informs
http://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–30
ISSN 0364-765X (print), ISSN 1526-5471 (online)

Hiring Secretaries over Time: The Benefit of Concurrent Employment

Yann Disser,[a] John Fearnley,[b] Martin Gairing,[b] Oliver Göbel,[c] Max Klimm,[d] Daniel Schmand,[e] Alexander Skopalik,[f] Andreas Tönnis[g]

[a] Graduate School of Computational Engineering, Technical University of Darmstadt, Darmstadt 64293, Germany; [b] Department of Computer Science, University of Liverpool, Liverpool L69 3BX, United Kingdom; [c] Department of Computer Science, RWTH Aachen University, Aachen 52062, Germany; [d] School of Business and Economics, Humboldt University of Berlin, Berlin 10099, Germany; [e] Department of Computer Science, Goethe University Frankfurt, Frankfurt am Main 60054, Germany; [f] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede 7500 AE, Netherlands; [g] Institute of Computer Science, University of Bonn, Bonn 53115, Germany

Contact: disser@mathematik.tu-darmstadt.de, https://orcid.org/0000-0002-2085-0454 (YD); john.fearnley@liverpool.ac.uk (JF); gairing@liverpool.ac.uk (MG); goebel@cs.rwth-aachen.de (OG); max.klimm@hu-berlin.de, https://orcid.org/0000-0002-9061-2267 (MK); schmand@em.uni-frankfurt.de, https://orcid.org/0000-0001-7776-3426 (DS); a.skopalik@utwente.nl (AS); atoennis@uni-bonn.de (AT)

Received: May 30, 2017
Revised: May 30, 2018
Accepted: December 23, 2018
Published Online in Articles in Advance: August 29, 2019

MSC2000 Subject Classification: Primary: 60G40, 62L15, 68W27; secondary: 68W40, 68Q87

OR/MS Subject Classification: Primary: computers/computer science, analysis of algorithms; secondary: probability: Markov processes

https://doi.org/10.1287/moor.2019.0993

Abstract. We consider a stochastic online problem where n applicants arrive over time, one per time step. Upon the arrival of each applicant, their cost per time step is revealed, and we have to fix the duration of employment, starting immediately. This decision is irrevocable; that is, we can neither extend a contract nor dismiss a candidate once hired. In every time step, at least one candidate needs to be under contract, and our goal is to minimize the total hiring cost, which is the sum of the applicants' costs multiplied with their respective employment durations. We provide a competitive online algorithm for the case that the applicants' costs are drawn independently from a known distribution. Specifically, the algorithm achieves a competitive ratio of 2.965 for the case of uniform distributions. For this case, we give an analytical lower bound of 2 and a computational lower bound of 2.148. We then adapt our algorithm to stay competitive even in settings with one or more of the following restrictions: (i) at most two applicants can be hired concurrently; (ii) the distribution of the applicants' costs is unknown; (iii) the total number n of time steps is unknown. On the other hand, we show that concurrent employment is a necessary feature of competitive algorithms by proving that no algorithm has a competitive ratio better than $\Omega(\sqrt{n}/\log n)$ if concurrent employment is forbidden.

Funding: Y. Disser was supported by the Excellence Initiative of the German federal and state governments and the Graduate School of Computational Engineering at the Technical University of Darmstadt [GSC 233/2]. M. Klimm's research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ [EXC-2046/1, project ID: 390685689].

Keywords: online algorithm • stopping problem • prophet inequality • Markov chain • secretary problem

1. Introduction

The theory of optimal stopping is concerned with problems of finding the best points in time to take a certain action based on a sequence of sequentially observed random variables. Problems of this kind are ubiquitous in the area of operations research, for example, when hiring, selling, purchasing, or procurement decisions are made based on the partial observation of a sequence of offers with known statistical properties. In one of the most basic stopping problems, a gambler sequentially observes realizations $x_1 \sim X_1$, $x_2 \sim X_2$, ... of a series of independent random variables. After being presented a realization $x_i \sim X_i$, the gambler has to decide immediately whether to keep the realization x_i as a prize or to continue gambling, hoping for a better realization. For this setting, the famous prophet inequality due to Krengel, Sucheston, and Garling (see Krengel and Sucheston [23, 24]) asserts that the best stopping rule of the gambler achieves in expectation at least half the optimal outcome of a prophet that foresees the realizations of all random variables and, thus, gains the expected maximal realization of all variables.

After the surprising result of Krengel, Sucheston, and Garling (Krengel and Sucheston [23, 24]), prophet-type inequalities were provided for several generalizations of their model, including settings where both the gambler and the prophet may stop multiple times (see Alaei [1], Kennedy [20]) and settings where both choose a set subject to a matroid constraint (see Kleinberg and Weinberg [22]), polymatroid constraints (see Dütting and Kleinberg [7]), and general constraints (see Rubinstein [28]).

In light of this remarkable progress in establishing prophet-type inequalities for various stochastic environments, two remarks are in order. First, the known results consider maximization problems only. Although obviously important as a model for situations where, for example, items are to be sold and offers for the items arrive over time, they do not capture the "dual" problem, where items need to be procured. In fact, minimization problems in stochastic environments are sparsely studied. The only work on minimization in the prophet inequality/secretary context is by Esfandiari et al. [8]. They show that there is no stopping rule that allows for a constant factor approximation compared with the prophet's outcome, even in the most basic case of single stopping and independent and identically distributed (i.i.d.) distributions. Second, the models above are inherently *static* in the sense that the objective depends only on the set of chosen realizations *at the end* of the sequence. This is a reasonable assumption when the underlying selling or purchase decisions have a long-term impact and the time during which the sequence of random variables is observed can be neglected. On the other hand, they fail to capture the natural situation where realizations are observed for a long period of time and selling or procurement decisions are taking effect even while further offers are observed.

To illustrate the key differences between static and dynamic settings, consider a firm that in each time step needs to be able to perform a certain task to be operational. Traditionally, the firm could advertise a position and hire an applicant able to perform the task. Assuming that the firm strives to minimize labour cost, this leads to a (static) prophet-type problem where the costs of the applicants are drawn from a distribution and the firm strives to minimize the realized costs. Alternatively, online marketplaces like oDesk.com and Freelance.com provide the opportunity to hire applicants with a limited contract duration and to possibly hire another contractor when a new offer with lower cost arrives. The constant rise of the revenue generated by these platforms (reaching USD 1 billion in 2014) suggests that the latter approach has growing economic importance (Verroios et al. [31]).

Hiring employees for a limited amount of time leads to a new kind of stopping problem where the ongoing observation period overlaps with the duration of contracts, and active contracts need to be maintained over time while receiving new offers. To model these situations, we study a natural setting where at least one contract needs to be active at each point in time and there is no additional benefit of having more than one active contract.[1] This covering constraint renders it beneficial to accept good offers even when other contracts are still active, and a key challenge is to manage the trade-off between accepting good offers and avoiding contract overlaps.

Specifically, we assume that in every time step $i \in [n]$, we observe the cost of the ith applicant x_i, where the values x_i are drawn i.i.d. from a common distribution X. In each time step i, we have to decide on a number of time steps t_i for which to hire the ith applicant. This duration is fixed irrevocably at time i, and extension or shortening of this duration is impossible later on. Hiring applicant i with realized cost x_i results in costs of $x_i t_i$. We are interested in minimizing the expected total hiring cost $\mathbf{E}_{x_1 \sim X,\ldots,x_n \sim X}\left[\sum_{i=1}^{n} t_i x_i\right]$, subject to the constraint that at least one applicant is under employment at all times.

1.1. Results and Outline

When the total number of time steps and the distribution are known, the dynamic stopping problem considered in this paper can be solved by a straightforward dynamic program (DP). The DP maintains a table of n^2 optimal threshold values depending on the number of remaining covered and uncovered time steps. Like other optimal solutions for similar stochastic optimization problems, the DP suffers from the fact that it relies on the exact knowledge of the distribution and the number of time steps, and does not allow us to quantify the optimal competitive ratio.

The results we give in this paper address these shortcomings. We give online algorithms with constant competitive ratios, and in doing so, we prove that the optimal online algorithm also gives a constant competitive ratio for any cost distribution that is known up front. Our techniques are robust with respect to incomplete information and can be extended to the case where the cost distribution and/or the total number of time steps is unknown, while still providing a constant competitive ratio. Furthermore, our approach is conceptually simple, efficient, and not tailored to specific distributions.

For ease of exposition, we present our algorithm in incremental fashion starting with a simplified version for uniform distributions in Section 4.1. The algorithm maintains different threshold values over time and hires applicants when their realized cost is below the threshold. By relating the execution of the algorithm with a Markov chain and by analyzing its hitting time, we bound the competitive ratio of the algorithm. In Section 4.2, we refine the algorithm and its analysis to show that it is 2.965-competitive in the uniform case. We provide an analytical lower bound of 2 for the best possible competitive ratio via a relaxation to the Cayley–Moser problem (see Moser [26]), and we give a computational lower bound of 2.14. For the analytical lower bound, we consider a relaxation of the problem where an applicant can be hired for any subset of future time steps (not necessarily contiguous and not necessarily starting immediately). We analyze the optimal online algorithm for the relaxation and show that it is 2-competitive for the relaxation, which implies a lower bound of 2 on the competitive ration of any online algorithm for the original problem. We further show that the optimal online algorithm for the relaxation is 2-competitive for a large class of distributions.

Subsequently, in Section 5, we generalize the algorithm to arbitrary distributions. Here, the main technical difficulty is to obtain a good estimation of the offline optimum. As we bound the offline optimum by a sum of conditional expectations given that the value lies in an interval bounded by exponentially decreasing quantiles of the distribution, we are able to derive a competitive ratio of 6.052.

In Section 6, we further generalize our techniques to give a constant competitive algorithm for the case where the distribution is unknown a priori. The main idea of the algorithm is to approximate the quantiles of the distribution by sampling.

Finally, in Section 7, we show that our algorithms remain competitive in the case that at most two applicants may be employed concurrently. We also extend our results to the case where the total number of applicants is unknown. In contrast to this, we show that the best possible online algorithm without concurrent employment has competitive ratio $\Theta(\sqrt{n}/\log n)$, even for uniform distributions.

To improve readability, we relegate the formal analysis of the underlying Markov chains to Section 8.

1.2. Related Work

The interest in optimal stopping rules for sequentially observed random experiments dates back at least as far as to Cayley [5], who asked for the optimal stopping rule when n tickets are drawn without replacement from a known pool of N tickets with different rewards (for more historical notes on this problem, see also Ferguson [9]). Cayley [5] solved this problem by backward induction, an approach later formalized by Bellman [4]. Moser [26] studied Cayley's [5] problem for the case that N is large and the N rewards are equal to the first N natural numbers. In that case, the problem can be approximated by n draws from the uniform distribution, and Moser [26] provided an approximation of the corresponding threshold values of the optimal stopping rule. For similar results for other distributions, see Gilbert and Mosteller [12], Guttman [15], and Karlin [19]. In Section 4.3, we will use the asymptotic behavior of the threshold due to Gilbert and Mosteller [12] to obtain a lower bound for our problem.

Krengel, Sucheston, and Garling (see Krengel and Sucheston [23, 24]) studied optimal stopping rules for arbitrary independent, nonnegative, but not necessarily identical random variables. Their famous prophet inequality asserts that the expected reward of a gambler who follows the optimal stopping rule (that can still be found using backward induction) is at least half the expected reward of a prophet who knows all realizations beforehand and will stop the sequence at the highest realization. Samuel-Cahn [30] showed that the same guarantee can be obtained by a simple stopping rule that uses a single threshold rather than n different thresholds as the solution of the dynamic program. Hill and Kertz [17] surveyed some variations of the problem.

More recently, Alaei [1] considered the setting where both the prophet and the gambler stop $k \in \mathbb{N}$ times and receive the sum of their realizations as rewards, and they gave an algorithm with competitive ratio $1-(k+3)^{-1/2}$. For a more general setting in which the selection of both the gambler and the prophet is restricted by a matroid constraint, Kleinberg and Weinberg [22] showed a tight competitive ratio of 1/2. Dütting and Kleinberg [7] generalized this result further to polymatroid constraints. Göbel et al. [14] studied a prophet inequality setting where a solution is feasible if it forms an independent set in an underlying network. They gave an online algorithm that achieves an $\mathcal{O}(\rho^2 \log n)$ approximation, where ρ is a structural parameter of the network. Very recently, Rubinstein [28] studied the problem for general downward-closed constraints. He gave an $\mathcal{O}(\log n \log r)$ approximation where r is the cardinality of the largest feasible set and showed that no online algorithm can be better than an $\mathcal{O}(\log n/\log\log n)$ approximation. For a generalization toward nonlinear valuation functions, see Rubinstein and Singla [29].

The recent interest in prophet inequalities is due to an interesting connection to mechanism design problems that was first made by Hajiaghayi et al. [16]. They remarked that threshold rules used to prove prophet inequalities correspond to truthful online mechanisms with the same approximation guarantee as the prophet inequality. Chawla et al. [6] noted that posted pricing mechanisms for revenue maximization can be derived from prophet inequalities by using the framework of virtual values due to Myerson [27]. As our algorithms operate on the basis of threshold values as well, they can also be turned into truthful mechanisms. However, the exact properties of these mechanisms deserve further investigation.

Esfandiari et al. [8] considered the minimization version of the classical prophet inequality setting. They showed that even for i.i.d. random variables, no stopping rule can achieve a constant approximation to the cost of a prophet. This is in contrast to our results for the dynamic prophet inequality setting, as we obtain a constant factor approximation even without knowledge of the distributions or n.

Further related are secretary problems (for a review, see Ferguson [9]), and in particular, secretary problems where the values are drawn from i.i.d. distributions as considered by Bearden [3]. The main difference from our model is that in secretary problems, the objective is to maximize the probability of selecting the best outcome. Yet, our algorithm developed in Section 6 for solving the case of unknown distributions is reminiscent of the optimal stopping rules for secretary problems, as it also employs a sampling phase in which the distribution is learned before hiring an applicant. Very recently, Fiat et al. [10] studied a dynamic secretary problem where secretaries are hired over time. In contrast to our work, they considered a maximization problem, and the contract duration was fixed.

2. Preliminaries

For a natural number $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. We consider a sequence $x_1 \sim X, x_2 \sim X, \ldots, x_n \sim X$ of n i.i.d. random variables drawn from a probability distribution X. Throughout this work, we assume that X is a continuous distribution with cumulative distribution F and probability density function f. Moreover, we assume that X assigns positive probability to nonnegative values only, that is, $F(0) = 0$. In every time step $i \in [n]$, the cost x_i of the ith applicant is revealed, and we must decide the number of time steps ti for which the applicant is hired. If $t_i = 0$, the applicant is not hired at all. The duration of the employment t_i is fixed irrevocably at time i; no extension or shortening of this duration at any further point in time is possible. Hiring applicant i with realized cost x_i for t_i time steps results in costs of $x_i t_i$. The objective is to minimize the expected total cost of hired applicants $\mathbf{E}[\sum_{i\in[n]} t_i x_i] := \mathbf{E}_{x_1\sim X,\ldots,x_n\sim X}\big[\sum_{i\in[n]} t_i x_i\big]$ subject to the constraint that at least one applicant is employed at each point in time $i \in [n]$, that is, $\max_{j\le i}\{j + t_j\} \ge i + 1$ for all $i \in [n]$.

This is an online problem because, at time i, we only know about the realizations $x_1, \ldots, x_i$ up to time i and have to base our decision about the hiring duration t_i of the ith applicant only on this information and previous hiring decisions $t_1, \ldots, t_{i-1}$. We are interested in obtaining online algorithms that perform well compared with an omniscient prophet. Let $\textsc{Opt}_n$ be the cost of an *optimal offline algorithm* (i.e., a prophet) knowing the n realizations in advance, and let $\textsc{Alg}_n$ be the cost of a solution of an online algorithm. Then the competitive ratio of the online algorithm $\textsc{Alg}_n$ is defined as $\limsup_{n\in\mathbb{N}} \mathbf{E}[\textsc{Alg}_n]/\mathbf{E}[\textsc{Opt}_n]$. We call an algorithm competitive if its competitive ratio is constant, and call it *strictly competitive* if even $\sup_{n\in\mathbb{N}} \mathbf{E}[\textsc{Alg}_n]/\mathbf{E}[\textsc{Opt}_n]$ is constant.

We use well-known facts from higher-order statistics of random variables to obtain the following.

Proposition 1. *The expected total cost of an optimal offline algorithm is* $\mathbf{E}[\textsc{Opt}_n] = \sum_{i\in[n]} \int_0^\infty (1 - F(x))^i \,\mathrm{d}x$.

Proof. In every step, the optimal offline algorithm employs the applicant with the lowest cost that has arrived so far. We have

$$\begin{aligned}
\mathbf{E}[\textsc{Opt}_n] &= \mathbf{E}\left[\sum_{i\in[n]} \min_{j\in[i]}\{x_j\}\right] \\
&= \sum_{i\in[n]} \mathbf{E}\left[\min_{j\in[i]}\{x_j\}\right] \\
&= \sum_{i\in[n]} \int_0^\infty \mathbf{Pr}\left[\min_{j\in[i]}\{x_j\} > x\right] \mathrm{d}x \\
&= \sum_{i\in[n]} \int_0^\infty (1 - F(x))^i \,\mathrm{d}x,
\end{aligned}$$

as claimed. □

Note that for every nontrivial probability distribution with nonzero probability mass around 0, the expected optimal offline cost of step k approaches 0 for $k \to \infty$. Consequently, any competitive online algorithm must also have a vanishing expected cost per step (where the hiring cost is distributed evenly over the hiring period).

3. An Optimal Online Algorithm

We begin by describing an optimal online algorithm that uses dynamic programming. Let $C(i, j)$ denote the expected overall cost if there are i time steps remaining and if the next j time steps are already covered by an existing contract. As a boundary condition, we have that $C(i,i) = 0$ for all i, because in this case, no further applicants need to be hired.

