgrade: whenever the game reaches segment j, we pretend that the algorithm is offered a deterministic value exactly v_j (even if the segment contains no such value). This yields a compressed game in which reaching segment j offers v_j, and the process proceeds to segment $j+1$ iff $K \geq j+1$.

In our analysis, we will credit the algorithm with all gifts G in addition to whatever payoff it obtains from its acceptance decision in the upgraded segment game. Since this only increases the payoff on every realization, it yields the upper bound

$$\mathbb{E}[\mathsf{ALG}] \;\leq\; \mathbb{E}[G] \;+\; \sup_{\pi} \mathbb{E}\big[(\text{payoff of policy } \pi \text{ in the upgraded segment game})\big], \tag{4}$$

where the supremum is over all (possibly randomized) online policies in the upgraded segment game (and the expectation is over K, n, and the policy's internal randomness).

Bounding the total gift $\mathbb{E}[G]$. Denote G_j as the contribution from segment j to the gift. For segment 0, since $N_0 \sim \mathrm{Poisson}(n_0) = \mathrm{Poisson}(1)$ and hence $\mathbb{E}[N_0] = 1$, we have

$$\mathbb{E}[G_0] = \mathbb{E}\big[v \cdot \mathbb{I}\Big[v \geq v_1\Big]\,\big] = \sum_{t=1}^{m} \mathbf{Pr}\left[v = v_t\right] \cdot v_t = \sum_{t=1}^{m} \varepsilon^{2t} \cdot \varepsilon^{-t} = \sum_{t=1}^{m} \varepsilon^t \leq \frac{\varepsilon}{1-\varepsilon} \leq 2\varepsilon,$$

using $\varepsilon \in (0, \frac{1}{4})$ in the last inequality.

Now fix $j \in \{1, \ldots, m-1\}$. Since $N_j = 0$ whenever $K < j$, we have

$$\mathbb{E}[G_j] = \mathbf{Pr}\left[K \geq j\right] \cdot \mathbb{E}[G_j \mid K \geq j] = \mathbf{Pr}\left[K \geq j\right] \cdot \mathbb{E}[N_j \mid K \geq j] \cdot \mathbb{E}\Big[v \cdot \mathbb{I}\Big[v \geq v_{j+1}\Big]\Big],$$

where the last equality uses independence between N_j and values conditional on $K \geq j$. We bound each factor as follows:

$$\mathbf{Pr}\left[K \geq j\right] = \sum_{i=j}^{m} \varepsilon^i \leq \frac{\varepsilon^j}{1-\varepsilon}, \qquad \mathbb{E}[N_j \mid K \geq j] = n_j - n_{j-1} \leq n_j = \varepsilon^{-2j},$$

where the identity for $\mathbb{E}[N_j \mid K \geq j]$ follows from Lemma A.1. Finally,

$$\mathbb{E}\Big[v \cdot \mathbb{I}\Big[v \geq v_{j+1}\Big]\Big] = \sum_{t=j+1}^{m} \mathbf{Pr}\left[v = v_t\right] \cdot v_t = \sum_{t=j+1}^{m} \varepsilon^{2t} \cdot \varepsilon^{-t} = \sum_{t=j+1}^{m} \varepsilon^t \leq \frac{\varepsilon^{j+1}}{1-\varepsilon}.$$

Combining these inequalities yields

$$\mathbb{E}[G_j] \leq \frac{\varepsilon^j}{1-\varepsilon} \cdot \varepsilon^{-2j} \cdot \frac{\varepsilon^{j+1}}{1-\varepsilon} = \frac{\varepsilon}{(1-\varepsilon)^2} \leq 2\varepsilon.$$

Summing over $j = 0, \ldots, m-1$ gives

$$\mathbb{E}[G] \;=\; \sum_{j=0}^{m-1} \mathbb{E}[G_j] \;\leq\; (2m+2)\varepsilon. \tag{5}$$

Bounding the upgraded main payoff via backward induction. We will use Lemma A.1 to argue that, upon reaching segment j, the algorithm cannot infer anything about whether $K \geq j+1$ beyond the fact that $K \geq j$. Hence, we obtain

$$\mathbf{Pr}\left[K \geq j+1 \mid K \geq j\right] \;=\; \frac{\sum_{i=j+1}^{m} \varepsilon^i}{\sum_{i=j}^{m} \varepsilon^i} \;\leq\; \varepsilon. \tag{6}$$

Now consider the upgraded segment game. For $j \in \{0, \ldots, m\}$, let U_j denote the optimal (over all policies) of the expected payoff obtainable starting from segment j, conditioned on reaching segment j. We claim that $U_j \leq v_j$ for all $j \in \{0, \ldots, m\}$, and prove this by backward induction. The base case is $U_m \leq v_m$. Now fix $j \leq m-1$ and assume inductively that $U_{j+1} \leq v_{j+1}$. Upon reaching segment j the algorithm is offered v_j and either accepts (obtaining v_j) or rejects. If it rejects, then regardless of the realized history, the conditional probability that the process reaches segment $j+1$ is at most ε by (6), and in that event the expected continuation payoff is at most U_{j+1}. Thus,

$$U_j \;\leq\; \max\big\{v_j,\, \varepsilon \cdot U_{j+1}\big\} \;\leq\; \max\big\{v_j,\, \varepsilon \cdot v_{j+1}\big\} \;=\; v_j,$$

where the last equality uses $v_{j+1} = \varepsilon^{-1} v_j$. This completes the induction, and in particular $U_0 \leq v_0 = 1$. Therefore,

$$\sup_{\pi} \; \mathbb{E}\big[(\text{payoff of policy } \pi \text{ in the upgraded segment game})\big] \;\leq\; 1. \tag{7}$$

Putting things together. Combining (4), (5), (7), and (3) gives

$$\frac{\mathbb{E}[\mathsf{ALG}]}{\mathbb{E}[\mathsf{OFF}]} \;\leq\; \frac{1 + (2m+2)\varepsilon}{(1-1/e)\, m} = \frac{2\varepsilon}{1-1/e} + \frac{2\varepsilon + 1}{(1-1/e)m}.$$

Given any target $c > 0$, choose $\varepsilon > 0$ sufficiently small so that $\frac{2\varepsilon}{1-1/e} \leq \frac{c}{2}$, and then choose m sufficiently large so that $\frac{2\varepsilon+1}{(1-1/e)m} \leq \frac{c}{2}$. With these choices we obtain $\mathbb{E}[\mathsf{ALG}(\mathbf{t}, \mathbf{v})] \leq c \cdot \mathbb{E}[\mathsf{OFF}(\mathbf{t}, \mathbf{v})]$ under our construction. □

B Proof for Theorem 2.2

In this appendix, we provide the full proof of Theorem 2.2.

