

Probing to Minimize

Weina Wang¹ 

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

Anupam Gupta¹ 

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

Jalani K. Williams¹ 

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We develop approximation algorithms for set-selection problems with deterministic constraints, but random objective values, i.e., stochastic probing problems. When the goal is to maximize the objective, approximation algorithms for probing problems are well-studied. On the other hand, few techniques are known for minimizing the objective, especially in the adaptive setting, where information about the random objective is revealed during the set-selection process and allowed to influence it. For minimization problems in particular, incorporating adaptivity can have a considerable effect on performance. In this work, we seek approximation algorithms that compare well to the optimal adaptive policy.

We develop new techniques for adaptive minimization, applying them to a few problems of interest. The core technique we develop here is an approximate reduction from an adaptive expectation minimization problem to a set of adaptive probability minimization problems which we call *threshold problems*. By providing near-optimal solutions to these threshold problems, we obtain bicriteria adaptive policies.

We apply this method to obtain an adaptive approximation algorithm for the MIN-ELEMENT problem, where the goal is to adaptively pick random variables to minimize the expected minimum value seen among them, subject to a knapsack constraint. This partially resolves an open problem raised in [5]. We further consider three extensions on the MIN-ELEMENT problem, where our objective is the sum of the smallest k element-weights, or the weight of the min-weight basis of a given matroid, or where the constraint is not given by a knapsack but by a matroid constraint. For all three of the variations we explore, we develop adaptive approximation algorithms for their corresponding threshold problems, and prove their near-optimality via coupling arguments.

2012 ACM Subject Classification Theory of computation → Stochastic approximation; Mathematics of computing → Combinatorial optimization

Keywords and phrases approximation algorithms, stochastic probing, minimization

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.120

Related Version *Full Version:* <https://arxiv.org/abs/2111.01955> [14]

Funding AG was supported in part by NSF awards CCF-1907820, CCF-1955785, and CCF-2006953. WW and JW were supported in part by NSF award CNS-2007733. JW was supported in part by the GEM Fellowship and the Gates Millennium Scholars Program.

1 Introduction

Consider the following stochastic optimization problem: given a collection X_1, X_2, \dots, X_n of independent non-negative r.v.s (random variables), with each r.v. X_e having an associated cost c_e and a known probability distribution, pick a subset $S \subseteq [n]$ (from a collection of

¹ Denotes random author order.

120:2 Probing to Minimize

feasible sets $\mathcal{F} \subseteq 2^{[n]}$) that minimizes

$$f_{\min}(S) \triangleq \mathbb{E} \left[\min_{e \in S} X_e \right];$$

more generally, given a “well-behaved” function f , pick a set $S \in \mathcal{F}$ to minimize

$$\mathbb{E}[f(S)].$$

Such a *constrained stochastic minimization* problem may arise in many contexts. For example, the government might have \$100 billion to disburse for vaccine research, and seek to produce a viable vaccine as quickly as possible. Given that each company e requests a funding amount c_e and, upon receiving funding, produces a viable vaccine in time X_e , the government might seek to fund companies in a way that minimizes the time it takes for at least one company to produce a viable vaccine. Alternatively, consider a home-repair setting, where among a large pool of possible repairmen, a homeowner might be willing to ask for at most k price estimates for a home-repair, and seek to minimize her final price paid; or a collection setting, where among a pool of 4000 possible sellers, a collector might purchase multiple copies of the same collectible from various distributors, and seek to minimize the number of defects in the best copy. In all of these settings, there is a natural tradeoff between quality assurance and constraint impact which makes these problems hard: often, riskier prospects are “cheaper” but could pay off handsomely, while more stable prospects are either more expensive or have a smaller possible upside.

While there has been considerable work on *maximization* problems in this setting (see, e.g., [7, 6, 9]), few algorithms and techniques are known for *minimization* problems. Indeed, the minimization problems appear much harder. One exception is the work of Goel et al. In [5], they consider the MIN-ELEMENT problem, in which one chooses a set S to minimize $\mathbb{E}[\min_{e \in S} X_e]$, subject to S satisfying a knapsack constraint $\sum_{e \in S} c_e \leq B$. Goel et al. consider the *non-adaptive* version of this MIN-ELEMENT problem, where one commits to a set S of variables before seeing any of the outcomes $\{X_e\}_{e \in S}$. They give a non-adaptive *bicriteria* approximation to the optimal non-adaptive policy, i.e., a set S whose cost exceeds the budget B by an $O(\log \log m)$ factor (where the r.v.s take on values in the set $\{0, \dots, m\}$), such that $\mathbb{E}[\min_{e \in S} X_e]$ is at most $(1 + \varepsilon)$ times the optimum.

Such non-adaptive solutions are particularly interesting when they have a small *adaptivity gap*, i.e., when non-adaptive solutions have a performance close to the best adaptive solutions. An *adaptive* solution builds its set S element-by-element: the outcome of an r.v.’s weight is revealed immediately after it is added to the selection set S , and the selection policy can then *adapt* and make future selections based on all the outcomes seen so far. Unfortunately, simple examples show that the MIN-ELEMENT problem can have a large adaptivity gap: see §A for a case with three random variables such that the best non-adaptive solution is arbitrarily worse than the best adaptive one.

Given such a large gap, it becomes interesting to *find adaptive solutions which compare well to the optimal adaptive policy*, so that we do not pay the price of the huge adaptivity gap. In [5], Goel et al. mention getting good adaptive solutions as a (potentially challenging) open problem; our work does exactly this.

1.1 Our results

We focus on providing efficiently computable bicriteria guarantees. We call an algorithm π an (α, β) -approximation if the selection set $S^{(\pi)}$ is the union of at most β feasible sets, and the objective value $\mathbb{E}[f(S^{(\pi)})]$ is at most α times the optimal expected objective value. Henceforth, $[m]_{\geq 0} \triangleq \{0, 1, 2, \dots, m\}$.

Our first main result is for the MIN-ELEMENT problem in the adaptive setting; to differentiate it from the extensions we consider next, we call it the MIN-ELEMENT-KNAPSACK problem.

► **Theorem 1.** *For the MIN-ELEMENT-KNAPSACK problem where the random variables take values in $[m]_{\geq 0}$, there exists an adaptive policy that obtains a $(4, O(\log \log m))$ -approximation.*

Theorem 1 provides a partial answer to an open question in [5]. Interestingly, our $O(\log \log m)$ resource augmentation factor *matches* that obtained by the authors of [5] in their result for the non-adaptive setting. However, these factors seem to arise for different reasons: their non-adaptive factor comes from a use of greedy submodular optimization, while ours comes from a new *adaptive binary search* procedure. This procedure, which reduces the original adaptive *expectation minimization* problem to an adaptive *probability minimization* problem, is a basic building block in our approach.

The MIN-ELEMENT-KNAPSACK can be generalized in many ways: we can consider *richer function classes* (and not just the minimum function), and we can consider *richer constraint sets* (and not just the simplest knapsack setting). In the first extension we present, the objective function $f(S)$ is the sum of the smallest k outcomes in S . We call this the MIN- k -KNAPSACK problem.

► **Theorem 2.** *For the MIN- k -KNAPSACK problem where the random variables take values in $[m]_{\geq 0}$, there exists an adaptive policy that obtains an $(8, O(\log \log m \cdot \log^2 k))$ -approximation.*

To prove Theorem 2, we use the same adaptive binary search idea from Theorem 1, and relate the resultant probability minimization problem to a different probability maximization problem that we call the *i-heads problem*, which is an interesting problem in its own right. We then obtain a better-than-optimal solution for this problem by extending the simple greedy algorithm used for Theorem 1, albeit overspending by an $O(\log k)$ factor. A detailed discussion appears in the techniques section.

Next we consider the MIN- k -MATROID problem, where the objective function $f(S)$ is again the sum of the smallest k outcomes in the selection set S , but the constraint is now a matroid constraint.² Our solution for this setting has an even better approximation than for the case of knapsack constraints.

► **Theorem 3.** *For the MIN- k -MATROID problem where the random variables take values in $[m]_{\geq 0}$, there exists an adaptive policy that obtains a $(8, O(\log \log m \cdot \log k))$ -approximation.*

Theorem 3 takes the techniques developed for the MIN- k -KNAPSACK problem, and shows the core reason why those techniques work: framed correctly, both constraints admit a nice sense of interchangeability. For the adaptive probability minimization problem we reduce to here, a *non-adaptive* greedy algorithm actually becomes *optimal*.

Finally, we investigate the MINBASIS-CARDINALITY problem, where the constraint is a cardinality constraint (we can pick at most B elements), but we *generalize* the objective function from being the sum of the smallest k random variables to the setting of matroids. Specifically, we now have a matroid $\mathcal{M}_\mathbf{I} = (\mathcal{U}, \mathcal{I})$ of rank k , and we consider the minimum-weight basis in this matroid. One should note that taking a uniform matroid as the constraint here gives us back the MIN- k -KNAPSACK problem, albeit in the case where all items have unit cost.

² A matroid $\mathcal{M}_\mathbf{O} = (\mathcal{U}, \mathcal{I})$ is specified by a ground set \mathcal{U} and a family of independent sets \mathcal{I} . A selection set $S^{(\pi)}$ is feasible if and only if it is an independent set, i.e., $S^{(\pi)}$ lies in the family of feasible sets $\mathcal{F} = \mathcal{I}$.

► **Theorem 4.** *For the MINBASIS-CARDINALITY problem where the random variables takes value in $[m]_{\geq 0}$, there exists an adaptive policy that obtains a $(8, O(\log \log m \cdot \log k))$ -approximation.*

While the result of Theorem 3 shows the importance of interchangeability in the constraint, Theorem 4 shows the importance of interchangeability in the objective. When the matroid constraint moves into the objective, the previously optimal non-adaptive matroid greedy algorithm now needs to be made *adaptive* in order to maintain optimality. We note a peculiarity here: while solving the non-adaptive probability minimization problem is non-trivial, the *adaptive* version of this problem has a simpler optimal solution.

1.2 Techniques

We now give some more details about the main techniques behind our results.

1.2.1 Reduction to threshold problems

The primary technique we use is a reduction to threshold problems. The idea is a clean one, provided we observe that, since $f(S)$ is a random variable taking values in $[m]_{\geq 0}$,

$$\mathbb{E}[f(S^{(\pi)})] \leq \Pr(f(S^{(\pi)}) > 0) + \sum_{j=0}^{\lfloor \log m \rfloor} \Pr(f(S^{(\pi)}) > 2^j) 2^j \leq 2\mathbb{E}[f(S^{(\pi)})]; \quad (1)$$

This suggests that a good policy for minimizing $\mathbb{E}[f(S^{(\pi)})]$ should somehow simultaneously minimize all of these $\Pr(f(S^{(\pi)}) > t)$ terms for values of t that are powers of 2. Now if we have an algorithm, referred to as **THRESHOLD**, which can (α, β) -approximate these threshold problems for every power-of-2 threshold, we obtain a $(2\alpha, O(\log m)\beta)$ -approximation to the optimal adaptive policy: first, use **THRESHOLD** to approximate *all* of these threshold problems, then combine their solutions. Since we require that $f(\cdot)$ is non-increasing, our combined set inherits all the guarantees of its subsets. But this is incredibly wasteful, and does not use the adaptivity. To avoid this loss, we again use that $f(\cdot)$ is non-increasing; if we obtain $f(S^{(\pi)}) \leq t$, we no longer need to worry about solving threshold problems with thresholds larger than t . To make use of this observation, we instead perform an *adaptive binary search* to determine the next threshold to call **THRESHOLD** on.