Suppose that $C(i', j')$ has already been computed for all $i' < i$ and all $j' \le i'$. First we describe how to compute $C(i,0)$. Suppose that we draw an applicant with cost x. Because there are no existing contracts, we must hire this applicant for at least one time step, and we will obviously hire this applicant for at most i time steps. If we hire the applicant for r time steps, our overall cost will be $rx + C(i-1, r-1)$. Thus, the optimal cost for an applicant costing x can be written as

$$\min_{1\le r\le i}\{rx + C(i-1, r-1)\}.$$

Therefore, we have

$$C(i,0) = \int_0^\infty \min_{1\le r\le i}\{rx + C(i-1,r-1)\} f(x)\,\mathrm{d}x. \tag{1}$$

Now we suppose that $C(i,j)$ has been computed for $j < i$ and describe how to compute $C(i, j+1)$. The analysis is similar to the above, but in this case we have the additional option to reject an applicant and wait one more time step. The cost of waiting one step is given by $C(i-1, j)$, so we get the following expression:

$$C(i,j+1) = \int_0^\infty \min\left\{C(i-1,j), \min_{j+1<r\le i}\{rx + C(i-1,r-1)\}\right\} f(x)\,\mathrm{d}x. \tag{2}$$

If $C(i,j)$ has been computed for all $i \le n$ and all $j \le i$, then there is a straightforward online algorithm that achieves expected cost $C(n,0)$. This algorithm simply waits for the cost x of each applicant to be revealed and then chooses the action that minimizes the expression in the above equations.

3.1. Analysis

The computational efficiency of this algorithm depends on the difficulty of evaluating the integrals in Equations (1) and (2). For the simple case where the cost distributions are uniform, the right-hand sides of both equations boil down to finding the piecewise minimum over at most n linear functions, which can easily be computed. For other distributions, the algorithm may be slower. It is worth noting that the algorithm cannot be applied in the case where the distribution is unknown. For the case of a known distribution, we conclude the following.

Theorem 1. *The dynamic program given by Equations* (1) *and* (2) *yields an optimal online algorithm.*

Before we move on, we describe some shortcomings of this algorithm that we seek to address in the remainder of this paper. The first issue of the algorithm is that, although it does provide an optimal competitive ratio, it is unclear how to analyze the algorithm, and in particular, we do not know what competitive ratio the algorithm guarantees. Second, the algorithm is very complicated to describe, as it uses at least n^2 different threshold values to decide the hiring duration of an applicant, and these threshold values are specifically tailored to the distribution in question. In the subsequent sections, we show that there exist algorithms with constant competitive ratios, and in doing so we prove that the competitive ratio of the optimal online algorithm is also constant. Third, the optimal online algorithm requires both the cost distribution and the total number of time periods to be known ahead of time. In contrast, in the following, we develop an online algorithm with a constant competitive ratio that still works even if neither information is known.

4. Uniformly Distributed Costs

In this section, we give two algorithms with constant competitive ratios in the case where applicants' costs are distributed uniformly. By shifting/rescaling, we may assume without loss of generality that $X = U[0,1]$, that is, that the costs are distributed uniformly in the unit interval. Using Proposition 1, we obtain the following expression for the expected cost of the offline optimum.

Lemma 1. $\mathbf{E}[\textsc{Opt}_n] = \mathcal{H}_{n+1} - 1$ *for all* $n \in \mathbb{N}$, *where* $\mathcal{H}_n$ *is the nth harmonic number.*

Proof. By Proposition 1, $\mathbf{E}[\textsc{Opt}_n] = \sum_{i\in[n]} \int_0^1 (1-x)^i\,\mathrm{d}x = \sum_{i\in[n]} \frac{1}{i+1} = \mathcal{H}_{n+1} - 1$. □.

Algorithm 1 (An 8.122-Competitive Algorithm for Uniformly Distributed Costs)

```
τ ← 1 ;                                          // threshold cost
t ← 1 ;                     // remaining time with current threshold
for i ← 1,...,n do
    t ← t − 1 ;
    if x_i ≤ τ then
        hire applicant i for 4/τ time steps ;
        if i + 4/τ > n then
            stop ;
        τ ← τ/2; t ← 1/τ
    else if t = 0 then
        τ ← 2τ; t ← 1/τ ;
```

4.1. A First Competitive Algorithm

We start with our first online algorithm for uniform distributions (see Algorithm 1). The main idea of the algorithm is that whenever we hire an applicant of cost x, we afterward seek an applicant of cost $x/2$. The expected time until such an applicant arrives is $2/x$.

If we set our hiring time equal to this expectation, we leave a considerable probability that we will not encounter any cheaper applicants before the hiring time runs out. Instead, we hire the applicant for $4/x$ steps and iteratively relax our hiring threshold after a certain time.

More precisely, assume $x = 1/2^j$ for some integer j. We then hire the applicant for time

$$\frac{4}{x} > \frac{4}{x} - 1 = \frac{2}{x} + \frac{1}{x} + \frac{1}{2x} + \frac{1}{4x} + \cdots + 1. \tag{3}$$

This way, if we do not find an applicant of cost at most $x/2$ during the next $2/x$ time steps, we continue seeking for an applicant with cost x for $1/x$ time steps, and so on. The geometric sum (3) just leaves enough time until we eventually seek for an applicant with cost at most 1, who is surely found.

To accommodate the fact that the costs of applicants are not powers of 2, in general, we maintain a threshold cost τ that is a power of 2 and reduce the threshold, whenever a new applicant is hired. See Algorithm 1 for a formal description. Finally, once an applicant is employed long enough to cover all remaining time steps, we stop. Importantly, this allows us to bound the lowest possible value of τ to be $2^{-\lceil \log n \rceil + 2}$.[2] If an applicant is hired below this threshold, the hiring time is $4/\tau \geq n$.

If $n \leq 4$, the algorithm hires only once for four time units and is, thus, 4-competitive. For the following arguments, assume that $n \geq 5$. During the course of the algorithm, the threshold cost τ can only take values of the form 2^{-j} for $j \in \{0, \ldots, k-1\}$, where $k = \lceil \log(n) \rceil - 1 \geq 2$. This allows us to describe the evolution of τ with a Markov chain M with $k+1$ states as follows. State k is the absorbing state that corresponds to the event that we *succeeded* in hiring an applicant at cost at most the threshold value $2^{-(k-1)} = 2^{-\lceil \log n \rceil + 2}$. Each other state $j \in \{0, \ldots, k-1\}$ corresponds to the event that the threshold value reaches $\tau = \tau_j := 2^{-j}$; see Figure 1(a). Each transition of the Markov chain from a state j to a state $j-1$ corresponds to the failure of finding an applicant below the threshold $\tau_j = 2^{-j}$ for $1/\tau_j = 2^j$ time steps, resulting in a doubling of the threshold cost. Each transition of the Markov chain from a state j to a state $j+1$ corresponds to the hiring of an applicant resulting in the reduction of the threshold cost. We can therefore use the expected total number of state transitions of the Markov chain when starting at state 0 to bound the number of hired applicants overall.

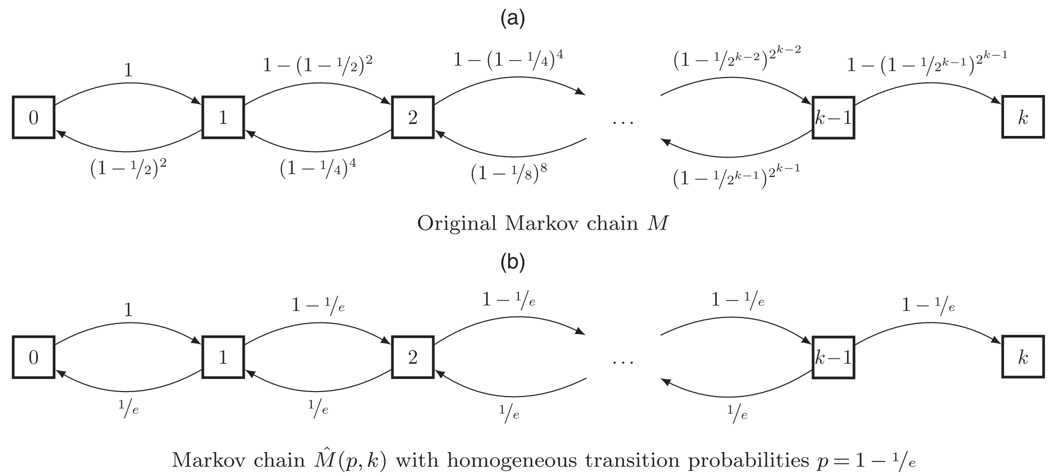
Let p_j denote the transition probability from state j to state $j+1$; that is, when in state j, the Markov chain transitions to state $j+1$ with probability p_j and to state $j-1$ with probability $1-p_j$. The probability that we fail to find an applicant with cost at most τ during $1/\tau$ time steps is bounded by

$$1 - p_j = (1-\tau)^{1/\tau} \leq \frac{1}{e},$$

that is, $p_j \geq 1 - \frac{1}{e}$. We set $p = 1 - \frac{1}{e}$ and consider the Markov chain $\hat{M}(p,k)$ with homogeneous transition probability p shown in Figure 1(b). As we will show in the following lemma, the total number of state transitions to reach state k in Markov chain $\hat{M}(p,k)$ provides an upper bound on the total number of state transitions to reach state k in Markov chain M. The analysis of $\hat{M}(p,k)$ then yields the following result.

Lemma 2. *For $n \geq 5$, starting in state 0 of Markov chain M with $k = \lceil \log(n) \rceil - 1$, the expected number of state transitions is at most $\frac{ek}{e-2}$.*

Figure 1. Markov chains M modeling the expected number of hired applicants of Algorithm 1. Nodes correspond to states. State $k = \lceil \log n \rceil - 1$ is absorbing.



Proof. Let $k = \lceil \log n \rceil - 1$ and $p = 1 - \frac{1}{e}$, and consider the Markov chain $\hat{M}(p,k)$ shown in Figure 1(b). We first claim that the expected number of state transitions when starting in state 0 in Markov chain M is bounded from above by that in Markov chain $\hat{M}(p,k)$. To see this, consider an arbitrary state j and consider the stochastic process that operates as M with the exception that the first time state j is visited, transition probabilities are as in $\hat{M}(p,k)$. Because $\hat{M}(p,k)$ has a higher probability to transition to a state with low index and the only absorbing state is k, this does not decrease the expected number of state transitions to state j in M. Iterating this argument, we derive that also the stochastic process where state j *always* transitions as in $\hat{M}(p,k)$ has a higher number of state transitions to state j. Iterating this argument over all states proves that the expected total number of state transitions in M is upper bounded by the expected total number of state transitions in $\hat{M}(p,k)$.

In Lemma 14 in Section 8.1, we show that the expected number of visits for each state in $\hat{M}(p,k)$ is bounded from above by $\frac{1}{2p-1} = \frac{e}{e-2}$. Because we start in state 0 and end after the first visit in state k, we conclude that the total number of state transitions of Markov chain $\hat{M}(p,k)$ is bounded by

$$\left(\frac{e}{e-2} - 1\right) + \sum_{i=1}^{k-1} \frac{e}{e-2} + 1 = \frac{ek}{e-2}.$$

This gives the claimed result. □

We proceed to use Lemmas 1 and 2 to obtain a first constant competitive algorithm for uniform costs.

Theorem 2. *Algorithm* 1 *is strictly* 8.122*-competitive for uniform distributions.*

Proof. If $n \in \{1, \dots, 4\}$, the algorithm hires the first applicant for n time units costing at most four times the expected cost of the optimum for the first times step; thus, the algorithm is 4-competitive. For the following arguments assume that $n \geq 5$. Because τ decreases whenever an applicant is hired, we can bound the number of hired applicants by the number of state transitions from a state j to state $j+1$ of the Markov chain. The algorithm terminates at the latest when state $k = \lceil \log(n) \rceil - 1$ is reached. If it ever reaches that point, it has hired at least k applicants, and every further hiring is mirrored by a state transition that decreases the current state. By using Lemma 2 and counting only the transitions that increase the state index, we can bound the expected number of hired applicants by

$$\frac{\frac{ek}{e-2} - k}{2} + k = \left(\frac{e}{e-2} + 1\right)\frac{k}{2} \leq \left(\frac{e}{e-2} + 1\right)\frac{\log n}{2}.$$

Whenever we hire an applicant below threshold τ, the cost of the applicant is uniform in $[0, \tau]$, so the expected cost is $\tau/2$. Because the hiring period is $4/\tau$, we get that each hired applicant incurs an expected total cost of 2.

The threshold τ for the next candidate is independent of the exact cost of the last hire. Therefore, we can combine the expected cost per candidate with Lemma 1, and we obtain

$$\frac{\mathbf{E}[\text{ALG}_n]}{\mathbf{E}[\text{OPT}_n]} \leq \frac{\ln n}{\mathcal{H}_{n+1} - 1} \cdot \frac{1}{\ln 2}\left(\frac{e}{e-2} + 1\right).$$

Using Lemma 3, proven below, where $\gamma \approx 0.577$ is the Euler–Mascheroni constant, this implies

$$\frac{\mathbf{E}[\text{ALG}_n]}{\mathbf{E}[\text{OPT}_n]} \leq \left(1 + \frac{20}{29}\left(\frac{5}{6} - \gamma\right)\right) \cdot \frac{1}{\ln 2}\left(\frac{e}{e-2} + 1\right) < 8.122,$$

as claimed. □

Lemma 3. *Let γ denote the Euler–Mascheroni constant. For any $n \in \mathbb{N}$, $\frac{\ln n}{\mathcal{H}_{n+1}-1} \leq 1 + \frac{20}{29}\left(\frac{5}{6} - \gamma\right)$.*

Proof. First, note that

$$\frac{\ln n}{\mathcal{H}_{n+1} - 1} \leq \frac{\mathcal{H}_n - \gamma}{\mathcal{H}_{n+1} - 1} = 1 + \frac{1 - \gamma - \frac{1}{n+1}}{\mathcal{H}_{n+1} - 1}.$$

It suffices to prove that

$$\sup_{n \in \mathbb{N}} \frac{1 - \gamma - \frac{1}{n+1}}{\mathcal{H}_{n+1} - 1} = \sup_{n \in \mathbb{N}, n \geq 2} \frac{1 - \gamma - \frac{1}{n}}{\mathcal{H}_n - 1} \leq \frac{20}{29}\left(\frac{5}{6} - \gamma\right).$$

To do so, we show that there is a unique $n' \in \mathbb{N}_{\geq 2}$ with

$$\frac{1 - \gamma - \frac{1}{n-1}}{\mathcal{H}_{n-1} - 1} \leq \frac{1 - \gamma - \frac{1}{n}}{\mathcal{H}_n - 1} \quad \text{for all } n \in \mathbb{N}_{\geq 2}, n \leq n', \text{ and}$$
$$\frac{1 - \gamma - \frac{1}{n}}{\mathcal{H}_n - 1} \geq \frac{1 - \gamma - \frac{1}{n+1}}{\mathcal{H}_{n+1} - 1} \quad \text{for all } n \in \mathbb{N}_{\geq 2}, n \geq n',$$

concluding that the supremum is attained at n'. Now we observe that

$$\frac{1 - \gamma - \frac{1}{n}}{\mathcal{H}_n - 1} - \frac{1 - \gamma - \frac{1}{n+1}}{\mathcal{H}_{n+1} - 1} \geq 0 \quad \Leftrightarrow \quad (\mathcal{H}_{n+1} - 1)\left(1 - \gamma - \frac{1}{n}\right) - (\mathcal{H}_n - 1)\left(1 - \gamma - \frac{1}{n+1}\right) \geq 0$$

and

$$\begin{aligned}
&(\mathcal{H}_{n+1} - 1)\left(1 - \gamma - \frac{1}{n}\right) - (\mathcal{H}_n - 1)\left(1 - \gamma - \frac{1}{n+1}\right)\\
&= \frac{1}{n+1}\left(1 - \gamma - \frac{1}{n}\right) - \frac{1}{n}(\mathcal{H}_n - 1) + \frac{1}{n+1}(\mathcal{H}_n - 1)\\
&= \frac{1}{n+1}\left(1 - \gamma - \frac{1}{n}\right) - (\mathcal{H}_n - 1)\frac{1}{n(n+1)} = \frac{1}{n(n+1)}(n(1-\gamma) - \mathcal{H}_n),
\end{aligned}$$

which is greater than or equal to 0 if and only if $n \geq 6$. We conclude that the supremum is attained at $n' = 6$. We finish the proof by observing

$$\frac{\ln n}{\mathcal{H}_{n+1} - 1} \leq 1 + \frac{1 - \gamma - \frac{1}{n+1}}{\mathcal{H}_{n+1} - 1} \leq 1 + \sup_{\tilde{n} \in \mathbb{N}_{\geq 2}} \frac{1 - \gamma - \frac{1}{\tilde{n}}}{\mathcal{H}_{\tilde{n}} - 1} \leq 1 + \frac{1 - \gamma - \frac{1}{6}}{\mathcal{H}_6 - 1} = 1 + \frac{20}{29}\left(\frac{5}{6} - \gamma\right),$$

for all $n \in \mathbb{N}$. □

4.2. Improving the Algorithm

We proceed to improve the competitive ratio of our algorithm as follows (see Algorithm 2). First, recall that in Algorithm 1, we hired an applicant below the current threshold of $\tau_j = 2^{-j}$ for $4/\tau_j$ time units with the rationale that

$$\sum_{i=0}^{j+1} \frac{1}{\tau_i} = \sum_{i=0}^{j+1} 2^i = 2^{j+2} - 1 = \frac{4}{\tau_j} - 1 < \frac{4}{\tau_j}.$$

With this inequality, it is ensured that we can afford $1/\tau_{j+1}$ time steps to look for an applicant below the threshold τ_{j+1} and, in case we do not find a suitable applicant, additional $1/\tau_j$ time steps to look for an applicant below the threshold τ_j, and so on, until the threshold is raised to $\tau_0 = 1$ and we find a suitable applicant with probability 1.

It turns out that it pays off to reduce the hiring time below threshold τ_j from $4/\tau_j$ to $3/\tau_j$. To compensate for that, we can only afford to look for an applicant below threshold τ_j for $\lceil \frac{3}{4\tau_j} \rceil$ time units. Note that for $\tau \in \{1/2, 1\}$, we round all times to the next integer. For $j \geq 3$, we then obtain

$$\sum_{i=0}^{j+1} \left\lceil \frac{3}{4\tau_j} \right\rceil = \left\lceil \frac{3}{4} \right\rceil + \left\lceil \frac{3}{2} \right\rceil + \sum_{i=2}^{j+1} \frac{3}{4\tau_j} = 1 + 2 + 3 \sum_{i=2}^{j+1} 2^{i-2} = 3 \cdot 2^j = \frac{3}{\tau_j}\ .$$

Similarly, we may check for $j = 0$ that $\lceil 3/4 \rceil + \lceil 3/2 \rceil = 3 = 3/\tau_0$, and for $j = 1$ that $\lceil 3/4 \rceil + \lceil 3/2 \rceil + 3 = 6 = 3/\tau_1$. Thus, we may conclude that the above choices ensure that an applicant is under contract at all times.

