Identifying Approximate Minimizers under Stochastic Uncertainty

Hessa Al-Thani* Viswanath Nagarajan †

Abstract

We study a fundamental stochastic selection problem involving n independent random variables, each of which can be queried at some cost. Given a tolerance level δ, the goal is to find a value that is δ-approximately minimum (or maximum) over all the random variables, at minimum expected cost. A solution to this problem is an adaptive sequence of queries, where the choice of the next query may depend on previously-observed values. Two variants arise, depending on whether the goal is to find a δ-minimum value or a δ-minimizer. When all query costs are uniform, we provide a 4-approximation algorithm for both variants. When query costs are non-uniform, we provide a 5.83-approximation algorithm for the δ-minimum value and a 7.47-approximation for the δ-minimizer. All our algorithms rely on non-adaptive policies (that perform a fixed sequence of queries), so we also upper bound the corresponding "adaptivity" gaps. Our analysis relates the stopping probabilities in the algorithm and optimal policies, where a key step is in proving and using certain stochastic dominance properties.

1 Introduction

We study a natural stochastic selection problem that involves querying a set of random variables so as to identify their minimum (or maximum) value within a desired precision. Consider a car manufacturer who wants to chose one design from n options so as to optimize some attribute (e.g., maximum velocity or energy efficiency). Each option i corresponds to an attribute value X_i which is uncertain and drawn from a known probability distribution. It is possible to determine the *exact* value of X_i by further testing— but this incurs some cost c_i. Identifying the exact minimum (or maximum) value among the X_is might be too expensive. Instead, our goal is to identify an *approximately* minimum (or maximum) value, within a prescribed tolerance level. For example, we might be satisfied with a value (and corresponding option) that is within 10% of the true minimum. The objective is to minimize the expected cost. In this paper, we provide the first constant-factor approximation algorithm for this problem.

Our problem is related to two lines of work: *stochastic combinatorial optimization* and *optimization under explorable uncertainty*. In stochastic combinatorial optimization, a solution makes selections incrementally and adaptively (i.e., the next selection can depend on previously observed random outcomes). An optimal solution here may even require exponential space to describe. Nevertheless, there has been much recent success in obtaining (efficient) approximation algorithms for such problems, see e.g., [DGV08, GM07, BGL+12, GKNR15, GGHK18, INvdZ16, JLLS20, HKP21, HLS24]. Optimization problems under explorable uncertainty involve querying values drawn from

*Funded by Graduate Sponsorship Research Award from the Qatar Research and Development Institute.

†Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, USA. Research supported in part by NSF grant CCF-2418495.

known intervals in order to identify a minimizer. Typically, these results focus on the *competitive ratio*, which relates the algorithm's (expected) query cost to the optimum query-cost in hindsight, see e.g., [Kah91, CHdT21, FMP+00, EHK+08, MMS17, EHK16, BDE+21, MS23]. In particular, for the problem of finding an *exact* minimizer among n intervals, [Kah91] obtained a 2-competitive algorithm in the adversarial setting and [CHdT21] obtained a 1.45-approximation algorithm in the stochastic setting. The problem we study is a significant generalization of the stochastic exact minimizer problem [CHdT21].

1.1 Problem Definition

In the *stochastic minimum query* (SMQ) problem, there are n independent discrete random variables $X_1, ..., X_n$ that lie in intervals $I_1, ..., I_n$ respectively. The random variables (r.v.s) may be negative. We assume that each interval is bounded and closed, i.e., $I_j = [\ell_j, r_j]$ for each $j \in [n]$. We also assume (without loss of generality) that each r.v. has non-zero probability at the endpoints of its interval, i.e., $\Pr[X_j = \ell_j] > 0$ and $\Pr[X_j = r_j] > 0$ for each $j \in [n]$.[1] We will use the terms random variable (r.v.) and interval interchangeably. The exact value of any r.v. X_j can only be determined by querying it, which incurs some cost $c_j \geq 0$. Additionally, we are given a "precision" value $\delta \geq 0$, where the goal is to identify the minimum value over *all* r.v.s up to an additive precision of δ. Formally, if $\mathsf{MIN} = \min_{j=1}^n X_j$ then we want to find a deterministic value VAL such that $\mathsf{MIN} \leq \mathsf{VAL} \leq \mathsf{MIN} + \delta$. Such a value VAL is called a δ-*minimum* value. The objective in SMQ is to minimize the expected cost of the queried intervals. Note that it may be sufficient to probe only a (small) subset of intervals before stopping.

We also consider a related, but harder, problem where the goal is to *identify* some δ-minimizer $i^* \in [n]$, i.e., an interval that satisfies $X_{i^*} \leq \mathsf{MIN} + \delta$. We refer to this problem as *stochastic minimum query for identification* (SMQI). If a δ-minimum value is found then it also provides a δ-minimizer (see §1.4). However, the converse is not true. So, an SMQI solution may return an un-queried a δ-minimizer i^* without determining a δ-minimum value.

Although our formulation above uses *additive* precision (we aim to find a value that is at most $\mathsf{MIN} + \delta$), we can also handle *multiplicative* precision where the goal is to find a value that is at most $\alpha \cdot \mathsf{MIN}$. This just requires a simple logarithmic transformation; see Appendix A. We can also handle the goal of finding the *maximum* value by working with negated r.v.s $\{-X_i\}_{i=1}^n$.

Throughout, we use $N := [n] = \{1, 2, \ldots, n\}$ to denote the index set of the r.v.s.

Adaptive and Non-adaptive policies Any solution to SMQ involves querying r.v.s sequentially until a δ-minimum value is found. In general, the sequence of queries may depend on the realizations of previously queried r.v.s. We refer to such solutions as *adaptive* policies. Formally, such a solution can be described as a decision tree where each node corresponds to the next r.v. to query and the branches out of a node represent the realization of the queried r.v. *Non-adaptive* policies are a special class of solutions where the sequence of queries is fixed upfront: the policy then performs queries in this order until a δ-minimum value is found. A central notion in stochastic optimization is the *adaptivity gap* [DGV08], which is the worst-case ratio between the optimal non-adaptive value and the optimal adaptive value. All our algorithms produce non-adaptive policies and hence also bound the adaptivity gap.

[1] Otherwise, we can just work with a smaller interval representing the same r.v.

1.2 Results

Our first result is on the SMQ problem with unit costs, for which we provide a 4-approximation algorithm. Moreover, we achieve this result via a non-adaptive policy, which also proves an upper bound of 4 on the adaptivity gap. This algorithm relies on combining two natural policies. The first policy simply queries the r.v. with the smallest left-endpoint. The second policy queries the r.v. that maximizes the probability of stopping in the very next step. When used in isolation, both these policies have unbounded approximation ratios. However, interleaving the two policies leads to a constant-factor approximation algorithm.

We also consider the (harder) unit-cost SMQI problem and show that the same policy leads to a 4-approximation algorithm: the only change is in the criterion to stop, which is now more relaxed. While the algorithm is the same as SMQ, the analysis for SMQI is significantly more complex due to the new stopping criterion, which allows us to infer a δ-minimizer i^* even when it has not been queried. Specifically, we prove a stochastic dominance property between r.v.s in our algorithm and the optimum (conditioned on the SMQ stopping criterion not occurring), and use this in relating the SMQI stopping-probability in the algorithm and the optimum.

Our next result is for the SMQ problem with non-uniform costs. We obtain a constant-factor approximation again, with a slightly worse ratio of 5.83. This is based on combining ideas from the unit-cost algorithm with a "power-of-two" approach. In particular, the algorithm proceeds in several iterations, where the i^{th} iteration incurs cost roughly 2^i. In each iteration i, the algorithm selects a subset of r.v.s with cost $O(2^i)$ based on the following two criteria (i) smallest left-endpoint and (ii) maximum probability of stopping in one step. In order to select the r.v.s for criterion (ii) we need to use a PTAS for an appropriate version of the knapsack problem.

Finally, we consider the SMQI problem with non-uniform costs. Directly using the SMQ algorithm for SMQI (as in the unit-cost case) does not work here: it leads to a poor approximation ratio. However, a modification of the SMQ algorithm works. Specifically, we modify step (i) above: instead of just selecting a prefix of intervals with the smallest left-endpoints, we select an "almost prefix" set by skipping some expensive intervals. We show that this approach leads to an approximation ratio of 7.47, which is slightly worse than what we obtain for SMQ. The analysis combines aspects of unit-cost SMQI and SMQ with non-uniform costs.