Adaptive binary search. Starting with the median threshold t , **THRESHOLD** probes a set S_t that aims to minimize the probability of $f(S_t) > t$. If **THRESHOLD** obtains $f(S_t) \leq t$, we say it succeeds and we recurse on the thresholds smaller than t ; if **THRESHOLD** fails to obtain $f(S_t) \leq t$, we recurse on the set of thresholds larger than t . While this seems simple, one should note the asymmetry between success and failure here: success at a threshold t guarantees that our final objective value is at most t , but failure at t *does not guarantee that the final objective value is greater than t* .

Nevertheless, we construct an upper bound **UB** on our algorithm's objective value, which allows us to analyze it as if failure at t *does* imply that the objective is at least t from then on. See §2 for details. This adaptive binary search now reduces the number of calls to $O(\log \log m)$.

1.2.2 The Min-Element-Knapsack problem

Armed with this reduction to a threshold problems, the proof of Theorem 1 follows from an approximation to the MIN-ELEMENT-KNAPSACK threshold problem below, where $\mathbf{Adm}(\mathcal{F})$ denotes the set of admissible policies with \mathcal{F} being the collection of feasible sets for selection:

$$\min_{\pi \in \mathbf{Adm}(\mathcal{F})} \Pr \left(\min_{e \in S^{(\pi)}} X_e > t \right).$$

Since any policy π succeeds at obtaining $\min_{e \in S} X_e \leq t$ as soon as it finds a single element for which $X_e \leq t$, adaptivity adds nothing to solving this problem. Hence we focus on the non-adaptive problem

$$\min_{S \in \mathcal{F}} \Pr \left(\min_{e \in S} X_e > t \right).$$

Taking the logarithm of the objective, we reduce to a knapsack instance where each element has *reward* equal to $-\log \Pr(X_e > t)$. Now a greedy procedure gives a $(1, 2)$ -approximation to the threshold problem. Combining all these together gives the $(4, O(\log \log m))$ -approximation from Theorem 1.

1.2.3 The Min- k -Knapsack problem

We turn next to a strict extension of the MIN-ELEMENT-KNAPSACK problem: the MIN- k -KNAPSACK problem. The first step, like in the MIN-ELEMENT-KNAPSACK problem, is to apply a threshold-based reduction. An idea similar to that from §1.2.1 allows us to reduce a slightly richer set of objective functions to threshold minimization problems.

Indeed, if $y_i(S)$ is the weight of the i -th smallest weight element in S , then our objective $f(S)$ is $\sum_{i=1}^k y_i(S)$, the sum of the k smallest weights in S . Writing the expectation as a sum of thresholds like in (1), we get that it suffices to find approximations to the threshold problem

$$\min_{\pi \in \mathbf{Adm}(\mathcal{F})} \Pr \left(y_i(S^{(\pi)}) > t \right)$$

for different values of t . This problem can be thought of in the following way.

► **Problem (The i -heads problem).** *We are given a set of coins \mathcal{U} , each having cost c_e and bias p_e . We can flip a coin e at most once, at a cost of c_e . Given a budget B , the i -heads problem seeks to adaptively flip coins to maximize $\Pr(\text{see at least } i \text{ heads})$.*

We extend the greedy knapsack strategy from §1.2.2 used to solve the case $i = 1$ to the case of a general i : the idea is again to pick items in increasing order of cost-per-reward, except that the budget is extended from B to $B + i\delta$, where δ is the maximum cost of any element. In order to control this δ term, we bucket the coins and choose the best coins at each of the top $O(\log i)$ levels until the cost of each coin becomes smaller than B/i .

The proof that this greedy policy G outperforms any admissible policy π requires some new ideas. The heart of the argument is to first construct a surrogate policy $\tilde{\pi}$ which strictly outperforms any policy π , and then to show an ordering of the coins of G such that

- The policies G and $\tilde{\pi}$ probe the same set after obtaining their first heads.
- For any amount of remaining budget B' , the probability that G obtains its first heads with B' budget remaining is greater than the probability that π obtains its first heads with B' budget remaining.

In the end, our argument shows that G stochastically dominates every admissible policy π , which completes the proof.

1.2.4 The Min- k -Matroid problem

We now discuss our first matroid-based variation on the MIN- k -KNAPSACK problem, which we call the MIN- k -MATROID problem. The same reduction used in the MIN- k -KNAPSACK problem implies that proving Theorem 3 reduces to providing a $(1, O(1))$ -approximation to threshold problem for matroids. We show that the matroid greedy algorithm MGREEDY is an optimal policy for this problem. In fact, the matroid greedy algorithm stochastically dominates any other policy. We argue this via induction on the matroid rank. The main observation we make here is that, after probing a single element, we find ourselves in a problem state nearly exactly like the starting state, but with a possibly reduced number of heads, and a resulting constraint family which is a matroid with a strictly smaller rank. Due to space limitations, we defer this section to the extended version [14].

1.2.5 The MinBasis-Cardinality problem

We now discuss the treatment of our final problem, the MINBASIS-CARDINALITY problem. The proof of Theorem 4 is again through the reduction to threshold problems for an objective $f(S)$ that is in the form of $\sum_{i=1}^k g_i(S)$. However, unlike the case of a MIN- k objective, now it is not obvious how we can write the MINBASIS objective $f(S)$ as $\sum_{i=1}^k g_i(S)$ for a monotonic function sequence $(g_i(\cdot))$ where each $g_i(\cdot)$ is a non-decreasing function taking values in $[m]_{\geq 0}$. In our proof, we first show that we can specify $g_i(S)$ to be the i -th smallest element in the minimum-weight basis contained in S generated by the standard matroid greedy algorithm. Then we show inductively that ADAPMGREEDY, a simple adaptive matroid greedy algorithm, is optimal for the resulting threshold problem. The key idea here is again to use our inductive assumption to reason about the future behavior of an optimal policy. As we did for the MIN- k -MATROID problem, we defer this section to the extended version in [14].

1.3 Related Work

The work closest to ours in content and aim is [5]. That paper allows a more general model, where element weights can take on non-negative discrete values $\mu_1, \mu_2, \dots, \mu_L$, but restricts it focus to the MIN-ELEMENT objective $f(S) \triangleq \min_{e \in S} X_e$. The authors give a bicriteria approximation scheme, showing for every $\varepsilon > 0$ how to obtain a set T such that $\text{cost}(T) \leq O\left(\log(L) + \log \frac{1}{\log(1+\varepsilon)}\right)$ with $\mathbb{E}[f(T)] \leq (1 + \varepsilon)\mathbb{E}[f(S^*)]$ where $S^* \triangleq \arg \min_{S \in \mathcal{F}} \mathbb{E}[f(S)]$. Moreover, they show that finding a $\text{poly}(L)$ approximation to the non-adaptive problem is NP-hard, by establishing an approximation-preserving reduction to a special class of covering integer programs.

Our results differ in several ways. First, the authors of [5] consider non-adaptive approximations to the optimal non-adaptive policy, whereas we provide adaptive approximations to the optimal adaptive policy, partially closing an open question posed by them. Second, they obtain a $(1 + \varepsilon)$ -approximation instead of a constant-factor approximation. Third, their techniques are quite different from ours: they use the submodularity of (a transformation of) the expected objective $\mathbb{E}[f(S)]$, while we adaptively explore elements through an adaptive binary search. Moreover, we consider several generalizations of the MIN-ELEMENT problem with more general objectives and constraints.

Stochastic probing *maximization* problems have been studied much more broadly: the results for these problems often show small adaptivity gaps and focus on particular classes of objective functions [1, 8, 2, 6]. In contrast, we show that one of our minimization problems has a large adaptivity gap, and our main technical result, the reduction to threshold problems, applies to a general class of objectives. Moreover, the approaches used for these problems and for ours are quite different.

There has been some work on adaptive minimization problems outside of the stochastic probing contexts. One example is the stochastic k-TSP model introduced by Ene et al. in [4]. In that setting, each vertex v in a given metric contains a stochastic reward R_v , and the goal is to adaptively minimize the expected cumulative distance one travels before obtaining k reward. In that introductory paper, Ene et al. gave an adaptive $O(\log k)$ -approximation and a non-adaptive $O(\log^2 k)$ -approximation for this problem. Much like we do, they solve their problem by repeatedly approximating an associated “dual” problem; in our case, this is a threshold problem, and in their case, this is an orienteering problem. In a later work [11], Jiang et al. extended this result, using some clever modifications to obtain an $O(1)$ -approximation.

Another good example is the general disutility-minimization problem covered by the Price-of-Information framework, introduced by Singla in [13]. In their setting, like ours, one must choose a set of items to probe, where each item has a fixed probing price and a random disutility value, and, once done probing, one must choose a minimum disutility subset from within the probed set, where this subset must satisfy some given covering constraint. In contrast to our setting, though, in the Price-of-Information framework the probing constraints are soft constraints; one seeks to minimize the *sum* of the probing costs and the final disutility value. As such, their main result, showing that simple modifications of standard greedy algorithms form good approximations, can no longer be applied.

With respect to solving threshold problems, to the authors’ knowledge, generally little is known. However, the threshold problem associated with the MIN- k -KNAPSACK problem (see § 4) has been heuristically treated before. In particular, it is a Bernoulli version of a more general adaptive knapsack problem, where rewards are independently random but the cost of each item is fixed, and the goal is to achieve a cumulative reward above some threshold with optimal probability. Previous work ([3], [10]) on this problem has focused on the development of heuristics and on optimizing the dynamic programming solution, in the setting where rewards are normally distributed. This work is instead concerned with the development of bicriteria approximation algorithms in the setting where rewards are Bernoulli distributed, though it may be possible to extend our methods to the more general case.

1.4 Open Problems

Our results leave open many questions, a few of which we explicitly pose here. Our first focus is the tightness of Lemma 5, our reduction to threshold problems. Is the $\log \log m$ in our resource augmentation factor tight? That is, assuming we use THRESHOLD in a black-box way, is there a better way to choose our threshold values than adaptive binary search? Alternatively, is it possible to modify our algorithm to obtain a bicriteria approximation scheme, as was done in [5]?