B.1 Preparation

Let t_1 be the first arrival time among n i.i.d. $\mathrm{Uni}[0,1]$ arrivals. Its density is

$$f_{t_1}(u) = n(1-u)^{n-1}, \qquad u \in [0,1].$$

Define the induced (random) quantile threshold

$$X \stackrel{\text{def}}{=\!=} q(t_1) = \max\{1-2t_1, 0\} \in [0,1].$$

Then $\mathbf{Pr}[X=0] = \mathbf{Pr}[t_1 \geq 1/2] = 2^{-n}$, and on $(0,1]$ the density of X is obtained by the change of variables $u = (1-x)/2$:

$$f_X(x) = \frac{n}{2}\Big(\frac{1+x}{2}\Big)^{n-1}, \qquad x \in (0,1].$$

The probability that OPT gets a quantile of at least q is $1-q^n$. For any fixed $q \in [0,1]$, let $f(q)$ denote the probability that the algorithm selects an agent whose quantile is at least q. Conditioned on $X = x > 0$, by symmetry of the random arrival order, the accepted quantile is distributed as an independent $\mathrm{Uni}[x,1]$ draw. Hence

$$\mathbf{Pr}\left[Q_{\mathsf{ALG}} \geq q \mid X = x\right] = \begin{cases} (1-x^n)\cdot \dfrac{1-q}{1-x}, & x \leq q, \\ 1-x^n, & x > q. \end{cases}$$

For the atom $X=0$, the algorithm accepts the first arrival unconditionally, and thus $\mathbf{Pr}[Q_{\mathsf{ALG}} \geq q \mid X = 0] = 1-q$. Combining gives

$$f(q) = 2^{-n}(1-q) \;+\; \int_0^q f_X(x)\,(1-x^n)\frac{1-q}{1-x}\,\mathrm{d}x \;+\; \int_q^1 f_X(x)\,(1-x^n)\,\mathrm{d}x. \tag{8}$$

Since F^{-1} is an arbitrary nondecreasing function of the quantile, it suffices to prove the tail inequality

$$f(q) \;\geq\; \frac{\ln 3}{2}\cdot(1-q^n) \qquad \forall q \in [0,1].$$

B.2 Large $n \geq 18$ case

Here, we prove the following lemma for large $n \geq 18$:

Lemma B.1 (Large n)**.** *For every* $n \geq 18$,

$$\inf_{q\in[0,1]} \frac{f(q)}{1-q^n} \;\geq\; \frac{\ln 3}{2}.$$

For convenience denote

$$g(x) \stackrel{\text{def}}{=\!=} f_X(x)\,(1-x^n), \qquad h(x) \stackrel{\text{def}}{=\!=} \frac{g(x)}{1-x} = f_X(x)\,\frac{1-x^n}{1-x}, \qquad A(q) \stackrel{\text{def}}{=\!=} 2^{-n} + \int_0^q h(x)\,\mathrm{d}x.$$

Then (8) can be written as

$$f(q) = (1-q)A(q) + \int_q^1 g(x)\,\mathrm{d}x. \tag{9}$$

Differentiating gives

$$f'(q) = -A(q).$$

In particular, both $f(q)$ and $1-q^n$ are strictly decreasing on $[0,1)$.

Cauchy mean-value form. For any $q \in [0,1)$, there exists $\xi \in (q,1)$ such that

$$\frac{f(q)}{1-q^n} = \frac{f'(\xi)}{-n\,\xi^{n-1}} = \frac{A(\xi)}{n\,\xi^{n-1}} \stackrel{\text{def}}{=\!=} H(\xi).$$

Apply Cauchy's mean value theorem to $F = f$ and $G(q) = 1 - q^n$ on the interval $[q,1]$. Using $F(1) = G(1) = 0$ and $G'(\xi) \neq 0$ for $\xi \in (q,1)$ gives the claimed representation.

Next, we will show that H is nonincreasing on $(0,1)$. This implies the worst case over q occurs at the endpoint $q \to 1$, i.e.,

$$\inf_{q\in[0,1]} \frac{f(q)}{1-q^n} = \lim_{q\to 1} \frac{f(q)}{1-q^n} = \frac{A(1)}{n}.$$

Finally, we lower bound $A(1)/n$ by $\ln 3/2$ for $n \geq 18$ using an explicit expression for $A(1)$. The remainder of this subsection fills in the details.

B.2.1 Monotonicity of H via unimodality of an auxiliary function $D(\cdot)$

From the definition $H(t) = A(t)/(nt^{n-1})$ we obtain

$$H'(t) = \frac{t\,h(t) - (n-1)A(t)}{n\,t^n}.$$

Define

$$D(t) \stackrel{\text{def}}{=\!=} (n-1)A(t) - t\,h(t).$$

Then $H'(t) \leq 0$ is equivalent to $D(t) \geq 0$ for all $t \in (0,1)$. Differentiating D yields

$$D'(t) = (n-2)h(t) - t\,h'(t) = h(t)\big[(n-2) - \phi(t)\big],$$

where

$$\phi(t) \stackrel{\text{def}}{=\!=} \frac{t\,h'(t)}{h(t)} = \frac{(n-1)t}{1+t} + \frac{t}{1-t} - \frac{n\,t^n}{1-t^n}. \tag{10}$$

Then we show that $\phi(\cdot)$ is monotone increasing, which implies the unimodality of $D(\cdot)$. Actually, differentiate (10):

$$\phi'(t) = \frac{n-1}{(1+t)^2} + \frac{1}{(1-t)^2} - \frac{n^2 t^{n-1}}{(1-t^n)^2}.$$

Using AM–GM on $1, t, \ldots, t^{n-1}$ yields

$$\frac{1-t^n}{1-t} = \sum_{k=0}^{n-1} t^k \;\geq\; n\,t^{(n-1)/2} \quad \Longrightarrow \quad \frac{n^2 t^{n-1}}{(1-t^n)^2} \leq \frac{1}{(1-t)^2}.$$

Substituting gives $\phi'(t) \geq \frac{n-1}{(1+t)^2} > 0$.