1.3 Related Work

Computing an approximately minimum or maximum value by querying a set of random variables is a central question in stochastic optimization. Most of the prior works on this topic have focused on *budgeted* variants. Here, one wants to select a subset of queries of total cost within some budget so as to maximize or minimize the value *among the queried* r.v.s. The results for the minimization and maximization versions are drastically different. A $1 - \frac{1}{e}$ approximation algorithm for the budgeted max-value problem follows from results on stochastic submodular maximization [AN16]; more complex "budget" constraints can also be handled in this setting [ASW16, GNS17]. These results also bound the adaptivity gap. In addition, PTASes are known for non-adaptive and adaptive versions of budgeted max-value [FLX18, SS21]. For the budgeted min-value problem, it is known that the adaptivity gap is unbounded and results for the non-adaptive and adaptive versions are based on entirely different techniques. [GGM10] obtained a bi-criteria approximation algorithm for the non-adaptive problem (the queried subset must be fixed upfront) that achieves a $1 + \epsilon$ approximation to the optimal value while exceeding the budget by at most an $O(\log \log m)$ factor, where each r.v.

takes an integer value in the range $\{0, 1, \ldots, m\}$. Subsequently, [WGW22] studied the adaptive setting (the queried subset may depend on observed realizations) and obtained a 4-approximation while exceeding the budget by at most an $O(\log \log m)$ factor. In contrast to these results, the goal in SMQ is to achieve a value close to the true minimum/maximum taken over *all* random variables $X_1, X_2, \ldots, X_n$ (not just the queried ones). Moreover, we want to find an approximately min/max value with probability one, as opposed to optimizing the expected min/max value.

A different formulation of the minimum-element problem is studied in [Sin18]: this combines the query-cost and the value of the minimum-queried element into a single objective. They obtain an exact algorithm for this setting, which also extends to a wider class of constrained problems.

Closely related to our work, [CHdT21] studied the SMQI problem with exact precision, i.e., $\delta = 0$. In particular, their goal is to identify an *interval* that is an exact minimizer. [CHdT21] obtained a 1.45-approximation ratio for general query costs. The SMQI problem that we study allows for arbitrary precision δ, and is significantly more complex than the setting in [CHdT21]. One indication of the difficulty of handling arbitrary δ is that the simpler SMQ problem with $\delta = 0$ (where we want to find the exact minimum value) admits a straightforward exact algorithm that queries by increasing left-endpoint; however, this algorithm has an unbounded ratio for SMQ with arbitrary δ (see §2 for an example).

As mentioned earlier, the SMQ problem is also related to optimization problems under explorable uncertainty. Apart from the minimum-value problem [Kah91], various other problems like computing the median [FMP+00], minimum spanning tree [EHK+08, MMS17] and set selection [EHK16, BDE+21, MS23] have been studied in this setting. The key difference from our work is that these results focus on the competitive ratio. In contrast, we compare to the optimal policy that is limited in the same manner as the algorithm. We note that there is an $\tilde{\Omega}(n)$ lower bound on the competitive ratio for SMQ and SMQI; see Appendix B. Our results show that much better (constant) approximation ratios are achievable for SMQ and SMQI in the stochastic setting, relative to an optimal policy.

1.4 Preliminaries

Stopping rule for SMQ. Even without querying any interval, we know that the minimum value is at most $R := \min_{i \in N}\{r_i\}$, the minimum right-endpoint. In order to simplify notation, we incorporate this information using a dummy r.v. $X_0 = [R, R]$ that is queried at the start of any policy and incurs no cost. We now formally define the condition under which a policy for SMQ is allowed to stop. We will refer to the partial observations at any point in a policy (i.e., values of r.v.s queried so far) as the *state*. Consider any state, given by a subset $S \subseteq N$ of queried r.v.s along with their observations $\{x_i\}_{i \in S}$. The minimum observed value is $\min_{i \in S} x_i$ and the minimum possible value among the un-queried r.v.s is $\min_{j \in N \setminus S} \ell_j$. The stopping criterion is:

$$\min_{i \in S} x_i \quad \leq \quad \min_{j \in N \setminus S} \ell_j \, + \, \delta. \tag{1}$$

If this criterion is met then $\mathsf{VAL} = \min_{i \in S} x_i$ is guaranteed to satisfy $\mathsf{MIN} \leq \mathsf{VAL} \leq \mathsf{MIN} + \delta$, where $\mathsf{MIN} = \min_{j \in N} X_j$. Also, $\arg\min_{i \in S} x_i$ is a δ-minimizer. On the other hand, if this criterion is not met then there is no value v that guarantees $\mathsf{MIN} \leq v \leq \mathsf{MIN} + \delta$: there is a non-zero probability that the minimum value is $\min_{j \in N \setminus S} \ell_j$ or $\min_{i \in S} x_i$ (and these values are more than δ apart). So,

Proposition 1.1. *A policy for* SMQ *can stop if and only if criterion* (1) *holds.*

The stopping rule for SMQI is described in §2.2. An SMQI policy can stop either due to the SMQ stopping rule (above) or by inferring an un-queried interval i^* as a δ-minimizer.

Adaptivity gap. We show that the adaptivity gap for the SMQ problem is more than one: so adaptive policies may indeed perform better. This example also builds some intuition for the problem. Consider an instance $\mathcal{I}$ with three intervals as shown in Figure 1. In particular, $X_1 \in \{0, 3, \infty\}$, $X_2 \in \{1, \infty\}$, $X_3 \in \{2, \infty\}$ and $\delta = 1$. Let $\Pr(X_1 = 0) = \frac{1}{3}, \Pr(X_1 = 3) = \frac{1}{3}, \Pr(X_1 = \infty) = \frac{1}{3}, \Pr(X_2 = 1) = \epsilon, \Pr(X_2 = \infty) = 1 - \epsilon, \Pr(X_3 = 2) = 1 - \epsilon, \Pr(X_3 = \infty) = \epsilon$. An adaptive policy is shown in Figure 2, which has cost at most $1 + \frac{2}{3} + \frac{\epsilon}{3} = \frac{5+\epsilon}{3}$. By a case analysis (see Appendix C) the best non-adaptive cost is $\min\left\{\frac{6-\epsilon}{3}, \frac{5+2\epsilon}{3}\right\}$. Setting $\epsilon = \frac{1}{3}$, we obtain an adaptivity gap of $\frac{17}{16}$. We can also modify this instance slightly to get a worse adaptivity gap of $\frac{12}{11}$.

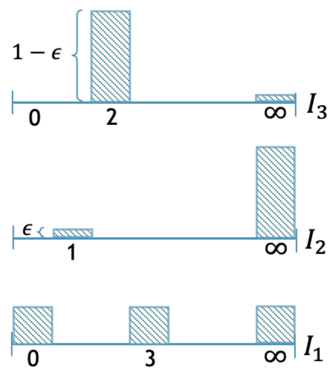


Figure 1: Adaptivity gap instance for SMQ.

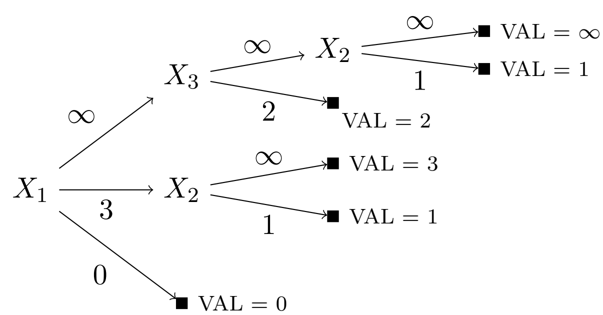


Figure 2: Optimal adaptive policy

Fixed threshold problem. In our analysis, we relate SMQ to the following simpler problem. Given n r.v.s $\{X_i : i \in N\}$ with costs as before, a *fixed* threshold θ and budget k, find a policy having query-cost at most k that maximizes the probability of observing a realization less than θ. A useful property of this fixed threshold problem is that it has adaptivity gap one; see Appendix D.

Proposition 1.2. *Consider any instance of the fixed threshold problem. Let V^* and F^* denote the maximum success probabilities over adaptive and non-adaptive policies respectively. Then, $V^* = F^*$*

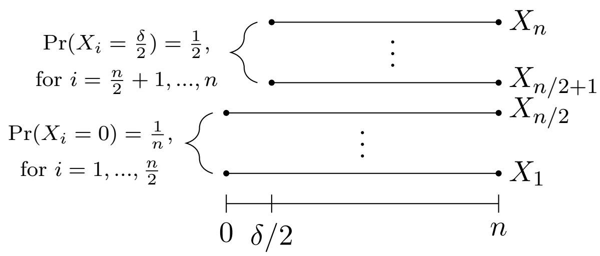
2 Algorithm for Unit Costs

Before presenting our algorithm, we discuss two simple greedy policies and show why they fail to achieve a good approximation.