Our next focus is on hardness. In [5], the non-adaptive version of the MIN-ELEMENT problem was shown to be hard to approximate without relaxing the budget, by giving an approximation-preserving reduction to a class of hard integer programs. Does a similar result exist for the adaptive version, or any of its extensions? We leave all of these questions as challenging open problems.

2 General Framework with Threshold Problems

In this section, we present the full details of the reduction from the stochastic minimization problem to threshold problems, first alluded to in Section 1.2.

120:8 Probing to Minimize

Formally, the f - \mathcal{F} threshold problem with threshold t is an adaptive probability minimization problem, wherein the policy π tries to avoid $f(S^{(\pi)}) > t$ with as high a probability as possible, while abiding by the constraint that $S^{(\pi)} \in \mathcal{F}$. Lemma 5 says that, given an (α, β) -approximation for the f - \mathcal{F} threshold problem, one can construct a $(4\alpha, O(\log \log m)\beta)$ -approximation for the adaptive expectation minimization problem $\min_{\pi \in \mathbf{Adm}(\mathcal{F})} \mathbb{E}[f(S^{(\pi)})]$. Our results are summarized in Lemma 5 and Corollary 6, which we state now.

► **Lemma 5 (Reduction to threshold problems).** *Let $f(S)$ be a non-increasing objective function taking values in $[m]_{\geq 0}$ for a positive integer m . Let \mathcal{F} be the constraint family and $\mathbf{Adm}(\mathcal{F})$ be the set of feasible policies under \mathcal{F} . Suppose that for any $t \in [m]_{\geq 0}$, the f - \mathcal{F} threshold problem*

$$\min_{\pi \in \mathbf{Adm}(\mathcal{F})} \Pr(f(S^{(\pi)}) > t)$$

admits an (α, β) -approximation. Then the stochastic minimization problem

$$\min_{\pi \in \mathbf{Adm}(\mathcal{F})} \mathbb{E}[f(S^{(\pi)})]$$

admits a $(4\alpha, O(\log \log m) \cdot \beta)$ -approximation.

► **Corollary 6 (Sum-of- k reduction).** *Let $g_i(S)$, $i \in [k]$, be a non-increasing function taking values in $[m]_{\geq 0}$, and assume that the function sequence (g_i) is monotonic in i . Let $f(S) = \sum_{i=1}^k g_i(S)$. Suppose that for any $t \in [m]_{\geq 0}$, the g_i - \mathcal{F} threshold problem*

$$\min_{\pi \in \mathbf{Adm}(\mathcal{F})} \Pr(g_i(S^{(\pi)}) > t)$$

admits an (α, β) -approximation. Then the stochastic minimization problem

$$\min_{\pi \in \mathbf{Adm}(\mathcal{F})} \mathbb{E}[f(S^{(\pi)})]$$

admits an $(8\alpha, O(\log \log m \cdot \log k) \cdot \beta)$ approximation.

As described in Section 1.2, to build the intuition for the reduction, we start with the decomposition (1): $\mathbb{E}[f(S^{(\pi)})] \leq \Pr(f(S^{(\pi)}) > 0) + \sum_{j=0}^{\lfloor \log m \rfloor} \Pr(f(S^{(\pi)}) > 2^j)2^j \leq 2\mathbb{E}[f(S^{(\pi)})]$. This bound follows from two basic observations: the relation $\mathbb{E}[f(S^{(\pi)})] = \sum_{t=0}^m \Pr(f(S^{(\pi)}) > t)$, which follows from the fact that $f(S) \in [m]_{\geq 0}$, and the bound $\sum_{t=0}^m a_t \leq a_0 + \sum_{j=0}^{\lfloor \log m \rfloor} a_{2^j}2^j \leq 2\sum_{t=0}^m a_t$, which applies to any non-increasing non-negative sequence $(a_t)_{t=0}^m$. This powers-of-2 bound is a slight modification of the bound used in the proof of Cauchy's condensation test in [12].

With (1) in mind, consider the following policy $\widehat{\pi}$. Let \widehat{S}_t be an (α, β) -approximation to the threshold problem $\min_{\pi \in \mathbf{Adm}(\mathcal{F})} \Pr(f(S^{(\pi)}) > t)$, and let $S^{(\widehat{\pi})} = \widehat{S}_0 \cup \widehat{S}_{2^0} \cup \widehat{S}_{2^1} \cup \dots \cup \widehat{S}_{2^{\lfloor \log m \rfloor}}$. Then

$$\begin{aligned} \mathbb{E}[f(S^{(\widehat{\pi})})] &\leq \Pr(f(S^{(\widehat{\pi})}) > 0) + \sum_{j=0}^{\lfloor \log m \rfloor} \Pr(f(S^{(\widehat{\pi})}) > 2^j)2^j \\ &\leq \Pr(f(\widehat{S}_0) > 0) + \sum_{j=0}^{\lfloor \log m \rfloor} \Pr(f(\widehat{S}_{2^j}) > 2^j)2^j \end{aligned}$$

$$\begin{aligned} &\leq \alpha \Pr(\text{OPT} > 0) + \sum_{j=0}^{\lfloor \log m \rfloor} \alpha \Pr(\text{OPT} > 2^j) 2^j \\ &\leq 2\alpha \mathbb{E}[\text{OPT}], \end{aligned}$$

where OPT is the value of an optimal policy for $\min_{\pi \in \text{Adm}(\mathcal{F})} \mathbb{E}[f(S^{(\pi)})]$, and we have used the fact that $f(\cdot)$ is non-increasing. Unfortunately, the policy $\hat{\pi}$ uses a resource augmentation factor of the order of $\log m$. In the proof of Lemma 5, we further reduce the resource augmentation factor using the policy METAMIN, described in Algorithm 1, which performs an adaptive binary search on the threshold values $R = \{0, 2^0, 2^1, \dots, 2^{\lfloor \log m \rfloor}\}$. Note that METAMIN requires as input a function $\text{THRESHOLD}(t)$, which, for any $t \in [m]_{\geq 0}$, generates an (α, β) -approximation to the $f - \mathcal{F}$ threshold problem with threshold value t .

■ **Algorithm 1** METAMIN(THRESHOLD).

```

1: Initialization:  $R \leftarrow \{0\} \cup \{2^j : j = 0, 1, \dots, \lfloor \log m \rfloor\}; \hat{S} \leftarrow \emptyset$ 
2:  $\hat{S}_0 \leftarrow \text{THRESHOLD}(0)$  ▷ Boundary case
3:  $\hat{S} \leftarrow \hat{S} \cup \hat{S}_0$ 
4: if  $f(\hat{S}_0) = 0$  then ▷ Succeeds the threshold test at 0
5:   return  $\hat{S}$ 
6: else ▷ Fails the threshold test at 0
7:    $R \leftarrow R \setminus \{0\}$ 
8: while  $R \neq \emptyset$  do ▷ Adaptive threshold testing
9:    $t \leftarrow \text{median of } R$ 
10:   $\hat{S}_t \leftarrow \text{THRESHOLD}(t)$ 
11:   $\hat{S} \leftarrow \hat{S} \cup \hat{S}_t$ 
12:  if  $f(\hat{S}_t) \leq t$  then ▷ Succeeds the threshold test at  $t$ 
13:     $R \leftarrow R \setminus \{\tau \in R : \tau \geq t\}$ 
14:  else ▷ Fails the threshold test at  $t$ 
15:     $R \leftarrow R \setminus \{\tau \in R : \tau \leq t\}$ 
16: return  $\hat{S}$ 

```

2.1 Proof of Lemma 5 (Reduction to threshold problems)

Proof. We prove Lemma 5 by showing that the policy METAMIN in Algorithm 1 is a $(4\alpha, O(\log \log m) \cdot \beta)$ -approximation. It is easy to see that the final set $S^{(\text{METAMIN})}$ is the union of at most $(O(\log \log m) \cdot \beta)$ \mathcal{F} -feasible sets, since METAMIN performs a binary search on $|R| = O(\log m)$ threshold values. Using METAMIN to refer to the random objective value obtained by running METAMIN, it suffices to show that $\mathbb{E}[\text{METAMIN}] \leq 4\alpha \mathbb{E}[\text{OPT}]$.

We first construct an upper bound UB on the objective value METAMIN whose value directly reflects the failing or succeeding of threshold tests. Specifically, let τ be the last threshold at which METAMIN fails, and we define $\text{UB} = \max\{\text{METAMIN}, \tau + 1\}$. Clearly, $\text{METAMIN} \leq \text{UB}$. Note that τ is also the *largest* threshold at which METAMIN fails during the binary search procedure, so if METAMIN fails the test at threshold value t , then $\text{UB} > t$.

We now show how the constructed upper bound UB reflects the test results. For notational convenience, we partition the interval $[0, m]$ into the following intervals:

$$[0, m] = \{0\} \cup (0, 1] \cup (1, 2] \cup (2, 2^2] \cup \dots \cup (2^{\lfloor \log m \rfloor - 1}, 2^{\lfloor \log m \rfloor}] \cup (2^{\lfloor \log m \rfloor}, m],$$

120:10 Probing to Minimize

which we denote as

$$\begin{aligned}\mathcal{I}_{-1} &= (0, 1] \triangleq (a_{-1}, b_{-1}]; \quad \mathcal{I}_j = (2^j, 2^{j+1}] \triangleq (a_j, b_j], \quad j = 0, 1, \dots, \lfloor \log m \rfloor - 1; \\ \mathcal{I}_{\lfloor \log m \rfloor} &= (2^{\lfloor \log m \rfloor}, m] \triangleq (a_{\lfloor \log m \rfloor}, b_{\lfloor \log m \rfloor}].\end{aligned}$$

We use the phrase that “METAMIN succeeds/fails the test at a threshold” to mean that “METAMIN indeed *performs* the test and *succeeds/fails*”.

The key to our proof is the claim below, which directly links the value of UB with behavior of these threshold tests. Note that the same relation does not generally hold when UB is replaced by METAMIN, i.e., with $\Pr(\text{METAMIN} \in (a_j, b_j])$ on the left side. It is possible that METAMIN fails at some a_j but ultimately attains a value smaller than a_j , if it happens to observe a small $f(\hat{S}_t)$ when it succeeds at later threshold t .

▷ **Claim.** For any $j \in \{-1, 0, 1, \dots, \lfloor \log m \rfloor\}$,

$$\Pr(\text{UB} \in (a_j, b_j]) = \Pr(\text{METAMIN fails at } a_j \text{ and succeeds at } b_j).$$

Proof of Claim. It suffices to focus on the case where METAMIN fails the test at threshold 0. We first argue that if METAMIN fails at a_j and succeeds at b_j , then $\text{UB} \in (a_j, b_j]$. By the construction of UB, we have seen that if METAMIN fails the test at a_j , then $\text{UB} \geq a_j + 1 > a_j$. If METAMIN succeeds the test at b_j , then $\text{METAMIN} \leq b_j$, and METAMIN does not perform tests at threshold values above b_j and thus does not have a chance to fail at these threshold values. Therefore, $\text{UB} \leq b_j$.