B.2.2 Endpoint analysis of D

We have $D(0) = (n-1)2^{-n} > 0$. For $D(1)$, we have $h(1) = \frac{n^2}{2}$ and thus

$$D(1) = (n-1)A(1) - \frac{n^2}{2}.$$

Now we show that for all $n \geq 18$,

$$(n-1)A(1) > \frac{n^2}{2}.$$

We begin with an explicit expression for $A(1)$. Recall

$$h(x) = f_X(x)\,\frac{1-x^n}{1-x} \;=\; \frac{n}{2}\Big(\frac{1+x}{2}\Big)^{n-1}\sum_{k=0}^{n-1} x^k.$$

Expanding $\big(\frac{1+x}{2}\big)^{n-1} = 2^{-(n-1)}\sum_{j=0}^{n-1}\binom{n-1}{j}x^j$ and integrating term-by-term yields

$$\int_0^1 h(x)\,\mathrm{d}x = \frac{n}{2}\cdot 2^{-(n-1)}\sum_{j=0}^{n-1}\binom{n-1}{j}\sum_{k=0}^{n-1}\frac{1}{j+k+1}.$$

Let $J \sim \mathrm{Binomial}(n-1,\frac{1}{2})$, i.e., $\mathbf{Pr}[J=j] = 2^{-(n-1)}\binom{n-1}{j}$. Then

$$\int_0^1 h(x)\,\mathrm{d}x = \frac{n}{2}\,\mathbb{E}_J\left[\sum_{k=0}^{n-1}\frac{1}{J+k+1}\right] = \frac{n}{2}\,\mathbb{E}_J\left[\sum_{k=1}^{n}\frac{1}{J+k}\right].$$

By definition $A(1) = 2^{-n} + \int_0^1 h(x)\,\mathrm{d}x$, hence

$$A(1) = 2^{-n} + \frac{n}{2}\,\mathbb{E}_J\left[\sum_{k=1}^{n}\frac{1}{J+k}\right]. \tag{11}$$

Define the function

$$G(x) \stackrel{\text{def}}{=\!=} \sum_{k=1}^{n}\frac{1}{x+k}, \qquad x \ge 0.$$

For integer $x = J$, we have $G(J) = H_{J+n} - H_J$. Moreover, G is convex on $[0,\infty)$ since

$$G''(x) = 2\sum_{k=1}^{n}\frac{1}{(x+k)^3} > 0.$$

Therefore, by Jensen's inequality and $\mathbb{E}[J] = (n-1)/2$,

$$\mathbb{E}_J\left[G(J)\right] \;\ge\; G(\mathbb{E}[J]) = \sum_{k=1}^{n}\frac{1}{\frac{n-1}{2}+k} \ge \int_0^n \frac{1}{\frac{n+1}{2}+x}\,\mathrm{d}x = \ln\frac{\frac{n+1}{2}+n}{\frac{n+1}{2}} = \ln\frac{3n+1}{n+1}. \tag{12}$$

Combining (11) and (12) yields

$$A(1) \;\ge\; 2^{-n} + \frac{n}{2}\ln\frac{3n+1}{n+1}.$$

It suffices to show

$$\ln\frac{3n+1}{n+1} \;\ge\; \frac{n}{n-1}.$$

The LHS is strictly increasing in n since

$$\frac{\mathrm{d}}{\mathrm{d}n}\ln\frac{3n+1}{n+1} = \frac{2}{(3n+1)(n+1)} > 0,$$

while the RHS is strictly decreasing for $n > 1$. Therefore, it is enough to verify the inequality at $n = 18$:

$$\ln\frac{55}{19} \approx 1.063 \;>\; \frac{18}{17} \approx 1.059.$$

This proves $(n-1)A(1) > \frac{n^2}{2}$ and hence $D(1) > 0$ for all $n \ge 18$.

B.2.3 Verify that $\frac{A(1)}{n} \geq \frac{\ln 3}{2}$

It remains to justify that for every $n \geq 18$ we indeed have $\frac{A(1)}{n} \geq \frac{\ln 3}{2}$. Recall from (11) that

$$\frac{A(1)}{n} = \frac{2^{-n}}{n} + \frac{1}{2}\,\mathbb{E}_J\left[G(J)\right], \qquad G(x) \stackrel{\text{def}}{=} \sum_{k=1}^{n} \frac{1}{x+k}, \qquad J \sim \text{Binomial}(n-1, \tfrac{1}{2}).$$

Thus, it suffices to show $\mathbb{E}_J[G(J)] \geq \ln 3$. Let $\mu \stackrel{\text{def}}{=} \mathbb{E}[J] = (n-1)/2$ and note that $\mathbf{Var}[J] = (n-1)/4$. Since

$$G''(x) = 2\sum_{k=1}^{n} \frac{1}{(x+k)^3},$$

we have $G''(x) \geq G''(n-1) = 2\sum_{k=n}^{2n-1} \frac{1}{k^3} \geq 2\int_n^{2n} \frac{1}{z^3}\,\mathrm{d}z = \frac{3}{4n^2}$ for all $x \in [0, n-1]$. By Taylor's theorem (using the uniform lower bound on the second derivative), for all $x \in [0, n-1]$,

$$G(x) \geq G(\mu) + G'(\mu)(x-\mu) + \frac{3}{8n^2}(x-\mu)^2.$$

Taking expectation over $x = J$ cancels the linear term and gives

$$\mathbb{E}_J\left[G(J)\right] \geq G(\mu) + \frac{3}{8n^2}\,\mathbf{Var}\left[J\right] = G(\mu) + \frac{3(n-1)}{32n^2}. \tag{13}$$

