

Stochastic Probing with Increasing Precision

Martin Hofer¹, Kevin Schewior², Daniel Schmand³

¹Goethe University Frankfurt, Germany

²University of Cologne, Germany

³University of Bremen, Germany

mhofer@cs.uni-frankfurt.de, schewior@cs.uni-koeln.de, schmand@uni-bremen.de

Abstract

We consider a selection problem with stochastic probing. There is a set of items whose values are drawn from independent distributions. The distributions are known in advance. Each item can be *tested* repeatedly. Each test reduces the uncertainty about the realization of its value. We study a testing model, where the first test reveals if the realized value is smaller or larger than the median of the underlying distribution. Subsequent tests allow to further narrow down the interval in which the realization is located. There is a limited number of possible tests, and our goal is to design near-optimal testing strategies that allow to maximize the expected value of the chosen item. We study both identical and non-identical distributions and develop polynomial-time algorithms with constant approximation factors in both scenarios.

1 Introduction

In recent years, there has been a surge of interest in learning problems with probing. There is a set of n items, and each item has an independent distribution over its value. The goal of the learner is to select an item with a value as large as possible. In the standard model, the learner can *probe* a bounded number of k items. Upon probing an item, the learner sees its realized value.

A variety of applications are captured by this approach and its extensions. For example, in a hiring process, an “item” is a candidate. The application material implies a stochastic belief over the quality of each candidate. Probing corresponds to an interview of a candidate, and the capacity of the interviewer is limited to k interviews. The probing problem corresponds to the selection of candidates to be interviewed to optimize the value of the candidate that is hired eventually. Additional applications arise, for example, in online dating or kidney exchange. The problem is to probe pairs of agents for compatibility and eventually match the population to maximize some objective function, e.g., the number of compatible pairs or the overall quality of matches. Probing has further applications in domains like influence maximization or Bayesian mechanism design [Gupta and Nagarajan, 2013; Asadpour and Nazerzadeh, 2016].

Computing an optimal probing decision is a non-trivial task as each of the subsequent probing decisions may depend on the outcomes of previous probes. A standard technique for doing so is an often intractable dynamic-programming approach. Beyond this, one commonly resorts to finding polynomial-time approximation algorithms (e.g., [Chen *et al.*, 2009; Bansal *et al.*, 2012; Dean *et al.*, 2008; Bhalgat *et al.*, 2011; Ma, 2018]).

In the vast majority of approaches studied in computer science, probing reveals the *exact realization* of the underlying random variable; probing an item completely eradicates the uncertainty. In contrast, many applications give rise to probing problems in which we only obtain *some limited information* about the item. Consider for example an interviewer in a hiring process. Instead of interpreting an interview as a single probe that reveals all information, it is usually the case that the interviewer can ask questions or request information that will partially reveal the qualifications of the respective candidate. More realistically, a question or exercise in an interview can be seen as a probe, but only by asking multiple questions of varying levels of difficulty the interviewer can eventually estimate the exact qualification of each candidate. As another example, consider a Bayesian single-item auction with posted prices. By setting a posted price, the auctioneer learns if the bidders have a value above or below that price, but that does not directly reveal the valuation of each bidder. Only repeated probing with different prices can reveal the exact value of each bidder.

In this paper, we introduce selection problems with *repeated probing*. In the beginning, nature makes a single independent draw for each item to determine the realized value. We can test an item, but the exact realized value stays unknown and each test only reveals limited information about the realized value. Subsequent testing can be used to obtain more and more fine-grained information about the same realized value. There are many ways to express this condition formally, i.e., how exactly the result of a test changes the conditional distribution of the item. In our model, we take a simple and intuitive approach: The first test reveals if the realized value of an item is above or below the median of the distribution. Each subsequent test reveals if the realized value of the item is above or below the median of the conditional distribution, where the condition is the binary feedback of previous tests.

Example 1. Suppose we can perform $k = 3$ tests on $n = 2$ items. Initially, the realized values of the items are drawn i.i.d. from the uniform distribution over the set $\{1, 2/3, 1/3, 0\}$.

W.l.o.g. we first test item 1. The result of the test is either A (realization above the median) or B (below the median). Suppose it is A, then we know that the realized value of item 1 is either 1 or $2/3$, both with probability $1/2$. Next, we test item 2. If the result for item 2 is B, then the realized value of item 2 must be either $1/3$ or 0, both with probability $1/2$. Hence, the optimum must be item 1. Instead, assume the result of the test for item 2 is A, then the realized value of item 2 is either 1 or $2/3$, both with probability $1/2$. We apply the third test again to item 1. If the result is A, it is clear that the realization of item 1 is 1, and so item 1 is an optimum. Otherwise, the realization is $2/3$, and then item 2 is an optimum.