We now show that if $\text{UB} \in (a_j, b_j]$, then METAMIN fails at a_j and succeeds at b_j . After METAMIN terminates, let t_F be the largest threshold that METAMIN has failed at, and let t_S be the smallest threshold that METAMIN has succeeded at. We argue that it must hold that $t_F = a_j$ and $t_S = b_j$, which is sufficient for the claim. By the termination condition, $(t_F, t_S]$ must be an interval among the class of intervals $\{(a_u, b_u] : u = -1, 0, 1, \dots, \lfloor \log m \rfloor\}$. By the argument in the last paragraph, the fact that METAMIN fails at t_F and succeeds at t_S implies that $\text{UB} \in (t_F, t_S]$. Since the intervals $(a_u, b_u]$'s are disjoint, it must be that $(t_F, t_S]$ is the same interval as $(a_j, b_j]$, which completes the proof. \triangleleft

With this claim, we complete the proof of Lemma 5 through the following inequalities. First, it is easy to see that the claim implies that

$$\Pr(\text{UB} \in (a_j, b_j]) \leq \Pr(\text{METAMIN fails at } a_j) \leq \Pr(f(\hat{S}_{a_j}) > a_j),$$

where \hat{S}_{a_j} is the set chosen by THRESHOLD for the threshold problem at threshold a_j . Then

$$\begin{aligned}\mathbb{E}[\text{UB}] &\leq \sum_{j=-1}^{\lfloor \log m \rfloor} \Pr(\text{UB} \in (a_j, b_j]) \cdot b_j \leq \sum_{j=-1}^{\lfloor \log m \rfloor} \Pr(f(\hat{S}_{a_j}) > a_j) \cdot 2^{j+1} \\ &\leq \sum_{j=-1}^{\lfloor \log m \rfloor} \alpha \Pr(\text{OPT} > a_j) \cdot 2^{j+1} = \alpha \Pr(\text{OPT} > 0) + 2\alpha \sum_{j=0}^{\lfloor \log m \rfloor} \Pr(\text{OPT} > 2^j) \cdot 2^j \\ &\leq 4\alpha \mathbb{E}[\text{OPT}],\end{aligned}$$

where the last line follows from the decomposition in (1). Recalling that $\mathbb{E}[\text{METAMIN}] \leq \mathbb{E}[\text{UB}]$, we have completed the proof that METAMIN is a $(4\alpha, O(\log \log m) \cdot \beta)$ -approximation. \blacktriangleleft

2.2 Proof of Corollary 6 (Sum-of- k reduction)

Proof. Without loss of generality, we assume that the function sequence (g_i) is non-increasing in i since otherwise we can reverse their indices. Applying the same powers-of-2 condensation trick for the non-increasing sequence $g_1(S), g_2(S), \dots, g_k(S)$, we have that for any policy π ,

$$\mathbb{E}\left[\sum_{i=1}^k g_i(S^{(\pi)})\right] \leq \mathbb{E}\left[\sum_{j=0}^{\lfloor \log(k) \rfloor} g_{2^j}(S^{(\pi)}) 2^j\right] \leq 2\mathbb{E}\left[\sum_{i=1}^k g_i(S^{(\pi)})\right].$$

We now consider the following policy $\tilde{\pi}$. For each $j = 0, 1, \dots, \lfloor \log(k) \rfloor$, let S_j be the output of the policy METAMIN in Algorithm 1 for $\min_{\pi \in \text{Adm}(\mathcal{F})} \mathbb{E}[g_{2^j}(S^{(\pi)})]$. Then let $S^{(\tilde{\pi})} = S_0 \cup S_1 \cup \dots \cup S_{2^{\lfloor \log k \rfloor}}$. Recall that METAMIN is a $(4\alpha, O(\log \log m) \cdot \beta)$ -approximation. Clearly, the policy $\tilde{\pi}$ has an augmentation factor of $O(\log \log m \cdot \log k) \cdot \beta$. Using linearity of expectation along with the inequality above, it follows neatly that $\mathbb{E}\left[\sum_{i=1}^k g_i(S^{(\tilde{\pi})})\right] \leq 8\alpha \mathbb{E}[\text{OPT}]$. This completes the proof that the policy $\tilde{\pi}$ is a $(8\alpha, O(\log \log m \cdot \log k) \cdot \beta)$ -approximation. \blacktriangleleft

3 The Min-Element-Knapsack Problem

In this section, we focus on the MIN-ELEMENT-KNAPSACK problem, where the objective function is the minimum weight and the constraint is a knapsack constraint. Specifically, letting $\mathcal{F} = \{S \subseteq \mathcal{U} : \text{cost}(S) \leq B\}$, the MIN-ELEMENT-KNAPSACK problem can be written as:

$$\min_{\pi \in \text{Adm}(\mathcal{F})} \mathbb{E}\left[\min_{e \in S^{(\pi)}} X_e\right]. \quad (2)$$

Our main result is Theorem 1, restated below for convenience.

► **Theorem 1.** *For the MIN-ELEMENT-KNAPSACK problem where the random variables take values in $[m]_{\geq 0}$, there exists an adaptive policy that obtains a $(4, O(\log \log m))$ -approximation.*

By the reduction to threshold problems in Lemma 5, to prove Theorem 1, it suffices to give a $(1, 2)$ -approximation to the MIN-ELEMENT-KNAPSACK threshold problem:

$$\min_{\pi \in \text{Adm}(\mathcal{F})} \Pr\left(\min_{e \in S^{(\pi)}} X_e > t\right). \quad (3)$$

In the remainder of this section, we show that a non-adaptive algorithm for this threshold problem achieves the desired $(1, 2)$ approximation ratio. More specifically, we first show that for the MIN-ELEMENT-KNAPSACK threshold problem, a non-adaptive feasible policy achieves the adaptive optimum, i.e., the adaptivity gap is 1, or equivalently, we say that there is no adaptivity gap. We then give a $(1, 2)$ -approximation for the non-adaptive version of the threshold problem.

3.1 No adaptivity gap

► **Lemma 7.** *The MIN-ELEMENT-KNAPSACK threshold problem in (3) has an adaptivity gap of 1 for any $t \in [m]_{\geq 0}$.*

Proof. Consider an arbitrary $t \in [m]_{\geq 0}$. We induct on the size of the universe \mathcal{U} . Clearly, the adaptivity gap is 1 when \mathcal{U} consists of one element.

120:12 Probing to Minimize

Assume that the adaptivity gap is 1 when the universe consists of n elements for some $n \geq 1$; i.e., for any input of the threshold problem (3) such that the universe consists of n elements, an optimal non-adaptive policy achieves the same objective value as an optimal adaptive policy. Now consider any input such that the universe \mathcal{U} has $n + 1$ elements and let π^* be an optimal adaptive policy for it. We will construct a non-adaptive policy, represented by a set $S \subseteq \mathcal{U}$, such that

$$\mathbb{P}\left(\min_{e \in S} X_e > t\right) = \mathbb{P}\left(\min_{e \in S^{(\pi^*)}} X_e > t\right). \quad (4)$$

Without loss of generality, we can assume that X_1 is the first element probed by the optimal adaptive policy π^* since the first probing decision does not depend on realizations of the random variables. After X_1 is probed, we consider the threshold problem whose input consists of a universe $\mathcal{U}' = \mathcal{U} \setminus \{1\}$ and a budget $B' = B - c_1$. Since $|\mathcal{U}'| = n$, we know that this problem has an adaptivity gap of 1. Let $S' \subseteq \mathcal{U}'$ be the set chosen by an optimal non-adaptive policy for this problem.

We claim that $S = \{1\} \cup S'$ satisfies (4). To see this, let π_x^* for $x \in [m]_{\geq 0}$ be the subsequent policy of π^* after seeing $X_1 = x$, and let $S^{(\pi_x^*)} \subseteq \mathcal{U}'$ be the set chosen by π_x^* . Then

$$\begin{aligned} \mathbb{P}\left(\min_{e \in S^{(\pi^*)}} X_e > t\right) &= \sum_{x=t+1}^m \mathbb{P}(X_1 = x) \mathbb{P}\left(\min_{e \in S^{(\pi^*)}} X_e > t \mid X_1 = x\right) \\ &= \sum_{x=t+1}^m \mathbb{P}(X_1 = x) \mathbb{P}\left(\min_{e \in S^{(\pi_x^*)}} X_e > t\right) \end{aligned} \quad (5)$$

$$\begin{aligned} &\geq \sum_{x=t+1}^m \mathbb{P}(X_1 = x) \mathbb{P}\left(\min_{e \in S'} X_e > t\right) \\ &= \mathbb{P}(X_1 > t) \mathbb{P}\left(\min_{e \in S'} X_e > t\right) \\ &= \mathbb{P}\left(\min_{e \in S} X_e > t\right), \end{aligned} \quad (6)$$

where (5) is due to the property of the minimum value, and (6) follows from the induction assumption that the non-adaptive choice of S' achieves optimality. Since π^* is an (adaptive) optimal policy, we know that this inequality is in fact an equality, which completes the proof. \blacktriangleleft

3.2 Reduction to knapsack

By Lemma 7, to solve the threshold problem (3), it suffices to solve its non-adaptive counterpart:

$$\min_{S \in \mathcal{F}} \Pr\left(\min_{e \in S} X_e > t\right) \quad (7)$$

Note that $\Pr(\min_{e \in S} X_e > t) = \prod_{e \in S} \Pr(X_e > t)$. Taking a log of this probability, we rewrite (7) as:

$$\max_{S \in \mathcal{F}} \sum_{e \in S} (-\log \Pr(X_e > t)) \quad (8)$$

This is a knapsack problem where the reward of each element e is $-\log \Pr(X_e > t)$. Therefore, a $(1, 2)$ -approximation is given by greedily adding elements in decreasing order of $\frac{-\log \Pr(X_e > t)}{c_e}$ until the first time the total cost exceeds the budget. This completes the proof of Theorem 1.

4 The Min- k -Knapsack Problem

In this section, we give the proof of Theorem 2 alluded to in § 1.2. We restate that theorem now, for the reader's convenience.

► **Theorem 2.** *For the MIN- k -KNAPSACK problem where the random variables take values in $[m]_{\geq 0}$, there exists an adaptive policy that obtains an $(8, O(\log \log m \cdot \log^2 k))$ -approximation.*

We begin by rewriting $f(S)$. For $i \in [k]$, if $|S| \geq k$, let $y_i(S)$ be the i -th smallest weight in S . Then $f(S) = \sum_{i=1}^k y_i(S)$. Noting that the functions $y_i(S)$ are monotonic in i , non-increasing in S , and take values in $[m]_{\geq 0}$, we apply Corollary 6, reducing the MIN- k -KNAPSACK problem to the y_i -KNAPSACK threshold problem. Let $C = \text{BIN}(\mathcal{U}, B, i, t)$ be the output of the binning procedure in Algorithm 2 and let $G = \text{EXTGREEDY}(\{e \in \mathcal{U}: c_e \leq \frac{B}{i}\}, B, i, t)$ be the output of the extended greedy algorithm in Algorithm 3. To complete the proof, we show that the non-adaptive policy $G \cup C$ is a $(1, O(\log k))$ -approximation for the y_i -KNAPSACK problem. Before that though, we give an equivalent formulation of the threshold problem.