Second, instead of reducing the threshold once by factor 2 when we hire a new applicant, we repeatedly halve the threshold for as long as it is still greater or equal to the actual cost of the new applicant. This way, we can ensure that the cost for which a new applicant is hired is uniformly distributed in $[\tau, 2\tau)$ for all hirings except the last and is uniformly distributed in $[0, 2\tau)$ for the last hiring, where τ denotes the threshold *after* the applicant is hired. The applicant is below the threshold of 2τ; thus, it is hired for $\frac{3}{2\tau}$ time units, and the expected total cost of each applicant is $\frac{3\tau}{2} \cdot \frac{3}{2\tau} = \frac{9}{4}$.

Algorithm 2 (A 2.965-Competitive Algorithm for Uniformly Distributed Costs)

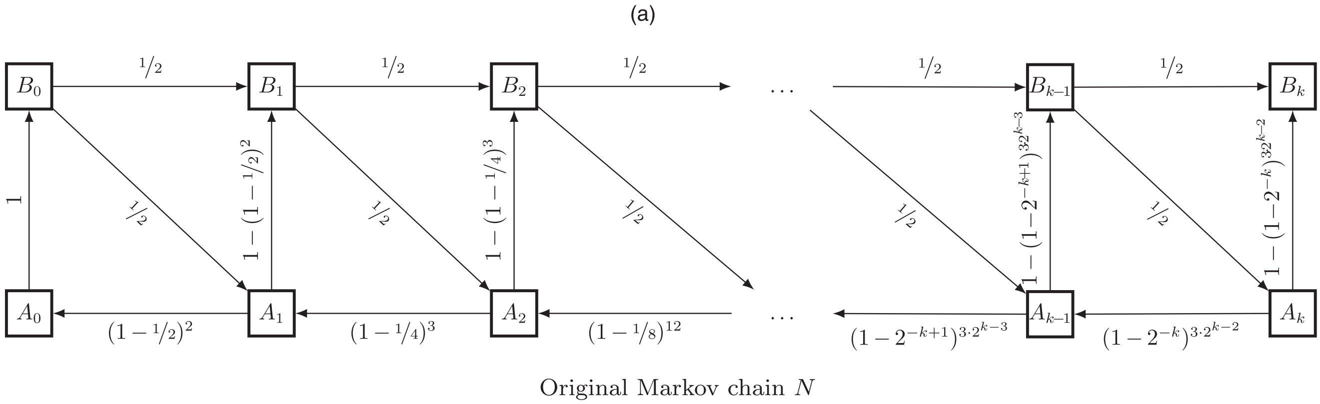
$\tau \leftarrow 1$; // threshold cost
$t \leftarrow 1$; // time with threshold
for $i \leftarrow 1, \ldots, n$ **do**
  $t \leftarrow t - 1$;
  if $x_i \leq \tau$ **then**
    while $x_i \leq \tau$ **do**
      $\tau \leftarrow \tau/2$;
    if $i + \frac{3}{2\tau} > n$ **then**
      hire applicant i until time n ;
      stop;
    hire applicant i for $\frac{3}{2\tau}$ time steps;
    $t \leftarrow \lceil \frac{3}{4\tau} \rceil$
  else if $t = 0$ **then**
    $\tau \leftarrow 2\tau$; $t \leftarrow \lceil \frac{3}{4\tau} \rceil$

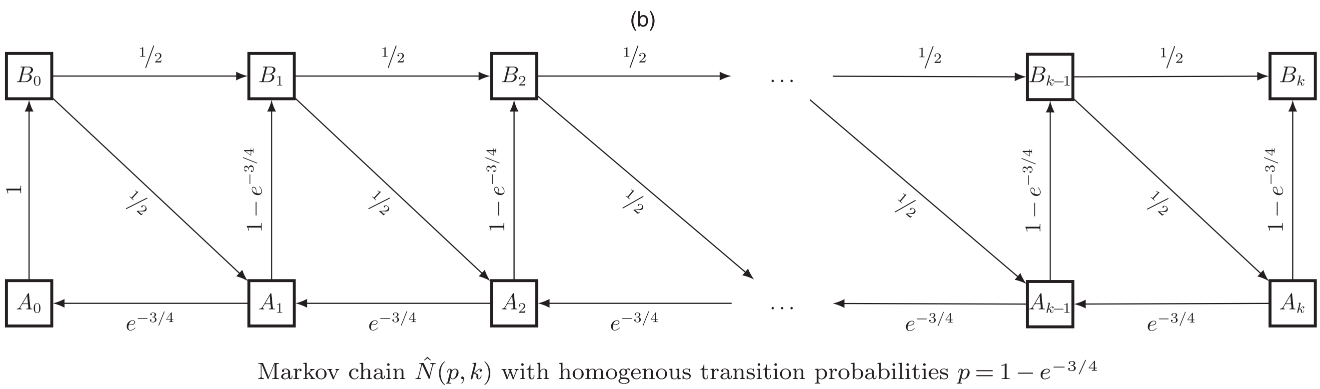
For $n \in \{1, 2\}$, the algorithm hires the first candidate for n time units and is, thus, 2-competitive. For the following arguments, assume that $n \geq 3$. Once we hire an applicant with a cost below 2^{-j}, the threshold τ after hiring is at most $2^{-(j+1)}$, so that the applicant is hired for at least $3 \cdot 2^j$ time steps. This implies that we need to account only for thresholds of the form $\tau_j = 2^{-j}$, where $j \in \{0, 1, \ldots, \lceil \log(n/3) \rceil\}$. We again capture the behavior of the algorithm with a Markov chain (see Figure 2). To this end, states $A_0, A_1, \ldots, A_k$ and $B_0, B_1, \ldots, B_k$ with $k = \lceil \log(n/3) \rceil$ are introduced. We distinguish between the states A_j that correspond to the algorithm looking for suitable applicants by comparing their costs with $\tau_j = 2^{-j}$ and the states B_j that correspond to the event that the cost of our current candidate is below the threshold $\tau_j = 2^{-j}$. Each state A_j with $j > 0$ either transitions to A_{j-1} with probability $(1 - p_j)$, when no applicant for the current threshold was found, or to B_j with probability p_j. As for the previous Markov chain, we have

$$(1 - p_j) = (1 - \tau_j)^{\lceil \frac{3}{4\tau_j} \rceil} \leq e^{-3/4}.$$

As in the previous section, we may consider the Markov chain with the homogenous transition probabilities $p = 1 - e^{-3/4}$ shown in Figure 2(b) instead, because we are interested only in upper bounding the number of hired applicants. Each state B_j with $j < k$ transitions to B_{j+1} or A_{j+1}, each with probability 1/2, because the cost x lies with equal probability in $[\tau, 2\tau)$ or $[0, \tau)$. State B_k is the only absorbing state of the Markov chain. Our analysis of the Markov chain in Section 8.2 yields the following result.

Figure 2. Markov chains modeling the expected number of hired applicants of Algorithm 2.





Lemma 4. *For $n \geq 3$, starting in state A_0 of Markov chain N, the expected number of transitions from an A state to a B state is at most*

$$h = \frac{kp}{3p-1} - \frac{4p(1-2p)}{(3p-1)^2} + \left(\frac{1-p}{3p-1}\right)^2 \left(\frac{2(1-p)}{1+p}\right)^k, \tag{4}$$

where $k = \lceil \log(n/3) \rceil$ and $p = 1 - e^{-3/4}$.

Proof. Let $k = \lceil \log(n/3) \rceil$, and let $p = 1 - e^{-3/4}$. We again argue that we overestimate the expected visiting times only when considering Markov chain $\hat{N}(p,k)$ instead of N. To see this, fix a state A_j, $j = 0, \ldots, k$, and consider the stochastic process N' that follows Markov chain N but, the first time state A_j is visited, transitions according to the probabilities of $\hat{N}(p,k)$. Because, in all stochastic processes we consider, the expected number of visits of each state is decreasing in the index of the starting state A_j, the expected number of visits of each state is not smaller in N' than in N. Iterating this argument, we conclude that the expected number of visits of all states in N does not exceed that in $\hat{N}(p,k)$.

In Lemma 15 in Section 8.2, we prove that the expected number of transitions from an A state to a B state of $\hat{N}(p,k)$ is bounded from above by (4). □

As every transition from an A state to a B state corresponds to the hiring of a candidate, bounding these transitions allows us to bound $\mathbf{E}[\textsc{Alg}_n]$. Together with the formula for $\mathbf{E}[\textsc{Opt}_n]$ proven in Lemma 1, we obtain an improved competitive ratio of 2.965.

Theorem 3. *Algorithm 2 is strictly 2.965-competitive for uniform distributions.*

Proof. For $n \in \{1, 2\}$, the algorithm hires the first applicant for n time steps and is, thus, 2-competitive. For the following arguments, assume that $n \geq 3$. Whenever an applicant is hired, the Markov chain transitions from A_j to B_j for some value $j \in [k]$. The algorithm terminates at the latest when state B_k is reached. We can thus bound the number of hired applicants by the expression h of Lemma 4. The last applicant hired when reaching the absorbing state B_k is hired for at most n time units and has costs uniformly distributed in $[0, 2^{-k})$ so that the expected cost is at most

$$n2^{-k-1} = n2^{-\lceil \log(n/3) \rceil - 1} \leq n2^{-\log(n/3)-1} = \frac{3}{2}.$$

Using Lemma 1 and the fact that the expected cost incurred by each hired applicant except the last is 9/4, we get

$$\frac{\mathbf{E}[\textsc{Alg}_n]}{\mathbf{E}[\textsc{Opt}_n]} \leq \frac{\frac{9}{4}(h-1)+\frac{3}{2}}{\mathcal{H}_{n+1}-1} \leq 2.965, \tag{5}$$

for all n. See Lemma 5 below for a proof of the last inequality. □

Lemma 5. *Let* $p = 1 - e^{-3/4}$, $k = \lceil \log(n/3) \rceil$, *and* $h = \frac{kp}{3p-1} - \frac{4p(1-2p)}{(3p-1)^2} + \left(\frac{1-p}{3p-1}\right)^2 \left(\frac{2(1-p)}{1+p}\right)^k$. *Then,* $\frac{\frac{9}{4}(h-1)+\frac{3}{2}}{\mathcal{H}_{n+1}-1} \leq 2.965$ *for all* n.

Proof. Because the expression $\frac{9}{4}(h-1)+\frac{3}{2}$ is constant as long as $k = \lceil \log(n/3) \rceil$ is constant, the ratio $\frac{\frac{9}{4}(h-1)+\frac{3}{2}}{\mathcal{H}_{n+1}-1}$ is maximized for some n of the form $n = 3 \cdot 2^{\ell-1} + 1$ with $\ell \in \mathbb{N}$. See also Figure 3, where the ratio is plotted as a function of n. The claim of the lemma is easily verified for $\ell = 1, \ldots, 6$. For $\ell \geq 7$, we obtain

$$\begin{aligned} h &= \frac{\ell p}{3p-1} - \frac{4p(1-2p)}{(3p-1)^2} + \left(\frac{1-p}{3p-1}\right)^2 \cdot \left(\frac{2(1-p)}{1+p}\right)^{\ell} \\ &\leq \frac{\ell p}{3p-1} - \frac{4p(1-2p)}{(3p-1)^2} + \left(\frac{1-p}{3p-1}\right)^2 \cdot \left(\frac{2(1-p)}{1+p}\right)^{7} \\ &< \frac{\ell p}{3p-1} + \frac{1}{e}, \end{aligned} \tag{6}$$

where for the first inequality we use that for $p = 1 - e^{-3/4}$, we have $\frac{2(1-p)}{1+p} = \frac{2}{2e^{3/4}-1} \approx 0.618 < 1$, and for the second inequality we evaluate (6) for $p = 1 - e^{-3/4}$. For the Euler–Mascheroni constant $\gamma \approx 0.577$, we obtain

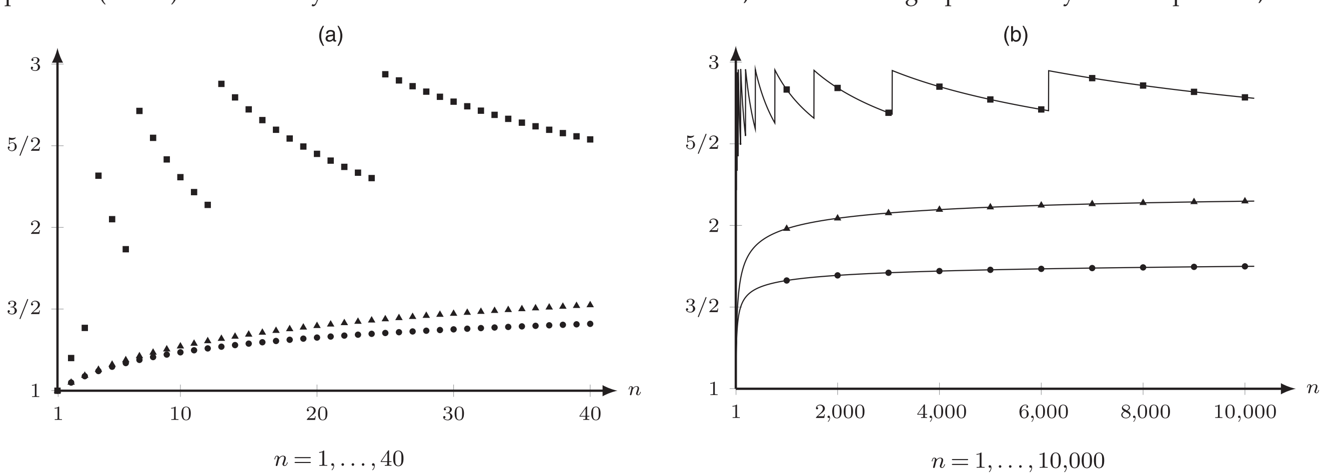
$$\begin{aligned} \frac{\frac{9}{4}(h-1)+\frac{3}{2}}{\mathcal{H}_{3\cdot 2^{\ell-1}+2}-1} &< \frac{\frac{9}{4}\left(\frac{\ell p}{3p-1}+\frac{1}{e}-1\right)+\frac{3}{2}}{\ln(3\cdot 2^{\ell-1}+2)-(1-\gamma)} \\ &\leq \frac{\frac{9}{4}\left(\frac{\ell p}{3p-1}+\frac{1}{e}-1\right)+\frac{3}{2}}{\ell \ln(2)+\ln(3)-\ln(2)-(1-\gamma)}, \end{aligned} \tag{7}$$

where we use that the denominator is positive. Using that $\ln(3) - \ln(2) - (1-\gamma) \approx -0.017 < 0$, elementary calculus shows that the expression in (7) is decreasing in ℓ. Evaluating it for $\ell = 7$, we obtain

$$\frac{\frac{9}{4}(h-1)+\frac{3}{2}}{\mathcal{H}_{3\cdot 2^{\ell-1}+2}-1} \leq 2.965,$$

as claimed. □

Figure 3. Upper bound on the competitive ratio of our algorithm (squares) from (5), the competitive ratio of the optimal online algorithm (triangle), and the lower bound on the competitive ratio of any online algorithm via the Gilbert and Mosteller [12] problem (circles) for uniformly distributed cost. For better visualization, marks in the right plot are only for multiples of 1,000.



4.3. Analytical Lower Bound

To obtain a lower bound on the competitive ratio of any online algorithm, we study in this section a relaxation of the problem. The relaxation allows us to exploit an interesting connection to the classical stopping problem with uniformly distributed random variables, which is known as the Cayley–Moser problem (see Gilbert and Mosteller [12], Moser [26]).

Theorem 4. *Asymptotically, for a uniform distribution, no online algorithm has a competitive ratio below* 2.

Proof. Consider the relaxation where we are allowed to hire an applicant for *any* (not necessarily contiguous) subset of all future time steps, while still having to decide on this set immediately upon arrival of the applicant. Formally, upon the arrival of applicant i with revealed cost $x_i \sim X$, we immediately decide to hire i for a set of time steps $T_i \subseteq \{k \in \mathbb{N} : k \geq i\}$ that realizes cost of $x_i\,|T_i|$. The objective is again to minimize the expected total cost

$$\mathbf{E}_{x_1\sim X,\dots,x_n\sim X}\Big[\sum\nolimits_{i\in[n]} x_i\,|T_i|\Big] \tag{8}$$

under the condition that $j \in \bigcup_{i\in[j]} T_i$ for all $j \in [n]$. In this setting, there is obviously no advantage to concurrent employment—once we hire an applicant for some time slot, there is no benefit of hiring additional applicants for the same time slot. Let $y_{ij} = 1$ if $j \in T_i$ and $y_{ij} = 0$ otherwise. Then we can rewrite (8) as

$$\mathbf{E}_{x_1\sim X,\dots,x_n\sim X}\Big[\sum\nolimits_{j\in[n]}\sum\nolimits_{i\in[j]} x_i\,y_{ij}\Big] = \sum\nolimits_{j\in[n]}\mathbf{E}_{x_1\sim X,\dots,x_j\sim X}\Big[\sum\nolimits_{i\in[j]} x_i\,y_{ij}\Big], \tag{9}$$

where we use linearity of expectation. The objective of the relaxation, then, is to minimize (9) under the condition that $\sum_{i\in[j]} y_{ij} = 1$ for all $j \in [n]$. Put differently, the decision of whether to hire an applicant for some time slot j is independent of the decision for other time slots. Thus, solving the relaxation reduces to simultaneously solving a stopping problem for each time slot j; that is, we need to hire exactly one of the first j applicants for this time slot, while applicants appear one by one and we need to irrevocably hire or discard each applicant upon their arrival.

Gilbert and Mosteller [12] showed that in the maximization version of the single stopping problem with uniformly distributed values, the optimal stopping rule is a threshold rule parametrized by t thresholds $\tau_0, \dots, \tau_{t-1}$. The rule stops at a time step when there are i remaining (unobserved) random variables and the realization is above τ_i. The threshold values follow the recursion $\tau_0 = 0$ and $\tau_{i+1} = (1+\tau_i^2)/2$ for all $i \geq 1$. Gilbert and Mosteller [12] showed that the value τ_t is also the expected revenue for the stopping problem with t slots when following the optimal strategy. They bounded the expected revenue for all t by

$$\tau_t \geq 1 - \frac{2}{t + \ln(t+1) + 1.767}.$$

By symmetry of the uniform distribution, for the corresponding single stopping problem with uniformly distributed costs and minimization objective, this immediately yields that the optimum expected cost $1-\tau_t$ is lower bounded by $h(t) := \frac{2}{t+\ln(t+1)+1.767}$.