Interestingly, by repeating the analysis for the case when the result of the first item is B, we see that we can always identify the item with the best realization. Note that we cannot achieve this with $k \leq 2$ tests. In this paper, we present provably good probing algorithms, even in this case, when the maximum realization cannot be fully identified. ■

Our main results are two probing algorithms, one for identically distributed items and one for non-identical distributions. We show that both run in polynomial time and obtain a constant-factor approximation of the optimal probing strategy that maximizes the expected value of the selected item. In fact, the constant approximation factors hold even with respect to an upper bound on the optimum given by the expected value of the best strategy in the *standard probing* model, where each of the k tests completely reveals the realization. Moreover, our algorithms and analysis also apply to a sequential variant of the problem, where all tests on a single item must be applied consecutively, and the order of items for probing is externally given. For this problem, we provide an efficient algorithm to compute the optimal testing strategy.

1.1 Related Work

Stochastic probing problems in which probing eradicates all uncertainty about the tested item have been extensively studied. A prominent line of work [Asadpour *et al.*, 2008; Gupta and Nagarajan, 2013; Adamczyk *et al.*, 2014; Gupta *et al.*, 2016; Gupta *et al.*, 2017] is concerned with fairly general models in which an—according to some given downward-closed set system—feasible set of (often Bernoulli) variables can be adaptively probed. When probing is done, a set of items that is feasible according to another given downward-closed set system can be selected, and the obtained value is an (e.g., submodular) function of the selected items. The goal is typically to develop algorithms that approximate the best strategy and whose guarantees are parameterized by the respective instance, e.g., parameters of the set systems corresponding to the constraints [Chen *et al.*, 2009; Bansal *et al.*, 2012; Dean *et al.*, 2008; Bhalgat *et al.*, 2011; Ma, 2018]. One approach to achieve constant-factor approximations is bounding both the adaptivity gap and the approximation factor of some non-adaptive algorithm by a constant; see, e.g., [Gupta *et al.*, 2016]. In this light, our approximation results, which are based on algorithms that work in the

sequential setting, can be viewed as a bound on the “sequentiality gap” of our problem.

Instead of having to obey a hard constraint on the set of items that can be probed, one is charged for probing a (typically arbitrarily but independently distributed) item in the *Pandora’s box* problem [Weitzman, 1979]. The goal is to maximize the expected difference between the value of the (originally single) picked item and the probing cost. While in the standard model the picked item must be a previously probed item, in [Beyhaghi and Kleinberg, 2019] any single item can be picked, but again probing eradicates all uncertainty. In a Markovian model [Gupta *et al.*, 2019], each probe only advances a Markov chain associated with the respective item by a single step. This is a model with limited information, but in contrast to our model an item may only be picked once the Markov chain has reached a terminal state, i.e., once *all* uncertainty has been eradicated. Some of these models have been generalized to multi-item variants; see, e.g., [Singla, 2018]. In the standard model, an optimal algorithm is known; more generally, one often resorts to approximating, sometimes to an arbitrary precision [Fu *et al.*, 2018].

The prophet-inequality setting [Krengel and Sucheston, 1977] is different in that the values of all items are revealed eventually and there is no probing cost. In the classic version, items are revealed in an adversarial order, and a single item can only be picked at the time of its revelation. Then, the best strategy can be computed via a simple dynamic program, but the challenge is typically to compare the performance with that of an all-knowing prophet. When the order of revelation can be chosen, computing the best strategy becomes less tractable [Agrawal *et al.*, 2020]. We also refer to surveys on this topic [Lucier, 2017; Correa *et al.*, 2018].

Let us emphasize that our problem is quite different from multi-armed bandit models (e.g., [Gittins, 1979]), in which typically actions have random payoffs from *unknown distributions*, from which samples are *repeatedly drawn and revealed*. In contrast, here each value for an item comes from a *known distribution*, is sampled only *once in the beginning* and only *revealed gradually* (upon testing). This setting calls for analyses different from the “regret”-style analysis typically applied for multi-armed bandit models.

1.2 Probing Model

We are given a set $N = \{1, \dots, n\}$ of items and a test capacity $k \in \mathbb{N}$. The value v_i of each item $i \in N$ is non-negative, drawn independently from a known distribution D_i over \mathbb{R}_+ , and unknown upfront. A *probing algorithm* can perform up to k tests. Each test is performed on one of the items. In contrast to most of the related literature, we assume that a test does not *reveal* the exact realization of the item’s value. Instead, each test results in an improved estimation of the realization.

We assume that the first test shows whether the value is above or strictly below the median of the distribution. For simplicity, let us first assume a continuous distribution. Let $D_i^{-1}(q)$ be the smallest value such that $\Pr[v_i < D_i^{-1}(q)] = q$. We call $D_i^{-1}(q)$ the q -quantile of D_i . The first test is called positive if $v_i \geq D_i^{-1}(1/2)$, and negative otherwise. Given this result, the conditional distribution of the item can then be tested for the new median in the same manner. In terms of the

original distribution, if the first test was positive, the next test reveals if $v_i \geq D_i^{-1}(3/4)$ or $v_i \in [D_i^{-1}(1/2), D_i^{-1}(3/4)]$; if the first test was negative, the next test reveals if $v_i \in [D_i^{-1}(1/4), D_i^{-1}(1/2))$ or $v_i < D_i^{-1}(1/4)$.

In this way, repeated testing leads to an improved estimation of the realized value – each subsequent test informs the algorithm whether the value is above (positive) or strictly below (negative) the median of the conditional distribution, where the condition is on the outcomes of all previous tests on that item. The algorithm can perform k tests in total. It can choose the next item to be tested adaptively. In the end, it selects one item. The goal is to maximize the value of the selected item.