1. A natural approach is to select intervals by increasing left-endpoint. Indeed, [Kah91] shows that this algorithm is optimal when $\delta = 0$, even in an online setting (with open intervals). Consider the instance with two types of intervals as shown in Figure 3. The r.v.s $X_1, \ldots, X_{n/2}$ are identically distributed with $X_i = 0$ w.p. $\frac{1}{n}$ and $X_i = n$ otherwise. The remaining r.v.s $X_{n/2+1}, \ldots, X_n$ are identically distributed with $X_i = \frac{\delta}{2}$ w.p. $\frac{1}{2}$ and $X_i = n$ otherwise. The greedy policy queries r.v.s in the order $1, 2, \ldots, n$, resulting in an expected cost of $\Omega(n)$ as it can stop only when it observes a "low" realization for some r.v. However, the policy that

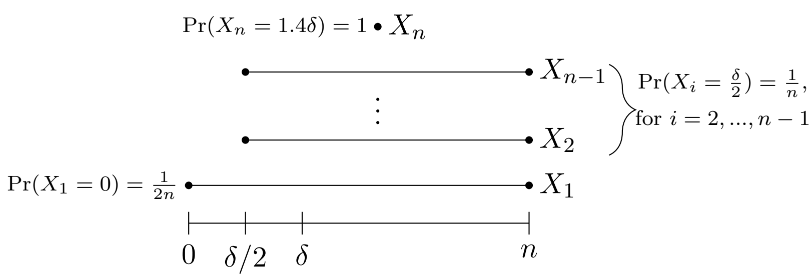
probes in the reverse order $n, n-1, \ldots, 1$ has constant expected cost: the policy can stop upon observing *any* "low" realization (even if a value of $\delta/2$ is observed, it is guaranteed to be within δ of the true minimum). So the approximation ratio of this greedy policy is $\Omega(n)$.

Figure 3: Bad example for greedy by left-endpoint.



2. A different greedy policy (based on the instance in Figure 3) is to always select the interval that maximizes the likelihood of stopping in one step. Now consider another instance with three types of intervals; see Figure 4. The r.v. X_n is always 1.4δ. The r.v. X_1 takes value 0 w.p. $\frac{1}{2n}$ and has value n otherwise. The remaining r.v.s $X_2, \ldots, X_{n-1}$ are identically distributed with $X_i = \frac{\delta}{2}$ w.p. $\frac{1}{n}$ and $X_i = n$ otherwise. As long as X_1 is not queried, the probability of stopping (in one step) is as follows: $\frac{1}{2n}$ for X_1, $\frac{1}{n}$ for $X_2, \ldots, X_{n-1}$ and zero for X_n. So this greedy policy will query in the order $2, 3, \ldots, n-1, 1, n$ resulting in an $\Omega(n)$ expected cost. On the other hand, querying the r.v.s X_1 and X_n guarantees that the policy can stop. So the optimal cost is at most 2, implying an $\Omega(n)$ approximation ratio.

Figure 4: Bad example for greedy by stopping probability.



Our approach is to interleave the above two greedy criteria. In particular, each iteration of our algorithm makes two queries: the interval with the smallest left-endpoint and the interval that maximizes the probability of stopping in one step. We will show that this leads to a constant-factor approximation. We first re-number intervals by increasing order of their left-endpoint, i.e., $\ell_1 \le \ell_2 \le \cdots \le \ell_n$. For each $k \in N$, let $\theta_k := \ell_{k+1} + \delta$. Algorithm 1 describes our algorithm formally.

Equivalently, we can view Algorithm 1 as first computing the permutation π (without querying) and then performing queries in the order given by π until the stopping criterion is met. Note that Algorithm 1 is non-adaptive because it uses observations only to determine when to stop. So, our analysis also upper bounds the adaptivity gap.

We overload notation slightly and use π to also denote the non-adaptive policy given in Algorithm 1. Note that each *iteration* in this policy involves *two* queries. We use σ to denote the optimal (adaptive) policy. Let $c_{exp}(\pi)$ and $c_{exp}(\sigma)$ denote the expected number of queries in policies π and

Algorithm 1 Non-Adaptive Double Greedy

1: Let $\ell^* = \min_{i\in N} \ell_i$, $m^* = R := \min_{i=1}^{n} r_i$, and $\pi \leftarrow \emptyset$.
2: **for** $j = 1, \ldots, n$ **do** ▷ iterations
3: Query interval j (if not already in π).
4: Query interval $b(j) = \text{argmax}_{i\in N\setminus(\pi\circ j)} \Pr[X_i \leq \theta_j]$.
5: Update list $\pi \leftarrow \pi \circ j \circ b(j)$. ▷ skip j if it was already in π
6: Update $m^* = \min\{m^*, X_j, X_{b(j)}\}$ and $\ell^* = \min_{i\in N\setminus\pi}\{\ell_i\}$.
7: **if** $m^* - \ell^* \leq \delta$ **then** stop.

σ, respectively. The key step in the analysis is to relate the termination probabilities in these two policies, formalized below.

Lemma 2.1. *For any $k \geq 1$, we have*

$$\Pr[\sigma \textit{ finishes in } k \textit{ queries}] \leq \Pr[\pi \textit{ finishes in } 2k \textit{ iterations}].$$

We will prove this lemma in the next subsection. First, we complete the analysis using this.

Theorem 2.2. *We have $c_{exp}(\pi) \leq 4 \cdot c_{exp}(\sigma)$.*

Proof. TOPROVE 0 □

2.1 Proof of Key Lemma

We now prove Lemma 2.1. Fix any $k \geq 1$ and define threshold $\theta := \theta_k = \ell_{k+1} + \delta$.

Let $T^* \subseteq N$ denote the optimal solution to the non-adaptive "fixed threshold" problem:

$$\max_{T\subseteq N, |T|\leq k} \quad \Pr\left[\min_{i\in T} X_i \leq \theta\right]. \tag{2}$$

We then proceed in two steps, as follows.

$$\Pr[\sigma \text{ finishes in } k \text{ queries}] \leq \Pr\left[\min_{i\in T^*} X_i \leq \theta\right] \leq \Pr[\pi \text{ finishes in } 2k \text{ iterations}]$$

The first inequality is shown in Lemma 2.3: this uses the fact that the fixed-threshold problem has adaptivity gap one (Proposition 1.2). The second inequality is shown in Lemma 2.4: this relies on the greedy criteria used in our algorithm.

Lemma 2.3. $\Pr[\sigma$ *finishes in k queries*$] \leq \Pr\left[\min_{i\in T^*} X_i \leq \theta\right]$.

Proof. TOPROVE 1 □

Lemma 2.4. $\Pr\left[\min_{i\in T^*} X_i \leq \theta\right] \leq \Pr[\pi$ *finishes in $2k$ iterations*$]$.

Proof. TOPROVE 2 □

2.2 Finding the minimum interval

In this section, we consider the SMQI problem, where the goal is to identify *an interval* that is guaranteed to be a δ-minimizer.

Unlike the previous SMQ setting (where we find a δ-minimum value), for SMQI we just want to identify some interval $i^* \in N$ such that $X_{i^*} \leq \mathsf{MIN} + \delta$. Recall that $\mathsf{MIN} = \min_{i \in N} X_i$. It is important to note that the interval i^* may not have been queried. It is easy to see that any SMQ policy is also feasible to SMQI. Indeed, by the stopping rule (1) for SMQ, the δ-minimum value returned is always the minimum value of a queried interval: so we also identify i^*. However, an SMQI policy may return an interval i^* without querying it. So the optimal value of SMQI may be strictly smaller than SMQ.

Remark: We note that the optimal values of SMQ and SMQI differ by at most the maximum query cost c_{max}. As noted above, the optimal SMQI value is at most that of SMQ. On the other hand, the optimal SMQ value is at most the optimal SMQI value plus the cost to query i^*. In the unit-cost setting, $c_{max} = 1$ and any policy has expected cost at least 1: so the optimal values of SMQ and SMQI are within a factor two of each other. This immediately implies that Algorithm 1 is also an 8-approximation for unit-cost SMQI. In the rest of this subsection, we will prove a stronger result, that Algorithm 1 is a 4-approximation for SMQI. Apart from the improved constant factor, these ideas will also be helpful for SMQI with general costs. We note that under general costs, the optimal SMQ and SMQI values may differ by an arbitrarily large factor because c_{max} is not a lower bound on the optimal value.

Stopping criteria for SMQI Consider any state, given by a subset $S \subseteq N$ of queried r.v.s along with their observations $\{x_i\}_{i \in S}$. There are two conditions under which the SMQI policy can stop.

- The first stopping rule is just the one for SMQ, Equation (1). This corresponds to the situation that interval i^* is queried. We restate this rule below for easy reference:

$$\min_{i \in S} x_i \quad \leq \quad \min_{j \in N \setminus S} \ell_j + \delta. \tag{3}$$

  In this case, we return $i^* = \arg\min_{i \in S} x_i$. We refer to this as the old stopping rule.

