

Algorithms and Adaptivity Gaps for Stochastic Probing

Anupam Gupta *

Viswanath Nagarajan †

Sahil Singla*

Abstract

A stochastic probing problem consists of a set of elements whose values are independent random variables. The algorithm knows the distributions of these variables, but not the actual outcomes. The only way to learn the actual outcomes is to probe these elements. However, there are constraints on which set of elements may be probed. (E.g., we may have to travel in some metric to probe elements but have limited time.) These constraints are called outer constraints. We want to develop an algorithm that picks some set of elements to maximize the (expected) value, subject to the picked subset of elements satisfying some other set of constraints, called the inner constraints. In the past, probing problems were studied for the case when both inner and outer constraints were intersections of matroids; these modeled kidney matching and Bayesian auctions applications. One limitation of past work was their reliance on linear-programming-like techniques, which made going beyond matroid-like structures difficult.

In this work, we give a very general adaptivity gap result that holds for all prefix-closed outer constraints, as long as the inner constraints are intersections of matroids. The adaptivity gap is $O(\log n)$ for any constant number of inner matroid constraints. The prefix-closedness captures most “reasonable” outer constraints, like orienteering, connectivity, and precedence. Based on this we obtain the first approximation algorithms for a number of stochastic probing problems, which have applications, e.g., to path-planning and precedence-constrained scheduling.

1 Introduction

Consider the following problem: we are given a set U of elements. Each element e has a random value V_e drawn from a known probability distribution π_e . (The element values are independent of each other.) Our goal is to output a set S , which satisfies some constraints described below, so as to maximize the expected value $\mathbb{E}[\sum_{e \in S} V_e]$ of the set. The problem is: we do not know the element values (i.e., the random outcomes); we only know the underlying probability distributions, and have to probe the elements to find out their realized values. Moreover, the set S can only contain elements that we have probed. There are two kinds of constraints (given by two set systems \mathcal{F}_{out} and \mathcal{F}_{in}):

- *Outer constraints:* these tell us which sequences of elements we can probe during the run of the algorithm. If we probe some ordered set P , then we

require $P \in \mathcal{F}_{out}$. Crucially, the outer constraints do not depend on the outcomes of the probes.¹

- *Inner constraints:* these tell us which subsets of elements can be finally output. If S is the final output, then we must have $S \in \mathcal{F}_{in}$.

Finally, we work in an *online decision-making* model: once we probe an element to find out its value, we must immediately make an irrevocable decision about its fate; i.e., we must either add it to the chosen set S , or discard it forever. Thus we want a strategy to adaptively probe and select elements so as to maximize the expected value. Such a strategy can be described by a decision tree where nodes correspond to probed elements and branches correspond to observed random outcomes (see Section 2 for a formal definition).

To make things vivid, consider the following stochastic version of the orienteering problem. We are given a metric space (U, d) , where each location $i \in U$ either contains a reward v_i (with probability p_i) or not (i.e., it has reward 0 with the remaining probability $1 - p_i$), and this is independent of the reward at other locations. Say the inner constraint is that we can collect the reward from at most K locations (a cardinality constraint). The outer constraint is that the probed locations must be visited by a tour of total length at most B that starts and ends at a “root” location r in the metric space.

This arises, e.g., in a path-planning application where a robot has to plan a route starting from the home base r , picking up packages distributed over the metric space and bringing them back to r . The robot can only carry K packages at a time (hence the inner constraint), and has a total battery life of B time units (hence the outer constraint). Not all packages are present every day: the robot merely knows the chance that the package at location i will be present (hence the stochasticity); for simplicity we assume that if there is a package at location i it has a fixed value. Finally, when the robot reaches a location i and finds that the package is indeed present, it must immediately decide whether to pick it or not — hence the online decision-making model. Finally, the goal is to find a strategy that

*Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

†Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA.

¹Formally, we will generalize \mathcal{F}_{out} to be constraints on *sequences* of elements that are probed, not just sets.

maximizes the expected value of the elements collected. Observe that if every probability $p_i = 1$ then we get the classical orienteering problem for which a $2 + \varepsilon$ -approximation algorithm is known [11, 15].

One can imagine other kinds of inner or outer constraints, of course. Here are some examples.

- *Inner constraints:* instead of picking at most K elements (which is a uniform matroid constraint), the elements may have some k different matroids defined on them, and the inner constraint may require us to output a set S that is independent in all k of these matroids. In other words, we now want to maximize the weighted “rank function” of the intersection of k matroids. We can even handle a slight generalization to the case of maximizing *sums of* weighted rank functions. This allows us to capture, e.g., the practically interesting case of coverage functions: given a set system (U, \mathcal{F}) and random elements, pick a subset $S \subseteq U$ of elements to maximize the expected weight of sets hit by S , subject to the outer constraints.
- *Outer Constraints:* Similarly, instead of the probed elements lying on a tour of length at most B (which is an orienteering constraint), one can imagine other outer constraints — say the probed elements need to belong to some other matroids. Or we may have precedence (some elements can be probed only after others have been probed), or other packing constraints.