To aid the discussion of computational complexity, distributions are discrete and given in explicit representation. For applying the tests for such distributions, we assume that ties are broken consistently, e.g., by initially drawing a random number $x_i \in [0, 1]$ for each item i , extending D_i to a continuous distribution over tuples (v_i, x_i) , and using a lexicographic comparison for tuples (v_i, x_i) .

Our algorithms are polynomial-time in the input given by the n discrete distributions in explicit representation and k . An algorithm can perform k tests on the items. Each test is executed via an oracle call that takes constant time.

We analyze approximation ratios of our algorithms, i.e., the worst-case ratio of the expected value of the algorithm over the expected value of an optimal (finite-time) probing algorithm. In fact, the bounds for our algorithms hold even in comparison to the expected value of the best algorithm in the *standard probing* model, where each test completely reveals the realization.

Outline. In Section 2 we describe our algorithm and the analysis for the case of independent and identically distributed (i.i.d.) items with $D_i = D_j = D$ for all $i, j \in N$. For i.i.d. we use the short notation $\delta_q = D^{-1}(q)$. Our algorithm for the general case is considered in Section 3. In Section 4 we consider the sequential probing problem, where items have to be probed sequentially in a given order. We conclude in Section 5.

2 Identical Distributions

As the main result in this section we present algorithm ALG_{iid} that has a constant approximation ratio for identical distributions. The algorithm only depends on the test results and *uses no additional information about the distribution*. In contrast, an optimal algorithm seems to be complicated and must have a further dependence on the distribution, as we show in the full version.

Even though our algorithm is quite simple, it still achieves a small constant approximation guarantee – even with respect to the optimum in the standard probing model, where each test reveals the realization.

Algorithm ALG_{iid} . Let $k' = 2^{\lceil \log_2 \min\{n, k+1\} \rceil}$ the smallest power of 2 that is larger or equal to $\min\{n, k+1\}$. Our algorithm ALG_{iid} performs tests on the items sequentially. For each item i , it repeatedly tests the item until it is clear whether its value v_i is larger or equal to $\delta_{(k'-1)/k'}$ or not,

i.e., until there are $\log_2(k')$ positive tests in a row or until there is a single negative test. If $v_i \geq \delta_{(k'-1)/k'}$, we call this item a *good item*. In this case, the algorithm selects i and terminates. Otherwise, it continues by testing item $i+1$. If the algorithm fails to find a good item or runs out of tests, it selects a random item.

We slightly abuse notation and use ALG_{iid} to denote our algorithm and $\mathbf{E}[\text{ALG}_{\text{iid}}]$ for the expected value of the chosen item. Similarly, we use OPT to denote the optimal probing strategy and $\mathbf{E}[\text{OPT}]$ for its expected value. Our guarantee will be in terms of $\mathbf{E}[\text{UB}_{k+1}]$, which is the expected value obtained by seeing the *exact* realization of the first $k+1$ items and selecting the one with the best realization. For UB we use tests from the standard probing model where they reveal the exact realization. Also, instead of testing k items and then possibly taking an (untested) item $k+1$ (in case $k < n$ and all observed realizations are below the expectation of D), we allow UB_{k+1} to also reveal the realization of item $k+1$ and then select the best realization from the $k+1$ tested items. Clearly, observing exact realizations and the additional test imply $\mathbf{E}[\text{UB}_{k+1}] \geq \mathbf{E}[\text{OPT}]$. Our main result in this section is the following.

Theorem 2. ALG_{iid} runs in polynomial time and achieves an expected value of

$$\mathbf{E}[\text{ALG}_{\text{iid}}] \geq \left(1 - \frac{1}{\sqrt[e]{e}} - o(1)\right) \cdot \mathbf{E}[\text{UB}_{k+1}] ,$$

where the asymptotics is in $\min(n, k)$.

Proof. We first assume $k < n$ and discuss the case $k \geq n$ below. In the subsequent Lemma 3, we prove a lower bound on the probability that ALG_{iid} finds a good item.

It is easy to see that the expected value of a good item is at least $\mathbf{E}[\text{UB}_{k+1}]$: Clearly, UB_{k+1} has probability $1/(k+1)$ to select each of the first $k+1$ items. Under the condition that the probability of selecting an item is $1/(k+1)$, by stochastic dominance, the largest-possible expectation of the item's value is $\mathbf{E}[v_i \mid v_i \geq \delta_{k/(k+1)}]$. Additionally, $\mathbf{E}[v_i \mid v_i \geq \delta_x]$ is increasing in x and $k' \geq k+1$. Hence, $\mathbf{E}[\text{UB}_{k+1}] \leq \mathbf{E}[v_i \mid v_i \geq \delta_{(k'-1)/k'}]$, the expected value of a good item.