Next we lower bound

$$G(\mu) = \sum_{k=1}^{n} \frac{1}{\mu+k} = \sum_{k=0}^{n-1} \frac{1}{\frac{n+1}{2}+k}.$$

Using that $1/x$ is convex, the trapezoidal rule yields

$$\int_0^n \frac{1}{\frac{n+1}{2}+x}\,\mathrm{d}x \leq \frac{1}{2}\left(\frac{1}{\frac{n+1}{2}} + \frac{1}{\frac{n+1}{2}+n}\right) + \sum_{k=1}^{n-1} \frac{1}{\frac{n+1}{2}+k},$$

and therefore

$$G(\mu) = \sum_{k=0}^{n-1} \frac{1}{\frac{n+1}{2}+k} \geq \ln \frac{\frac{n+1}{2}+n}{\frac{n+1}{2}} + \frac{1}{2}\left(\frac{1}{\frac{n+1}{2}} - \frac{1}{\frac{n+1}{2}+n}\right) = \ln\frac{3n+1}{n+1} + \left(\frac{1}{n+1} - \frac{1}{3n+1}\right). \tag{14}$$

Combining (13) and (14) gives

$$\mathbb{E}_J\left[G(J)\right] \geq \ln\frac{3n+1}{n+1} + \frac{2n}{(n+1)(3n+1)} + \frac{3(n-1)}{32n^2}.$$

Finally, since $\ln 3 = \ln\frac{3n+1}{n+1} + \ln\!\big(1 + \frac{2}{3n+1}\big)$ and $\ln(1+x) \leq x$ for $x \geq 0$, it suffices that

$$\frac{2n}{(n+1)(3n+1)} + \frac{3(n-1)}{32n^2} \;\geq\; \frac{2}{3n+1},$$

which is equivalent to

$$3(n-1)(3n+1)(n+1) \;\geq\; 64n^2.$$

The left-hand side equals $9n^3 + 3n^2 - 9n - 3$, so the inequality holds for all $n \geq 18$ (indeed, $9n^3 - 61n^2 - 9n - 3$ is increasing for $n \geq 5$ and positive at $n = 18$). Thus $\mathbb{E}_J[G(J)] \geq \ln 3$ for all $n \geq 18$, implying $A(1)/n \geq \ln 3/2$ and completing the large-n case.

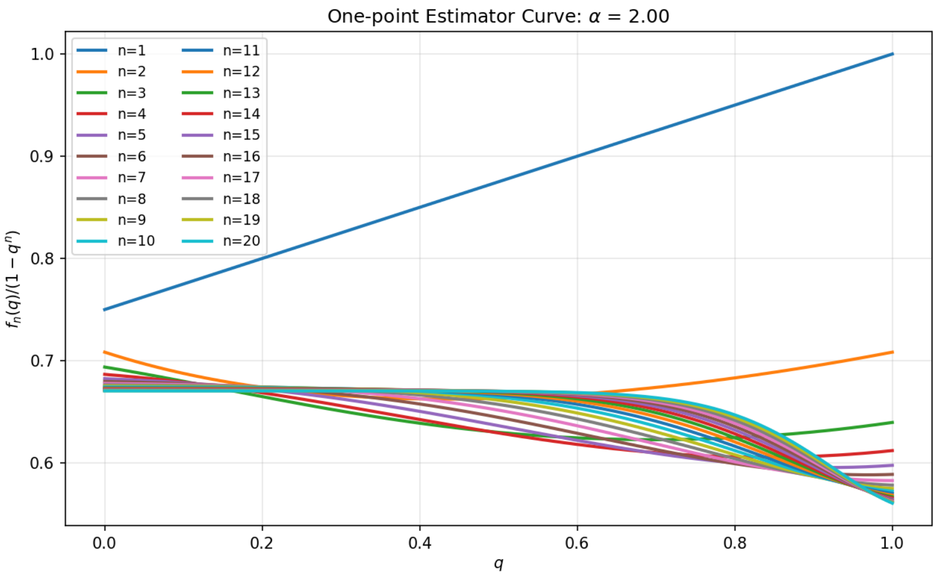

Figure 3: Ratio curves $f(q)/(1-q^n)$ for $n = 1, \dots, 20$.

B.3 Small n case

For $n < 18$, we verify $\inf_{q\in[0,1]} \frac{f(q)}{1-q^n} \geq \ln 3/2$ by a computer-assisted check based on symbolic integration, followed by exact polynomial manipulation and high-precision evaluation on a finite candidate set.

Lemma B.2 (Small n)**.** *For each $n \in \{1, 2, \dots, 17\}$, we have*

$$\inf_{q\in[0,1]} \frac{f(q)}{1-q^n} \geq \frac{\ln 3}{2}.$$

Computer-assisted proof. We provide a compact record of the verification methodology; the code is short and uses only standard symbolic and numerical routines (SymPy + high-precision evaluation).

Step 1: obtain an exact polynomial representation. For a fixed n, we symbolically compute the integral expression (8) for $f(q)$. We then divide by $(1-q)$ and form

$$P_n(q) \stackrel{\text{def}}{=\!=} \frac{f(q)}{1-q}.$$

For the specific integral structure here, $P_n(q)$ simplifies to a polynomial in q for each $n \in \{1, 2, \dots, 17\}$; for example, for $n = 1, \dots, 5$ we obtain:

$$\begin{aligned}
n = 1, \quad & P_n(q) = \tfrac{1}{4}q + \tfrac{3}{4}, \\
n = 2, \quad & P_n(q) = \tfrac{1}{24}q^3 + \tfrac{5}{24}q^2 + \tfrac{11}{24}q + \tfrac{17}{24}, \\
n = 3, \quad & P_n(q) = \tfrac{1}{80}q^5 + \tfrac{11}{160}q^4 + \tfrac{31}{160}q^3 + \tfrac{61}{160}q^2 + \tfrac{91}{160}q + \tfrac{111}{160}, \\
n = 4, \quad & P_n(q) = \tfrac{1}{224}q^7 + \tfrac{19}{672}q^6 + \tfrac{97}{1120}q^5 + \tfrac{209}{1120}q^4 + \tfrac{1117}{3360}q^3 \\
& \qquad + \tfrac{559}{1120}q^2 + \tfrac{699}{1120}q + \tfrac{769}{1120}, \\
n = 5, \quad & P_n(q) = \tfrac{1}{576}q^9 + \tfrac{29}{2304}q^8 + \tfrac{349}{8064}q^7 + \tfrac{799}{8064}q^6 + \tfrac{1471}{8064}q^5 \\
& \qquad + \tfrac{151}{504}q^4 + \tfrac{3571}{8064}q^3 + \tfrac{4621}{8064}q^2 + \tfrac{5251}{8064}q + \tfrac{5503}{8064}.
\end{aligned}$$