Because we need to solve a stopping problem for each time slot $1, 2, \dots, n$, and by linearity of expectation, we get a lower bound on the expected cost of $\sum_{t=1}^{n} h(t)$ for the relaxed problem. On the other hand, by Lemma 1, for the offline optimum of our original problem, we have $\mathbf{E}_{x_1\sim U[0,1],\dots,x_n\sim U[0,1]}[\textsc{Opt}_n] = \sum_{t=1}^{n} g(t) := \sum_{t=1}^{n}\frac{1}{1+t}$. Because $h(t)$ and $g(t)$ are both monotonically decreasing, we can estimate $\sum_{t=1}^{n} h(t) \geq \int_1^{n+1} h(t)\,\mathrm{d}t$ and $\sum_{t=1}^{n} g(t) \leq \int_0^{n} g(t)\,\mathrm{d}t$. Also, because both integrals tend to infinity for growing n, we can apply l'Hôpital's rule and obtain

$$\lim_{n\to\infty}\frac{\sum_{t=1}^{n} h(t)}{\sum_{t=1}^{n} g(t)} \geq \lim_{n\to\infty}\frac{\int_1^{n+1} h(t)\,\mathrm{d}t}{\int_0^{n} g(t)\,\mathrm{d}t} = \lim_{n\to\infty}\frac{h(n+1)}{g(n)} = \lim_{n\to\infty}\frac{2(n+1)}{n+1+\log(n+2)+1.767} = 2.$$

As $\sum_{t=1}^{n} h(t)$ is the expected cost of an optimum online solution to the relaxed problem, it is a lower bound on the expected cost of an optimum online solution to the original problem, and we get the desired bound. □

A plot of the lower bound $\frac{\sum_{t=1}^{n} h(t)}{H_{n+1}-1}$ as a function of n, shown in Figure 3, reveals that the lower bound converges very slowly. Even for $n = 10{,}000$, the lower bound is still below 9/5.

One may wonder whether this relaxation gives rise to a lower bound strictly larger than 2 for distributions other than the uniform distribution. We proceed to show that this is not the case for a large class of distributions. Formally, we show that the optimal online algorithm for the relaxation where applicants can be hired for any subset of future time steps is asymptotically 2-competitive for any continuous distribution that is in the domain of attraction of an extreme value distribution and for which the optimum is unbounded as n grows. Note that these conditions are satisfied by the uniform distribution and many other distributions such as, for example, the exponential distribution.

Theorem 5. *Let X be a distribution that is in the domain of attraction of some extreme value distribution and such that $\lim_{n\to\infty} \mathbf{E}[\mathrm{O{\scriptsize PT}}_n] = \infty$. Then, for n large enough, there is a 2-competitive algorithm for the relaxation where applicants can be hired for any subset of future time steps.*

Proof. As argued in the proof of Theorem 4, the relaxation can be solved optimally by solving a separate stopping problem for each time step. Asymptotically, the worst-case competitive ratio of this optimal online algorithm is given by

$$\sup_{X\in\mathscr{X}} \lim_{n\to\infty} \left\{ \frac{\sum_{i=1}^n \tau_i}{\sum_{i=1}^n \mathbf{E}_{x_1,\ldots,x_i\sim X}[\min\{x_1,\ldots,x_i\}]} \right\}, \tag{10}$$

where $\mathscr{X}$ is the class of all continuous probability distributions on $\mathbb{R}_{\geq 0}$, and τ_i is the expected cost of the single stopping problem where i values are drawn from X when following the optimal stopping strategy. By the backward induction principle, the values τ_i follow the recursion $\tau_0 = \infty$ and $\tau_{i+1} = \mathbf{E}_{x\sim X}[\min\{x, \tau_i\}]$, and the optimal strategy is to stop at realization x when i draws are remaining and $x \leq \tau_i$.

To prove an upper bound on (10), we need bounds on the relative behavior of the threshold values τ_i and the expected minima of i independent draws. To utilize known results from the stopping literature, we first transform (10) into an equivalent maximization problem. To this end, let $\check{\mathscr{X}}$ be the class of all continuous probability distributions on $\mathbb{R}_{\leq 0}$. We then reformulate (10) as

$$\sup_{\check{X}\in\check{\mathscr{X}}} \lim_{n\to\infty} \left\{ \frac{\sum_{i=1}^n \check{\tau}_i}{\sum_{i=1}^n \mathbf{E}_{x_1,\ldots,x_i\sim \check{X}}[\max\{x_1,\ldots,x_i\}]} \right\}, \tag{11}$$

where $\check{\tau}_i$ are the optimal thresholds of the maximization version of the stopping problem with values drawn from $\check{X}$, that is, $\check{\tau}_0 = -\infty$ and $\check{\tau}_{i+1} = \mathbf{E}_{\check{x}\sim\check{X}}[\max\{\check{x}, \check{\tau}_i\}]$. The equivalence between (10) and (11) follows from the fact that a sequence $(X_1, n_1), (X_2, n_2), \ldots$ converges to the supremum in (10) if and only if the sequence $(-X_1, n_1), (-X_2, n_2), \ldots$ converges to the supremum in (11). Here, we denote by $-X$ the distribution of $-x$ where $x \sim X$.

In the following, we fix an arbitrary distribution $\check{X} \in \check{\mathscr{X}}$ satisfying the assumptions of the theorem and use the shorthand notation $\check{m}_i = \mathbf{E}_{x_1,\ldots,x_i\sim\check{X}}[\max\{x_1,\ldots,x_i\}]$. With this notation, the theorem can be proven by showing that $\lim_{n\to\infty} \sum_{i=1}^n \check{\tau}_i / \sum_{i=1}^n \check{m}_i \leq 2$. To this end, we recall a result from Kennedy and Kertz [21], who showed that the asymptotic behavior of the sequence $(\check{\tau}_n)_{n\in\mathbb{N}}$ depends on the domain of attraction of $\check{X}$ to the classical three classes of extreme value distributions. Specifically, Kennedy and Kertz [21] assumed that the underlying distribution $\check{X}$ lies in the domain of attraction of some limit distribution G, that is, there exist sequences of constants $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ with $a_n > 0$ such that $a_n(\max\{\check{x}_1, \ldots, \check{x}_n\} - b_n)$ converges in distribution to some nondegenerate function G as $n \longrightarrow \infty$. In this case, we say that $\check{X}$ is in the domain of attraction of G and write $\check{X} \in \mathscr{D}(G)$. The Fisher–Tippett–Gnedenko theorem (Fisher and Tippett [11], Gnedenko [13]) states that under these conditions, G must be one of the following types of extreme-value distributions:

$$\begin{aligned}
G_I(x) &= \exp(-e^{-x}) && \text{for } x \in \mathbb{R};\\
G_{II}^{\alpha}(x) &= \begin{cases} 0 \\ \exp(-x^{-\alpha}) \end{cases} && \begin{matrix} \text{if } x < 0, \\ \text{otherwise;} \end{matrix}\\
G_{III}^{\alpha}(x) &= \begin{cases} \exp(-(-x)^{\alpha}) \\ 1 \end{cases} && \begin{matrix} \text{if } x < 0, \\ \text{otherwise.} \end{matrix}
\end{aligned}$$

Necessary and sufficient conditions for $X \in \mathscr{D}(G)$ and $G \in \{G_I\} \cup \{G_{II}^{\alpha} : \alpha > 1\} \cup \{G_{III}^{\alpha} : \alpha > 0\}$ were given by Leadbetter et al. [25]. To state these conditions formally, we first introduce the notion of *regular variation* due to Karamata [18]. Formally, a function $f : \mathbb{R} \longrightarrow \mathbb{R}_{>0}$ is called *regularly varying* at infinity with index $\alpha \in \mathbb{R}_{\geq 0}$ if $\lim_{t\to\infty} \frac{f(tx)}{f(t)} = x^{\alpha}$ for all $x > 0$. It is further called regularly varying at $x_0 \in \mathbb{R}$ if the function $x \mapsto f(x_0 - \frac{1}{x})$ is regularly varying at infinity.

Let $\check{F}$ be the cumulative distribution of $\check{X}$, and let $\omega = \sup\{x : \check{F}(x) < 1\}$. The conditions given by Leadbetter et al. [25] imply that $\check{X} \in \mathscr{D}(G_{II}^{\alpha})$ if and only if $\omega = \infty$ and $1 - \check{F}(x)$ is regularly varying at infinity with index $-\alpha$. Furthermore, $\check{X} \in \mathscr{D}(G_{III}^{\alpha})$ if and only if $\omega < \infty$ and $1 - \check{F}(x)$ is regularly varying at ω with index $-\alpha$. The conditions for $\check{X} \in \mathscr{D}(G_I)$ are more involved to state and immaterial for our further analysis. Leadbetter et al. [25] further show that when the distributions lie in the domains of attraction of the respective extreme value distributions, the sequences of constants can be chosen as follows:

$$
\begin{aligned}
a_n &= \tfrac{1}{g(\gamma_n)} &&\text{and} && b_n = \gamma_n && \text{if } \check{X} \in \mathscr{D}(G_I),\\
a_n &= \tfrac{1}{\gamma_n} &&\text{and} && b_n = 0 && \text{if } \check{X} \in \mathscr{D}(G_{II}^{\alpha}),\\
a_n &= \tfrac{1}{\omega-\gamma_n} &&\text{and} && b_n = \omega && \text{if } \check{X} \in \mathscr{D}(G_{III}^{\alpha}),
\end{aligned}
$$

where $\gamma_n = \inf\{x : \check{F}(x) \geq 1 - \frac{1}{n}\}$ and $g(t) = \frac{\int_t^{\omega} 1-\check{F}(s)\,\mathrm{d}s}{1-\check{F}(t)}$.

We note that because we are interested only in distributions $\check{X}$ on $\mathbb{R}_{\leq 0}$, we need not consider the case $\check{X} \in \mathscr{D}(G_{II}^{\alpha})$ for the following arguments. Moreover, it is easy to see that among the distributions $\check{X} \in \mathscr{D}(G_{III}^{\alpha})$, the supremum in (11) is attained by a sequence with $\omega = 0$.

Kennedy and Kertz [21, theorem 1.2] established that

$$\lim_{n\to\infty} a_n(\check{\tau}_n - b_n) = \lim_{n\to\infty} \frac{\check{\tau}_n - \gamma_n}{g(\gamma_n)} = 0 \qquad \text{if } \check{X} \in \mathscr{D}(G_I), \tag{12a}$$

$$\lim_{n\to\infty} a_n(\check{\tau}_n - b_n) = \lim_{n\to\infty} \frac{\check{\tau}_n - \omega}{\omega - \gamma_n} = -\left(1 + \frac{1}{\alpha}\right)^{1/\alpha} \qquad \text{if } \check{X} \in \mathscr{D}(G_{III}^{\alpha}), \alpha > 0. \tag{12b}$$

Kennedy and Kertz [21] further stated that

$$\lim_{n\to\infty} a_n(\check{m}_n - b_n) = \lim_{n\to\infty} \frac{\check{m}_n - \gamma_n}{g(\gamma_n)} = \gamma \qquad \text{if } \check{X} \in \mathscr{D}(G_I), \tag{13a}$$

$$\lim_{n\to\infty} a_n(\check{m}_n - b_n) = \lim_{n\to\infty} \frac{\check{m}_n - \omega}{\omega - \gamma_n} = -\Gamma\left(1 + \frac{1}{\alpha}\right) \qquad \text{if } \check{X} \in \mathscr{D}(G_{III}^{\alpha}), \alpha > 0, \tag{13b}$$

where $\gamma \approx 0.5772$ is the Euler–Mascheroni constant and Γ denotes the gamma function. Combining (12a) with (13a) and (12b) with (13b), we obtain

$$\lim_{n\to\infty} \frac{\check{\tau}_n}{\check{m}_n} = \lim_{n\to\infty} \frac{\frac{\check{\tau}_n - \gamma_n}{g(\gamma_n)} + \frac{\gamma_n}{g(\gamma_n)}}{\frac{\check{m}_n - \gamma_n}{g(\gamma_n)} + \frac{\gamma_n}{g(\gamma_n)}} = 1 \qquad \text{if } \check{X} \in \mathscr{D}(G_I), \tag{14a}$$

$$\lim_{n\to\infty} \frac{\check{\tau}_n}{\check{m}_n} = \frac{(1 + \frac{1}{\alpha})^{1/\alpha}}{\Gamma(1 + \frac{1}{\alpha})} \qquad \text{if } \check{X} \in \mathscr{D}(G_{III}^{\alpha}), \alpha > 0, \omega = 0, \tag{14b}$$

where for (14a), we use that $g(\gamma_n) \to 0$ as $n \to \infty$. Results very similar to (14) are also reported by Kennedy and Kertz [21, theorem 1.3]. Their theorem statement, however, has a small inaccuracy regarding the case $\omega = 0$, which is the most interesting for us, so we chose to give the results again.

To finish the proof, suppose we have a distribution $\check{X} \in \bigcup_{\alpha>0} \mathscr{D}(G_{III}^{\alpha}) \cup \mathscr{D}(G_I)$ such that $\lim_{n\to\infty} \textsc{Opt}_n = \sum_{n=1}^{\infty} -\check{m}_n = \infty$. Then, (14a) and (14b) imply that $\sum_{n=1}^{\infty} \check{\tau}_n = -\infty$ as well. By the Stolz–Cesàro theorem (Ash et al. [2]), we then obtain

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} \check{\tau}_i}{\sum_{i=1}^{n} \check{m}_i} = \lim_{n\to\infty} \frac{\check{\tau}_n}{\check{m}_n} = \begin{cases} 1 & \text{if } \check{X} \in \mathscr{D}(G_I),\\ \frac{(1+\frac{1}{\alpha})^{1/\alpha}}{\Gamma(1+\frac{1}{\alpha})} & \text{if } \check{X} \in \mathscr{D}(G_{III}^{\alpha}). \end{cases}$$

So to obtain a lower bound, we are interested in solving

$$\sup_{\substack{\check{X} \in \mathscr{D}(G_{III}^{\alpha})\\ \omega = 0}} \left\{ \frac{(1 + \frac{1}{\alpha})^{1/\alpha}}{\Gamma(1 + \frac{1}{\alpha})} : \sum_{i=n}^{\infty} \check{m}_n = -\infty \right\}. \tag{15}$$

To solve (15), recall that for a distribution on $\mathbb{R}_{\leq 0}$ with $\omega = 0$, it holds that $\check{X} \in \mathcal{D}(G^{\alpha}_{III})$ if and only if $\omega < \infty$ and $1 - \check{F}(x)$ is regularly varying at 0, that is,

$$\lim_{t\to\infty} \frac{1 - \check{F}(0 - \frac{1}{tx})}{1 - \check{F}(0 - \frac{1}{t})} = x^{-\alpha} \tag{16}$$

for all $x > 0$. Because $\lim_{n\to\infty} \gamma_n = 0$ and $\gamma_n < 0$, we can replace $t = -1/\gamma_n$ and $x = -\gamma_n$. Because for a regular varying function the convergence in (16) is uniform in x, we know that for every $\epsilon > 0$ there is $N_0 \in \mathbb{N}$ such that for all $n \geq N_0$ and some $\epsilon_n \in (-\epsilon, \epsilon)$, we have

$$\epsilon_n = \frac{1 - \check{F}(-1)}{1 - \check{F}(\gamma_n)} - (-\gamma_n)^{-\alpha} = n\Big(1 - \check{F}(-1)\Big) - (-\gamma_n)^{-\alpha},$$

where we use that $\check{F}$ is continuous, and thus $F(\gamma_n) = 1 - 1/n$. We obtain

$$\gamma_n = -\Big(n\Big(1 - \check{F}(-1)\Big) - \epsilon_n\Big)^{-1/\alpha}.$$

As a consequence, $\sum_{n=1}^{\infty} \gamma_n = -\infty$ if and only if $\alpha \geq 1$. Next recall that

$$\lim_{n\to\infty} \frac{\check{m}_n}{\gamma_n} = \Gamma\bigg(1 + \frac{1}{\alpha}\bigg),$$

implying that $\sum_{n=1}^{\infty} \check{m}_n = -\infty$ if and only if $\sum_{n=1}^{\infty} \check{\gamma}_n = -\infty$. We obtain

$$\sup_{\substack{\check{X}\in\mathcal{D}(G^{\alpha}_{III})\\ \omega=0}} \left\{ \frac{(1+\frac{1}{\alpha})^{1/\alpha}}{\Gamma(1+\frac{1}{\alpha})} : \sum_{i=n}^{\infty} \check{m}_n = -\infty \right\} = \sup_{\alpha>0} \left\{ \frac{(1+\frac{1}{\alpha})^{1/\alpha}}{\Gamma(1+\frac{1}{\alpha})} : \alpha \geq 1 \right\} = \frac{2}{\Gamma(2)} = 2,$$

which yields the claimed result. □

Note that the bound of 2 on the competitiveness of the optimal online algorithm obtained in Theorem 5 is tight, as we showed in Theorem 4 that, for the uniform distribution, the optimal online algorithm is not better than 2-competitive.

4.4. Computational Lower Bound

In this section, we give a computational lower bound based on an optimal online algorithm for uniformly distributed costs. This gives a slightly higher lower bound than the analytical bound from Section 4.3. We implemented the optimal online algorithm presented in Section 3 in exact arithmetic, using rounding to prevent numbers from getting too large. The algorithm achieves a competitive ratio of 2.148 for an instance with 10,000 time steps (see Figure 3). We describe the details on the computational lower bound in the following paragraphs.

For the uniform distribution, we know from Lemma 1 that the optimal offline algorithm has expected cost $\mathcal{H}_{n+1} - 1$. On the other hand, the entry $C(n,0)$ in the dynamic programming table of the optimal online algorithm gives the optimal cost for an instance with n days. Therefore, for every $n > 0$, the ratio $\frac{C(n,0)}{\mathcal{H}_{n+1}-1}$ provides a lower bound on the best strict competitive ratio achievable by any online algorithm.