By Lemma 3, we can conclude that the algorithm finds and selects a good item with probability at least

$$\alpha = 1 - \frac{1 - \frac{1}{2k'}}{1 - \frac{\log(k')}{k'}} \cdot \frac{1}{\sqrt[e]{e^{\frac{k+1}{k'+1}}}} - \frac{1}{2^{k-1}3} .$$

Since a good item has expected value at least $\mathbf{E}[\text{UB}_{k+1}]$ the approximation factor is at least α . Note that $\alpha = 1 - \frac{1}{\sqrt[e]{e}} - o(1)$, where the asymptotics is in $k = \min(n, k)$.

Finally, let us briefly discuss the case $k \geq n$. We can restrict ALG_{iid} to n tests and apply the same analysis, where n replaces k . Since $\mathbf{E}[\text{UB}_{n+1}] = \mathbf{E}[\text{UB}_n] \geq \mathbf{E}[\text{OPT}]$, the approximation guarantee follows, with asymptotics in $n = \min(n, k)$. \square

Lemma 3. The probability that ALG_{iid} runs out of tests before finding a good item can be upper bounded by

$$\frac{1 - \frac{1}{2k'}}{1 - \frac{\log(k')}{k'}} \cdot \frac{1}{\sqrt[e]{e^{\frac{k+1}{k'+1}}}} + \frac{1}{2^{k-1}3} .$$

Proof. The sequence of tests can be seen as a sequence of Benoulli trials. ALG_{id} can run k tests, and we are looking for a success run of length $r = \log_2(k')$ in a sequence of k Benoulli trials. This implies that for some item we have succeeded to verify that it is a good item. To avoid trivialities, we assume $r > 1$. Each trial has a success probability of $1/2$. [Feller, 1957, Volume 1, page 325] observes that the probability of no success run of length r is given by

$$q = A_1 + A_2 + \dots + A_r,$$

where

$$A_1 = \frac{1 - \frac{1}{2}x}{(r+1-rx) \cdot \frac{1}{2}} \cdot \frac{1}{x^{k+1}} \quad \text{and} \quad |A_i| \leq \frac{4}{2^{k+1}3^r},$$

for all $i = 2, \dots, r$. Here, x is the root with smallest absolute value of $f(y) = 1 - y + \frac{1}{2^{r+1}} \cdot y^{r+1}$. The unique positive root of $f(y)$ that is different from 2 happens to be the one with smallest absolute value. In the full version we show that for this root $1 + \frac{1}{2k'} \leq x \leq 1 + \frac{1}{k'}$. This allows to conclude

$$\begin{aligned} q &= A_1 + (r-1) \frac{4}{2^{k+1}3^r} \\ &= \frac{1 - \frac{1}{2}x}{(r+1-rx) \cdot \frac{1}{2}} \cdot \frac{1}{x^{k+1}} + \frac{r-1}{2^{k-1}3^r} \\ &\leq \frac{1 - \frac{1}{2k'}}{1 - \frac{r}{k'}} \cdot \frac{1}{(1 + \frac{1}{2k'})^{k+1}} + \frac{1}{2^{k-1}3} \\ &\leq \frac{1 - \frac{1}{2k'}}{1 - \frac{\log(k')}{k'}} \cdot \frac{1}{(1 + \frac{1}{2k'})^{(2k'+1) \frac{k+1}{2k'+1}}} + \frac{1}{2^{k-1}3} \\ &\leq \frac{1 - \frac{1}{2k'}}{1 - \frac{\log(k')}{k'}} \cdot \frac{1}{e^{\frac{k+1}{2k'+1}}} + \frac{1}{2^{k-1}3} \\ &\leq \frac{1 - \frac{1}{2k'}}{1 - \frac{\log(k')}{k'}} \cdot \frac{1}{\sqrt{e^{\frac{k+1}{k'+1}}}} + \frac{1}{2^{k-1}3}, \end{aligned}$$

for all $k > 1$. Here, we used $(1 + 1/x)^{x+1} > e$ in the second to last inequality. \square

3 General Distributions

Our main result in this section is an algorithm that has a constant approximation ratio for non-identical, independent distributions D_i . We again use OPT to denote the optimal probing strategy and $\mathbf{E}[\text{OPT}]$ for the expected value of the chosen item. As in the previous section, we first concentrate on the case $k < n$. We again overestimate $\mathbf{E}[\text{OPT}]$ by an upper bound resulting from a probing strategy in the standard probing model where tests reveal exact realizations. In particular, $\mathbf{E}[\text{UB}_\ell]$ is the expected value from the optimal strategy that can *adaptively* inspect $\ell \leq n$ of the items, learns their *exact* realization and then picks the best realization it has seen.

Again, $\mathbf{E}[\text{UB}_{k+1}] \geq \mathbf{E}[\text{OPT}]$ by similar arguments as above. As UB_{k+1} can adaptively learn the exact realizations, its decisions are clearly much more informed than those of OPT . Moreover, when adaptively inspecting the exact value of k items, we might eventually want to resort to an uninspected item with the maximum expected value (if all realizations are below that expectation). Instead, for UB_{k+1} we

can also learn the realization of the last uninspected item and then pick the best one among the $k+1$ items seen. This is clearly stronger than $\mathbf{E}[\text{OPT}]$. Our main result is to provide an algorithm with constant approximation w.r.t. $\mathbf{E}[\text{UB}_{k+1}]$.