Step 2: reduce to a nonnegativity check on $[0,1]$. Fix a target constant α. To prove $\inf_{q\in[0,1]} \frac{f(q)}{1-q^n} \geq \alpha$, it suffices to prove

$$h_n(q) \stackrel{\text{def}}{=\!=} f(q) - \alpha(1-q^n) \geq 0 \qquad \forall q \in [0,1].$$

On $[0,1)$ we can equivalently divide by $(1-q) > 0$ and use $1-q^n = (1-q)\sum_{k=0}^{n-1} q^k$ to obtain the polynomial inequality

$$\widetilde{h}_n(q) \stackrel{\text{def}}{=\!=} P_n(q) - \alpha \sum_{k=0}^{n-1} q^k \geq 0 \qquad \forall q \in [0,1).$$

The endpoint $q=1$ is handled separately by taking the limit $q \uparrow 1$ (both sides admit finite limits). In the implementation, we take $\alpha = \frac{11}{20} = 0.55$, which is *strictly larger* than $\ln 3/2 \approx 0.5493$; therefore, this verified bound implies the desired $\ln 3/2$ guarantee.

Step 3: global minimization via critical points. Since $\widetilde{h}_n$ is a polynomial, we compute its derivative $\widetilde{h}'_n$ exactly, find all real roots of $\widetilde{h}'_n$ in $[0,1]$ (using high-precision root finding), and evaluate $\widetilde{h}_n$ at the endpoints and all such critical points with high precision. The minimum over this finite candidate set equals the global minimum on $[0,1]$.

Step 4: numerical cross-check. As an independent sanity check, we also evaluate the defining integrals of $f(q)$ numerically using high-order Gauss–Legendre quadrature on a dense mesh of q values, including points very close to 1; this matches the symbolic computations and consistently yields a positive margin above 0 for $\widetilde{h}_n(q)$.

Applying the above verification procedure for each $n = 1, 2, \ldots, 17$ yields $\min_{q\in[0,1)} \widetilde{h}_n(q) \geq 0$ and $\lim_{q\uparrow 1} \widetilde{h}_n(q) \geq 0$ with a nontrivial numerical margin (well above floating-point noise), which establishes the claim. $\square$

Conclusion of the proof of Theorem 2.2. Combining Lemmas B.1 and B.2 yields that

$$\inf_{q\in[0,1]} \frac{f(q)}{1-q^n} \geq \frac{\ln 3}{2}$$

for all n, which implies the claimed competitive ratio in value space.

C Deferred Proofs from Section 4

We collect here the proofs deferred from the analysis of Algorithm 1: Claim 4.3 (Case (A)), the Case (B) derivations (inequality (2) for subcase B.1 and Claim 4.4 for subcase B.2), and Claim 4.5 (birthday-paradox reduction). We first present the following simple properties of the optimal online and offline algorithms.

Lemma C.1. *For any positive integers $n > m$*

$$\mathit{ON}(m) \geq \frac{m}{n} \cdot \mathit{ON}(n), \quad \mathit{OFF}(m) \geq \frac{m}{n} \cdot \mathit{OFF}(n).$$

Proof. We begin with the property of the offline benchmark. $\mathsf{OFF}(n)$ represents the expected maximum value among n agents, however, it is in the first m agents with probability $\frac{m}{n}$. Formally, denote $i^* = \operatorname{argmax}_{i\in[n]} v_i$, we have

$$\begin{aligned}
\mathsf{OFF}(m) &= \mathbb{E}_{\mathbf{v}\sim\mathbf{F}}\left[\max_{i\in[m]} v_i\right] \\
&\geq \mathbb{E}_{\mathbf{v}\sim\mathbf{F}}\left[\mathbb{I}\left[i^* \in [m]\right] \cdot \max_{i\in[n]} v_i\right] \\
&= \mathbb{E}_{\mathbf{v}\sim\mathbf{F}}\left[\max_{i\in[n]} v_i\right] \cdot \mathbf{Pr}\left[i^* \in [m]\right] = \frac{m}{n} \cdot \mathsf{OFF}(n).
\end{aligned}$$

Then we turn to the online benchmark property. We prove it by induction, it suffices to prove for the case $n = m+1$. Assume towards a contradiction that $\mathsf{ON}(m+1) > \frac{m+1}{m} \cdot \mathsf{ON}(m)$. Then we construct a strategy for m agents with utility at least $\frac{m \cdot \mathsf{ON}(m+1)}{m+1}$ based on the $m+1$ optimal thresholds. This would contradict the optimality of $\mathsf{ON}(m)$.