► **Definition 8.** *For a fixed threshold t , call an element e **below-threshold** if $X_e \leq t$. Define $\text{rank}(S)$ as the number of below-threshold elements contained in S , i.e., $\text{rank}(S) = |\{e \in S : X_e \leq t\}|$.*

It follows from definitions that $y_i(S) \leq t$ if and only if $\text{rank}(S) \geq i$, since both conditions imply and are implied by the presence of i below-threshold elements in S . Using the rank-based condition, the y_i -KNAPSACK threshold problem

$$\min_{\pi \in \text{Adm}(\mathcal{F})} \Pr\left(y_i(S^{(\pi)}) > t\right). \quad (9)$$

is equivalent to the following problem, which we call the *i-th rank problem*

$$\max_{\pi \in \text{Adm}(\mathcal{F})} \Pr\left(\text{rank}(S^{(\pi)}) \geq i\right). \quad (10)$$

Note that the *i-th rank problem* is an instance of the *i-heads* problem defined in § 1.2.3 with heads-probability for element e set to be $\Pr(X_e \leq t)$. To complete our proof, it suffices to show that, for any $i \in [k]$ and $\pi \in \text{Adm}(\mathcal{F})$,

$$\Pr(\text{rank}(G \cup C) \geq i) \geq \Pr\left(\text{rank}(S^{(\pi)}) \geq i\right), \quad (11)$$

and that $\text{cost}(G \cup C) \leq O(\log k)B$.

We refer to these as the value inequality and cost inequality, respectively. To see cost inequality, note that for the C_j in the binning procedure in Algorithm 2, $C_j \leq 2^j \cdot \frac{B}{2^{j-1}} = 2B$ and $\text{cost}(G) \leq B + (i+1) \cdot \frac{B}{i} \leq 3B$. Since $C = \bigcup_{j=1}^{\lceil \log i \rceil} C_j$, it follows that $\text{cost}(G \cup C) \leq 3B + \lceil \log i \rceil \cdot 2B \leq O(\log k)B$, where the final step comes from the fact that $i \leq k$.

We prove (11) by showing the following lemmas.

► **Lemma 9.** *For the *i-th rank problem* with knapsack constraint \mathcal{F} in (10), suppose that there is a set of policies $\{\pi_\ell : \ell \in [i]\}$ such that for any $\ell \in [i]$ and any $\widehat{\pi} \in \text{Adm}(\mathcal{F})$ whose $S^{(\widehat{\pi})}$ only consists of elements from $\{e \in \mathcal{U} : c_e \leq \frac{B}{i}\}$,*

$$\Pr\left(\text{rank}(S^{(\pi_\ell)}) \geq \ell\right) \geq \Pr\left(\text{rank}(S^{(\widehat{\pi})}) \geq \ell\right).$$

Let $C = \text{BIN}(\mathcal{U}, B, i, t)$ be the output of the binning procedure. If one probes C and then executes π_{ℓ^} with $\ell^* = \max\{i - \text{rank}(C), 1\}$, then the final selection set $C \cup S^{(\pi_{\ell^*})}$ obtains an*

120:14 Probing to Minimize

objective value in the i -th rank problem that is as good as the optimum, i.e., for any policy $\pi \in \mathbf{Adm}(\mathcal{F})$,

$$\Pr(\text{rank}(C \cup S^{(\pi_{\ell^*})}) \geq i) \geq \Pr(\text{rank}(S^{(\pi)}) \geq i).$$

► **Lemma 10.** Define G_ℓ as $\text{EXTGREEDY}(\{e \in \mathcal{U} : c_e \leq \frac{B}{i}\}, B, \ell, t)$, and consider the sequence of sets $\{G_\ell : \ell \in [i]\}$. Then for any $\ell \in [i]$ and any $\hat{\pi} \in \mathbf{Adm}(\mathcal{F})$ whose $S^{(\hat{\pi})} \subseteq \{e \in \mathcal{U} : c_e \leq \frac{B}{i}\}$,

$$\Pr(\text{rank}(G_\ell) \geq \ell) \geq \Pr(\text{rank}(S^{(\hat{\pi})}) \geq \ell).$$

Algorithm 2 BIN(\mathcal{U}, B, i, t).

```

1: Initialization:  $z \leftarrow \lceil \log i \rceil$ ;  $C \leftarrow \emptyset$ 
2: for  $j$  from 1 to  $z$  do ▷ Construct the cost bucket
3:    $\text{low} \leftarrow \max\left\{\frac{B}{2^j}, \frac{B}{i}\right\}$ ;  $\text{high} \leftarrow \frac{B}{2^{j-1}}$ 
4:    $\text{BUCKET}_j \leftarrow \{e \in \mathcal{U} : \text{low} < c_e \leq \text{high}\}$ 
5:    $C_j \leftarrow \emptyset$  ▷ Perform greedy selection
6:   while  $|C_j| < 2^j$  and  $\text{BUCKET}_j \neq \emptyset$  do
7:      $\ell \leftarrow \arg \max_{e \in \text{BUCKET}_j} \Pr(X_e \leq t)$ 
8:      $\text{BUCKET}_j \leftarrow \text{BUCKET}_j - \{\ell\}$ 
9:      $C_j \leftarrow C_j \cup \{\ell\}$ 
10:     $C \leftarrow C \cup C_j$ 
11: return  $C$ .

```

Algorithm 3 EXTGREEDY(\mathcal{U}', B', i', t).

```

1: Initialization:  $\delta \leftarrow \max_{e \in \mathcal{U}'} c_e$ ;  $G \leftarrow \emptyset$ ;  $\text{POOL} \leftarrow \mathcal{U}'$ 
2: for  $e \in \mathcal{U}'$ : do ▷ Compute density of each element
3:    $a_e \leftarrow \frac{-\log \Pr(X_e > t)}{c_e}$ 
4: while  $\text{cost}(G) < B' + i'\delta$  do ▷ Perform greedy selection
5:    $\ell \leftarrow \arg \max_{e \in \text{POOL}} a_e$ 
6:    $G \leftarrow G \cup \{\ell\}$ 
7:    $\text{POOL} \leftarrow \text{POOL} - \{\ell\}$ 
8: return  $G$ 

```

Combining Lemmas 9 and 10 completes the proof of the value inequality (11) once we notice that the $\{G_\ell : \ell \in [i]\}$ in Lemma 10 serves as the $\{\pi_\ell : \ell \in [i]\}$ in Lemma 9 and $G_\ell \subseteq G$ for all $\ell \in [i]$, where one should recall that $G = \text{EXTGREEDY}(\{e \in \mathcal{U} : c_e \leq \frac{B}{i}\}, B, i, t)$. We prove each in turn.

4.1 Proof of Lemma 9 (Analysis of BIN)

We prove this lemma via two claims. First, in Claim 11, we show that the rank of the selection set $S^{(\pi)}$ under any policy π is stochastically smaller than the rank of $S_{\leq \frac{B}{i}}^{(\pi)} \cup C$ where $S_{\leq \frac{B}{i}}^{(\pi)} \triangleq S^{(\pi)} \cap \{e \in \mathcal{U} : c_e \leq \frac{B}{i}\}$. In other words, we can replace the high-cost elements of $S^{(\pi)}$ with C and *strictly* improve its performance.

Second, in Claim 12 we argue that, for any policy π , given knowledge of the set C 's rank, we can choose a policy π' which selects only elements from $\{e \in \mathcal{U} : c_e \leq \frac{B}{i}\}$ whose set in expectation outperforms $S_{\leq \frac{B}{i}}^{(\pi)}$. Note that the set $S_{\leq \frac{B}{i}}^{(\pi)}$ is a random set whose composition may depend on the weights of elements with costs larger than $\frac{B}{i}$. Formally, we show the following claims.

▷ **Claim 11.** For any selection set $S^{(\pi)}$, $\Pr(\text{rank}(S^{(\pi)}) \geq i) \leq \Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)} \cup C) \geq i)$.

▷ **Claim 12.** Let $\mathcal{F}' = \{S \in \mathcal{F} : c_e \leq \frac{B}{i}, \forall e \in S\}$ be the constraint family \mathcal{F} but with all the high-cost elements removed. For any $\pi \in \text{Adm}(\mathcal{F})$ and $\ell \in [i]$, there exists a $\pi'_\ell \in \text{Adm}(\mathcal{F}')$ such that

$$\Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)} \cup C) \geq i \mid \text{rank}(C) = i - \ell) \leq \Pr(\text{rank}(S^{(\pi'_\ell)}) \geq \ell)$$

From these two claims, the proof of Lemma 9 flows quite directly:

$$\Pr(\text{rank}(S^{(\pi)}) \geq i) \leq \Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)} \cup C) \geq i) \quad (12)$$

$$= \sum_{\ell \in [i]} \Pr(\text{rank}(C) = i - \ell) \Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)} \cup C) \geq i \mid \text{rank}(C) = i - \ell) \quad (13)$$

$$\leq \sum_{\ell \in [i]} \Pr(\text{rank}(C) = i - \ell) \Pr(\text{rank}(S^{(\pi'_\ell)}) \geq \ell) \quad (13)$$

$$\leq \sum_{\ell \in [i]} \Pr(\text{rank}(C) = i - \ell) \Pr(\text{rank}(S^{(\pi'_\ell)}) \geq \ell) \quad (14)$$

$$= \Pr(\text{rank}(C \cup S^{(\pi_{i - \text{rank}(C)})}) \geq i),$$

where (12) follows from Claim 11, (13) follows from Claim 12, and (14) follows from Lemma 9's main assumption on $\{\pi_\ell : \ell \in [i]\}$. This completes our proof of Lemma 9, contingent on the claims.

4.1.1 Proof of Claim 11

To show that

$$\Pr(\text{rank}(S^{(\pi)}) \geq i) \leq \Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)} \cup C) \geq i),$$

we make a simple coupling argument. Consider the following alteration of π . Execute the policy π as normal but, for any j , every time π would probe an element e in BUCKET_j , instead make π do the following:

1. If $e \notin C_j$, then we instead make π probe the unprobed element in C_j of smallest below-threshold probability. In other words, π probes $\arg \min_{v \in C_j - S^{(\pi)}} \Pr(X_v \leq t)$, where $S^{(\pi)}$ is the policy π 's current selection set (instead of its final selection set, as in the usual definition).
2. If $e \in C_j$ and still unselected, then we allow π to proceed as usual, i.e., we allow π to probe e .
3. If $e \in C_j$ but has already been probed (due to the first line in this list), we direct π to probe the element $v \in C_j - S^{(\pi)}$ of immediately higher $\Pr(X_v \leq t)$; this is always possible, since π probes at most 2^j elements from BUCKET_j , and we swap out elements in ascending order of $\Pr(X_v \leq t)$.