Note that this also implies an $\Omega(1)$ -approximation for $k > n-1$. With $n-1$ tests we achieve a $\Omega(1)$ -approximation with respect to $\mathbf{E}[\text{UB}_n]$, which always learns and selects the best item – a trivial upper bound on what can be achieved with any kind of testing. As such, we can simply run our strategy using $n-1$ tests and ignore the additional $k-n+1$ tests. For the rest of the section, we therefore concentrate on the case $k \leq n-1$.

As a first step, we apply a reduction to concentrate on a smaller number relevant items. We do so using the following result from the literature, rephrased for our needs.

Theorem 4 (Theorem 2 in [Asadpour and Nazerzadeh, 2016]). *There exists an algorithm that, given $k \in \mathbb{N}$, in polynomial time selects a subset $N_{k+1} \subseteq N$ of the items with $|N_{k+1}| = k+1$ and*

$$\mathbf{E} \left[\max_{i \in N_{k+1}} v_i \right] \geq \left(1 - \frac{1}{e}\right) \cdot \mathbf{E}[\text{UB}_{k+1}].$$

In contrast to Asadpour and Nazerzadeh we have direct access to the distributions. By inspecting their analysis, we see that this implies the stated approximation without reduction by an $\varepsilon > 0$.

Now given the subset N_{k+1} , we apply a further random sampling step – we pick a uniform random subset $N' \subseteq N_{k+1}$ of k' items. Clearly, we sample the item with the best realization from N_{k+1} with probability $k'/(k+1)$. Thus,

$$\mathbf{E} \left[\max_{i \in N'} v_i \right] \geq \frac{k'}{k+1} \cdot \mathbf{E} \left[\max_{i \in N_{k+1}} v_i \right].$$

We choose k' to be smaller than $k+1$ by a large-enough constant factor; choosing k' to be the largest power of 2 that is at most $k/10$ will suffice. For now, assume that k is relatively large so that $k' \in \mathbb{N}$. Then, since $k' \in \Omega(k)$, we get $\mathbf{E}[\max_{i \in N'} v_i] = \Omega(1) \cdot \mathbf{E}[\text{UB}_{k+1}]$. At the same time, since $k \geq 10 \cdot k'$, we have many more tests available than there are relevant items in N' . For convenience, we renumber the items such that $N' = [k'] = \{1, \dots, k'\}$.

It remains to achieve a constant approximation to $\mathbf{E}[\max_{i \in [k']} v_i]$. Let \mathcal{E}_i be the event that i has the largest value of all items in N' . Here, we break ties in order of lower item numbers. We can write

$$\mathbf{E} \left[\max_{i \in [k']} v_i \right] = \sum_{i=1}^{k'} \mathbf{Pr}[\mathcal{E}_i] \cdot \mathbf{E}[v_i | \mathcal{E}_i]. \quad (1)$$

In the following we will use p_i as shorthand for $\mathbf{Pr}[\mathcal{E}_i]$ for all $i \in [k']$. Given explicit representations of the discrete distributions D_i for items in $[k']$, the values p_i can be computed easily in polynomial time¹.

¹For each possible realization v_i of item i , compute the probability that item i has value v_i , all items $j = 1, \dots, i-1$ have a realization $v_j < v_i$, and all items $j = i+1, \dots, k'$ have a realization $v_j \leq v_i$. The product of these numbers is the probability that v_i constitutes the maximum of all realizations. p_i is the sum of probabilities computed for all realizations of item i .

We try to pick each item $i \in [k']$ that realizes to any fixed value above the $1 - \Omega(p_i)$ -quantile with constant probability. Then, with (1) and a similar argument as for identical distributions, we indeed get an $\Omega(1)$ -approximation. Our algorithm again operates sequentially over the items. It considers items $1, \dots, k'$ in arbitrary order, say, in ascending order of their indices. Upon considering item i , it (approximately) checks if v_i realizes above the $1 - p_i$ quantile of D_i . If this check succeeds, it simply selects item i ; otherwise it discards i and proceeds with the next item.

Assuming we could perform the check for the $1 - p_i$ quantile not only approximately but exactly in our model (say, using $\Theta(\log(1/p_i))$ tests), this algorithm would *not* obtain all realizations above the $1 - \Omega(p_i)$ quantile with constant probability for all i ; indeed, we need a specific approximate check. First, p_1 may be arbitrarily close to 1. Then we are unable to guarantee to arrive at item 2 with a constant probability and thereby fail to select v_2 with constant probability when v_2 realizes to a value above the $1 - \Omega(p_2)$ quantile of D_2 . Second, p_1 may be so small that $\Theta(-\log p_1)$ exceeds k , the number of available tests. Then we never select v_1 .

We address both issues by defining

$$q_i = \frac{\max\{p_i, 1/k'\}}{4} \in \Omega(p_i)$$

and using q_i in place of p_i . Lifting values smaller than $1/k'$ to $1/k'$ can be seen as an idea borrowed from the setting of identical distributions. Dividing the resulting probability by 4 makes sure that there is a constant lower bound on the probability that for any given item i the algorithm eventually considers i . A similar idea is used in Bayesian mechanism design [Chawla *et al.*, 2010; Alaei, 2014].