We implemented the optimal online algorithm for the uniform case and computed the expression above for increasing values of n. To obtain a conclusive proof, one needs to implement the algorithm in exact rational arithmetic. However, in doing so, we found that the sizes of the numerators and denominators grow very quickly in n, and already for $n = 22$ both the numerator and the denominator have over a million digits. This makes it computationally intractable to compute $\frac{C(n,0)}{\mathcal{H}_{n+1}-1}$ for large n.

To address this, we adopted a rounding scheme: after computing $C(i,j)$ for some i and j, we rounded the number down to another rational with a smaller numerator and denominator, and then stored the rounded number in the dynamic programming table. Because we only ever round down, the resulting costs computed by the algorithm must always be cheaper than the expected cost of the optimal online algorithm. Therefore, the computed value of $\frac{C(n,0)}{\mathcal{H}_{n+1}-1}$ is still a lower bound on the strict competitive ratio that can be achieved.

Ultimately, we found that for $n = 10{,}000$, the competitive ratio can be no better than 2.148. The following remark summarizes the results.

Remark 1. For a uniform distribution, no online algorithm has a strict competitive ratio below 2.148.

5. Arbitrary Distributions

Algorithm 3 (A 6.052-Competitive Algorithm for Arbitrary Distributions)

```
q ← 1 ;                                              // threshold quantile index
τ ← δ_q ;                                            // threshold cost
t ← 1 ;                                              // time with threshold
for i ← 1,...,n do
    t ← t − 1 ;
    if x_i ≤ τ then
        while x_i ≤ τ do
            q ← q/2; τ ← δ_q ;
        if i + 2/q > n then
            hire applicant i until time n ;
            stop ;
        hire applicant i for 2/q time steps ;
        t ← 1/q ;
    else if t = 0 then
        q ← 2q; τ ← δ_q; t ← 1/q ;
```

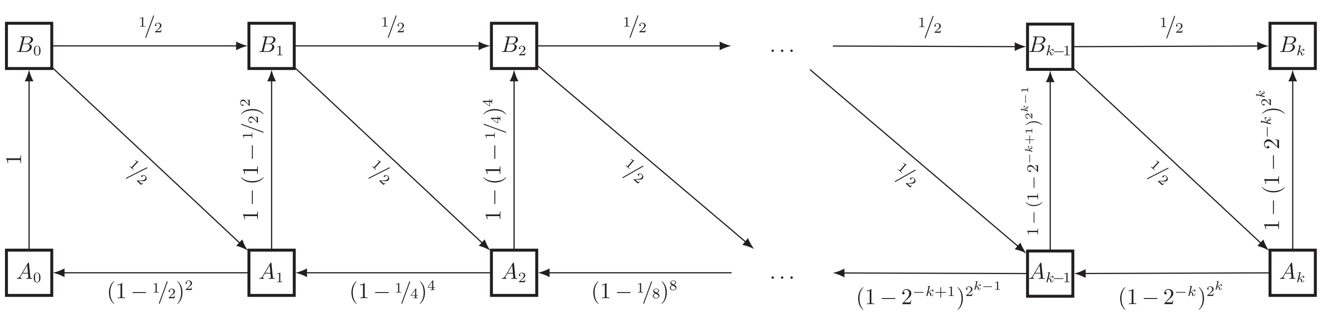
In this section, we propose an algorithm similar to Algorithm 2 for an arbitrary distribution X (see Algorithm 3). Whenever we halve our threshold in the course of Algorithm 2, we essentially halve the probability mass of X below the threshold (i.e., the probability that a drawn value lies below τ). To achieve the same effect with respect to an arbitrary distribution X, we consider *quantiles* δ_q of X, defined by the property that $\mathbf{Pr}\left[x \le \delta_q\right] = q$ for continuous distributions.[3] Algorithm 3 changes the threshold by halving and doubling q and using $\tau = \delta_q$, which results in the same behavior as Algorithm 2 when X is uniform. Therefore, we can, in principle, analyze the algorithm for general distributions using a Markov chain similar to that in Section 4.2. Specifically, the Markov chain again governs the evolution of the value $q = 2^{-j}$, $j = 0, 1, 2, \ldots$, and the corresponding threshold value $\tau = \delta_q$ in the course of Algorithm 3. After finding an applicant with a cost x_i below the threshold value τ, the value of q is halved until $\delta_q < x_i \le \delta_{q-1}$. Because the applicant is then hired for $2/q$ time steps, where q is the value after the halving, we conclude that when hiring an applicant below the threshold of δ_q, it is hired for $4/q$ time units. As the process stops at the latest when $4/q \ge n$, the Markov chain N' has states $A_0, A_1, \ldots, A_k$ and $B_0, B_1, \ldots, B_k$ with $k = \lceil \log n \rceil - 2$ (see Figure 4).

Again, we start in state A_0, B_k is the absorbing state, and states A_j, B_j correspond to states of the algorithm where $\tau = \delta_{2^{-j}}$. The transition probability from any state A_j to state B_j is bounded from below by $p = 1 - 1/e$, because the probability of not finding any applicant of cost at most δ_q within $1/q$ steps is

$$1 - \mathbf{Pr}[x > \delta_q]^{1/q} = 1 - (1 - \mathbf{Pr}[x \le \delta_q])^{1/q} \ge 1 - (1-q)^{1/q} \ge 1 - \frac{1}{e}.$$

The analysis of the algorithm for arbitrary distributions, however, turns out to be more intricate than for the uniform case, for two main reasons. First, for uniform distributions, it was sufficient to count the total number of transitions from an A state to a B state, as any such transition corresponds to the hiring of a candidate with a total cost of 9/4. On the other hand, for general distributions, we need to bound the number of transitions from

Figure 4. Markov chain N' modeling the expected number of hired applicants of Algorithm 3.



state B_j to state A_{j+1} for each j individually, as the resulting costs may differ among the different values of j. The following lemma provides a bound independent of j.

Lemma 6. *Starting in state A_0 of Markov chain N', for each $j \in \{0, \dots, k-1\}$, the expected number of transitions from B_j to A_{j+1} is at most $\frac{p}{3p-1}$, where $p = 1 - \frac{1}{e}$.*

Proof. With the same arguments as in the proof of Lemma 4, we obtain an upper bound on the expected number of transitions from B_j to A_{j+1} by considering the Markov chain $\hat{N}(p,k)$ with homogenous transition probability $p = 1 - \frac{1}{e}$ and $k = \lceil \log n \rceil - 2$. For the latter Markov chain, Lemma 16 proven in Section 8.2 establishes the result. □

The second main issue when analyzing the competitive ratio of the algorithm is the lack of a concrete value for $\textsc{Opt}_n$ for general distributions. Thus, we need the following lemma that expresses $\mathbf{E}[\textsc{Opt}_n]$ as a sum over conditional expectations of the form $\mathbf{E}[x \mid \delta_{2^{-(r+1)}} < x \le \delta_{2^{-r}}]$ with $r \in \mathbb{N}$.

Lemma 7. *Let $n \ge 5$, $k = \lceil \log n \rceil - 2$, and $\eta := \frac{5}{2} - \frac{55}{6e^2} \approx 1.259$. Then, we have*

$$\mathbf{E}[\textsc{Opt}_n] \ge \sum_{r=0}^{k-1} 2^{r-1} \mathbf{E}[x \mid \delta_{2^{-(r+1)}} < x \le \delta_{2^{-r}}] + \eta 2^{k-1} \mathbf{E}[x \mid x \le \delta_{2^{-k}}].$$

Proof. By linearity of expectation, $\mathbf{E}[\textsc{Opt}_n] = \sum_{i \in [n]} \mathbf{E}[\min\{x_1, \dots, x_i\}]$, where, for $i \in [n]$, the random variables $x_1, \dots, x_i$ are drawn independently from X. To prove the claim, we proceed to express, for fixed $i \in [n]$, the expectation $\mathbf{E}[\min\{x_1, \dots, x_i\}]$ in terms of $\mathbf{E}[x \mid x \le \delta_{2^{-k}}]$ and $\mathbf{E}[x \mid \delta_{2^{-(r+1)}} < x \le \delta_{2^{-r}}]$ with $r \in \{0, \dots, k-1\}$. To this end, for $i \in [n]$ and $r \in \{0, \dots, k-1\}$, let

$$\mathscr{E}_{i,r,=1} = \Big[\big|\{x_1, \dots, x_i\} \cap (\delta_{2^{-(r+1)}}, \delta_{2^{-r}}]\big| = 1 \text{ and } \big|\{x_1, \dots, x_i\} \cap (\delta_{2^{-(r+1)}}, \delta_1]\big| = i\Big]$$

be the stochastic event that the minimum of the i draws $x_1 \sim X, \dots, x_i \sim X$ is in the interval $(\delta_{2^{-(r+1)}}, \delta_{2^{-r}}]$ and none of the other $i-1$ draws is in that interval. Additionally, let

$$\mathscr{E}_{i,k,=1} = [|\{x_1, \dots, x_i\} \cap [0, \delta_{2^{-k}}]| = 1].$$

Furthermore, for $r \in \{0, \dots, k-1\}$, let

$$\mathscr{E}_{i,r,>1} = [|\{x_1, \dots, x_i\} \cap (\delta_{2^{-(r+1)}}, \delta_{2^{-r}}]| > 1 \text{ and } |\{x_1, \dots, x_i\} \cap (\delta_{2^{-(r+1)}}, \delta_1]| = i]$$

be the stochastic event that the minimum of the i draws is in the interval $(\delta_{2^{-(r+1)}}, \delta_{2^{-r}}]$ and at least one of the other $i-1$ draws is in that interval. Similarly, let

$$\mathscr{E}_{i,k,>1} = [|\{x_1, \dots, x_i\} \cap [0, \delta_{2^{-k}}]| > 1].$$

For fixed i, the events $\mathscr{E}_{i,r,=1}$ and $\mathscr{E}_{i,r,>1}$ for $r \in \{0, \dots, k\}$ are clearly disjoint. Because $\sum_{r=0}^{k} (\mathbf{Pr}[\mathscr{E}_{i,r,=1}] + \mathbf{Pr}[\mathscr{E}_{i,r,>1}]) = 1$, by the law of total expectation, we have

$$\begin{aligned}\mathbf{E}[\min\{x_1, \dots, x_i\}] = \sum_{r=0}^{k} \Big(&\mathbf{E}\big[\min\{x_1, \dots, x_i\} \mid \mathscr{E}_{i,r,=1}\big] \mathbf{Pr}\big[\mathscr{E}_{i,r,=1}\big] \\ &+ \mathbf{E}\big[\min\{x_1, \dots, x_i\} \mid \mathscr{E}_{i,r,>1}\big] \mathbf{Pr}\big[\mathscr{E}_{i,r,>1}\big]\Big).\end{aligned}$$

We observe that $\mathbf{E}\big[\min\{x_1, \dots, x_i\} \mid \mathscr{E}_{i,r,=1}\big] = \mathbf{E}[x \mid \delta_{2^{-(r+1)}} < x \le \delta_{2^{-r}}]$ for all $r \in \{0, \dots, k-1\}$, and, similarly, $\mathbf{E}[\min\{x_1 \dots, x_i\} \mid \mathscr{E}_{i,k,=1}] = \mathbf{E}[x \mid x \le \delta_{2^{-k}}]$. In addition, we have $\mathbf{E}\big[\min\{x_1, \dots, x_i\} \mid \mathscr{E}_{i,r,>1}\big] \ge \delta_{2^{-(r+1)}} \ge \mathbf{E}[x \mid \delta_{2^{-(r+2)}} < x \le \delta_{2^{-(r+1)}}]$ for all $r \in \{0, \dots, k-1\}$. We then obtain

$$\begin{aligned}\mathbf{E}[\min\{x_1, \dots, x_i\}] \ge\ & \mathbf{E}[x \mid \delta_{2^{-1}} < x \le \delta_1] \mathbf{Pr}\big[\mathscr{E}_{i,0,=1}\big] \\ &+ \sum_{r=1}^{k-1} \mathbf{E}[x \mid \delta_{2^{-(r+1)}} < x \le \delta_{2^{-r}}] \Big(\mathbf{Pr}\big[\mathscr{E}_{i,r,=1}\big] + \mathbf{Pr}\big[\mathscr{E}_{i,r-1,>1}\big]\Big) \\ &+ \mathbf{E}[x \mid x \le \delta_{2^{-k}}] \Big(\mathbf{Pr}\big[\mathscr{E}_{i,k,=1}\big] + \mathbf{Pr}\big[\mathscr{E}_{i,k-1,>1}\big]\Big),\end{aligned}$$

and hence

$$\begin{aligned}\mathbf{E}[\mathrm{O}\text{PT}_n] \geq{}& \mathbf{E}[x \mid \delta_{2^{-1}} < x \leq \delta_1] \sum_{i=1}^{n} \mathbf{Pr}\big[\mathscr{E}_{i,0,=1}\big] \\ &+ \sum_{r=1}^{k-1}\left(\mathbf{E}[x \mid \delta_{2^{-(r+1)}} < x \leq \delta_{2^{-r}}] \sum_{i=1}^{n}\Big(\mathbf{Pr}\big[\mathscr{E}_{i,r,=1}\big] + \mathbf{Pr}\big[\mathscr{E}_{i,r-1,>1}\big]\Big)\right) \\ &+ \mathbf{E}[x \mid x \leq \delta_{2^{-k}}] \sum_{i=1}^{n}\Big(\mathbf{Pr}\big[\mathscr{E}_{i,k,=1}\big] + \mathbf{Pr}\big[\mathscr{E}_{i,k-1,>1}\big]\Big).\end{aligned}$$

The probability that a single draw falls in the range $(\delta_{2^{-(r+1)}}, \delta_{2^{-r}}]$ and $i-1$ draws are larger than $\delta_{2^{-r}}$ is $2^{-(r+1)}(1-2^{-r})^{i-1}$. Because there are i possibilities for which of the draws falls in this range, we have

$$\mathbf{Pr}\big[\mathscr{E}_{i,r,=1}\big] = \begin{cases} 0 & \text{if } r = 0 \text{ and } i > 1, \\ 1/2 & \text{if } r = 0 \text{ and } i = 1, \\ i2^{-(r+1)}(1-2^{-r})^{i-1} & \text{if } r \in \{1, \ldots, k-1\}, \\ i2^{-r}(1-2^{-r})^{i-1} & \text{if } r = k. \end{cases}$$

Similarly, for $r \in \{1, \ldots, k\}$, we have

$$\begin{aligned}\mathbf{Pr}\big[\mathscr{E}_{i,r-1,>1}\big] &= (1-2^{-r})^i - (1-2^{-(r-1)})^i - \mathbf{Pr}\big[\mathscr{E}_{i,r-1,=1}\big] \\ &= (1-2^{-r})^i - (1-2^{-(r-1)})^i - i2^{-r}(1-2^{-(r-1)})^{i-1}.\end{aligned}$$

We then obtain

$$\begin{aligned}\mathbf{E}[\mathrm{O}\text{PT}_n] \geq{}& \mathbf{E}[x \mid \delta_{2^{-1}} < x \leq \delta_1] \cdot 2^{-1} \\ &+ \sum_{r=1}^{k-1} \mathbf{E}[x \mid \delta_{2^{-(r+1)}} < x \leq \delta_{2^{-r}}] \cdot \alpha(r,n) \\ &+ \mathbf{E}[x \mid x \leq \delta_{2^{-k}}] \cdot \alpha(k,n)\end{aligned}$$

with

$$\alpha(r,n) = \sum_{i=1}^{n}\Big(\mathbf{Pr}\big[\mathscr{E}_{i,r,=1}\big] + \mathbf{Pr}\big[\mathscr{E}_{i,r-1,>1}\big]\Big). \tag{17}$$

It remains to show that $\alpha(r,n) \geq 2^{r-1}$ for $1 \leq r < k$ and $\alpha(k,n) \geq \eta 2^{k-1}$. We have

$$\begin{aligned}\alpha(r,n) &= \sum_{i=1}^{n}\Big(\mathbf{Pr}\big[\mathscr{E}_{i,r,=1}\big] + \mathbf{Pr}\big[\mathscr{E}_{i,r-1,>1}\big]\Big) \\ &= \begin{cases} \sum_{i=1}^{n}\big(i2^{-r}(1-2^{-r})^{i-1} + (1-2^{-r})^i - (1-2^{-(r-1)})^i - i2^{-r}(1-2^{-(r-1)})^{i-1}\big) & \text{if } r = k, \\ \sum_{i=1}^{n}\big(i2^{-(r+1)}(1-2^{-r})^{i-1} + (1-2^{-r})^i - (1-2^{-(r-1)})^i - i2^{-r}(1-2^{-(r-1)})^{i-1}\big) & \text{otherwise.} \end{cases}\end{aligned} \tag{18}$$

To prove the lemma, we proceed to show that $\inf_{n\in\mathbb{N}} \inf_{r\in\{0,\ldots,\lceil\log n\rceil-3\}} \frac{\alpha(r,n)}{2^{r-1}} \geq 1$ and $\inf_{n\in\mathbb{N}} \frac{\alpha(\lceil\log n\rceil-2,n)}{2^{\lceil\log n\rceil-3}} \geq \eta$. Differentiating the well-known formula for the geometric sum $\sum_{i=1}^{n} a^i = \frac{a-a^{n+1}}{1-a}$, we obtain $\sum_{i=1}^{n} ia^{i-1} = \frac{na^{n+1}-(n+1)a^n+1}{(1-a)^2}$. We use both formulas to simplify all partial sums. For a binary event $\mathscr{E}$, we denote by $\chi_{\mathscr{E}}$ the indicator variable for event $\mathscr{E}$; that is, $\chi_{\mathscr{E}} = 1$ if $\mathscr{E}$ is true, and $\chi_{\mathscr{E}} = 0$ otherwise. For $r \in \{1, \ldots, k\}$, we then obtain

$$\begin{aligned}\alpha(r,n) ={}& (1+\chi_{r=k})2^{r-1}\big[n(1-2^{-r})^{n+1} - (n+1)(1-2^{-r})^n + 1\big] + 2^r\big[1-2^{-r} - (1-2^{-r})^{n+1}\big] \\ &- 2^{r-1}\Big[1-2^{-(r-1)} - (1-2^{-(r-1)})^{n+1}\Big] - 2^{r-2}\Big[n(1-2^{-(r-1)})^{n+1} - (n+1)(1-2^{-(r-1)})^n + 1\Big] \\ ={}& 2^{r-1}\Big[(1+\chi_{r=k})\big(n(1-2^{-r})^{n+1} - (n+1)(1-2^{-r})^n + 1\big) + 2 - 2^{-(r-1)} - 2(1-2^{-r})^{n+1} \\ &-1 + 2^{-(r-1)} + (1-2^{-(r-1)})^{n+1} - \frac{n}{2}(1-2^{-(r-1)})^{n+1} + \frac{n+1}{2}(1-2^{-(r-1)})^n - \frac{1}{2}\Big]\end{aligned}$$

$$
\begin{aligned}
&= 2^{r-1}\Bigg[\frac{3}{2} + \chi_{r=k} + (1-2^{-r})^n(n(1+\chi_{r=k})(1-2^{-r}) - (n+1)(1+\chi_{r=k}) - 2(1-2^{-r}))\\
&\quad + (1-2^{-(r-1)})^n\left((1-2^{-(r-1)}) - \frac{n}{2}(1-2^{-(r-1)}) + \frac{n+1}{2}\right)\Bigg]\\
&= 2^{r-1}\Bigg[\frac{3}{2} + \chi_{r=k} + (1-2^{-r})^n(2^{-(r-1)} - n2^{-r}(1+\chi_{r=k}) - 3 - \chi_{r=k})\\
&\quad + (1-2^{-(r-1)})^n\left(\frac{3}{2} + n2^{-r} - 2^{-(r-1)}\right)\Bigg]\\
&= 2^{r-1}\Bigg[\frac{3}{2} + \chi_{r=k} + \left((1-2^{-r})^n - (1-2^{-(r-1)})^n\right)\left(2^{-(r-1)} - n2^{-r} - \frac{3}{2}\right)\\
&\quad - \left(\frac{3}{2} + \chi_{r=k}(1+n2^{-r})\right)(1-2^{-r})^n\Bigg].
\end{aligned}
$$