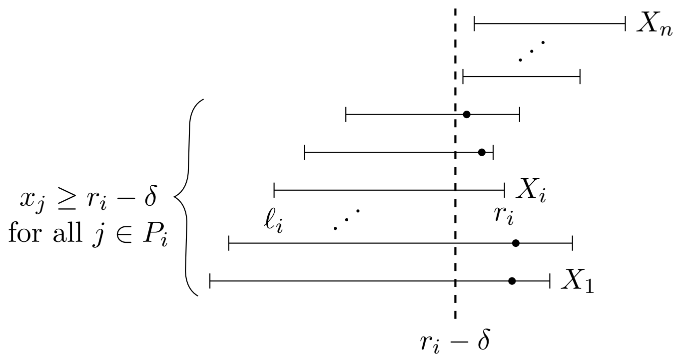
- The second stopping rule handles the situation where an un-queried interval i^* is returned. For any $i \in N$, define the "almost prefix" set $P_i := \{j \in N \setminus i : \ell_j < r_i - \delta\}$. Note that either P_i or $P_i \cup i$ is a *prefix* of $[n]$. (As before, we assume that intervals are indexed by increasing order of their left-endpoint, i.e., $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_n$.) The new rule is:

$$\exists\, i \in N \text{ such that } P_i \subseteq S \text{ and } \min_{j \in P_i} x_j \geq r_i - \delta. \tag{4}$$

  In other words, there is some interval i where (1) all intervals $j \neq i$ with left-endpoint $\ell_j < r_i - \delta$ have been queried, and (2) the minimum value of these r.v.s is at least $r_i - \delta$. In this case, we return $i^* = i$ (we may not know a δ-minimum value). We refer to this as the new stopping rule. See Figure 5 for an example.

Proposition 2.5. *A policy for* SMQI *can stop if and only if either criterion* (3) *or* (4) *holds.*

Figure 5: Illustration of new SMQI stopping criterion.



Our algorithm for SMQI with unit costs remains the same as for SMQ (Algorithm 1). The only difference is in the new stopping criterion (described above). Recall that π is the permutation used by our non-adaptive policy. When it is clear from the context, we will also use π to denote our SMQI policy that performs queries in the order of π until stopping criteria (3) or (4) applies.

Theorem 2.6. *The non-adaptive policy π is a 4-approximation algorithm for* SMQI.

We now prove this result. Let σ denote an optimal adaptive policy for SMQI. For any $k \geq 1$, we will show:

$$\Pr[\sigma \text{ finishes in } k \text{ queries}] \leq \Pr[\pi \text{ finishes in } 2k \text{ iterations}]. \tag{5}$$

This would suffice to prove the 4-approximation, exactly as in Theorem 2.2.

In order to prove (5), we fix some $k \geq 1$. As in the previous proof, let $\theta = \theta_k = \ell_{k+1} + \delta$ and let T^* be defined as in (2). To reduce notation, define the following events.

$$\mathcal{A}_1 : \text{ our policy } \pi \text{ finishes within } 2k \text{ iterations due to (3).}$$

$$\mathcal{A}_2 : \text{ our policy } \pi \text{ finishes within } 2k \text{ iterations due to (4).}$$

$$\mathcal{O}_1 : \text{ optimal policy } \sigma \text{ finishes within } k \text{ queries due to (3).}$$

$$\mathcal{O}_2 : \text{ optimal policy } \sigma \text{ finishes within } k \text{ queries due to (4).}$$

Handling the old stopping criterion. Let L denote the smallest un-queried left-endpoint at the end of iteration $2k$ in π. Note that L is a deterministic value as π is a non-adaptive policy. Moreover, $L \geq \ell_{2k+1}$ as π would have queried the first $2k$ r.v.s. Let $\mathcal{G}$ be the event that $X_i > L + \delta$ for all intervals i queried by π in its first $2k$ iterations. In other words, $\mathcal{G}$ is precisely the event that stopping criterion (3) *does not* apply at the end of iteration $2k$ in π, i.e., $\mathcal{G} = \neg\mathcal{A}_1$. By Lemma 2.4,

$$\Pr[\neg\mathcal{G}] = \Pr[\mathcal{A}_1] \geq \Pr\left[\min_{i \in T^*} X_i \leq \theta\right].$$

Similarly, let $\mathcal{G}^*$ be that event that $X_i > \theta$ for all intervals i in the first k queries of σ. From the proof of Lemma 2.3, we obtain $\mathcal{O}_1 \subseteq \neg\mathcal{G}^*$ and

$$\Pr[\neg\mathcal{G}^*] \leq \Pr\left[\min_{i \in T^*} X_i \leq \theta\right].$$

Combining the above two inequalities, we have

$$\Pr[\neg\mathcal{G}^*] \leq \Pr[\neg\mathcal{G}]. \tag{6}$$

Handling the new stopping criterion. Let $\mathcal{G}_A$ be the event that $X_j > L+\delta$ for *all* r.v.s $j \in N$. Similarly, let $\mathcal{G}_A^*$ be the event that $X_j > \theta$ for *all* $j \in N$. Clearly,

$$\Pr\left[\mathcal{A}_2 \,|\, \mathcal{G}\right] \;=\; \Pr\left[\mathcal{A}_2 \,|\, \mathcal{G}_A\right] \quad \text{and} \quad \Pr\left[\mathcal{O}_2 \,|\, \mathcal{G}^*\right] \;=\; \Pr\left[\mathcal{O}_2 \,|\, \mathcal{G}_A^*\right]. \tag{7}$$

We will now prove that

$$\Pr\left[\mathcal{A}_2 \,|\, \mathcal{G}_A\right] \;\geq\; \Pr\left[\mathcal{O}_2 \,|\, \mathcal{G}_A^*\right]. \tag{8}$$

If σ finishes due to (4) in k queries then the almost-prefix set $P_{i^*} \subseteq [k+1]$: otherwise $|P_{i^*}| > k$ which contradicts with the fact that all r.v.s in P_{i^*} must be queried. Let $R = \{i \in N : P_i \subseteq [k+1]\}$ be all such intervals. It now follows that the event $\mathcal{O}_2$ (which corresponds to policy σ) is *contained in* the event

$$\mathcal{E} := \bigvee_{i \in R} \left(\wedge_{j \in P_i} (X_j \geq r_i - \delta)\right). \tag{9}$$

Note that $\mathcal{E}$ is independent of the policy: it only depends on the realizations of the r.v.s (and doesn't depend on whether/not an interval has been queried).

Moreover, our policy π queries all the r.v.s in $[2k] \supseteq [k+1]$ within $2k$ iterations. So, for all $i \in R$, the r.v.s in $P_i \subseteq [k+1]$ are queried by π in $2k$ iterations. Hence, event $\mathcal{A}_2$ (which corresponds to policy π) *contains* event $\mathcal{E}$.

Recall that the event $\mathcal{G}_A$ (resp. $\mathcal{G}_A^*$) in policy π (resp. σ) means that every r.v. is more than $L+\delta$ (resp. θ). Also, $\theta \leq L+\delta$, which means

$$\Pr[X_j \geq u | X_j > L+\delta] \geq \Pr[X_j \geq u | X_j > \theta], \quad \forall u \in \mathbb{R}, \forall j \in N.$$

In other words, for any $j \in N$, if Y_j (resp. Z_j) is the r.v. X_j conditioned on $\mathcal{G}_A$ (resp. $\mathcal{G}_A^*$) then Y_j *stochastically dominates* Z_j.[2] Note also that the r.v.s Y_js (resp. Z_js) are independent. Using the fact that event $\mathcal{E}$ corresponds to a *monotone* function, we obtain:

Lemma 2.7. *Let $\{Y_j : j \in N\}$ and $\{Z_j : j \in N\}$ be independent r.v.s such that Y_j stochastically dominates Z_j for each $j \in N$. Then, $\Pr[\mathcal{E}(Y_1, ..., Y_n)] \geq \Pr[\mathcal{E}(Z_1, ..., Z_n)]$ where event $\mathcal{E}$ is a function of independent r.v.s as defined in (9).*

Proof. TOPROVE 3 □

Using Lemma 2.7, we obtain $\Pr[\mathcal{E}|\mathcal{G}_A] \geq \Pr[\mathcal{E}|\mathcal{G}_A^*]$, which proves (8). Combined with (7),

$$\Pr\left[\mathcal{A}_2 \,|\, \mathcal{G}\right] \;\geq\; \Pr\left[\mathcal{O}_2 \,|\, \mathcal{G}^*\right]. \tag{10}$$

Wrapping up. We have

$$\begin{aligned}
\Pr[\mathcal{A}_1 \vee \mathcal{A}_2] &= \Pr[\mathcal{A}_1] + \Pr[\mathcal{A}_2 \wedge \neg\mathcal{A}_1] = \Pr[\neg\mathcal{G}] + \Pr[\mathcal{A}_2 \wedge \mathcal{G}] \\
&= \Pr[\neg\mathcal{G}] + \Pr[\mathcal{A}_2|\mathcal{G}] \cdot \Pr[\mathcal{G}] = 1 - (1 - \Pr[\mathcal{A}_2|\mathcal{G}]) \cdot \Pr[\mathcal{G}] \\
&\geq 1 - (1 - \Pr[\mathcal{O}_2|\mathcal{G}^*]) \cdot \Pr[\mathcal{G}^*] \qquad \text{by (6) and (10)} \\
&= \Pr[\neg\mathcal{G}^*] + \Pr[\mathcal{O}_2 \wedge \mathcal{G}^*] \\
&\geq \Pr[\mathcal{O}_1] + \Pr[\mathcal{O}_2 \wedge \neg\mathcal{O}_1] \qquad \text{using } \mathcal{O}_1 \subseteq \neg\mathcal{G}^* \\
&= \Pr[\mathcal{O}_1 \vee \mathcal{O}_2].
\end{aligned}$$

This completes the proof of (5) and the theorem.