120:16 Probing to Minimize

If this procedure is applied, every probed element of cost $> \frac{B}{i}$ must lie in C . Note also that, in all cases, the replacement element must have a below-threshold probability at least as big as the element e being replaced, since C_j contains the 2^j elements from BUCKET_j with the highest below-threshold probability. For any optimal policy, it follows that this would result in a strict increase in success probability. But every element in our procedure must lie in C , and probing the rest of C can only improve things, which implies the claim.

4.1.2 Proof of Claim 12

To see that

$$\Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)} \cup C) \geq i \mid \text{rank}(C) = i - \ell) \leq \Pr(\text{rank}(S^{(\pi'_\ell)}) \geq \ell)$$

it suffices to consider the following thought experiment: Give a policy free access to C , and full knowledge of $\{X_e : e \in C\}$, but restrict it to selecting new elements of cost $> \frac{B}{i}$; how would that policy maximize $\Pr(\text{its selection set reaches rank } i)$? One approach is to run π using the replacement procedure from the proof of Claim 11; whenever π would probe an element e of cost $> \frac{B}{i}$, we pretend π looked at X_e while using the weight of its replacement element. Doing this, we can *simulate* full access to all high-cost elements, while *actually* only having access to C .

Call a policy low-cost if it does not choose elements of cost $> \frac{B}{i}$. A better way is to take the remaining rank required, $\ell = i - \text{rank}(C)$, and find a low-cost policy π' which maximizes $\Pr(\text{rank}(S^{(\pi')}) \geq \ell \mid \cap_{e \in C} \{X_e = x_e\})$. Clearly though, from the independence of weights, the performance of any valid policy which probes no elements in C must be independent of the outcomes in C , i.e., $\Pr(\text{rank}(S^{(\pi')}) \geq \ell \mid \cap_{e \in C} \{X_e = x_e\}) = \Pr(\text{rank}(S^{(\pi')}) \geq \ell)$. Take L to be the set of $(x_e)_{e \in C}$ values where $\text{rank}(C) = i - \ell$. Since by assumption π'_ℓ is *better* than π' , it follows from the above that

$$\begin{aligned} \Pr(\text{rank}(C) = i - \ell) \Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)} \cup C) \geq i \mid \text{rank}(C) = i - \ell) \\ = \Pr(\text{rank}(C) = i - \ell) \Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)}) \geq \ell \mid \text{rank}(C) = i - \ell) \\ = \sum_{(x_e)_{e \in C} \in L} \Pr(\cap_{e \in C} \{X_e = x_e\}) \Pr(\text{rank}(S_{\leq \frac{B}{i}}^{(\pi)}) \geq \ell \mid \cap_{e \in C} \{X_e = x_e\}) \\ \leq \sum_{(x_e)_{e \in C} \in L} \Pr(\cap_{e \in C} \{X_e = x_e\}) \Pr(\text{rank}(S^{(\pi')}) \geq \ell \mid \cap_{e \in C} \{X_e = x_e\}) \\ = \sum_{(x_e)_{e \in C} \in L} \Pr(\cap_{e \in C} \{X_e = x_e\}) \Pr(\text{rank}(S^{(\pi')}) \geq \ell) \end{aligned}$$

Since $\Pr(\text{rank}(S^{(\pi')}) \geq \ell) \leq \Pr(\text{rank}(S^{(\pi'_\ell)}) \geq \ell)$ and $\sum_{(x_e)_{e \in C} \in L} \Pr(\cap_{e \in C} \{X_e = x_e\})$ is simply $\Pr(\text{rank}(C) = i - \ell)$, the claim follows.

4.2 Proof of Lemma 10 (Analysis of ExtGreedy)

We now prove to prove the remaining lemma, which concerns EXTGREEDY, Algorithm 3.

Throughout this proof, we assume, for brevity's sake, that \mathcal{U} contains no elements of cost $> \frac{B}{i}$ and that the constraint \mathcal{F} is a knapsack constraint using the original budget B and this limited-cost universe \mathcal{U} . This in no way affects the proofs here, since G is drawn from a limited cost universe and the policy it is competing against, $\hat{\pi}$, must be admissible in this low-cost setting.

We proceed in four steps. Recall that we say a policy π succeeds if $\text{rank}(S^{(\pi)}) \geq \ell$. First, we provide a characterization of $\Pr(\pi \text{ succeeds})$ based on a structural decomposition of π 's decision tree. Second, we use that characterization to derive a sufficient condition on two policies π and π' which implies that $\Pr(\pi' \text{ succeeds}) \geq \Pr(\pi \text{ succeeds})$. In the last two steps, we prove that this sufficient condition holds for G_ℓ and π .

4.2.1 Characterization of success probability

In the first step of our proof, we characterize the success probability of policies for the ℓ -th rank problem. First, we decompose the structure of a policy π , then we describe how to use that decomposition to represent the probability that π succeeds.

4.2.1.1 Policies and decision trees

We assume without loss of generality that a policy is constructed as a decision tree, where nodes correspond to elements probed, and edges correspond to the particular value of an element's weight upon probing. Because we are concerned only with $\text{rank}(S)$, and element-weights are independent of each other, one can, without lowering a policy's probability of success, separate the $(m + 1)$ -many possible outcomes into two particular meta-outcomes: whether the element e probed has $X_e \leq t$, or $X_e > t$. We also associate a particular directionality to outcomes: the root is the first element probed, the left child of the root is the element probed if the root is below-threshold, and so on.

► **Definition 13** (The Left Chain, $L^{(\pi)}$). *Define the “left chain” of a decision tree to be the sequence of elements probed when every previous element probed is above-threshold. Let $1_j^{(\pi)}$ be the j -th element in this sequence. Then, the left chain of a policy π is simply the sequence $L^{(\pi)} \triangleq (1_0^{(\pi)}, 1_1^{(\pi)}, \dots)$. Further, let $L_j^{(\pi)} \triangleq (1_0^{(\pi)}, 1_1^{(\pi)}, \dots, 1_j^{(\pi)})$ be the left chain of π , truncated at the j -th element.*

► **Definition 14** (Exit trees, $T_j^{(\pi)}$). *We say the policy π “exits the left chain” at j if $1_j^{(\pi)}$ is the first below-threshold element probed. It follows that, for every j , one can talk about the decision sub-tree corresponding to “exiting” at j ; we call this sub-tree $T_j^{(\pi)}$.*

With $L^{(\pi)}$, these $T_j^{(\pi)}$ give a neat decomposition of the policy π ; we visualize this in Figure 1b.

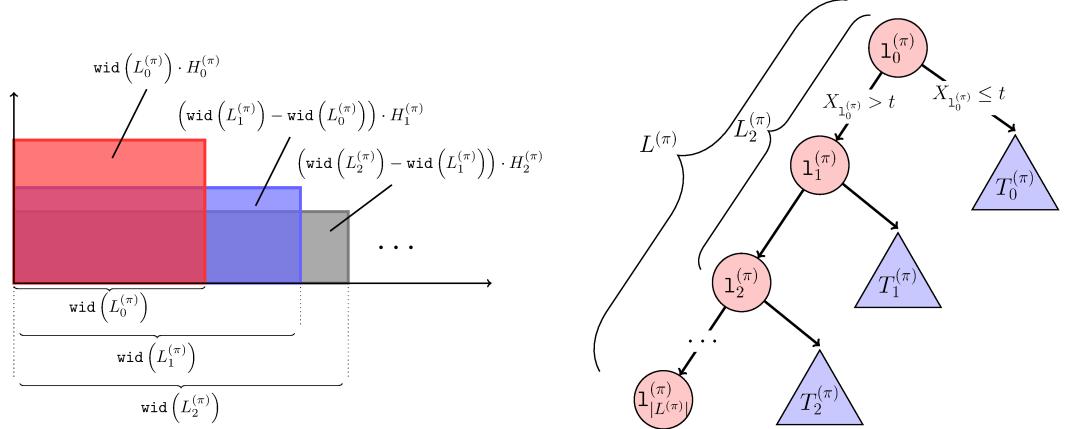
4.2.1.2 Non-adaptive policies

One can also apply this decomposition to non-adaptive policies (deterministic sets) G . In fact, one special characteristic of non-adaptive policies is that there is no fixed left chain: one can probe the deterministic set G in any order, without changing the probability that $\text{rank}(G) \geq \ell$. In other words, one can define $L^{(G)}$ to be an arbitrary permutation of G without affecting the probability that G succeeds. Note that, by definition, the “exit trees” of G are simply the policies which probe whatever elements in G remain unprobed. In other words, $T_j^{(G)} \triangleq G - L_j^{(G)}$.

We next characterize the success probability $\Pr(\text{rank}(S^{(\pi)}) \geq \ell)$ under a policy π based on the structural decomposition.

► **Definition 15** (The Exit Value, $H_j^{(\pi)}$). *For a policy π , let the exit value $H_j^{(\pi)}$ be the probability of the policy π 's success given it has exited the left chain after probing the element $1_j^{(\pi)}$, i.e.,*

$$H_j^{(\pi)} \triangleq \Pr\left(\text{rank}(S^{(\pi)}) \geq \ell \mid \text{rank}(L_{j-1}^{(\pi)}) = 0, 1_j^{(\pi)} \text{ is below-threshold}\right).$$



(a) A visualization of the success probability decomposition found in (15). The j -th rectangle has width $\text{wid}(L_j^{(\pi)})$ and height $H_j^{(\pi)}$. The total area covered by these rectangles is the success probability $\Pr(\text{rank}(S^{(\pi)}) \geq \ell)$. Full details can be found in § 4.2.1.

(b) A visualization of the structural decomposition. Upon probing an element $1_j^{(\pi)}$, if $X_{1_j^{(\pi)}} \leq t$, then the policy π proceeds to the right child of the node $(1_j^{(\pi)})$; otherwise, π proceeds to the left child. The left child of the node $(1_j^{(\pi)})$ is $1_{j+1}^{(\pi)}$, the next node in the left chain $L^{(\pi)}$. The right child is the exit tree $T_j^{(\pi)}$.

■ **Figure 1** Illustrations of the probability decomposition and structural decomposition, respectively, of § 4.2.1.