As the probabilities are nonnegative, $\alpha(r,n) = \sum_{i=1}^{n}(\mathbf{Pr}[\mathscr{E}_{i,r,=1}] + \mathbf{Pr}[\mathscr{E}_{i,r-1,>1}])$ is nondecreasing in n for all $r \in \{1,\ldots,k\}$. We proceed to show that $\alpha(r,n) \geq 2^{r-1}$ for all $r \in \{0,\ldots,k-1\} = \{0,\ldots,\lceil \log n\rceil - 3\}$. Because r and n are integral, $r \leq \lceil \log n \rceil - 3$ implies $n \geq 2^{r+2}+1$. Using the monotonicity of $\alpha(r,n)$ and substituting $t := 2^r$, we have

$$
\begin{aligned}
&\inf_{n\in\mathbb{N}}\ \inf_{r\in\{1,\ldots,\lceil\log n\rceil-3\}} \frac{\alpha(r,n)}{2^{r-1}}\\
&= \inf_{r\in\mathbb{N}}\ \inf_{n\in\{2^{r+2}+1,\ldots\}} \frac{\alpha(r,n)}{2^{r-1}}\\
&= \inf_{r\in\mathbb{N}} \frac{\alpha(r,2^{r+2}+1)}{2^{r-1}}\\
&= \inf_{r\in\mathbb{N}}\Bigg\{\frac{3}{2} + \left((1-2^{-r})^{2^{r+2}+1} - (1-2^{-(r-1)})^{2^{r+2}+1}\right)\left(2^{-(r-1)} - (2^{r+2}+1)2^{-r} - \frac{3}{2}\right)\\
&\qquad -\frac{3}{2}(1-2^{-r})^{2^{r+2}+1}\Bigg\}\\
&\geq \inf_{t\in\mathbb{N}}\left\{\frac{3}{2} + \left(\left(1-\frac{1}{t}\right)^{4t+1} - \left(1-\frac{2}{t}\right)^{4t+1}\right)\left(\frac{2}{t} - 4 - \frac{1}{t} - \frac{3}{2}\right) - \frac{3}{2}\left(1-\frac{1}{t}\right)^{4t+1}\right\}\\
&= \inf_{t\in\mathbb{N}}\left\{\frac{3}{2} + \left(\left(1-\frac{1}{t}\right)^{4t+1} - \left(1-\frac{2}{t}\right)^{4t+1}\right)\left(\frac{1}{t} - \frac{11}{2}\right) - \frac{3}{2}\left(1-\frac{1}{t}\right)^{4t+1}\right\}.
\end{aligned}
$$

The first-order Taylor approximation of the function $f(x) = x^{4t+1}$ at $x = 1 - {}^{1}\!/_{t}$ gives $f(1-{}^{2}\!/_{t}) = (1-{}^{2}\!/_{t})^{4t+1} = (1-{}^{1}\!/_{t})^{4t+1} - \frac{4t+1}{t}(1-{}^{1}\!/_{t})^{4t} + R_2$, with $R_2 \geq 0$ as f is convex. This implies

$$
\begin{aligned}
&\inf_{n\in\mathbb{N}}\ \inf_{r\in\{1,\ldots,\lceil\log n\rceil-2\}} \frac{\alpha(r,n)}{2^{r-1}}\\
&\geq \inf_{t\in\mathbb{N}}\left\{\frac{3}{2} - \frac{4t+1}{t}\left(\frac{11}{2} - \frac{1}{t}\right)\left(1-\frac{1}{t}\right)^{4t} - \frac{3}{2}\left(1-\frac{1}{t}\right)\left(1-\frac{1}{t}\right)^{4t}\right\}\\
&= \inf_{t\in\mathbb{N}}\left\{\frac{3}{2} - \left(\frac{47}{2} - \frac{1}{t^2}\right)\left(1-\frac{1}{t}\right)^{4t}\right\}\\
&\geq \inf_{t\in\mathbb{N}}\left\{\frac{3}{2} - \frac{47}{2}\left(1-\frac{1}{t}\right)^{4t}\right\}.
\end{aligned}
$$

As the latter expression is decreasing in t, we have

$$
\inf_{n\in\mathbb{N}}\ \inf_{r\in\{1,\ldots,\lceil\log n\rceil-3\}} \frac{\alpha(r,n)}{2^{r-1}} \geq \lim_{t\to\infty}\left\{\frac{3}{2} - \frac{47}{2}(1-\frac{1}{t})^{4t}\right\} = \frac{3}{2} - \frac{47}{2e^4} \approx 1.069 > 1.
$$

It remains to show that $\frac{\alpha(k,n)}{2^{k-1}} \geq \eta$. For $r = k = \lceil \log n \rceil - 2$, we have

$$\alpha(k,n) = 2^{k-1}\left[\frac{5}{2} + \left((1-2^{-k})^n - (1-2^{-(k-1)})^n\right)\left(2^{-(k-1)} - n2^{-k} - \frac{3}{2}\right) - \left(\frac{5}{2} + n2^{-k}\right)(1-2^{-k})^n\right].$$

Again, as $\alpha(r,n)$ is nondecreasing in n, this value is minimal for $n = 2^{k+1}+1$. Substituting $t = 2^k$, we obtain

$$\begin{aligned}
&\inf_{n\in\mathbb{N}} \frac{\alpha(\lceil \log n\rceil - 2, n)}{2^{\lceil \log n\rceil - 3}}\\
&= \inf_{k\in\mathbb{N}} \frac{\alpha(k, 2^{k+1}+1)}{2^{k-1}}\\
&= \inf_{k\in\mathbb{N}} \left\{\frac{5}{2} + \left((1-2^{-k})^{2^{k+1}+1} - (1-2^{-(k-1)})^{2^{k+1}+1}\right)\left(2^{-(k-1)} - (2^{k+1}+1)2^{-k} - \frac{3}{2}\right) - \left(\frac{5}{2} + (2^{k+1}+1)2^{-k}\right)(1-2^{-k})^{2^{k+1}+1}\right\}\\
&\geq \inf_{\substack{t\in\mathbb{N}\\ t\geq 2}} \left\{\frac{5}{2} + \left(\left(1-\frac{1}{t}\right)^{2t+1} - \left(1-\frac{2}{t}\right)^{2t+1}\right)\left(\frac{2}{t} - 2 - \frac{1}{t} - \frac{3}{2}\right) - \left(\frac{5}{2} + 2 + \frac{1}{t}\right)\left(1-\frac{1}{t}\right)^{2t+1}\right\}\\
&= \inf_{\substack{t\in\mathbb{N}\\ t\geq 2}} \left\{\frac{5}{2} + \left(\left(1-\frac{1}{t}\right)^{2t+1} - \left(1-\frac{2}{t}\right)^{2t+1}\right)\left(\frac{1}{t} - \frac{7}{2}\right) - \left(\frac{9}{2} + \frac{1}{t}\right)\left(1-\frac{1}{t}\right)^{2t+1}\right\}.
\end{aligned}$$

By second-order Taylor approximation of the function $f(x) = x^{2t+1}$ at $x = (1 - 1/t)$, we obtain

$$\begin{aligned}
f(1-\tfrac{2}{t}) = \left(1-\frac{2}{t}\right)^{2t+1} &= \left(1-\frac{1}{t}\right)^{2t+1} - \frac{2t+1}{t}\left(1-\frac{1}{t}\right)^{2t} + \frac{2t(2t+1)}{2t^2}\left(1-\frac{1}{t}\right)^{2t-1}\\
&\quad - \frac{2t(2t+1)(2t-1)}{6t^3}\left(1-\frac{1}{t}\right)^{2t-2} + R_4,
\end{aligned}$$

where the remainder is $R_4 \geq 0$, as the fourth derivative is nonnegative. (This can easily be seen when expressing the remainder in Lagrange form.) We then obtain

$$\begin{aligned}
&\inf_{n\in\mathbb{N}} \frac{\alpha(\lceil \log n\rceil - 2, n)}{2^{\lceil \log n\rceil - 3}}\\
&\geq \inf_{\substack{t\in\mathbb{N}\\ t\geq 2}} \left\{\frac{5}{2} + \left(\frac{2t+1}{t}\left(1-\frac{1}{t}\right)^{2t} - \frac{2t+1}{t}\left(1-\frac{1}{t}\right)^{2t-1} + \frac{(2t+1)(2t-1)}{3t^2}\left(1-\frac{1}{t}\right)^{2t-2}\right)\left(\frac{1}{t} - \frac{7}{2}\right) - \left(\frac{9}{2} + \frac{1}{t}\right)\left(1-\frac{1}{t}\right)^{2t+1}\right\}\\
&= \inf_{\substack{t\in\mathbb{N}\\ t\geq 2}} \left\{\frac{5}{2} + \left(1-\frac{1}{t}\right)^{2t}\left(\left(\frac{2t+1}{t} - \frac{2t+1}{t-1} + \frac{(2t+1)(2t-1)}{3(t-1)^2}\right)\left(\frac{1}{t} - \frac{7}{2}\right) - \left(\frac{9}{2} + \frac{1}{t}\right)\left(1-\frac{1}{t}\right)\right)\right\}\\
&= \inf_{\substack{t\in\mathbb{N}\\ t\geq 2}} \left\{\frac{5}{2} - \frac{55t^4 - 125t^3 + 89t^2 + 8t - 12}{6t^2(t-1)^2}\left(1-\frac{1}{t}\right)^{2t}\right\}.
\end{aligned}$$

It is straightforward to check that $(1 - 1/t)^{2t}$ and $\frac{55t^4-125t^3+89t^2+8t-12}{6t^2(t-1)^2}$ are increasing in t. This implies

$$\begin{aligned}
\inf_{k\in\mathbb{N}} \frac{\alpha(k, 2^{k+1}+1)}{2^{k-1}} &= \lim_{t\to\infty} \left\{\frac{5}{2} - \frac{55t^4 - 125t^3 + 89t^2 + 8t - 12}{6t^2(t-1)^2}\left(1-\frac{1}{t}\right)^{2t}\right\}\\
&= \frac{5}{2} - \frac{55}{6e^2} \approx 1.259,
\end{aligned}$$

which finishes the proof. □

Combining Lemmas 6 and 7, we obtain the main result of this section.

Theorem 6. *Algorithm* 3 *is* 6.052*-competitive for arbitrary distributions.*

Proof. For $n \in \{1, \ldots, 4\}$, Algorithm 3 hires the first applicant for n time units and is, thus, 4-competitive. For the following arguments, assume that $n \geq 5$, and let $k = \lceil \log n \rceil - 2 \geq 1$.

Algorithm 3 hires an applicant whenever the Markov chain transitions from a state B_j to A_{j+1}, and hires the final applicant when it reaches state B_k. By Lemma 6 for each j, the expected number of transitions from state B_j to A_{j+1} is at most $\frac{p}{3p-1}$, where $p = 1 - \frac{1}{e}$. Each applicant who is hired while transitioning from B_j to A_{j+1} is hired for 2^{j+2} time units, and its expected cost value is $\mathbf{E}[x \mid \delta_{2^{-(j+1)}} \le x \le \delta_{2^{-j}}]$. The final applicant hired when state B_k is reached is hired for at most n time units and has expected cost $\mathbf{E}[x \mid x \le \delta_{2^{-k}}]$.

Because the number of visits to a state and the cost for hiring an applicant in the state are stochastically independent, we obtain

$$\begin{aligned}
\mathbf{E}[\mathrm{ALG}_n] &\le \frac{p}{3p-1}\sum_{j=0}^{k-1}\left(2^{j+2}\mathbf{E}[x \mid \delta_{2^{-(j+1)}} < x \le \delta_{2^{-j}}]\right) + n\mathbf{E}[x \mid x \le \delta_{2^{-k}}] \\
&= \frac{1-{}^1/_e}{2-{}^3/_e}\sum_{j=0}^{k-1}\left(2^{j+2}\mathbf{E}[x \mid \delta_{2^{-(j+1)}} < x \le \delta_{2^{-j}}]\right) + n\mathbf{E}[x \mid x \le \delta_{2^{-k}}] \\
&\le \frac{8e-8}{2e-3}\mathbf{E}[\mathrm{OPT}_n] + n\left(1 - \frac{e-1}{2e-3}\eta\right)\mathbf{E}[x \mid x \le \delta_{2^{-k}}], \qquad (19)
\end{aligned}$$

where we use Lemma 7 and where $\eta = \frac{5}{2} - \frac{55}{6e^2}$. Furthermore, recall that $\mathbf{E}[\mathrm{OPT}_n] = \sum_{i=1}^{n}\mathbf{E}[\min\{x_1, \ldots, x_i\}]$. For $i \in [n]$, we have

$$\begin{aligned}
\mathbf{E}[\min\{x_1,\ldots,x_i\}] &\ge \mathbf{E}[x \mid x \le \delta_{2^{-k}}]\mathbf{Pr}[|\{x_1,\ldots,x_i\} \cap [0,\delta_{2^{-k}}]| \le 1] \\
&\ge \mathbf{E}[x \mid x \le \delta_{2^{-k}}]\mathbf{Pr}\left[\left|\{x_1,\ldots,x_i\} \cap [0,\delta_{4/n}]\right| \le 1\right] \\
&= \mathbf{E}[x \mid x \le \delta_{2^{-k}}]\Big(\mathbf{Pr}\left[\left|\{x_1,\ldots,x_i\} \cap [0,\delta_{4/n}]\right| = 0\right] \\
&\qquad + \mathbf{Pr}\left[\left|\{x_1,\ldots,x_i\} \cap [0,\delta_{4/n}]\right| = 1\right]\Big) \\
&= \mathbf{E}[x \mid x \le \delta_{2^{-k}}]\left(\left(1-\frac{4}{n}\right)^i + \frac{4i}{n}\left(1-\frac{4}{n}\right)^{i-1}\right),
\end{aligned}$$

which implies (for $n \ge 5$)

$$\begin{aligned}
\mathbf{E}[\mathrm{OPT}_n] &\ge \mathbf{E}[x \mid x \le \delta_{2^{-k}}]\sum_{i=1}^{n}\left(\left(1-\frac{4}{n}\right)^i + \frac{4i}{n}\left(1-\frac{4}{n}\right)^{i-1}\right) \\
&= \mathbf{E}[x \mid x \le \delta_{2^{-k}}]\left(\frac{n}{2}\left(1 - 3\left(1-\frac{4}{n}\right)^n\right) + \left(1-\frac{4}{n}\right)^n - 1\right) \\
&\ge \mathbf{E}[x \mid x \le \delta_{2^{-k}}]\left(\frac{n}{2}\left(1 - \frac{3}{e^4}\right) + \left(1-\frac{4}{n}\right)^n - 1\right). \qquad (20)
\end{aligned}$$

Combining (20) with (19) and using $n \ge 5$, we obtain

$$\begin{aligned}
\mathbf{E}[\mathrm{ALG}_n] &\le \frac{8e-8}{2e-3}\mathbf{E}[\mathrm{OPT}_n] + \frac{1 - \frac{e-1}{2e-3}\left(\frac{5}{2} - \frac{55}{6e^2}\right)}{\frac{1}{2}\left(1-\frac{3}{e^4}\right) + \left(1-\frac{4}{n}\right)^n - \frac{1}{n}}\mathbf{E}[\mathrm{OPT}_n] \\
&\le \left(\frac{8e-8}{2e-3} + \frac{1 - \frac{e-1}{2e-3}\left(\frac{5}{2} - \frac{55}{6e^2}\right)}{\frac{1}{2}\left(1-\frac{3}{e^4}\right) + \left(1-\frac{4}{5}\right)^5 - \frac{1}{5}}\right)\mathbf{E}[\mathrm{OPT}_n] \le 6.052 \cdot \mathbf{E}[\mathrm{OPT}_n]
\end{aligned}$$

as claimed. □

6. Unknown Distributions

In this section, we again consider an arbitrary distribution X with distribution function F. In contrast to before, we assume that X is unknown to us. In particular, we do not have access to the quantiles of X. We first give a bound for the cost of the offline optimum that does not rely on quantiles. In the following, we let $\mathbf{E}[x] := \mathbf{E}_{x\sim X}[x]$.