To (approximately) check more easily if v_i realizes above $D_i^{-1}(1 - q_i)$, we round q_i to a power of 2 (with negative exponent). Note that the lowest possible value, that is, $1/(4k')$, is already a power of 2. In general, we define \tilde{q}_i to be the largest power of 2 which is at most q_i . Having arrived at item i , our algorithm tests item i at most $-\log \tilde{q}_i \in \mathbb{N}$ times. As soon as one of the tests is negative, we stop testing item i and continue with the next item; if all tests are positive, we select item i . This concludes the description of our algorithm, which we summarize as ALG_{gen} . For a formal and precise description, see Algorithm 1. Recall that the analysis for the case $k > n - 1$ follows from restricting attention to $\min(k, n)$ tests.

The main result is the following theorem. For most of the proof we assume $k > k_0$ for a suitable constant ($k_0 = 50$ is sufficient), since our analysis relies on concentration bounds and we need to ensure $k' \in \mathbb{N}$. Otherwise, for constant $k \leq k_0$, selecting an item with the best (a priori) expectation ALG_{gen} trivially guarantees a constant-factor approximation. By slight misuse of notation, we use $\mathbf{E}[\text{ALG}_{\text{gen}}]$ to denote the expected value of the item selected by our algorithm.

Theorem 5. ALG_{gen} runs in polynomial time and achieves an expected value of

$$\mathbf{E}[\text{ALG}_{\text{gen}}] \geq \Omega(1) \cdot \mathbf{E}[\text{UB}_{k+1}] .$$

To prove this theorem, we first show the following lemma.

Algorithm 1: ALG_{gen} for General Distributions

Input : Distributions D_1, \dots, D_n over \mathbb{R}_+ , $k \in \mathbb{N}$.

Output: The index of the picked item.

- 1 **if** $k \leq k_0$, **return** $i \in \arg \max_{i \in [n]} \mathbf{E}[v_i]$.
 - 2 Required tests: $k \leftarrow \min(k, n - 1)$.
 - 3 Select set N_{k+1} of items using Theorem 4.
 - 4 $k' \leftarrow 2^{\lfloor \log(k/10) \rfloor}$.
 - 5 Select set N' of k' items from N_{k+1} uniformly at random; w.l.o.g. $N' = [k']$.
 - 6 **for** i **in** $[k']$:
 - 7 \mathcal{E}_i is the event that $\arg \max_{i \in [k']} v_i$ is item i (breaking ties arbitrarily).
 - 8 $q_i \leftarrow \max\{\mathbf{Pr}[\mathcal{E}_i], 1/k'\}/4$.
 - 9 $\tilde{q}_i \leftarrow 2^{\lfloor -\log q_i \rfloor}$.
 - 10 **for** j **in** $[-\log \tilde{q}_i]$:
 - 11 **if** test is available:
 - 12 **test** distribution D_i .
 - 13 **if** negative test result: **break** inner loop.
 - 14 **if** $j = -\log \tilde{q}_i$: **return** i .
-

Lemma 6. Suppose $k > k_0$ and $k' = 2^{\lfloor \log_2 k/10 \rfloor}$. There is a constant $r > 0$ such that, for any $i \in [k']$, the probability that ALG_{gen} arrives at item i with at least $\log k' + 2$ unused tests is at least r .

Proof. It suffices to consider the event that ALG_{gen} arrives at the last item, i.e., item k' , with $\log k' + 2$ unused tests, called \mathcal{F} in the following, and bound its probability from below by a constant. By the union bound, we can write

$$\mathbf{Pr}[\mathcal{F}] \geq 1 - \mathbf{Pr}[\mathcal{F}_1] - \mathbf{Pr}[\mathcal{F}_2] . \tag{2}$$

Here, \mathcal{F}_1 is the event that the algorithm picks any v_i prior to even considering $v_{k'}$. To define \mathcal{F}_2 , we view the tests as independent, unbiased coins and realize *all* of them, even those that are potentially not used by the algorithm. Now \mathcal{F}_2 is the event that among the first $k - \log k'$ tests, fewer than $k' - 1$ have result 0. Indeed, whenever \mathcal{F} does not occur, at least one of \mathcal{F}_1 and \mathcal{F}_2 occurs.

We first consider \mathcal{F}_1 . Note that $\sum_{i \in [k']} p_i = 1$. Since $\max\{p_i, 1/k'\} \leq p_i + 1/k'$ for all $i \in [k']$, it follows that $\sum_{i \in [k']} \max\{p_i, 1/k'\} \leq 2$, so $\sum_{i \in [k']} q_i \leq 1/2$ by definition of q_i . Then, using $\tilde{q}_i \leq q_i$ for all $i \in [k']$, we have $\sum_{i \in [k']} \tilde{q}_i \leq \frac{1}{2}$.