For each possible exit j , let \mathcal{F}' be a knapsack constraint with universe $\mathcal{U}' = \mathcal{U} - L_j^{(\pi)}$ and budget $B' = B - \text{cost}(L_j^{(\pi)})$. By element-weight independence, the behavior of policy π after leaving the left chain at j can be treated, without loss of generality, as the behavior of a some new policy π' with constraint \mathcal{F}' . In other words,

$$H_j^{(\pi)} = \Pr(\text{rank}(S^{(\pi)}) \geq \ell \mid \pi \text{ exits at } j) = \Pr(\text{rank}(S^{(\pi')}) \geq \ell - 1).$$

4.2.1.3 Basic characterization of a policy π

Let $p_j^{(\pi)}$ be the probability that the left-element $1_j^{(\pi)}$ is below-threshold. Together with the left chain and exit values of a policy π defined, we can characterize the success probability of a policy as

$$\Pr(\text{rank}(S^{(\pi)}) \geq \ell) = p_0^{(\pi)} \cdot H_0^{(\pi)} + \sum_{j=1}^{|L^{(\pi)}|} \left[\prod_{y=0}^{j-1} (1 - p_y^{(\pi)}) \right] p_j^{(\pi)} \cdot H_j^{(\pi)}.$$

We develop this decomposition further.

► **Definition 16** (The width, $\text{wid}(S)$). For a set S , define the width of S , denoted $\text{wid}(S)$, to be the probability that at least one element in S is below-threshold.

Define the reward of an element e as $\text{reward}(e) = -\log(\Pr(X_e > t))$, and the total reward of a set S by extension as $\text{reward}(S) = \sum_{e \in S} \text{reward}(e)$. Note that $\text{wid}(S)$ can be written as

$$\text{wid}(S) = 1 - \prod_{e \in S} (\Pr(X_e > t)) = 1 - \exp\left(\sum_{e \in S} \log(\Pr(X_e > t))\right) = 1 - \exp\left(-\text{reward}(S)\right),$$

which implies that $\text{wid}(S)$ is an increasing function of $\text{reward}(S)$. Notice also that, for any element $u \notin S$, the difference $\text{wid}(S \cup \{u\}) - \text{wid}(S) = [\prod_{e \in S} (1 - p_e)] p_u$. It follows that, for any policy π with a left chain of length x ,

$$\Pr(\text{rank}(S^{(\pi)}) \geq \ell) = \text{wid}(L_0^{(\pi)}) \cdot \mathbf{H}_0^{(\pi)} + \sum_{j=1}^{|L^{(\pi)}|} [\text{wid}(L_j^{(\pi)}) - \text{wid}(L_{j-1}^{(\pi)})] \cdot \mathbf{H}_j^{(\pi)} \quad (15)$$

4.2.2 A sufficient condition: Rectangle covering

In the second step of our proof, we use the characterization just developed to derive a sufficient condition on two policies π and π' which implies that $\Pr(\pi' \text{ succeeds}) \geq \Pr(\pi \text{ succeeds})$. We make a visual argument. One can characterize the summation in (15) as the total area covered by a sequence of rectangles all nestled in the corner of the positive orthant, where the j -th rectangle in the sequence has height $\mathbf{H}_j^{(\pi)}$ and width $\text{wid}(L_j^{(\pi)})$; we visualize this in Figure 1a.

Likewise, for any *non-adaptive* policy G , for *any* left-chain ordering $L^{(G)}$ on the set G , one has precisely the previous decomposition of (15), where $\mathbf{H}_j^{(G)} = \Pr(\text{rank}(G - L_j^{(G)}) \geq \ell - 1)$.

4.2.2.1 Main idea

Geometrically, to show that a non-adaptive policy G succeeds more often than an optimal policy π , it suffices to show that the area covered by G in such a diagram is larger than the area covered by π in such a diagram. Further, it suffices to show that the each of π 's rectangles is *completely covered* by at least one G 's rectangles, for some ordering of the elements on the left chain $L^{(G)}$. We shall do precisely this, with $G = \text{EXTGREEDY}(\mathcal{U}, B, \ell, t)$. Formally, we will show the following lemma.

► **Lemma 17** (Rectangle-covering lemma). *Let $G = \text{EXTGREEDY}(\mathcal{U}, B, \ell, t)$. Given an optimal policy π 's left chain $L^{(\pi)}$, one can construct a fixed ordering $L^{(G)}$ such that there exists an increasing sequence of indices $(\sigma(j))_{j \in \{0, 1, \dots, |L^{(\pi)}|\}}$ satisfying for every j*

$$\text{wid}(L_{\sigma(j)}^{(G)}) \geq \text{wid}(L_j^{(\pi)}) \quad (16)$$

and

$$\mathbf{H}_{\sigma(j)}^{(G)} \geq \mathbf{H}_j^{(\pi)}. \quad (17)$$

In other words, the $\sigma(j)$ -th rectangle of G completely covers the j -th rectangle of π .

Note that the sequence of $\sigma(j)$'s in Lemma 17 need not be consecutive. For example, the first rectangle of G might completely cover the first 4 rectangles of π . In that case, one could set $\sigma(1), \sigma(2), \sigma(3)$, and $\sigma(4)$ all to be 1. The $\sigma(\cdot)$ sequence is simply a formalism by which we can explicitly assign coverage of a particular π -rectangle to a particular G -rectangle.

4.2.3 Proof of Lemma 17

All that remains to prove Lemma 10 is to prove Lemma 17. To do so, we induct on the target rank ℓ .

In the base case, $\ell = 1$, and whenever a policy exits its left chain, it has succeeded. In other words, for any policy π and index j , the height $\mathbf{H}_j^{(\pi)} = 1$. To prove the base case, since all rectangle heights are the same, it suffices to show the width inequality $\text{wid}(L^{(G)}) \geq \text{wid}(L^{(\pi)})$;

equivalently, we must show that $\text{reward}(L^{(G)}) \geq \text{reward}(L^{(\pi)})$. But this follows directly from our greedy-until-overflowing strategy for selecting G , proving the base case.

For the remainder of this section, we assume the following inductive assumption.

Inductive assumption. Given any universe \mathcal{U}' , budget B' , target $\ell' < \ell$, and threshold t , for any policy $\pi' \in \mathbf{Adm}(\mathcal{F}')$, one can construct a fixed ordering $L^{(G')}$ of the set $G' = \text{EXTGREEDY}(\mathcal{U}', B', \ell', t)$ such that there exists an increasing sequence of indices $(\beta(j))_{j \in (0, 1, \dots, |L^{(\pi')}|)}$ satisfying the analogous height and width conditions (17) and (16).

4.2.3.1 Inductive step

For the remainder of our proof, we use our inductive assumption for lower-rank problems to construct a left chain ordering $L^{(G)}$ and assignment sequence $\sigma(\cdot)$ which satisfies these conditions for the ℓ -th rank problem. Before we continue with our construction, we define a surrogate policy $\tilde{\pi}$ for the policy π which simplifies our analysis.

Construction of the surrogate policy $\tilde{\pi}$.

► **Definition 18** (The residual sets, T_j^*). *Let*

$$T_j^* = \text{EXTGREEDY}(\mathcal{U} - L_j^{(\pi)}, B - \text{cost}(L_j^{(\pi)}), \ell - 1, t)$$

be the extended greedy set one would choose using the resources available after exiting the left chain at j while executing π . As usual, we refer to both the non-adaptive policy which probes T_j^ and the set T_j^* by the same name.*

Given a policy π and the residual sets T_j^* , we now construct a simpler-to-analyze policy $\tilde{\pi}$ such that $\Pr(\tilde{\pi} \text{ succeeds}) \geq \Pr(\pi \text{ succeeds})$.

► **Claim 19.** Recall the decomposition of π into $L^{(\pi)}$ and sub-trees $T_j^{(\pi)}$. Likewise, define $\tilde{\pi}$ as the policy which has the same left chain as π , but replaces each sub-tree $T_j^{(\pi)}$ with the set T_j^* . Then,

$$\Pr(\text{rank}(S^{(\tilde{\pi})}) \geq \ell) \geq \Pr(\text{rank}(S^{(\pi)}) \geq \ell). \quad (18)$$

Proof. We show first that $\tilde{\pi}$ is a valid policy. Let $\mathcal{U}' = \mathcal{U} - L_j^{(\pi)}$, let $B' = B - \text{cost}(L_j^{(\pi)})$, and let \mathcal{F}' be a knapsack constraint formed with universe \mathcal{U}' with budget B' . Since the behavior of policy π after leaving the left chain at j is equivalent to the behavior of some new policy π' with constraint \mathcal{F}' , and the set $G' = \text{EXTGREEDY}(\mathcal{U}', B', \ell - 1, t)$ must be disjoint from $L_j^{(\pi)}$, it follows that $\tilde{\pi}$, which probes the left chain $L^{(\pi)}$ until we exit at j then probes G' , is a perfectly valid policy; it probes no elements twice, and uses no foreknowledge of any weight outcomes.

We now argue that $\Pr(\tilde{\pi} \text{ succeeds}) \geq \Pr(\pi \text{ succeeds})$ from the success probability characterization of (15). Since $\tilde{\pi}$ and π share the same left chain $L^{(\pi)}$, they must have the same width values $\text{wid}(L_j^{(\pi)})$. Thus, it suffices to show that replacing π' with G' can only increase the height of $\tilde{\pi}$, i.e., that $\mathbf{H}_j^{(\tilde{\pi})} \geq \mathbf{H}_j^{(\pi)}$. To see this, note that, by the same justification that says Lemma 17 (rectangle covering) implies Lemma 10 (EXTGREEDY is better than π), our inductive assumption immediately implies that, since $\pi' \in \mathbf{Adm}(\mathcal{F}')$,

$$\mathbf{H}_j^{(\tilde{\pi})} = \Pr(\text{rank}(G') \geq \ell - 1) \geq \Pr(\text{rank}(S^{(\pi')}) \geq \ell - 1).$$

By the definition of π' and $\mathbf{H}_j^{(\pi)}$,

$$\Pr\left(\text{rank}\left(S^{(\pi')}\right) \geq \ell - 1\right) = \Pr\left(\text{rank}\left(S^{(\pi)}\right) \geq \ell \mid \pi \text{ exits the left chain at } j\right) = \mathbf{H}_j^{(\pi)}.$$

Combining these observations, we find that $\mathbf{H}_j^{(\tilde{\pi})} \geq \mathbf{H}_j^{(\pi)}$, as desired. \triangleleft

Construction of $L^{(G)}$ and $\sigma(\cdot)$. We now proceed with our construction. We first make the following claim about the sets T_j^* .

▷ **Claim 20.** Let π be any policy in $\text{Adm}(\mathcal{F})$ and let $L^{(\pi)}$ be its left chain. Let

$$T_j^* = \text{EXTGREEDY}\left(\mathcal{U} - L_j^{(\pi)}, B - \text{cost}\left(L_j^{(\pi)}\right), \ell - 1, t\right).$$

Then $T_{j+1}^* \subseteq T_j^* \subseteq G$, for all $j \in [|L^{(\pi)}|]$.