Lemma 8. *For arbitrary distributions X, $\mathbf{E}[\mathrm{OPT}_n] \ge \mathbf{E}[x] + \sum_{i=1}^{\lfloor \log n \rfloor} 2^{i-1}\int_0^\infty (1-F(x))^{2^i}\,dx$.*

Proof. Because the left-hand side of the inequality to prove is increasing in n, whereas the right-hand side increases only when n is a power of 2, we may assume without loss of generality that n is a power of 2. By Proposition 1, we have $\mathbf{E}[\textsc{Opt}_n] = \sum_{i\in[n]} \int_0^\infty (1-F(x))^i \, \mathrm{d}x$. Using that $(1-F(x))^i$ is decreasing with i, we split the sum into the ranges $(n/2, n], (n/4, n/2], (n/8, n/4], \ldots$ and bound each part by the last term in the corresponding range, that is,

$$\begin{aligned}
\mathbf{E}[\textsc{Opt}_n] &= \mathbf{E}[x] + \sum_{i=2}^{n} \int_0^\infty (1-F(x))^i \, \mathrm{d}x \\
&\geq \mathbf{E}[x] + \frac{n}{2}\int_0^\infty (1-F(x))^n \, \mathrm{d}x + \frac{n}{4}\int_0^\infty (1-F(x))^{n/2} \, \mathrm{d}x + \cdots + \int_0^\infty (1-F(x))^2 \, \mathrm{d}x \\
&= \mathbf{E}[x] + \sum_{i=1}^{\log n} \frac{n}{2^i} \int_0^\infty (1-F(x))^{n/2^{i-1}} \, \mathrm{d}x \\
&= \mathbf{E}[x] + \sum_{i=1}^{\log n} 2^{\log n - i} \int_0^\infty (1-F(x))^{2^{\log n - i+1}} \, \mathrm{d}x \\
&= \mathbf{E}[x] + \sum_{i=0}^{\log n - 1} 2^{i} \int_0^\infty (1-F(x))^{2^{i+1}} \, \mathrm{d}x \\
&= \mathbf{E}[x] + \sum_{i=1}^{\log n} 2^{i-1} \int_0^\infty (1-F(x))^{2^{i}} \, \mathrm{d}x,
\end{aligned}$$

as claimed. □

We now describe our algorithm for unknown distributions (see Algorithm 4). Without knowledge of the quantiles of X, we have no good way to directly adjust the cost threshold τ. Instead, for some integral value $\lambda > 1$ to be fixed later, we devote a $\frac{1}{\lambda+1}$ fraction of the time spent in each state j to sample X to estimate a suitable value for τ and then wait for an appropriate candidate to appear. Specifically, in state j, we sample for $2^j - 1$ time units and then observe the applicants for another $\lambda(2^j - 1)$ time units. Thus, the maximum number of time units spent in state j is $\bar{t}_j = (1+\lambda)(2^j - 1)$. When observing the applicants, we hire any candidate whose cost does not exceed the minimum cost while sampling. The hiring time is $t_j = (1+\lambda)2^{j+2}$ time units. Because

$$\sum_{i=0}^{j+1} \bar{t}_i = (1+\lambda)\sum_{i=0}^{j+1}(2^i - 1) = (1+\lambda)(2^{j+2} - j - 3) \leq t_j,$$

we are guaranteed to hire a new applicant (or terminate the algorithm) during the hiring time.

Algorithm 4 (A 48-Competitive Algorithm for Unknown Distributions)

```
τ ← ∞ ;                                              // threshold cost
t_sample ← 0 ;                     // remaining time until threshold is fixed
t_wait ← 1 ;                        // remaining time once threshold is fixed
j ← 0 ;                                           // state of the algorithm
for i ← 1, ..., n do
    if t_sample > 0 then
        τ ← min{τ, x_i} ;
        t_sample ← t_sample − 1 ;
    else if t_wait > 0 then
        t_wait ← t_wait − 1 ;
        if x_i ≤ τ then
            if i + (1 + λ)2^{j+2} > n then
                hire applicant i until time n ;
                stop;
            hire applicant i for (1 + λ)2^{j+2} time steps ;
            j ← j + 1; τ ← ∞; t_sample ← 2^j − 1; t_wait ← λ t_sample ;
    else
        j ← j − 1; τ ← ∞; t_sample ← 2^j − 1; t_wait ← λ t_sample ;
```

The maximum value of j that can be reached during the execution of the algorithm is bounded by the fact that $(1+\lambda)2^{j+2} \leq n$, that is, $j \leq \lceil \log \frac{n}{1+\lambda} \rceil - 2$.

Again, we introduce a Markov chain that has one state for each possible value of j and an absorbing state k; see Figure 5. The probability that we do not hire an applicant in state j with $0 < j < k$ equals the probability that the smallest cost observed while sampling is lower than the smallest cost observed while waiting. Because $t_{\text{wait}} = \lambda t_{\text{sample}}$, we have a hiring probability of $p = \frac{\lambda}{\lambda+1}$. With this probability, the Markov chain transitions to state $j+1$, and otherwise to state $j-1$.

As the Markov chain already has homogenous transition probabilities equal to $p = \frac{\lambda}{\lambda+1}$, Lemma 14 directly implies the following result.

Lemma 9. *The expected number of visits to each state j of the Markov chain is at most $\frac{1}{2p-1}$.*

Combining Lemma 8 and Lemma 9 yields the main result of this section.

Theorem 7. *For $\lambda = 3$, Algorithm 4 is strictly 48-competitive for unknown distributions.*

Proof. For $n \in \{1, \dots, 4\}$, the first applicant is hired for n time units, and the algorithm is, thus, 4-competitive. For the following arguments, assume that $n \geq 5$. Using Lemma 9 with $p = \frac{\lambda}{\lambda+1}$, we conclude that the algorithm visits each state at most $\frac{1}{2p-1} = \frac{\lambda+1}{\lambda-1}$ times in expectation. In state j with $0 < j < k$, with probability $p = \frac{\lambda}{\lambda+1}$, an applicant is hired for $(1+\lambda)2^{j+2}$ units of time. The cost of the applicant is determined by drawing $2^j - 1$ numbers to determine a minimum τ, and then continuing to draw until we find the first cost smaller than τ. We can bound the expected cost of the applicant by the expected cost when drawing 2^j numbers and taking the minimum, that is,

$$\mathbf{E}[x \mid x \leq \tau] \leq \mathbf{E}_{x_i \sim X}\left[\min_{i \in \{1,\dots,2^j\}} \{x_i\}\right] = \int_0^\infty (1 - F(x))^{2^j}\, \mathrm{d}x.$$

The algorithm stops at the latest when an applicant is hired in state $k - 1 = \lceil \log \frac{n}{\lambda+1} \rceil - 2$, as the applicant is hired for at least n time steps. Because the number of visits to a state, the probability of hiring in a state, and the expected cost when hiring are independent, we obtain

$$\mathbf{E}[\textsc{Alg}_n] \leq \frac{\lambda(\lambda+1)}{\lambda-1} \sum_{j=0}^{k} \left(2^{j+2} \int_0^\infty (1 - F(x))^{2^j}\, \mathrm{d}x\right).$$

Together with Lemma 8 and $k - 1 \leq \lfloor \log n \rfloor - 2$, this yields

$$\begin{aligned}
\frac{\mathbf{E}[\textsc{Alg}_n]}{\mathbf{E}[\textsc{Opt}_n]} &\leq \frac{4\mathbf{E}[x]\frac{(\lambda+1)^2}{\lambda-1} + \sum_{j=1}^{k-1}\left(\frac{\lambda(\lambda+1)}{\lambda-1} 2^{j+2} \int_0^\infty (1-F(x))^{2^j}\, \mathrm{d}x\right)}{\mathbf{E}[x] + \sum_{j=1}^{\lfloor \log n \rfloor} 2^{j-1} \int_0^\infty (1-F(x))^{2^j}\, \mathrm{d}x} \\
&\leq \max\left\{\frac{4\mathbf{E}[x]\frac{(\lambda+1)^2}{\lambda-1}}{\mathbf{E}[x]}, \frac{\sum_{j=1}^{k-1}\left(\frac{\lambda(\lambda+1)}{\lambda-1} 2^{j+2} \int_0^\infty (1-F(x))^{2^j}\, \mathrm{d}x\right)}{\sum_{j=1}^{\lfloor \log n \rfloor} 2^{j-1} \int_0^\infty (1-F(x))^{2^j}\, \mathrm{d}x}\right\} \\
&\leq \max\left\{4\frac{(\lambda+1)^2}{\lambda-1}, 8\frac{\lambda(\lambda+1)}{\lambda-1}\right\} \leq 48,
\end{aligned}$$

as claimed. □

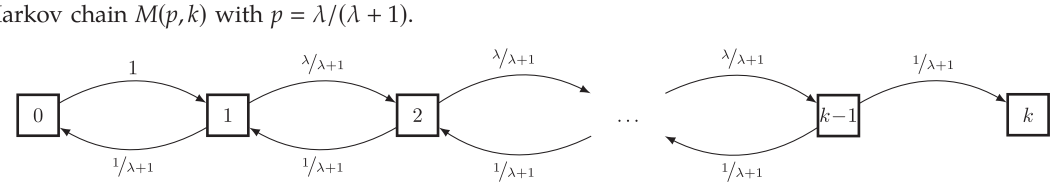
7. Sequential Employment

We now turn our attention to the number of applicants that are concurrently under employment. We show that there is no constant competitive algorithm for the problem that the covering constraint for the required number of employed candidates is fulfilled with equality in every step.

We can easily adapt the algorithms in the previous sections to be competitive in a setting where not more than two applicants may be employed during any period of time.

Lemma 10. *Algorithms 1–4 can be adapted to employ not more than two applicants concurrently while loosing at most a factor of 2 in their competitive ratio.*

Figure 5. Markov chain $M(p,k)$ with $p = \lambda/(\lambda+1)$.



Proof. We double the hiring times of the algorithms and stay idle during the first half of the hiring period; that is, we discard all applicants encountered during that period. This doubling causes a loss of a factor not larger than 2. Furthermore, it has the effect that after waiting for half of the hiring time, effectively, the remaining hiring time is as before. This, in turn, implies that the employment period of any previously hired applicant runs out while staying idle for a new applicant. This is because the hiring time of a new applicant was defined to be larger than the remaining hiring time of the previous one, and thus only ever two applicants are employed concurrently. □

Lemma 10 allows us to generalize our algorithms for input sequences of unknown length. Without knowledge of n, we cannot stop our algorithm once an applicant is hired for more than the remaining time, and, in addition, the last applicant may be hired for time steps after time step n. However, if no more than two applicants are employed concurrently, it is guaranteed that we never employ more than a single additional applicant. It can be shown that the expected cost of hiring the last two applicants hired by our algorithms is never larger than a constant times the cost of an optimal solution. (This is trivial for the uniform distribution where the cost for hiring an applicant is bounded by 9/4; for arbitrary and unknown distributions, it follows from Lemmas 7 and 8, respectively.) Because, up to the hiring of the last two hired applicants, the algorithms do not need knowledge on n, we obtain the following corollary.

Corollary 1. *Algorithms* 1–4 *can be adapted to be competitive even when n is not known.*

The question remains whether we can stay competitive when only a single applicant may be employed at a time. We refer to this setting as the setting of *sequential employment.* In the remaining part of this section, we show that the competitive ratio is $\Omega(\sqrt{n}/\log n)$ for any online algorithm. To do so, we give an optimal online algorithm for the sequential employment for any distribution and show that the competitive ratio is in the order of $\Theta(\sqrt{n}/\log n)$ for the special case of $X = U[0,1]$. This implies that, in contrast to *Algorithms* 1–4, there is no constant competitive algorithm for arbitrary distributions. Note that the offline optimum uses only sequential employment.

Let $\mathbf{E}_n$ denote the expected cost of the best online algorithm for n applicants under sequential employment. We give an optimal online algorithm (see Algorithm 5) based on the values $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_{n-1}$. Because a single applicant needs to be employed at any time, the only decision of the algorithm regards the respective hiring times. Interestingly, our algorithm hires all but the last applicant only for a single unit of time.

Before we prove this result, we need the following technical lemma.

Lemma 11. *The function $G(\tau) := \mathbf{Pr}[x \geq \tau](\tau - \mathbf{E}[x \mid x \geq \tau])$ is nondecreasing.*

Proof. We rewrite $G(\tau) = \tau\mathbf{Pr}[x \geq \tau] - \int_\tau^\infty xf(x)\,\mathrm{d}x$, where f is the density of X. Then, for $\tau' > \tau$, we have

$$\begin{aligned} G(\tau') - G(\tau) &= \tau'\mathbf{Pr}[x \geq \tau'] - \tau\mathbf{Pr}[x \geq \tau] + \int_\tau^{\tau'} xf(x)\,\mathrm{d}x \\ &\geq \tau' \int_{\tau'}^\infty f(x)\,\mathrm{d}x - \tau\int_\tau^\infty f(x)\,\mathrm{d}x + \tau\int_\tau^{\tau'} f(x)\,\mathrm{d}x \\ &\geq 0, \end{aligned}$$

which concludes the proof. □

Algorithm 5 (An Optimal Online Algorithm for Sequential Employment)

```
for i ← 1, ..., n do
    if x_i < τ_{n−i} = E_{n−i}/(n−i) then
        hire applicant i for remaining time n − i + 1 ;
        stop ;
    else
        hire applicant i for one unit of time ;
```

We are now in position to prove that Algorithm 5 is optimal.

Theorem 8. *Algorithm* 5 *is an optimal online algorithm for sequential employment.*

Proof. Let $\tau_i := \mathbf{E}_i/i$ be the threshold employed by Algorithm 5 when i applicants remain. For technical reasons, let τ_0 be any constant greater than τ_1. We prove the theorem by induction on n, additionally showing that $\tau_n \leq \tau_{n-1}$.

For $n = 1$, the algorithm is obviously optimal, and $\tau_1 \le \tau_0$ by definition. Consider the first applicant of cost x_1. With $\mathbf{E}_0 := 0$, the expected cost of the optimal online algorithm follows the recursion

$$\min_{t \in \{1,\ldots,n\}} \{x_1 t + \mathbf{E}_{n-t}\}. \tag{21}$$

Consider the case $x_1 < \tau_{n-1} = \mathbf{E}_{n-1}/(n-1)$. We proceed to show that the minimum (21) is attained for $t = n$. By induction, for all $t \in \{1,\ldots,n-1\}$, we have $x_1 < \tau_{n-1} \le \tau_t$, and thus

$$nx_1 = tx_1 + (n-t)x_1 < tx_1 + \mathbf{E}_{n-t}.$$

Now consider the case $x_1 \ge \tau_{n-1}$. We need to show that the minimum (21) is attained for $t = 1$. By induction, for all $t \in \{2,\ldots,n\}$, we have $\tau_{n-1} \le \tau_{n-t}$, and thus

$$\begin{aligned} tx_1 + \mathbf{E}_{n-t} &= x_1 + (t-1)x_1 + (n-t)\tau_{n-t} \\ &\ge x_1 + (t-1)\tau_{n-1} + (n-t)\tau_{n-1} \\ &= x_1 + \mathbf{E}_{n-1}. \end{aligned}$$

It remains to show $\tau_n \le \tau_{n-1}$. From the above, we have

$$\mathbf{E}_n = n\mathbf{Pr}[x < \tau_{n-1}]\mathbf{E}[x \mid x < \tau_{n-1}] + \mathbf{Pr}[x \ge \tau_{n-1}](\mathbf{E}[x \mid x \ge \tau_{n-1}] + \mathbf{E}_{n-1}).$$

Using

$$\mathbf{E}[x] = \mathbf{Pr}[x < \tau_{n-1}]\mathbf{E}[x \mid x < \tau_{n-1}] + \mathbf{Pr}[x \ge \tau_{n-1}]\mathbf{E}[x \mid x \ge \tau_{n-1}],$$

this yields

$$\begin{aligned} \tau_n &= \mathbf{E}[x] + \frac{1}{n}\mathbf{Pr}[x \ge \tau_{n-1}](\mathbf{E}_{n-1} - (n-1)\mathbf{E}[x \mid x \ge \tau_{n-1}]) \\ &= \mathbf{E}[x] + \frac{n-1}{n}\mathbf{Pr}[x \ge \tau_{n-1}](\tau_{n-1} - \mathbf{E}[x \mid x \ge \tau_{n-1}]). \end{aligned}$$

Using Lemma 11 (with $\tau_{n-1} \le \tau_{n-2}$ by induction) and the fact that the second term is negative, we obtain

$$\begin{aligned} \tau_n &\le \mathbf{E}[x] + \frac{n-2}{n-1}\mathbf{Pr}[x \ge \tau_{n-1}](\tau_{n-1} - \mathbf{E}[x \mid x \ge \tau_{n-1}]) \\ &\le \mathbf{E}[x] + \frac{n-2}{n-1}\mathbf{Pr}[x \ge \tau_{n-2}](\tau_{n-2} - \mathbf{E}[x \mid x \ge \tau_{n-2}]) \\ &= \tau_{n-1}, \end{aligned}$$

which concludes the proof. □

We derive the optimal competitive ratio for the case where $X = U[0,1]$.

Lemma 12. *For $X = U[0,1]$, we have*

$$\mathbf{E}_n = \begin{cases} \dfrac{1}{2}, & \text{for } n = 1, \\ \mathbf{E}_{n-1} + \dfrac{1}{2} - \dfrac{\mathbf{E}_{n-1}^2}{2(n-1)}, & \text{for } n > 1. \end{cases}$$

Proof. The case $n = 1$ follows from $\mathbf{E}[x] = 1/2$. For $n > 1$, we use the fact that Algorithm 5 is optimal. We obtain

$$\begin{aligned} \mathbf{E}_n &= n\mathbf{Pr}[x < \tau_{n-1}]\mathbf{E}[x \mid x < \tau_{n-1}] + \mathbf{Pr}[x \ge \tau_{n-1}](\mathbf{E}[x \mid x \ge \tau_{n-1}] + \mathbf{E}_{n-1}) \\ &= n\tau_{n-1} \cdot \frac{1}{2}\tau_{n-1} + (1 - \tau_{n-1})\left(\frac{1 + \tau_{n-1}}{2} + \mathbf{E}_{n-1}\right) \\ &= \frac{n\tau_{n-1}^2}{2} + \frac{1}{2} + \mathbf{E}_{n-1} - \frac{1}{2}\tau_{n-1}^2 - \mathbf{E}_{n-1}\tau_{n-1} \\ &= \mathbf{E}_{n-1} + \frac{1}{2} - \frac{\mathbf{E}_{n-1}^2}{2(n-1)}, \end{aligned}$$

which concludes the proof. □

With this, we can bound the expected cost of any online algorithm.

Lemma 13. *For $X = \mathcal{U}[0,1]$, we have $\sqrt{n+1} - 1 \le \mathbf{E}_n \le \sqrt{n}$.*

Proof. For the sake of contradiction, assume that $\mathbf{E}_n > \sqrt{n}$ for some value of n. With Lemma 12, we obtain

$$\mathbf{E}_{n+1} = \mathbf{E}_n + \frac{1}{2} - \frac{\mathbf{E}_n^2}{2n} < \mathbf{E}_n + \frac{1}{2} - \frac{n}{2n} = \mathbf{E}_n,$$

which is a contradiction with $\mathbf{E}_n$ being nondecreasing.