Since the probability that we pick item i is at most \tilde{q}_i (for that to happen, v_i has to realize above the $1 - \tilde{q}_i$ quantile of D_i), again by the union bound, the probability that we pick any item at all is at most $1/2$. Therefore

$$\mathbf{Pr}[\mathcal{F}_1] \leq 1/2. \tag{3}$$

It remains to bound $\mathbf{Pr}[\mathcal{F}_2]$ from above and away from $1/2$. Towards applying a Chernoff bound, for $i \in [k]$ we define X_i to be the Bernoulli variable whose value is precisely the result of the i -th test. Then X_1, \dots, X_k are independent, and $X := \sum_{i=1}^{4k/5} X_i$ has expectation $2k/5$. We get

$$\mathbf{Pr}[\mathcal{F}_2] < \mathbf{Pr}\left[X \geq \frac{7k}{10}\right] = \mathbf{Pr}\left[X \geq \left(1 + \frac{3}{4}\right) \cdot \mathbf{E}[X]\right]$$

$$\leq \exp\left(-\frac{\left(\frac{3}{4}\right)^2 \cdot \frac{2k}{5}}{3}\right) < \frac{1}{4}, \quad (4)$$

where the first inequality follows from the fact that, when \mathcal{F}_2 occurs, we must not have more than $k' \leq k/10$ tests with result 0 among the first $4k/5 < k - (\log k + 2)$, so $X \geq 7k/10$ follows. The second inequality follows by plugging in the expected value of X , and the third one by applying the corresponding one-sided Chernoff bound. The last inequality follows because $k > k_0 = 50$.

The claim follows by using Inequalities (3) and (4) in (2). \square

With this lemma at hand, we can prove the main theorem.

Proof of Theorem 5. First consider the case $k \leq k_0 = 50$. Let $i^* \in \arg \max_{i \in [n]} \mathbf{E}[v_i]$ be the returned index. Here we overestimate $\mathbf{E}[\text{UB}_{k+1}]$ by selecting all $k+1$ observed realizations and obtaining the sum of the values. For this objective, it is trivially optimal to select the set I_{k+1}^* which we define to be the set of $k+1$ items with highest expectation. Since ALG_{gen} selects the single item with highest expectation, it recovers at least

$$\mathbf{E}[v_{i^*}] \geq \frac{1}{k+1} \cdot \sum_{i \in I_{k+1}^*} \mathbf{E}[v_i] \geq \frac{1}{k_0+1} \cdot \mathbf{E}[\text{UB}_{k+1}],$$

implying our claim.

Now consider the case $k > k_0 = 50$. By Lemma 6, there exists a constant $r > 0$ such that with probability at least r , for any given item i , the algorithm arrives at i with at least $-\log \tilde{q}_i \leq \log k' + 2$ unused tests. Hence,

$$\begin{aligned} \mathbf{E}[\text{ALG}_{\text{gen}}] &\geq \sum_{i=1}^{k'} r \cdot \Pr[v_i \geq D_i^{-1}(1 - \tilde{q}_i)] \\ &\quad \cdot \mathbf{E}[v_i \mid v_i \geq D_i^{-1}(1 - \tilde{q}_i)] \\ &= r \cdot \sum_{i=1}^{k'} \tilde{q}_i \cdot \mathbf{E}[v_i \mid v_i \geq D_i^{-1}(1 - \tilde{q}_i)] \\ &\geq r \cdot \sum_{i=1}^{k'} \frac{p_i}{8} \cdot \mathbf{E}\left[v_i \mid v_i \geq D_i^{-1}\left(1 - \frac{p_i}{8}\right)\right] \\ &\geq r \cdot \sum_{i=1}^{k'} \frac{1}{8} \cdot \Pr[\mathcal{E}_i] \cdot \mathbf{E}[v_i \mid \mathcal{E}_i] = \frac{r}{8} \cdot \mathbf{E}\left[\max_{i \in [k']} v_i\right]. \end{aligned} \quad (5)$$

In the first step, we use the independence of arriving at item i and its realization v_i . The second step uses the definition of D_i . The third step follows by monotonicity of $x \cdot \mathbf{E}[v_i \mid v_i \geq D_i^{-1}(1 - x)]$ as a function of x and $\tilde{q}_i \geq q_i/2 \geq p_i/8$. In the fourth step, we use that $p_i = \Pr[\mathcal{E}_i]/8$ for the first part and stochastic dominance to compare the two expected values. The fifth step uses the definition of \mathcal{E}_i .

Recalling the discussion of Theorem 4 and the random sampling step, we note

$$\mathbf{E}\left[\max_{i \in [k']} v_i\right] \geq \left(1 - \frac{1}{e}\right) \cdot \frac{k'}{k+1} \cdot \mathbf{E}[\text{UB}_{k+1}]. \quad (6)$$

The ratio follows by combining (5) and (6) with $k' \geq k/20$.

The running time of ALG_{gen} is dominated by applying the algorithm by [Asadpour and Nazerzadeh, 2016] and computing the values $p_i = \Pr[\mathcal{E}_i]$. Both steps run in time polynomial in the input size. \square

4 Sequential Probing

We consider a sequential scenario of the probing problem, in which tests for the same item must be conducted consecutively, and items must be tested in a given order. This restricts the algorithm and the optimal probing strategy in two ways.

First, if a test series for an item j is stopped, j cannot be tested anymore. This restriction is very natural in many practical applications such as the hiring process discussed above. Typically, a candidate cannot be interviewed again after the job interview is finished. Additional applications for this assumption include flat viewings, inspection of second hand articles, or test series with time consuming test setups.