[2] We say that r.v. Y stochastically dominates Z if $\Pr[Y \geq u] \geq \Pr[Z \geq u]$ for all $u \in \mathbb{R}$.

3 Algorithm for General Costs

We now consider the SMQ problem with non-uniform query costs. We assume (without loss of generality, by scaling) that costs are at least one, i.e., $\min_{i\in N} c_i \geq 1$. The high-level idea is similar to the unit-cost case: interleaving the two greedy criteria of smallest left-endpoint and highest probability of stopping. However, we need to incorporate the costs carefully. To this end, we use an iterative algorithm that in every iteration g, makes a *batch* of queries having total cost about 2^g. (In order to optimize the approximation ratio, we use a generic base y for the exponential costs.)

For any subset $S \subseteq N$, let $c(S) := \sum_{j\in S} c_j$ denote the cost of querying all intervals in S. Again, we renumber intervals so that $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_n$.

Definition 3.1. *For any $g \geq 0$, let T_g be the maximal prefix of intervals having cost at most y^g.*

Algorithm 2 Double Greedy for General Cost

1: Let $\ell^* = \min_{j\in N} \ell_j$, $m^* = R := \min_{j\in N} r_j$, and $\pi \leftarrow \emptyset$.
2: **for** $g = 0, 1, 2, \ldots,$ **do** $\triangleright$ iteration
3: Query intervals $T_g \setminus \pi$ and update list $\pi \leftarrow \pi \circ T_g$.
4: Update $\ell^* = \min_{j\in N\setminus\pi}\{\ell_j\}$ and let threshold $\theta_g = \ell^* + \delta$.
5: Compute a $(1, 1+\epsilon)$ bicriteria approximate solution U_g for:

$$p_g^* = \min_{T\subseteq N\setminus\pi} \left\{ \Pr\left[\min_{j\in T} X_j > \theta_g\right] : c(T) \leq y^g \right\}. \tag{KP}$$

6: Query intervals U_g and update list $\pi \leftarrow \pi \circ U_g$.
7: Update $\ell^* = \min_{j\in N\setminus\pi}\{\ell_j\}$ and $m^* = \min\left\{m^*, \min_{j\in T_g\cup U_g} X_j\right\}$.
8: **if** $m^* - \ell^* \leq \delta$ **then** stop.

The complete algorithm is given in Algorithm 2. The optimization problem (KP) solved in Step 5 is a variant of the classic knapsack problem: in Theorem E.1 (see Appendix E) we provide a $(1, 1+\epsilon)$ bicriteria approximation algorithm for (KP) for any constant $\epsilon > 0$. In particular, this ensures that $c(U_g) \leq y^g(1+\epsilon)$ and

$$\Pr\left[\min_{j\in U_g} X_j > \theta_g\right] \leq p_g^*.$$

Note that the left-hand-side above equals $\prod_{j\in U_g} \Pr[X_j > \theta_g]$ as all r.v.s are independent.

Furthermore, just like Algorithm 1, we can view Algorithm 2 as first computing the permutation π (without querying) and then performing queries in that order until the stopping criterion. So, our algorithm is a non-adaptive policy and our analysis also upper-bounds the adaptivity gap.

3.1 Analysis

We use σ to denote the optimal (adaptive) policy and π to denote our non-adaptive policy.

Definition 3.2. *For any $g \geq 0$, let $o_g := \Pr[\sigma$ does not finish by cost $y^g]$. Similarly, for our policy we define $v_g := \Pr[\pi$ does not finish by iteration $g]$. We also define σ_g to be the optimal policy truncated at cost y^g, i.e., the total cost of queried intervals is always at most y^g. Similarly, we define π_g to be our policy truncated at the end of iteration g.*

The key part of the analysis lies in relating the non-stopping probabilities o_g and a_g in the optimal and algorithmic policies: see Lemma 3.4. Our first lemma bounds the (worst-case) cost incurred in g iterations of our policy.

Lemma 3.3. *The cost of our policy until the end of iteration g is*

$$c(\pi_g) \le (1+\epsilon)\left(1+\frac{y}{y-1}\right) y^g.$$

Proof. TOPROVE 4 □

Lemma 3.4. *For all $g \ge 0$, we have $v_g \le o_g$.*

Proof. TOPROVE 5 □

In Lemma 3.5 we lower bound the expected cost of the optimal policy. Let $c_{exp}(\pi)$ and $c_{exp}(\sigma)$ denote the expected cost of our greedy policy and the optimal policy, respectively.

Lemma 3.5. *For any base $y \ge 1$, we have $\sum_{g\ge 0} y^g \cdot o_g \le \frac{y}{y-1} c_{exp}(\sigma) - \frac{1}{y-1}$.*

Proof. TOPROVE 6 □

Theorem 3.6. *There is a $(3+2\sqrt{2}+\epsilon)$-approximation for the* SMQ *problem with general costs.*

Proof. TOPROVE 7 □

4 SMQI under Non-uniform Costs

We now consider the (harder) problem of identifying a δ-minimum interval. Recall that the goal here is to identify some interval $i^* \in N$ such that $X_{i^*} \le \mathsf{MIN} + \delta$ where $\mathsf{MIN} = \min_{i\in N} X_i$. The interval i^* may not have been queried by the policy. Unlike the unit-cost case, we can no longer rely on the SMQ algorithm itself (see the example below).

Bad example for the SMQ policy. Consider an instance with the following r.v.s.

- X_1 is distributed over the interval $[0, 1.5\,\delta]$. (The exact distribution is irrelevant.)
- X_2 has $\Pr[X_2 = 0.3\,\delta] = \frac{1}{n^2}$ and $\Pr[X_2 = 2\,\delta] = 1 - \frac{1}{n^2}$.
- $X_3, \ldots, X_n$ are identically distributed with $\Pr[X_i = 0.7\,\delta] = \frac{1}{n}$ and $\Pr[X_i = 1.5\,\delta] = 1 - \frac{1}{n}$.

The cost $c_1 = n \gg 1$ and all other costs are unit. The SMQ policy from Algorithm 2 will first select at least $\Omega(n)$ r.v.s among $X_3, \ldots, X_n$ because these will optimize (KP). Crucially, the policy will not query X_1 or X_2 for a long time. Consequently, the expected cost of this policy is $\Omega(n)$. On the other hand, an optimal policy just queries X_2: if $X_2 = 2\,\delta$ then it returns $i^* = 1$ (stopping rule (4) applies); if $X_2 = 0.3\,\delta$ then it returns $i^* = 2$ (stopping rule (3) applies). So the SMQ algorithm has an $\Omega(n)$ approximation ratio when applied directly to SMQI.

This example shows that in the presence of non-uniform costs, additional work is needed to handle the new stopping criterion (4). In particular, we need to skip expensive intervals while querying in the order of left-endpoints. The SMQI algorithm for general costs has the same high-level structure as the one for SMQ (Algorithm 2). The only change is in Step 3 where we modify the queried set T_g by skipping some expensive intervals. We first define these new "almost prefix" sets that will be used in Step 3.

Definition 4.1. *For any iteration $g \geq 0$, an interval $j \in N$ is called g-**big** if its cost $c_j > y^g$ (otherwise, j is called g-small).*

Recall that the intervals are numbered according to their left-endpoint, i.e., $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_n$.