Let $h(n) := \sqrt{n+1} - 1$. It is easy to check that $\mathbf{E}_n \ge h(n)$ for $n < 7$. For $n \ge 7$, we use induction on n. To that end, assume $\mathbf{E}_n \ge h(n)$ holds, and consider $\mathbf{E}_{n+1}$. Clearly, $\mathbf{E}_{n+1} \ge \mathbf{E}_n$. If $\mathbf{E}_n \ge \sqrt{n+1} - 0.8$, it thus suffices to show that $h(n+1) - h(n) \le 0.2$. Because h is concave and $n \ge 7$, we indeed have

$$h(n+1) - h(n) \le h'(n) = \frac{1}{2\sqrt{n+1}} \le 0.2.$$

Finally, let $\mathbf{E}_n < \sqrt{n+1} - 0.8$. Using $n \ge 7$, we show that $\mathbf{E}_n$ grows faster than $h(n)$:

$$\begin{aligned}\mathbf{E}_{n+1} - \mathbf{E}_n &= \frac{1}{2} - \frac{\mathbf{E}_n^2}{2n} \ge \frac{1}{2} - \frac{(\sqrt{n+1} - 0.8)^2}{2n} = \frac{160\sqrt{n+1} - 164}{200n} \\ &\ge \frac{\sqrt{n+1}}{2(n+1)} = h'(n) \ge h(n+1) - h(n),\end{aligned}$$

which concludes the proof. □

Together with Lemma 1, we immediately get the following bound on the competitive ratio of any online algorithm.

Theorem 9. *The competitive ratio of the best online algorithm for sequential employment and a uniform distribution $X = U[0,1]$ is $\Theta(\sqrt{n}/\log n)$.*

Observe that Theorem 9 implies that every sequential algorithm has a competitive ratio of $\Omega(\sqrt{n}/\log n)$. This immediately disqualifies simple algorithms that never hire more than one applicant at every point in time from being constant competitive for the problems considered in Sections 4–6.

8. Analysis of the Markov Chains

In this section, we study the Markov chains that govern the evolution of the threshold values of our algorithms.

8.1. Markov Chain $\hat{M}(p,k)$

We start with the simple Markov chain $\hat{M}(p,k)$ used in Sections 4.1 and 6. The Markov chain has states $0, \ldots, k$ and transition probabilities as shown in Figure 6.

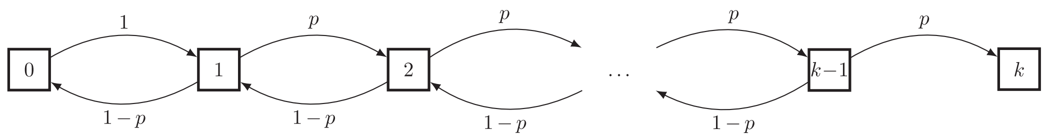
In the following, we compute the expected number of visits to each state.

Lemma 14. *Let $p > 1/2$ and $k \in \mathbb{N}$. Starting in state 0, the expected number of visits to each state j of the Markov chain $\hat{M}(p,k)$ is at most $\frac{1}{2p-1}$.*

Proof. Let v_j denote the expected number of visits to state j, when starting from state 0. We derive that the values v_j, $j \in \{0, \ldots, k\}$, satisfy the following equations:

$$v_k = 1, \tag{22a}$$
$$v_k = p v_{k-1}, \tag{22b}$$
$$v_{k-1} = p v_{k-2}, \tag{22c}$$
$$v_j = (1-p) v_{j+1} + p v_{j-1} \qquad \text{for all } j \in \{2, \ldots, k-2\}, \tag{22d}$$
$$v_1 = v_0 + (1-p) v_2, \tag{22e}$$
$$v_0 = 1 + (1-p) v_1, \tag{22f}$$

Figure 6. Markov chain used in Section 4.1 and Section 6. Nodes correspond to states.



where (22a) follows from the fact that k is the absorbing state, and (22b) uses that state k is reached only from state $k-1$. Equation (22c) follows because state $k-1$ can be reached from state $k-2$ only. Equation (22d) follows from the fact that we reach state j from $j-1$ and $j+1$ and leave states $j-1$ and $j+1$ to j with a probabilities of p and $1-p$, respectively. As state 0 is left with probability 1 toward its successor, Equation (22e) holds as a special case. Furthermore, for state 0, we obtain Equation (22f) because 0 is the starting state and can be reached only from state 1.

Note that (22a) and (22b) imply $v_{k-1} = 1/p$, which, by (22c), implies $v_{k-2} = 1/p^2$. With these start values, (22d) uniquely defines a homogenous recurrence relation on $v_1, \dots, v_{k-1}$ with

$$v_j = \frac{1}{p} v_{j+1} - \frac{1-p}{p} v_{j+2} \qquad \text{for all } j \in \{2, \dots, k-2\}.$$

Solving this recurrence by the method of characteristic equations yields that the characteristic polynomial $x^2 - \frac{1}{p}x + \frac{1-p}{p}$ has roots $\frac{1}{p} - 1$ and 1 so that the explicit solution is

$$v_j = \lambda_1 \left(\frac{1}{p} - 1\right)^{k-j-1} + \lambda_2$$

for some parameters $\lambda_1, \lambda_2 \in \mathbb{R}$. Choosing λ_1 and λ_2 such that the equations $v_{k-1} = 1/p$ and $v_{k-2} = 1/p^2$ are satisfied gives

$$\lambda_1 = \frac{1}{2p-1}\left(1 - \frac{1}{p}\right), \quad \lambda_2 = \frac{1}{p} - \frac{1}{2p-1}\left(1 - \frac{1}{p}\right).$$

As a result, for $j \in \{1, \dots, k\}$, we obtain

$$\begin{aligned} v_j &= \frac{1}{2p-1}\left(1 - \frac{1}{p}\right)\left(\frac{1}{p} - 1\right)^{k-j-1} + \frac{1}{p} - \frac{1}{2p-1}\left(1 - \frac{1}{p}\right) \\ &= \frac{1}{2p-1}\left[\left(\frac{1}{p} - 1\right) - \left(\frac{1}{p} - 1\right)^{k-j}\right] + \frac{1}{p}. \end{aligned} \tag{23}$$

Finally, v_0 is defined via (22f). Observe that, together with (23), this satisfies (22e) as required.

It remains to show that $v_j \le \frac{1}{2p-1}$ for all $j \in \{0, \dots, k\}$. For $j \in \{1, \dots, k-1\}$, we use Equation (23) and the fact that $p > 1/2$ to obtain

$$\begin{aligned} v_j &= \frac{1}{2p-1}\left[\left(\frac{1}{p} - 1\right) - \left(\frac{1}{p} - 1\right)^{k-j}\right] + \frac{1}{p} \\ &\le \frac{1}{2p-1}\left(\frac{1}{p} - 1\right) + \frac{1}{p} \\ &= \frac{p}{p(2p-1)} = \frac{1}{2p-1}. \end{aligned}$$

For $j = 0$ we have, by Equation (22f),

$$v_0 = 1 + (1-p)v_1 \le 1 + \frac{1-p}{2p-1} = \frac{p}{2p-1} \le \frac{1}{2p-1},$$

which completes the proof. □

8.2. Markov Chain $\hat{N}(p,k)$

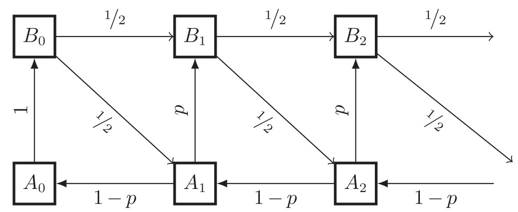
In this section, we study the Markov chain $\hat{N}(p,k)$ used in Sections 4.2 and 5. The Markov chain has states A_j and B_j, for $j \in \{0, \dots, k\}$, and transition probabilities as shown in Figure 7.

We start to bound the expected number of transitions from an A state to a B state.

Lemma 15. *Starting in state A_0 of Markov chain $\hat{N}(p,k)$, the expected number of transitions from an A state to a B state is at most*

$$h = \frac{kp}{3p-1} - \frac{4p(1-2p)}{(3p-1)^2} + \left(\frac{1-p}{3p-1}\right)^2 \left(\frac{2(1-p)}{1+p}\right)^k.$$

Figure 7. Markov chain $\hat{N}(p,k)$ with homogenous transition probability p and $k+1$ states used in Sections 4.2 and 5. Nodes correspond to states.



Proof. Let a_j (respectively b_j) denote the expected number of transitions from an A state to a B state, when starting from state A_j (respectively B_j). We get

$$b_k = 0, \tag{24a}$$
$$b_j = \frac{1}{2}b_{j+1} + \frac{1}{2}a_{j+1} \qquad \text{for all } j \in \{0,\dots,k-1\}, \tag{24b}$$
$$a_j = p(b_j+1) + (1-p)a_{j-1} \qquad \text{for all } j \in \{1,\dots,k\}, \tag{24c}$$
$$a_0 = 1 + b_0. \tag{24d}$$

Defining $\beta = \frac{2(1-p)}{1+p}$, for $j \in \{0,\dots,k\}$, it is straightforward to check that (24a), (24b), and (24c) are fulfilled by

$$a_j = \frac{(k-j+2)p}{3p-1} - \beta^j \frac{2p(1-p)}{(3p-1)^2} + \beta^k \frac{(1-p)^2}{(3p-1)^2} \quad \text{and}$$
$$b_j = \frac{(k-j)p}{3p-1} - \beta^j \frac{(1-p)^2}{(3p-1)^2} + \beta^k \frac{(1-p)^2}{(3p-1)^2}.$$

It follows that the expected number of transitions from an A state to a B state when starting at A_0 is

$$a_0 = \frac{(k+2)p}{3p-1} - \frac{2p(1-p)}{(3p-1)^2} + \beta^k \frac{(1-p)^2}{(3p-1)^2} = \frac{kp}{3p-1} - \frac{4p(1-2p)}{(3p-1)^2} + \frac{2^k(1-p)^{k+2}}{(3p-1)^2(1+p)^k},$$

which completes the proof. □

Lemma 16. *Starting in state A_0 of Markov chain $\hat{N}(p,k)$ for each $j \in \{0,\dots,k-1\}$, the expected number of transitions from B_j to A_{j+1} is at most $\frac{p}{3p-1}$.*

Proof. As the expected number of such transitions is half the expected number of visits to state B_j , it suffices to bound the latter quantity.

Suppose we are in state B_j. The probability of coming back to B_j equals the probability of hitting A_j from B_j. Denote by $a_i(j)$ and $b_i(j)$ the hitting probability of state A_i from A_j and B_j, respectively. We have

$$b_i(k) = 0, \tag{25a}$$
$$b_i(j) = \frac{1}{2}b_i(j+1) + \frac{1}{2}a_i(j+1) \qquad \text{for all } j \in \{i,\dots,k-1\}, \tag{25b}$$
$$a_i(j) = pb_i(j) + (1-p)a_i(j-1) \qquad \text{for all } j \in \{i+1,\dots,k\}, \tag{25c}$$
$$a_i(i) = 1. \tag{25d}$$

Let $\beta = \frac{2(1-p)}{p+1} < 1$ (as $p > \frac{1}{3}/$). It is easy to check that for $j \in \{i,\dots,k\}$,

$$a_i(j) \le \beta^{j-i},$$
$$b_i(j) \le \frac{1-p}{2p}\beta^{j-i}$$

gives an upper bound on the solution of (25), as these values satisfy equalities (25b), (25c), (25d), and only overestimate (25a). We can interpret the visits to state B_j after the first visit as a geometric random variable with success probability $1-b_j(j)$. Thus, the expected number of visits to B_j is given by

$$1 + \frac{1-(1-b_j(j))}{1-b_j(j)} = \frac{1}{1-b_j(j)} \le \frac{1}{1-\frac{1-p}{2p}} = \frac{2p}{3p-1}.$$

We conclude that the expected number of transitions from B_j to A_{j+1} is at most $\frac{p}{3p-1}$, proving the claim. □

9. Conclusion

We considered prophet inequalities with a covering constraint and a minimization objective. We gave constant competitive algorithms for this type of problem and established concurrent employment as a necessary feature of such algorithms.

We note that our results extend to slightly more general settings, where (a) we relax the covering constraint by associating a penalty $B < \infty$ with time steps where no contract is active, (b) multiple applicants arrive in each time step, and (c) applicants may be hired fractionally.

A crucial limitation of our model is the assumption that costs are distributed independently, and it remains an interesting question how to address correlated costs.

Acknowledgments

The authors thank the anonymous reviewers and the associate editor for their valuable comments that helped improve the presentation of the paper.

Endnotes

[1] We discuss a relaxation of the strict covering constraint in Section 9.

[2] Here and throughout, we denote the logarithm of n to base 2 by $\log(n)$ and the natural logarithm of n with $\ln(n)$.

[3] In general, we need to define δ_q more carefully via $\mathbf{Pr}[x \le \delta_q] \ge q$ and $\mathbf{Pr}[x \ge \delta_q] \ge 1 - q$.

References

[1] Alaei S (2014) Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM. J. Comput.* 43(2): 930–972.
[2] Ash JM, Berele A, Catoiu S (2012) Plausible and genuine extensions of l'Hospital's rule. *Math. Magazine* 85(1):52–60.
[3] Bearden JN (2006) A new secretary problem with rank-based selection and cardinal payoffs. *J. Math. Psych.* 50(1):58–59.
[4] Bellman R (1954) The theory of dynamic programming. *Bull. Amer. Math. Soc.* 60(6):503–514.
[5] Cayley AF (1896) *Mathematical Questions with Their Solutions. The Collected Mathematical Papers of Arthur Cayley*, vol. 10 (Cambridge University Press, Cambridge, UK), 587–588.
[6] Chawla S, Hartline JD, Malec DL, Sivan B (2010) Multi-parameter mechanism design and sequential posted pricing. *Proc. 41st Annual ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 311–320.
[7] Dütting P, Kleinberg R (2015) Polymatroid prophet inequalities. Bansal N, Finocchi I, eds. *Algorithms—ESA 2015*, Lecture Notes in Computer Science, vol. 9294 (Springer, Berlin), 437–449.
[8] Esfandiari H, Hajiaghayi M, Liaghat V, Monemizadeh M (2015) Prophet secretary. Bansal N, Finocchi I, eds. *Algorithms—ESA 2015*, Lecture Notes in Computer Science, vol. 9294 (Springer, Berlin), 496–508.
[9] Ferguson TS (1989) Who solved the secretary problem? *Statist. Sci.* 4(3):282–289.
[10] Fiat A, Gorelik I, Kaplan H, Novgorodov S (2015) The temp secretary problem. Bansal N, Finocchi I, eds. *Algorithms—ESA 2015*, Lecture Notes in Computer Science, vol. 9294 (Springer, Berlin), 631–642.
[11] Fisher RA, Tippett LHC (1928) Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Math. Proc. Cambridge Philos. Soc.* 24(2):180–190.
[12] Gilbert JP, Mosteller F (1966) Recognizing the maximum of a sequence. *J. Amer. Statist. Assoc.* 61(313):35–73.
[13] Gnedenko BV (1943) Sur la distribution limite du terme maximum dune série aléatoire. *Ann. Math.* 44(3):423–453.
[14] Göbel O, Hoefer M, Kesselheim T, Schleiden T, Vöcking B (2014) Online independent set beyond the worst-case: Secretaries, prophets, and periods. Esparza J, Fraigniaud P, Husfeldt T, Koutsoupias E, eds. *Automata, Languages, and Programming—ICALP 2014*, Lecture Notes in Computer Science, vol. 8573 (Springer, Berlin), 508–519.
[15] Guttman I (1960) On a problem of L. Moser. *Canadian Math. Bull.* 3(1):35–39.
[16] Hajiaghayi MT, Kleinberg R, Sandholm T (2007) Automated online mechanism design and prophet inequalities. *Proc. 22nd Natl. Conf. Artificial Intelligence* (AAAI Press, Palo Alto, CA), 58–65.
[17] Hill TP, Kertz RP (1992) A survey of prophet inequalities in optimal stopping theory. *Contemporary Math.* 125(1):191–207.
[18] Karamata J (1933) Sur un mode de croissance régulière. Théorèmes fondamentaux. *Bull. Soc. Math. France* 61:55–62.
[19] Karlin S (1962) Stochastic models and optimal policy for selling an asset. Arrow KJ, Karlin S, Scarf H, eds. *Studies in Applied Probability and Management Science* (Stanford University Press, Stanford, CA), 148–158.
[20] Kennedy DP (1987) Prophet-type inequalities for multi-choice optimal stopping. *Stochastic Processes Appl.* 24(1):77–88.
[21] Kennedy DP, Kertz RP (1991) The asymptotic behavior of the reward sequence in the optimal stopping of i.i.d. random variables. *Ann. Probab.* 19(1):329–341.
[22] Kleinberg R, Weinberg SM (2012) Matroid prophet inequalities. *Proc. 44th Annual ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 123–136.
[23] Krengel U, Sucheston L (1977) Semiamarts and finite values. *Bull. Amer. Math. Soc.* 83(4):745–747.
[24] Krengel U, Sucheston L (1978) On semiamarts, amarts, and processes with finite value. Kuelbs J, ed. *Probability on Banach Spaces*, Advances in Probability and Related Topics, vol. 4 (Dekker, New York), 197–266.
[25] Leadbetter MR, Lindgren G, Rootzen H (1983) *Extremes and Related Properties of Random Sequences and Processes* (Springer, New York).
[26] Moser L (1956) On a problem of Cayley. *Scripta Math.* 22:289–292.

[27] Myerson RB (1981) Optimal auction design. *Math. Oper. Res.* 6(1):58–73.

[28] Rubinstein A (2016) Beyond matroids: Secretary problem and prophet inequality with general constraints. *Proc. 48th Annual ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 324–332.

[29] Rubinstein A, Singla S (2017) Combinatorial prophet inequalities. *Proc. 28th Annual ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 1671–1687.

[30] Samuel-Cahn E (1984) Comparison of threshold stop rules and maximum for independent non-negative random variables. *Ann. Probab.* 12(4):1213–1216.

[31] Verroios V, Papadimitriou P, Johari R, Garcia-Molina H (2015) Client clustering for hiring modeling in work marketplaces. *Proc. 21st ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 2187–2196.