Proof. We first find a simpler condition to show. As usual, for a fixed index j , let $\mathcal{U}' = \mathcal{U} - L_j^{(\pi)}$ and let $B' = B - \text{cost}\left(L_j^{(\pi)}\right)$. Now, let δ' be the max cost of any element in \mathcal{U}' . By the definition of EXTGREEDY, $\text{EXTGREEDY}(\mathcal{U}', B', \ell - 1, t) = \text{EXTGREEDY}(\mathcal{U}', B' + (\ell - 1)\delta', 0, t)$. As the maximum cost of any element in \mathcal{U} can only decrease as elements are removed, $\delta' \leq \delta$, and, from a budget argument, $\text{EXTGREEDY}(\mathcal{U}', B' + (\ell - 1)\delta', 0, t) \subseteq \text{EXTGREEDY}(\mathcal{U}', B' + (\ell - 1)\delta, 0, t)$. As such, we have that $T_j^* = \text{EXTGREEDY}(\mathcal{U}', B' + (\ell - 1)\delta', 0, t)$ and, letting $e = 1_j^{(\pi)}$ be the j -th element in the left chain $L^{(\pi)}$, $T_{j+1}^* \subseteq \text{EXTGREEDY}(\mathcal{U}' - \{e\}, B' - c_e + (\ell - 1)\delta', 0, t)$. Thus, to show $T_{j+1}^* \subseteq T_j^* \subseteq G$, we prove a more general claim; that, for a general universe $\hat{\mathcal{U}}$, budget \hat{B} , and element $e \in \hat{\mathcal{U}}$,

$$\text{EXTGREEDY}\left(\hat{\mathcal{U}} - \{e\}, \hat{B} - c_e, 0, t\right) \subseteq \text{EXTGREEDY}\left(\hat{\mathcal{U}}, \hat{B}, 0, t\right).$$

For brevity, call the former set T' and the latter T . Without loss of generality, number the elements of $\hat{\mathcal{U}}$ from 1 to $|\hat{\mathcal{U}}|$ in descending order of reward density, maintaining that higher density elements have lower element number. Let τ be the smallest index for which $\sum_{x=1}^{\tau} c_x \geq \hat{B}$. Then, by definition, the set $T = \{1, 2, \dots, \tau\}$. Likewise, if τ' is the smallest index for which $\sum_{x=1, x \neq e}^{\tau'} c_x \geq \hat{B} - c_e$, then $T' = \{1, 2, \dots, \tau'\} - \{e\}$. Thus, to show $T' \subseteq T$, it suffices to show that $\tau' \leq \tau$.

There are two cases. If the removed element $e \leq \tau$, then the selection process for T' ends by τ at the latest, since $\sum_{x=1}^{\tau} c_x \geq \hat{B}$ implies that $\sum_{x=1, x \neq e}^{\tau} c_x = \sum_{x=1}^{\tau} c_x - c_e \geq \hat{B} - c_e$, and τ' is the smallest index which satisfies this condition. Hence, $\tau' \leq \tau$.

In the other case, the removed element $e > \tau$, and $\sum_{x=1}^{\tau} c_x = \sum_{x=1, x \neq e}^{\tau} c_x$ meaning $\sum_{x=1, x \neq e}^{\tau} c_x \geq \hat{B} \geq \hat{B} - c_e$. Since τ' is the smallest index satisfying the above, $\tau' \leq \tau$, proving the claim. \triangleleft

This claim gives the construction almost directly. First, it induces a total order on G , where the order of an element $e \in G$ is the smallest index j for which $e \notin T_j^*$, i.e., all the elements of $G - T_0^*$ are of order 0, all the elements of $T_0^* - T_1^*$ are of order 1, and so on. We can then take $L^{(G)}$ to be any permutation of G which proceeds in non-decreasing order. Second, it allows us to define $\sigma(j)$ as the index r for which $G - L_r^{(G)} = T_j^*$; in other words, we make $\sigma(j) = |\{e \in G : e\text{'s order} \leq j\}| - 1$.

Verifying the construction. Having given a construction for $L^{(G)}$ and $\sigma(\cdot)$, in this final part of our proof, we show that the given construction satisfies Lemma 17's width and height conditions. The height condition is satisfied by construction, since, if $\tilde{\pi}$ exits the left chain at j , then it probes T_j^* , the same set that G probes upon exiting its left chain at $\sigma(j)$. All that remains is to show that

$$\text{wid}\left(L_{\sigma(j)}^{(G)}\right) = \text{wid}(G - T_j^*) \geq \text{wid}\left(L_j^{(\pi)}\right) = \text{wid}\left(L_j^{(\tilde{\pi})}\right),$$

where the first and last equalities follow by construction. Our argument here centers on the *reward* characterization of the width, where $\text{reward}(S) = \sum_{e \in S} -\log \Pr(X_e > t)$. As explained in the definition of $\text{wid}(S)$, it is monotonically increasing in the reward of S . Thus, it suffices to show that $\text{reward}(G - T_j^*) \geq \text{reward}\left(L_j^{(\pi)}\right)$.

We prove this in three short steps. First, since the reward is a modular set function, we may eliminate the common elements between $L_j^{(\pi)}$ and $G - T_j^*$, giving us L' and G' respectively. Second, we argue that, since $L_j^{(\pi)}$ is disjoint from T_j^* , the intersection of these two sets satisfies $L_j^{(\pi)} \cap (G - T_j^*) = L_j^{(\pi)} \cap G$, and thus L' must also be disjoint from G . Because EXTGREEDY selects elements in order of decreasing reward density (i.e., in order of decreasing *reward per unit cost*), it follows that every element in G' has a higher reward density than every element in L' . From here, it suffices to show that $\text{cost}(G') \geq \text{cost}(L')$, or, since cost too is a modular set function, that $\text{cost}(G - T_j^*) \geq \text{cost}\left(L_j^{(\pi)}\right)$.

This third and final step follows from some basic facts about G , T_j^* , δ , and $L_j^{(\pi)}$. First, note that $\text{cost}(T_j^*) \leq B - \text{cost}\left(L_j^{(\pi)}\right) + (\ell - 1)\delta + \delta$, where the first 3 terms follow from the definition of EXTGREEDY and the final δ term follows from the fact that δ is the maximum cost of any element. From there, we note that $\text{cost}(G - T_j^*) = \text{cost}(G) - \text{cost}(T_j^*)$, since $T_j^* \subseteq G$. Once we note that $\text{cost}(G) \geq B + \ell\delta$ by EXTGREEDY's stopping condition, it follows that $\text{cost}(G - T_j^*) \geq \text{cost}\left(L_j^{(\pi)}\right)$, completing our proof of Lemma 17.

References

- 1 Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. In *Int. Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 29–40, Lyon, France, 2014. doi:10.4230/LIPIcs.STACS.2014.29.
- 2 Domagoj Bradac, Sahil Singla, and Goran Zuzic. (Near) Optimal Adaptivity Gaps for Stochastic Multi-Value Probing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, volume 145, pages 49:1–49:21, Boston, MA, 2019. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.49.
- 3 Robert L. Carraway, Robert L. Schmidt, and Lawrence R. Weatherford. An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns. *Nav. Res. Log.*, 40(2):161–173, 1993.
- 4 Alina Ene, Viswanath Nagarajan, and Rishi Saket. Approximation algorithms for stochastic k-TSP. In *IARCS Ann. Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, Kanpur, India, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.27.
- 5 Ashish Goel, Sudipto Guha, and Kamesh Munagala. How to probe for an extreme value. *ACM Trans. Algorithms*, 7(1):12:1–12:20, December 2010. doi:10.1145/1868237.1868250.
- 6 Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artif. Int. Res.*, 42:427–486, 2011. URL: <https://dl.acm.org/doi/10.5555/2208436.2208448>.

- 7 Sudipto Guha and Kamesh Munagala. Approximation algorithms for budgeted learning problems. In *Proc. Ann. ACM Symp. Theory of Computing (STOC)*, pages 104–113, San Diego, CA, 2007. doi:10.1145/1250790.1250807.
- 8 Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and XOS functions. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1688–1702, Barcelona, Spain, 2017. doi:10.5555/3039686.3039797.
- 9 Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and adaptivity gaps for stochastic probing. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1731–1747, Barcelona, Spain, 2017. doi:10.1137/1.9781611974782.111.
- 10 Taylan Ilhan, Seyed MR Iravani, and Mark S Daskin. The adaptive knapsack problem with stochastic rewards. *Oper. Res.*, 59(59):242–248, 2011. doi:10.1287/opre.1100.0857.
- 11 Haotian Jiang, Jian Li, Daogao Liu, and Sahil Singla. Algorithms and Adaptivity Gaps for Stochastic k-TSP. In *Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151, pages 45:1–45:25, Seattle, Washington, 2020. doi:10.4230/LIPIcs.ITCS.2020.45.
- 12 W. Rudin. *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1976.
- 13 Sahil Singla. The price of information in combinatorial optimization. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 2523–2532, 2018. doi:10.1137/1.9781611975031.161.
- 14 Weina Wang, Anupam Gupta, and Jalani Williams. Probing to minimize, 2021. arXiv: 2111.01955.

A Adaptivity Gap Example

► **Example 21.** Consider the following probing problem. We are given a universe of three elements, X_1 , X_2 , and X_3 to choose from, independently distributed from each other, with distributions we describe later. We are allowed to choose 2 of these random variables for our set S , and the end objective $f(S)$ is the minimum weight element in S . The distributions are as follows: X_1 is N^2 with probability (w.p.) $\frac{1}{N^2}$ and 1 otherwise, X_2 is N^2 w.p. $\frac{1}{N}$ and 0 otherwise, and X_3 is N w.p. 1.

Computing the expected value of each fixed two-element set, we find that

$$\begin{aligned}\mathbb{E}[f(\{1, 3\})] &= \mathbb{E}[\min(X_1, N)] = \left(1 - \frac{1}{N^2}\right) \cdot 1 + \left(\frac{1}{N^2}\right) \cdot N \geq 1, \\ \mathbb{E}[f(\{1, 2\})] &= \left(1 - \frac{1}{N}\right) \mathbb{E}[X_1] = \left(1 - \frac{1}{N}\right) \left[\left(1 - \frac{1}{N^2}\right) \cdot 1 + \left(\frac{1}{N^2}\right) \cdot N^2 \right] \geq 1, \\ \mathbb{E}[f(\{2, 3\})] &= \mathbb{E}[\min(X_2, N)] = \frac{1}{N} \cdot N = 1.\end{aligned}$$

Now consider the following adaptive policy, which we call π : First, probe element 1. If $X_1 = 1$, then probe element 2; otherwise probe element 3. Intuitively, this policy probes the ‘risky’ element 2 only if it has already secured an objective value $f(S^{(\pi)}) \leq 1$. Computing the expectation, we find $\mathbb{E}[f(S^{(\pi)})] = \left(1 - \frac{1}{N^2}\right) \cdot \frac{1}{N} \cdot 1 + \left(\frac{1}{N^2}\right) \cdot N \leq \frac{2}{N}$. Thus, the adaptivity gap of this instance is at least $\frac{N}{2}$, which can be made arbitrarily large.