Let the thresholds for $m+1$ agents be $\{\tau_i\}_{i \in [m+1]}$ Then we construct thresholds $\{\tau_i'\}_{i \in [m]}$ for m agents as follows: we first denote the contribution of the i-th agent as

$$Con(i) = \prod_{j=1}^{i-1} \mathbf{Pr}\left[v_j < \tau_j\right] \cdot \mathbb{E}_{v_i \sim F_v}\left[v_i \cdot \mathbb{I}\Big[v_i \geq \tau_i\Big]\right].$$

Since the sum of all $m+1$ agents' contribution is $\mathsf{ON}(m+1)$, there must exist an agent whose contribution is $\leq \frac{\mathsf{ON}(m+1)}{m+1}$. Suppose $k = \operatorname{argmin}_{i \in [m+1]} Con(i)$ and we "skip" it in the m agents case by setting

$$\tau_i' = \begin{cases} \tau_i & \text{if } i < k, \\ \tau_{i+1} & \text{if } i \geq k. \end{cases}$$

The contributions of the first $k-1$ agents remain the same. However, the contributions of other agents become larger without the risk that the player will accept the original k-th agent. Formally, $\forall i \geq k$ we have

$$\begin{aligned} Con'(i) &= \prod_{j=1}^{i-1} \mathbf{Pr}\left[v_j < t_j'\right] \cdot \mathbb{E}_{v_i \sim F_v}\left[v_i \cdot \mathbb{I}(v_i \geq t_i')\right] \\ &= \prod_{j=1}^{i} \mathbf{Pr}\left[v_j < \tau_j\right] \cdot \frac{\mathbb{E}_{v_{i+1} \sim F_v}[v_{i+1} \cdot \mathbb{I}(v_{i+1} \geq \tau_{i+1})]}{\mathbf{Pr}[v_k < \tau_k]} \\ &= \frac{Con(i+1)}{\mathbf{Pr}[v_k < \tau_k]} \geq Con(i+1). \end{aligned}$$

Thus, the utility of this policy for m agents is at least $\frac{m \cdot \mathsf{ON}(m+1)}{m+1} > \mathsf{ON}(m)$ which is a contradiction. □

C.1 Proof of Claim 4.3 (Case (A))

Proof. We present the proof only for the ON benchmark, as for the OFF benchmark it is almost the same. In the lower bound (1) on $ALG(n)$ we shall only use that

$$ALG(n) \geq \mathbb{E}_{\mathbf{t}}\left[\mathbb{I}\Big[n_\varepsilon(\mathbf{t}) < \theta\Big] \cdot A(n - n_\varepsilon(\mathbf{t}))\right],$$

as the probability $\mathbf{Pr}[n_\varepsilon < \theta] = 1 - O(\varepsilon)$ by the Chernoff bound. We get the desired bound as follows

$$\begin{aligned} \frac{ALG(n)}{\mathsf{ON}(n)} &\geq \mathbb{E}_{n_\varepsilon}\left[\mathbb{I}\Big[n_\varepsilon < \theta\Big] \cdot \frac{A(n-n_\varepsilon)}{\mathsf{ON}(n-n_\varepsilon)} \cdot \frac{\mathsf{ON}(n-n_\varepsilon)}{\mathsf{ON}(n)}\right] \\ &\geq \min_{k \in [N]} \frac{A(k)}{\mathsf{ON}(k)} \cdot \mathbb{E}_{n_\varepsilon}\left[\mathbb{I}\Big[n_\varepsilon < \theta\Big] \cdot \frac{\mathsf{ON}(n-n_\varepsilon)}{\mathsf{ON}(n)}\right] \\ &\geq \min_{k \in [N]} \frac{A(k)}{\mathsf{ON}(k)} \cdot \mathbb{E}_{n_\varepsilon}\left[\mathbb{I}\Big[n_\varepsilon < \theta\Big] \cdot \frac{n-n_\varepsilon}{n}\right] \\ &\geq \min_{k \in [N]} \frac{A(k)}{\mathsf{ON}(k)} \cdot \Big(\mathbf{Pr}\left[n_\varepsilon < \theta\right] - \mathbb{E}_{n_\varepsilon}\left[\frac{n_\varepsilon}{n}\right]\Big) \\ &= \min_{k \in [N]} \frac{A(k)}{\mathsf{ON}(k)} \cdot (1 - O(\varepsilon)), \end{aligned}$$

where the second inequality holds, as $\frac{A(n-n_\varepsilon)}{\mathsf{ON}(n-n_\varepsilon)} \geq \min_{k \in [N]} \frac{A(k)}{\mathsf{ON}(k)}$ for any value of $n_\varepsilon \geq 0$; the third inequality holds by Lemma C.1; the last inequality holds by linearity of expectation and since

$\mathbb{E}[\mathbb{I}[n_\varepsilon < \theta] \cdot \frac{n_\varepsilon}{n}] \le \mathbb{E}[\frac{n_\varepsilon}{n}]$; and the last equality holds since $\mathbb{E}[\frac{n_\varepsilon}{n}] = \varepsilon$ and $\mathbf{Pr}[n_\varepsilon < \theta] = 1 - O(\varepsilon)$ when $n < N - 2N^{2/3}$.

Thus, ALG loses at most $O(\varepsilon)$ in γ_{ON} (or in γ_{OFF}), compared to $\min_{k\in[N]} \frac{A(k)}{\mathsf{ON}(k)}$ (or $\min_{k\in[N]} \frac{A(k)}{\mathsf{OFF}(k)}$) in Case (A) when n is small. □

C.2 Case (B): Proof of inequality (2) and Claim 4.4

In Case (B), where $n > N - 2N^{2/3}$, we apply the Chernoff bound to the number of observed agents n_ε. Let $\mu \overset{\text{def}}{=\!=} \mathbb{E}[n_\varepsilon] = \varepsilon \cdot n$. Then the event $\mathcal{E}_1$ that n_ε is close to μ, $\mathcal{E}_1 \overset{\text{def}}{=\!=} \mathbb{I}\big[\,|n_\varepsilon - \mu| \le \mu^{2/3}\big]$, has probability $\mathbf{Pr}[\mathcal{E}_1] = 1 - O(\varepsilon)$. Note that in the event $\mathcal{E}_1(n_\varepsilon)$ we have:

$$n - n_\varepsilon \ge \eta(n_\varepsilon) \ge (1 - O(\varepsilon)) \cdot n. \tag{15}$$