Second, we restrict all testing to adhere to a fixed ordering of the items, i.e., the order of items, in which they can be tested, is given upfront. Note that this constraint has no bite for the i.i.d. scenario.

Interestingly, all our results from the previous sections directly carry over to the sequential probing problem. Both our algorithms test each item using a single consecutive test series and can be applied when given any fixed order of items.

Observation 7. *Algorithms ALG_{iid} and ALG_{gen} are polynomial-time algorithms with constant approximation factor for the sequential probing problem.*

As the main result in this section, we show how to compute the optimal probing strategy in polynomial time. For a proof, we refer to the full version.

Theorem 8. *The optimal strategy in the sequential probing problem can be computed in polynomial time.*

5 Conclusion

An arguably unrealistic assumption in existing stochastic probing models is that any test reveals full information about the probed item. We initiate research that addresses this shortcoming and introduce a first natural coarse-to-fine-grained model. For this model, we provide polynomial-time algorithms with constant approximation factors for both i.i.d. values and general independent values.

It is straightforward to see that our results extend (with different constants in the approximation guarantees) to the case when a test separates realizations in the top- c -quantile and the bottom $(1 - c)$ -quantile, for constant $c \in (0, 1)$. It is an open problem to analyze algorithms when c is not constant.

Another interesting direction is to consider correlated random variables. For related problems in online stopping, correlations have only recently started to be considered and tend to be technically very challenging.

More generally, there is potential for extending the rich theory on standard probing models towards tests that yield only limited information, including cases in which the learner can choose a set of items instead of a single one.

Another research direction are hardness results for stochastic probing problems with and without increasing precision.

References

- [Adamczyk *et al.*, 2014] Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 29–40, 2014.
- [Agrawal *et al.*, 2020] S. Agrawal, J. Sethuraman, and X. Zhang. On optimal ordering in the optimal stopping problem. In *ACM Conference on Economics and Computation (EC)*, pages 187–188, 2020.
- [Alaei, 2014] Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM J. Comput.*, 43(2):930–972, 2014.
- [Asadpour and Nazerzadeh, 2016] Arash Asadpour and Hamid Nazerzadeh. Maximizing stochastic monotone submodular functions. *Manag. Sci.*, 62(8):2374–2391, 2016.
- [Asadpour *et al.*, 2008] Arash Asadpour, Hamid Nazerzadeh, and Amin Saberi. Stochastic submodular maximization. In *Workshop on Internet and Network Economics (WINE)*, pages 477–489, 2008.
- [Bansal *et al.*, 2012] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.
- [Beyhaghi and Kleinberg, 2019] Hedyeh Beyhaghi and Robert Kleinberg. Pandora’s problem with nonobligatory inspection. In *ACM Conference on Economics and Computation (EC)*, pages 131–132, 2019.
- [Bhalgat *et al.*, 2011] Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. Improved approximation results for stochastic knapsack problems. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1647–1665, 2011.
- [Chawla *et al.*, 2010] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *ACM Symposium on Theory of Computing (STOC)*, pages 311–320, 2010.
- [Chen *et al.*, 2009] Ning Chen, Nicole Immorlica, Anna R. Karlin, Mohammad Mahdian, and Atri Rudra. Approximating matches made in heaven. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 266–278, 2009.
- [Correa *et al.*, 2018] José R. Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Recent developments in prophet inequalities. *SIGecom Exch.*, 17(1):61–70, 2018.
- [Dean *et al.*, 2008] Brian C. Dean, Michel X. Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.*, 33(4):945–964, 2008.
- [Feller, 1957] William Feller. *An introduction to probability theory and its applications*. John Wiley & Sons, Inc., 1957.
- [Fu *et al.*, 2018] Hao Fu, Jian Li, and Pan Xu. A PTAS for a class of stochastic dynamic programs. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 56:1–56:14, 2018.
- [Gittins, 1979] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, pages 148–177, 1979.
- [Gupta and Nagarajan, 2013] Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 205–216, 2013.
- [Gupta *et al.*, 2016] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and adaptivity gaps for stochastic probing. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1731–1747, 2016.
- [Gupta *et al.*, 2017] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and XOS functions. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1688–1702. SIAM, 2017.
- [Gupta *et al.*, 2019] Anupam Gupta, Haotian Jiang, Ziv Scully, and Sahil Singla. The markovian price of information. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 233–246, 2019.
- [Krengel and Sucheston, 1977] U. Krengel and L. Sucheston. Semiamarts and finite values. *Bull. Amer. Math. Soc.*, 83:745–747, 1977.
- [Lucier, 2017] Brendan Lucier. An economic view of prophet inequalities. *SIGecom Exch.*, 16(1):24–47, 2017.
- [Ma, 2018] Will Ma. Improvements and generalizations of stochastic knapsack and markovian bandits approximation algorithms. *Math. Oper. Res.*, 43(3):789–812, 2018.
- [Singla, 2018] Sahil Singla. The price of information in combinatorial optimization. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2523–2532. SIAM, 2018.
- [Weitzman, 1979] Martin L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47:641–654, 1979.