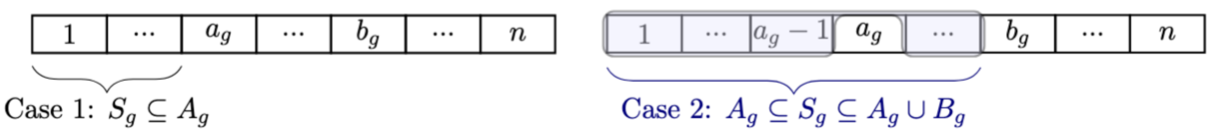
Definition 4.2. *Consider any iteration $g \geq 0$.*

- *Let a_g and b_g denote the first and second g-big intervals, respectively.*
- *$A_g := \{1, 2, \ldots, a_g - 1\}$ is the maximal prefix of $[n]$ that does not contain any g-big interval.*
- *$B_g := \{a_g + 1, \ldots, b_g - 1\}$ is the segment of $[n]$ between the first and second g-big intervals.*

We now define the almost-prefix query set S_g as follows:

1. *If $c(A_g) > y^g$ then S_g is the maximal prefix of A_g having cost at most $2y^g$. Here, $S_g \subseteq A_g$.*
2. *If $c(A_g) \leq y^g$ then S_g is the maximal prefix of $A_g \cup B_g$ having cost at most y^g. Here, $S_g \supseteq A_g$.*

Figure 6: Illustration of Definition 4.2.



SMQI **algorithm.** This involves replacing the "prefix set" T_g in Step 3 of the SMQ algorithm (Algorithm 2) by the almost-prefix set S_g defined above. The other steps remain the same as in Algorithm 2. The stopping criterion also changes: we will perform queries in the order of π until either (3) or (4) applies. We start with a useful lemma showing that the almost-prefix sets S_g are nested (as was the case for the sets T_g). We note that this lemma is just needed to obtain a tighter constant factor.

Lemma 4.3. *For each $g \geq 0$, we have $S_g \subseteq S_{g+1}$.*

Proof. TOPROVE 8 □

The rest of the analysis combines ideas from the non-uniform cost SMQ and the uniform cost SMQI. Let π denote our (non-adaptive) policy and σ the optimal adaptive policy. We re-use the terms from Definition 3.2:

$$o_g := \Pr[\sigma \text{ does not finish by cost } y^g].$$

$$v_g := \Pr[\pi \text{ does not finish by iteration } g].$$

$$\sigma_g \quad \text{is the optimal policy truncated at cost } y^g$$

$$\pi_g \quad \text{is our policy truncated at the end of iteration } g.$$

Lemma 4.4. *The cost of our policy until the end of iteration g is*

$$c(\pi_g) \le (1+\epsilon)\left(2 + \frac{y}{y-1}\right) y^g.$$

Proof. TOPROVE 9 ☐

Lemma 3.5 continues to hold here as well; so:

$$\sum_{g\ge 0} y^g o_g \le \frac{y}{y-1} c_{exp}(\sigma) - \frac{1}{y-1}. \tag{11}$$

The key step is the analogue of Lemma 3.4, which we prove in the next subsection.

Lemma 4.5. *For all $g \ge 0$, we have $v_g \le o_g$.*

We can now prove the main result.

Theorem 4.6. *There is a $(4 + 2\sqrt{3} + \epsilon)$-approximation for* SMQI *with general costs.*

Proof. TOPROVE 10 ☐

4.1 Proof of Lemma 4.5

Fix any iteration g. As in the unit-cost SMQI proof, we define the following events.

$$\begin{aligned}
\mathcal{A}_1 &: \text{ our policy } \pi \text{ finishes within } g \text{ iterations due to (3).}\\
\mathcal{A}_2 &: \text{ our policy } \pi \text{ finishes within } g \text{ iterations due to (4).}\\
\mathcal{O}_1 &: \text{ optimal policy } \sigma \text{ finishes by cost } y^g \text{ due to (3).}\\
\mathcal{O}_2 &: \text{ optimal policy } \sigma \text{ finishes by cost } y^g \text{ due to (4).}
\end{aligned}$$

Clearly, $1 - v_g = \Pr[\mathcal{A}_1 \vee \mathcal{A}_2]$ and $1 - o_g = \Pr[\mathcal{O}_1 \vee \mathcal{O}_2]$.

Handling the old stopping criterion. Let L denote the smallest un-queried left-endpoint at the end of iteration g in π. Note that L is a deterministic value. Let $\mathcal{G}$ be the event that $X_j > L + \delta$ for all intervals j queried by π_g. Note that $\mathcal{G} = \neg\mathcal{A}_1$, i.e., criterion (3) does *not* apply by the end of iteration g. Recall that threshold θ_g (Step 4 in Algorithm 2) is δ more that the smallest un-queried left-endpoint in that step. Clearly, $\theta_g \le L + \delta$ in iteration g.

Now consider the truncated optimal policy. Let $L(\sigma_g)$ be its smallest un-queried left-endpoint; this is a random value as σ_g is an adaptive policy. We claim that

$$L + \delta \;\ge\; \theta_g \;\ge\; \min_{j \in N \setminus S_g} \ell_j + \delta \;\ge\; L(\sigma_g) + \delta. \tag{12}$$

Above, the second inequality uses the fact that S_g is queried before Step 4. To see the last inequality, note that σ_g cannot query any g-big interval: so $L(\sigma_g) \le \ell_{a_g}$. We have two cases depending on the definition of S_g:

- If $S_g \supseteq A_g$ then clearly $\min_{j\in N\setminus S_g} \ell_j = \ell_{a_g} \ge L(\sigma_g)$.

- If $S_g \subseteq A_g$ then we must have $c(A_g) > y^g$, which means that S_g contains the maximal prefix of cost at most y^g. Again, this implies $\min_{j\in N\setminus S_g} \ell_j \geq L(\sigma_g)$.

This proves (12).

Now, let $\mathcal{G}^*$ be the that event that $X_j > \theta_g$ for all intervals j in σ_g. Using (12) it follows that $\mathcal{O}_1 \subseteq \neg\mathcal{G}^*$. We now obtain:

$$\Pr[\mathcal{G}^*] = \Pr\left[\min_{j\in\sigma_g} X_j > \theta_g\right] \geq \Pr\left[\min_{j\in\pi_g} X_j > \theta_g\right] \geq \Pr\left[\min_{j\in\pi_g} X_j > L+\delta\right] = \Pr[\mathcal{G}]. \tag{13}$$

The first inequality follows from the proof of Lemma 3.4: see (**??**) - (**??**). The second inequality above uses $\theta_g \leq L+\delta$.

Handling the new stopping criterion. Let $\mathcal{G}_A$ be the event that $X_j > L+\delta$ for *all* r.v.s $j \in N$. Similarly, let $\mathcal{G}_A^*$ be the event that $X_j > \theta_g$ for *all* $j \in N$. Clearly, $\Pr[\mathcal{A}_2 \mid \mathcal{G}] = \Pr[\mathcal{A}_2 \mid \mathcal{G}_A]$ and $\Pr[\mathcal{O}_2 \mid \mathcal{G}^*] = \Pr[\mathcal{O}_2 \mid \mathcal{G}_A^*]$. We will now prove that

$$\Pr[\mathcal{A}_2 \mid \mathcal{G}] = \Pr[\mathcal{A}_2 \mid \mathcal{G}_A] \geq \Pr[\mathcal{O}_2 \mid \mathcal{G}_A^*] = \Pr[\mathcal{O}_2 \mid \mathcal{G}^*]. \tag{14}$$

Using (13), exactly as in the proof of Theorem 2.6, this implies $\Pr[\mathcal{A}_1 \vee \mathcal{A}_2] \geq \Pr[\mathcal{O}_1 \vee \mathcal{O}_2]$. We repeat the argument below for completeness.

$$\begin{aligned}
\Pr[\mathcal{A}_1 \vee \mathcal{A}_2] &= \Pr[\mathcal{A}_1] + \Pr[\mathcal{A}_2 \wedge \neg\mathcal{A}_1] = \Pr[\neg\mathcal{G}] + \Pr[\mathcal{A}_2 \wedge \mathcal{G}] \\
&= \Pr[\neg\mathcal{G}] + \Pr[\mathcal{A}_2|\mathcal{G}] \cdot \Pr[\mathcal{G}] = 1 - (1 - \Pr[\mathcal{A}_2|\mathcal{G}]) \cdot \Pr[\mathcal{G}] \\
&\geq 1 - (1 - \Pr[\mathcal{O}_2|\mathcal{G}^*]) \cdot \Pr[\mathcal{G}^*] \qquad \text{by (13) and (14)} \\
&= \Pr[\neg\mathcal{G}^*] + \Pr[\mathcal{O}_2 \wedge \mathcal{G}^*] \\
&\geq \Pr[\mathcal{O}_1] + \Pr[\mathcal{O}_2 \wedge \neg\mathcal{O}_1] \qquad \text{using } \mathcal{O}_1 \subseteq \neg\mathcal{G}^* \\
&= \Pr[\mathcal{O}_1 \vee \mathcal{O}_2].
\end{aligned}$$

This proves Lemma 4.5.