The first inequality in (15) is equivalent to $n_\varepsilon - 2(1-\varepsilon)n_\varepsilon^{\frac{2}{3}} \le \mu$. We have:

$$n_\varepsilon - 2(1-\varepsilon)n_\varepsilon^{\frac{2}{3}} \le n_\varepsilon - \frac{4}{3}n_\varepsilon^{\frac{2}{3}} \le n_\varepsilon - \mu^{\frac{2}{3}} \le \mu,$$

where the first inequality holds by $\varepsilon \le \frac{1}{3}$; the second inequality is due to $n_\varepsilon \ge \mu - \mu^{\frac{2}{3}}$; the last inequality holds by $n_\varepsilon \le \mu + \mu^{\frac{2}{3}}$. The second inequality in (15) follows from

$$\begin{aligned}
\frac{n_\varepsilon - 2n_\varepsilon^{2/3}}{n} \cdot \frac{1-\varepsilon}{\varepsilon} &= \frac{n_\varepsilon - 2n_\varepsilon^{2/3}}{\mu} \cdot (1-\varepsilon) \\
&\ge \frac{\mu - 4\mu^{2/3}}{\mu} \cdot (1-\varepsilon) \\
&= 1 - O(\varepsilon + \mu^{-\frac{1}{3}}),
\end{aligned}$$

where the inequality is due to $n_\varepsilon \le \mu + \mu^{\frac{2}{3}}$. Since we assume $n > N - 2N^{2/3}$ in Case (B), we have $\mu = \varepsilon n = \Omega(\varepsilon^{-3})$, and therefore $O(\varepsilon + \mu^{-\frac{1}{3}}) = O(\varepsilon)$.

Thus, for the indicator $\mathbb{I}[n - n_\varepsilon \ge \eta(n_\varepsilon)] = 1$ in (1), we can simplify the bound (1) to

$$\begin{aligned}
ALG(n) &\ge \mathbb{E}_{n_\varepsilon}\left[ALG(n) \cdot \mathbb{I}\Big[\mathcal{E}_1\Big]\right] \\
&\ge \mathbf{E}_{n_\varepsilon}\left[\left(\mathbb{I}\Big[n_\varepsilon < \theta\Big] \cdot A(n - n_\varepsilon) + \mathbb{I}\Big[n_\varepsilon \ge \theta\Big] \cdot \mathsf{ON}(n) \cdot \frac{\mathsf{ON}(\eta(n_\varepsilon))}{\mathsf{ON}(n)}\right) \cdot \mathbb{I}\Big[\mathcal{E}_1\Big]\right].
\end{aligned} \tag{16}$$

Subcase B.1 ($n \ge N$): proof of (2). By the Chernoff bound, $\mathbf{Pr}[n_\varepsilon < \theta] = O(\varepsilon)$, so we can ignore the term $\mathbb{I}[n_\varepsilon < \theta]$ in (16) and obtain

$$\begin{aligned}
ALG(n) &\ge \mathbb{E}_{n_\varepsilon}\left[\mathbb{I}\Big[n_\varepsilon \ge \theta\Big] \cdot \mathbb{I}\Big[\mathcal{E}_1\Big] \cdot \frac{\eta(n_\varepsilon)}{n}\right] \cdot \mathsf{ON}(n) \\
&\ge (1 - O(\varepsilon)) \cdot \mathbb{E}_{n_\varepsilon}\left[\mathbb{I}\Big[n_\varepsilon \ge \theta\Big] \cdot \mathbb{I}\Big[\mathcal{E}_1\Big]\right] \cdot \mathsf{ON}(n) \\
&\ge (1 - O(\varepsilon)) \cdot \big(1 - \mathbf{Pr}\left[n_\varepsilon < \theta\right] - \mathbf{Pr}\left[\overline{\mathcal{E}_1}\right]\big) \cdot \mathsf{ON}(n) \\
&= (1 - O(\varepsilon)) \cdot \mathsf{ON}(n) \ge (1 - O(\varepsilon)) \cdot A(n),
\end{aligned}$$

where the first inequality holds by Lemma C.1; the second inequality holds, as under $\mathcal{E}_1$ the ratio $\frac{\eta(n_\varepsilon)}{n} \ge 1 - O(\varepsilon)$; the third inequality holds by the union bound; the last equality holds since $\mathbf{Pr}[\mathcal{E}_1] = 1 - O(\varepsilon)$ and $\mathbf{Pr}[n_\varepsilon < \theta] = O(\varepsilon)$ by the Chernoff bounds. This proves (2) and is in fact stronger than what we had in Claim 4.3.

Subcase B.2 ($N - 2N^{2/3} < n < N$)**: proof of Claim 4.4.** We prove the claim for the ON benchmark (for the OFF benchmark, the proof is almost identical). We continue to simplify the lower bound (16) on $ALG(n)$:

$$\begin{aligned}
ALG(n) &\geq \mathbf{E}_{n_\varepsilon}\left[\mathbb{I}\left[\mathcal{E}_1\right]\cdot\left(\mathbb{I}\left[n_\varepsilon < \theta\right]\cdot A(n-n_\varepsilon) + \mathbb{I}\left[n_\varepsilon \geq \theta\right]\cdot \mathsf{ON}(n)\cdot\frac{\eta(n_\varepsilon)}{n}\right)\right]\\
&\geq (1-O(\varepsilon))\cdot\mathbf{E}_{n_\varepsilon}\left[\mathbb{I}\left[\mathcal{E}_1\right]\cdot\left(\mathbb{I}\left[n_\varepsilon < \theta\right]\cdot A(n-n_\varepsilon) + \mathbb{I}\left[n_\varepsilon \geq \theta\right]\cdot \mathsf{ON}(n)\right)\right]\\
&\geq (1-O(\varepsilon))\cdot\mathbb{E}_{n_\varepsilon}\left[A(n-n_\varepsilon)\mathbb{I}\left[\mathcal{E}_1\right]\right],
\end{aligned}$$

where the first inequality holds by Lemma C.1; the second inequality holds as $\frac{\eta(n_\varepsilon)}{n} = 1 - O(\varepsilon)$ in the event $\mathcal{E}_1$; the third inequality holds as $\mathsf{ON}(n) \geq A(n-n_\varepsilon)$ and $\mathbb{I}[n_\varepsilon < \theta] + \mathbb{I}[n_\varepsilon \geq \theta] = 1$. We conclude the proof by observing that