Proving (14). The key property here is the following.

Lemma 4.7. *If σ finishes due to criterion (4) and identifies i^* by cost y^g then $P_{i^*} \subseteq S_g$.*

Proof. TOPROVE 11 □

Let $R = \{h \in N : P_h \subseteq S_g\}$. By Lemma 4.7 it follows that if event $\mathcal{O}_2$ occurs then $i^* \in R$. Hence, $\mathcal{O}_2$ is a subset of the event

$$\mathcal{E} := \bigvee_{h\in R} \left(\wedge_{j\in P_h}(X_j > r_h - \delta)\right).$$

Moreover, our policy π_g queries all the r.v.s in S_g. So, for all $h \in R$, the r.v.s in $P_h \subseteq S_g$ are queried by π_g. Hence, event $\mathcal{A}_2$ contains event $\mathcal{E}$.

Recall that the event $\mathcal{G}_A$ (resp. $\mathcal{G}_A^*$) in policy π (resp. σ) means that every r.v. is more than $L+\delta$ (resp. θ). Also, $\theta \leq L+\delta$, which means

$$\Pr[X_j > t | X_j > L+\delta] \geq \Pr[X_j > t | X_j > \theta], \quad \forall t \in \mathbb{R}, \forall j \in N.$$

In other words, for any $j \in N$, r.v. X_j conditioned on $\mathcal{G}_A$ stochastically dominates X_j conditioned on $\mathcal{G}_A^*$. Using Lemma 2.7 (which deals with the same event $\mathcal{E}$) with $Y_j = X_j|\mathcal{G}_A$ and $Z_j = X_j|\mathcal{G}_A^*$, we obtain $\Pr[\mathcal{E}|\mathcal{G}_A] \geq \Pr[\mathcal{E}|\mathcal{G}_A^*]$, which proves (14).

References

[AN16] Arash Asadpour and Hamid Nazerzadeh. Maximizing stochastic monotone submodular functions. *Manag. Sci.*, 62(8):2374–2391, 2016.

[ASW16] Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. *Math. Oper. Res.*, 41(3):1022–1038, 2016.

[BDE+21] Evripidis Bampis, Christoph Dürr, Thomas Erlebach, Murilo Santos de Lima, Nicole Megow, and Jens Schlöter. Orienting (hyper)graphs under explorable stochastic uncertainty. In *29th Annual European Symposium on Algorithms (ESA)*, volume 204 of *LIPIcs*, pages 10:1–10:18, 2021.

[BGL+12] N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.

[CHdT21] Steven Chaplick, Magnús M. Halldórsson, Murilo S. de Lima, and Tigran Tonoyan. Query minimization under stochastic uncertainty. *Theoretical Computer Science*, 895:75–95, 2021.

[DGV08] B. C. Dean, M. X. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.*, 33(4):945–964, 2008.

[EHK+08] Thomas Erlebach, Michael Hoffmann, Danny Krizanc, Matús Mihal'Ák, and Rajeev Raman. Computing minimum spanning trees with uncertainty. In *STACS 2008*, pages 277–288. IBFI Schloss Dagstuhl, 2008.

[EHK16] Thomas Erlebach, Michael Hoffmann, and Frank Kammer. Query-competitive algorithms for cheapest set problems under uncertainty. *Theoretical Computer Science*, 613:51–64, 2016.

[FLX18] Hao Fu, Jian Li, and Pan Xu. A PTAS for a class of stochastic dynamic programs. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107 of *LIPIcs*, pages 56:1–56:14, 2018.

[FMP+00] Tomás Feder, Rajeev Motwani, Rina Panigrahy, Chris Olston, and Jennifer Widom. Computing the median with uncertainty. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 602–607, 2000.

[GGHK18] Dimitrios Gkenosis, Nathaniel Grammel, Lisa Hellerstein, and Devorah Kletenik. The Stochastic Score Classification Problem. In *26th Annual European Symposium on Algorithms (ESA)*, pages 36:1–36:14, 2018.

[GGM10] Ashish Goel, Sudipto Guha, and Kamesh Munagala. How to probe for an extreme value. *ACM Transactions on Algorithms (TALG)*, 7(1):1–20, 2010.

[GKNR15] Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, and R. Ravi. Running errands in time: Approximation algorithms for stochastic orienteering. *Math. Oper. Res.*, 40(1):56–79, 2015.

[GM07] Sudipto Guha and Kamesh Munagala. Approximation algorithms for budgeted learning problems. In *39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 104–113. ACM, 2007.

[GNS17] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and XOS functions. In *Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1688–1702. SIAM, 2017.

[HKP21] Lisa Hellerstein, Devorah Kletenik, and Srinivasan Parthasarathy. A tight bound for stochastic submodular cover. *J. Artif. Intell. Res.*, 71:347–370, 2021.

[HLS24] Lisa Hellerstein, Naifeng Liu, and Kevin Schewior. Quickly determining who won an election. In *15th Innovations in Theoretical Computer Science Conference (ITCS)*, LIPIcs, pages 61:1–61:14, 2024.

[INvdZ16] Sungjin Im, Viswanath Nagarajan, and Ruben van der Zwaan. Minimum latency submodular cover. *ACM Trans. Algorithms*, 13(1):13:1–13:28, 2016.

[JLLS20] Haotian Jiang, Jian Li, Daogao Liu, and Sahil Singla. Algorithms and adaptivity gaps for stochastic k-tsp. In *11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151 of *LIPIcs*, pages 45:1–45:25, 2020.

[Kah91] Simon Kahan. A model for data in motion. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of computing*, pages 265–277, 1991.

[MMS17] Nicole Megow, Julie Meißner, and Martin Skutella. Randomization helps computing a minimum spanning tree under uncertainty. *SIAM Journal on Computing*, 46(4):1217–1240, 2017.

[MS23] Nicole Megow and Jens Schlöter. Set selection under explorable stochastic uncertainty via covering techniques. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 319–333. Springer, 2023.

[Sin18] Sahil Singla. The price of information in combinatorial optimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2523–2532. SIAM, 2018.

[SS21] Danny Segev and Sahil Singla. Efficient approximation schemes for stochastic probing and prophet problems. In *22nd ACM Conference on Economics and Computation (EC)*, pages 793–794. ACM, 2021.

[WGW22] Weina Wang, Anupam Gupta, and Jalani K Williams. Probing to minimize. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215, page 120. Schloss Dagstuhl–Leibniz-Zentrum f {\" u} r Informatik, 2022.

A Multiplicative Precision

Given an instance with non-negative r.v.s $\{X_i\}_{i=1}^n$ and multiplicative precision $\alpha \geq 1$, consider a new instance of SMQ with r.v.s $\{X'_i := \ln(X_i)\}_{i=1}^n$ and additive precision $\delta := \ln \alpha$. Note that

$$\mathsf{MIN}' = \min_{i=1}^{n} X'_i = \min_{i=1}^{n} \ln(X_i) = \ln\left(\min_{i=1}^{n} X_i\right) = \ln(\mathsf{MIN}).$$

An α-approximately minimum value W for the original instance satisfies $\mathsf{MIN} \leq W \leq \alpha \cdot \mathsf{MIN}$, where $\mathsf{MIN} = \min_{i=1}^n X_i$. Then, $\mathsf{VAL} = \ln(W)$ satisfies $\mathsf{MIN}' = \ln(\mathsf{MIN}) \leq \mathsf{VAL} \leq \ln(\mathsf{MIN}) + \ln\alpha = \mathsf{MIN}' + \delta$, i.e., VAL is a δ-minimum value for the new instance. Similarly, if VAL is a δ-minimum value for the new instance then $W := e^{\mathsf{VAL}}$ is an α-approximately minimum value for the original instance.

B Bad Example for Competitive Ratio

We provide an example that rules out any reasonable *competitive ratio* bound for SMQ and SMQI with precision $\delta > 0$. This is in sharp contrast to the corresponding problem with exact precision ($\delta = 0$) for which a constant competitive ratio is known [Kah91]. We note that results in the online setting assume open intervals, which in our setting (with discrete r.v.s) corresponds to all left-endpoints being distinct.[3] The benchmark in the online setting is the *hindsight optimum*, which is the minimum number (or cost) of queries that are needed to verify a δ-minimum value *conditioned* on the realizations $\{x_i\}_{i=1}^n$ of the r.v.s.

Consider an instance with n r.v.s with $\Pr[X_i = i] = p := \frac{\ln n}{n}$ and $\Pr[X_i = n^2] = 1 - p$ for all $i \in [n]$. All costs are unit and the precision $\delta = n$. We refer to the values $\{1, 2, \ldots, n\}$ as *low* values: note that any low value is a δ-minimum value for this instance.