$$\begin{aligned}
\frac{ALG(n)}{\mathsf{ON}(n)} \geq&(1-O(\varepsilon))\cdot\mathbf{E}_{n_\varepsilon}\left[\frac{A(n-n_\varepsilon)}{\mathsf{ON}(n-n_\varepsilon)}\cdot\frac{\mathsf{ON}(n-n_\varepsilon)}{\mathsf{ON}(n)}\mathbb{I}\left[\mathcal{E}_1\right]\right]\\
\geq&(1-O(\varepsilon))\cdot\min_{k\in[N]}\frac{A(k)}{\mathsf{ON}(k)}\cdot\mathbb{E}_{n_\varepsilon}\left[\frac{n-n_\varepsilon}{n}\mathbb{I}\left[\mathcal{E}_1\right]\right]\\
\geq&(1-O(\varepsilon))\cdot\min_{k\in[N]}\frac{A(k)}{\mathsf{ON}(k)}\cdot\left(\mathbf{Pr}_{n_\varepsilon}\left[\mathcal{E}_1\right] - \mathbb{E}_{n_\varepsilon}\left[\frac{n_\varepsilon}{n}\right]\right)\\
=&(1-O(\varepsilon))\cdot\min_{k\in[N]}\frac{A(k)}{\mathsf{ON}(k)},
\end{aligned}$$

where the first inequality follows from our bound on $ALG(n)$ above; the second inequality holds as $\frac{A(n-n_\varepsilon)}{\mathsf{ON}(n-n_\varepsilon)} \geq \min_{k\in[N]}\frac{A(k)}{\mathsf{ON}(k)}$; third inequality follows from the linearity of expectations and since $\mathbb{E}_{n_\varepsilon}[\frac{n_\varepsilon}{n}\mathbb{I}[\mathcal{E}_1]] \leq \mathbb{E}_{n_\varepsilon}[\frac{n_\varepsilon}{n}]$; the last equality holds as $\mathbb{E}_{n_\varepsilon}[\frac{n_\varepsilon}{n}] = \varepsilon$ and $\mathbf{Pr}[\mathcal{E}_1] = 1 - O(\varepsilon)$.

C.3 Proof of Claim 4.5

Proof. The second inequality is trivial, since the objective becomes smaller for any A and demand n. We now prove the first inequality. Similarly to the birthday paradox, we observe that $\mathcal{E}$ occurs with high probability for sufficiently large T and $\forall n \in [N]$. Specifically,

$$\mathbf{Pr}\left[\overline{\mathcal{E}}\mid n\right] = 1 - \prod_{i=1}^{n-1}\left(1-\frac{i}{T}\right) \leq \sum_{i=1}^{N-1}\frac{i}{T} = O(\frac{N^2}{T}) = O(\varepsilon).$$

Hence, for any algorithm A and demand n, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{t},\mathbf{v}}\left[A(\mathbf{t},\mathbf{v})\right] =&\mathbb{E}\left[A(\mathbf{t},\mathbf{v})\cdot\mathbb{I}\left[\mathcal{E}(\mathbf{t})\right] + A(\mathbf{t},\mathbf{v})\cdot\mathbb{I}\left[\overline{\mathcal{E}(\mathbf{t})}\right]\right]\\
\leq&\mathbb{E}\left[A(\mathbf{t},\mathbf{v})\cdot\mathbb{I}\left[\mathcal{E}\right]\right] + \mathsf{ON}(n)\cdot\mathbb{E}\left[\mathbb{I}\left[\overline{\mathcal{E}(\mathbf{t})}\right]\right]\\
=&\mathbb{E}\left[A(\mathbf{t},\mathbf{v})\cdot\mathbb{I}\left[\mathcal{E}\right]\right] + \mathsf{ON}(n)\cdot O(\varepsilon),
\end{aligned}$$

where the inequality follows from the fact that for any fixed time $\mathbf{t}$ the online algorithm $\mathbb{E}_{\mathbf{v}}[A(\mathbf{t},\mathbf{v})] \leq \mathsf{ON}(n)$ cannot do better than the optimum online algorithm that knows n. The claim follows after we divide the above by $\mathsf{ON}(n)$ and take the supremum sup over all possible algorithms A. □

D Hedge Algorithm Convergence for the Separation Instance

Figure 4 plots the per-round competitive ratio of the algorithm player's best response against the adversary's Hedge-updated mixed strategy on the separation instance of Theorem 5.1, starting from a uniform prior on $n \in \{1, \ldots, 10\}$. The ratio drops monotonically from about 0.80, crosses the I.I.D. prophet bound $\alpha \approx 0.745$ at round 342, and stabilizes below 0.739 by round 10^3—consistent with the $O(\sqrt{\log(N)/R})$ convergence to an approximate equilibrium established in Theorem 4.1.

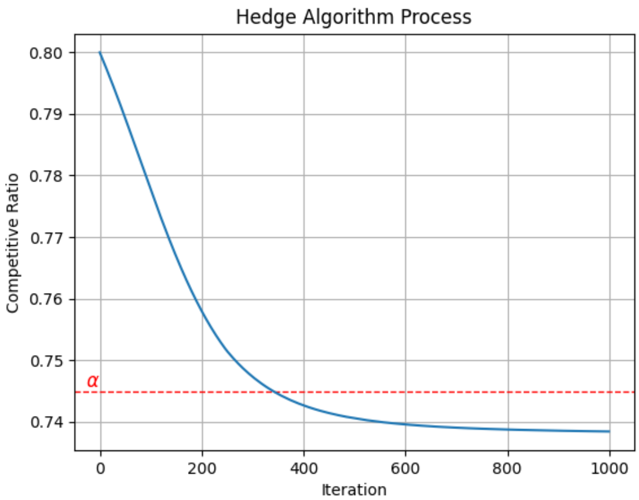

Figure 4: Competitive ratio of the algorithm player's best response against the adversary's mixed strategy over n, plotted across rounds of the Hedge dynamic on the separation instance. The dashed line marks the I.I.D. prophet bound $\alpha \approx 0.745$.