We first consider the hindsight optimum. If any of the n r.v.s (say k) realizes to a low value then verifying the δ-minimum value just requires querying k, which has cost 1. On the other hand, the probability that none of the n r.v.s realizes to a low value is $(1-p)^n \leq e^{-pn} = \frac{1}{n}$: in this case the optimal verification cost is n (querying all r.v.s). So the expected optimal cost is at most 2.

Now, consider any SMQ policy: this does not know the realizations before querying. The only way to stop querying is (1) when some low value is observed, or (2) all n r.v.s have been queried. The probability that the i^{th} r.v. is queried is exactly $(1-p)^{i-1}$, which corresponds no low realization among the previous $i-1$ r.v.s. So, the expected cost of any policy is:

$$\sum_{i=1}^{n}(1-p)^{i-1} = \sum_{i=0}^{\infty}(1-p)^i - \sum_{i=n}^{\infty}(1-p)^i = \frac{1}{p} - \frac{1}{p}(1-p)^n \geq \frac{1}{p}(1 - e^{-pn}),$$

where the second equality uses $\sum_{i=0}^{\infty}(1-p)^i = \frac{1}{p}$. Using $p = \frac{\ln n}{n}$, the expected cost is at least $\frac{n}{\ln n}(1 - \frac{1}{n})$.

Hence the competitive ratio for SMQ is $\Omega(\frac{n}{\ln n})$.

C Adaptivity Gap for SMQ

The instance has three intervals with $X_1 \in \{0, 3, \infty\}$, $X_2 \in \{1, \infty\}$, $X_3 \in \{2, \infty\}$ and $\delta = 1$. Let $\Pr(X_1 = 0) = \frac{1}{3}, \Pr(X_1 = 3) = \frac{1}{3}, \Pr(X_1 = \infty) = \frac{1}{3}, \Pr(X_2 = 1) = \epsilon, \Pr(X_2 = \infty) = 1 - \epsilon, \Pr(X_3 = 2) = 1 - \epsilon, \Pr(X_3 = \infty) = \epsilon$. Recall that the adaptive policy has cost at most $\frac{5+\epsilon}{3}$.

[3] Alternatively, our example can be modified into one with open intervals where the competitive ratio is still $\tilde{\Omega}(n)$.

We consider the cost of all possible non-adapative policies:

1. The cost of policy $\{1, 2, 3\}$ is,

$$NA \geq 1 + \frac{2}{3} + \frac{1-\epsilon}{3} = \frac{6-\epsilon}{3}.$$

   We query X_1 w.p. 1, X_2 w.p. 2/3 and X_3 w.p. $(1-\epsilon)/3$.

2. The cost of policy $\{1, 3, 2\}$ is,

$$NA \geq 1 + \frac{2}{3} + \frac{2\epsilon}{3} = \frac{5+2\epsilon}{3}.$$

   We query X_1 w.p. 1, X_3 w.p. 2/3 and X_2 w.p. $2\epsilon/3$.

3. The cost of policy $\{2, 1, 3\}$ is,

$$NA \geq 2 - \epsilon + \frac{1-\epsilon}{3} = \frac{7-4\epsilon}{3}.$$

   We query X_2 w.p. 1, X_1 w.p. $1-\epsilon$ and X_3 w.p. $(1-\epsilon)/3$.

4. The cost of policy $\{2, 3, 1\}$ is,

$$NA \geq 2 - \epsilon + 1 - \epsilon = 3 - 2\epsilon.$$

   We query X_2 w.p. 1, X_3 w.p. $1-\epsilon$ and X_1 w.p. $1-\epsilon$.

5. The remaining non-adaptive policies start with 3. Any such policy costs at least 2 because even if $X_3 = 2$ (its lowest value) we cannot stop.

So, the optimal non-adaptive value is $\frac{1}{3} \cdot \min\{6-\epsilon, 5+2\epsilon, 7-4\epsilon, 9-6\epsilon, 6\}$. Setting $\epsilon = \frac{1}{3}$, the non-adaptive optimum is $\frac{17}{9}$, whereas the adaptive optimum is $\frac{16}{9}$.

Hence, the adaptivity gap for SMQ is at least $\frac{17}{16}$.

Remark C.1. We note that if we allow $\Pr(X_2 = 1) = \epsilon_2 < \epsilon_3 = \Pr(X_3 = \infty)$ then we can achieve a ratio that is equal to $\frac{12}{11}$ as $\epsilon_2 \to 0$ and $\epsilon_3 = 0.5$.

D Fixed Threshold Problem

Here, we prove Proposition 1.2. We proceed by induction on the budget k. For any set S of r.v.s and budget k, let

$$V(S,k) := \max_{\mathcal{A} \subseteq S, c(\mathcal{A}) \leq k} \Pr_{\mathcal{A},X}\left[\min_{j \in \mathcal{A}} X_j \leq \theta\right],$$

denote the maximum success probability over adaptive policies (having cost at most k). Similarly,

$$F(S,k) = \max_{T \subseteq S : c(T) \leq k} \Pr_X\left[\min_{j \in T} X_j \leq \theta\right]$$

be the maximum over non-adaptive policies. We will show that $V(S,k) = F(S,k)$, which would prove Proposition 1.2. It suffices to show $V(S,k) \le F(S,k)$. (Clearly, $V(S,k) \ge F(S,k)$ as adaptive policies capture all non-adaptive policies.)

The base case ($k = 1$) is trivial because any policy is non-adaptive (it selects a single r.v.).

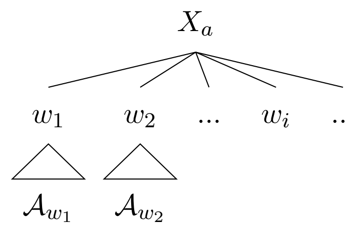


Figure 7: Adaptive policy $\mathcal{A}$ for the fixed threshold problem.

For the inductive step, we fix some budget ℓ and want to show $V(S,\ell) \le F(S,\ell)$. For any policy π, we will use $prob(\pi) := \Pr[\min_{i\in\pi} X_i \le \theta]$ to denote its success probability. Let $\mathcal{A}$ denote the optimal adaptive policy, which has $prob(\mathcal{A}) = V(S,\ell)$. Let $a \in S$ denote the first query in policy $\mathcal{A}$. Let T^+ (resp. T^-) represent all realizations of X_a that are at most (resp. more than) threshold θ. For any realization w of X_a, let $\mathcal{A}_w$ denote the rest of policy $\mathcal{A}$ *conditioned* on $X_a = w$; note that the cost $c(\mathcal{A}_w) \le \ell - c_a$ because policy $\mathcal{A}$ always has cost at most ℓ. See Figure 7. Below, we use $p_a := \Pr[X_a \le \theta] = \sum_{w\in T^+} \Pr[X_a = w]$; so $\sum_{w\in T^-} \Pr[X_a = w] = 1 - p_a$. We now have:

$$\begin{aligned} V(S,\ell) = prob(\mathcal{A}) &= p_a + \sum_{w\in T^-} \Pr[X_a = w]\cdot prob(\mathcal{A}_w) \\ &\le p_a + \sum_{w\in T^-} \Pr[X_a = w]\cdot V(S\setminus a, \ell - c_a) = p_a + (1-p_a)\cdot V(S\setminus a, \ell - c_a) \qquad (15) \\ &\le p_a + (1-p_a)\cdot F(S\setminus a, \ell - c_a) \le F(S,\ell) \qquad (16) \end{aligned}$$

The inequality in (15) uses the fact that each $\mathcal{A}_w$ is a feasible adaptive policy for the smaller instance on r.v.s $S \setminus a$ and budget $\ell - c_a$. The first inequality in (16) is by induction. The second inequality in (16) is by the following observation. Let $T \subseteq S \setminus a$ be an optimal non-adaptive policy for the instance $F(S\setminus a, \ell - c_a)$; then $T \cup a$ is a feasible non-adaptive policy for the instance $F(S,\ell)$ with success probability $p_a + (1-p_a)\cdot prob(T) = p_a + (1-p_a)\cdot F(S\setminus a, \ell - c_a)$.

E The Knapsack Subroutine (KP)

We now provide a bi-criteria approximation algorithm for the knapsack instance (KP).

Theorem E.1. *Given discrete random variables $\{X_i\}_{i=1}^n$ with costs $\{c_i\}_{i=1}^n$, budget d and threshold $\theta \in \mathbb{R}$, there is an $n^{O(1/\epsilon)}$ time algorithm that finds $T \subseteq N$ such that $\Pr[\min_{j\in T} X_j > \theta] \le p^*$ and $c(T) \le (1+\epsilon)d$, for any $\epsilon > 0$. Here,*

$$p^* = \min_{T\subseteq N} \left\{ \Pr\left[\min_{j\in T} X_j > \theta\right] : c(T) \le d \right\}. \qquad (*)$$

Proof. TOPROVE 12 $\square$