With suitable inner/outer constraints, we can capture a rich class of stochastic optimization problems including stochastic matching [18, 7] and sequential pricing in Bayesian auctions [14, 32].

In previous work on stochastic probing problems [27, 3], approximation algorithms were obtained for cases where both the inner and outer constraints are intersections of matroids.² One reason for considering matroid constraints was their rich polyhedral structure. In general, the optimal strategies for these stochastic problems may be exponential-sized decision trees, and hence difficult to argue about. However, if we only consider matroid constraints, we can capture these optimal adaptive strategies using good LP relaxations, and then round the LP solutions to get “non-adaptive” strategies that compare favorably with the optimal strategies. (A non-adaptive strategy is just an *a priori* fixed probing order satisfying the outer constraints.) This gives good approximation algorithms, and also shows non-adaptive strategies that are close to the best adaptive ones (i.e.,

²One could generalize these results slightly to k -systems, knapsack constraints etc.

bounds the “adaptivity gap”). This approach breaks down once we broaden the set of outer constraints to allow, say, orienteering constraints, for which we do not know any good LP relaxations. We can no longer rely on the crutch of LP techniques to compare to the adaptive optimal. What can we do in such cases?

A natural and ideal goal would be to prove an adaptivity gap result: to show that the gap between the best adaptive and non-adaptive policies is small. This would allow us to reduce the search space considerably at a small loss in performance. To complement this, we would like to obtain algorithms that find approximately optimal non-adaptive strategies for these problems. These are precisely the questions we address in this paper.

1.1 Our Results Our main result is a very general adaptivity gap result for *arbitrary* outer constraints (satisfying a minimal “prefix closed” condition—see Section 2) and matroid-intersection inner constraints.

THEOREM 1.1. *The adaptivity gap of any stochastic probing problem where the outer constraints are prefix-closed and the inner constraints are intersections of k matroids is at most $\alpha_{ad} = O(k^4 \log(nk))$, where n is the number of elements.*

We obtain a better result for partition matroids.

THEOREM 1.2. *For stochastic probing problems with prefix-closed outer constraints and inner constraints given by intersections of k partition matroids, the adaptivity gap is at most $O(k^4 \log k)$.*

Note that this implies a constant adaptivity gap for $k = O(1)$ partition matroids, improving over the $O(\log n)$ bound from Theorem 1.1. Moreover, both these results extend to the case where the objective is to maximize the sum of different weighted matroid rank functions. Appendix B.1 gives a simple example showing that the adaptivity gap is at least $\frac{7}{6}$ even with $k = 1$ partition matroid.

Our techniques allow us to derive the following useful result from the proof of Theorem 1.2.

COROLLARY 1.1. *For stochastic probing problems with prefix-closed outer constraints (and no inner constraints), the problem of maximizing a weighted coverage function has adaptivity gap at most 3.51.*

By the above theorems, it suffices to solve the non-adaptive versions of these problems up to a small loss in the objective function. The actual outer constraints play a role only in solving these non-adaptive versions, the results above just rely on the outer constraints

being prefix-closed. For some interesting classes of outer constraints, we show that existing algorithms for deterministic optimization can be used to obtain algorithms for non-adaptive (and hence, by the above theorems, for adaptive) stochastic probing. Here are two examples (more appear in Section 6).

Stochastic Probing on Metrics. This is a generalization of the orienteering-type problem discussed in the introduction. The elements U lie in a metric with root node r , and the chosen elements must be visited via a path of length B and must satisfy a k -matroid intersection constraint. The goal is to compute an adaptive route (originating from r) that maximizes the expected value of selected elements. Using Theorem 1.1 and a slight extension of an algorithm for “submodular orienteering” [16], we obtain:

THEOREM 1.3. *There is a quasi-polynomial time $O(k^5 \log^2 n)$ -approximation algorithm for stochastic probing on metrics with k -matroid intersection inner constraints.*

Interestingly, this result also partially answers an open question from [29] on approximating the *stochastic minimum latency submodular cover* (StocMLSC) problem: we defer details to Section 6. In particular Theorem 1.3 leads to a quasi-polynomial time $O(\log^3 n)$ -approximation algorithm for the special case of StocMLSC with matroid rank functions.

Stochastic Connected Dominating Set. In the budgeted connected dominating set (BCDS) problem, we are given an undirected graph $G = (V, E)$ and budget k , and the goal is to choose a set of k connected vertices that maximize the number of dominated vertices. [31] obtained an $O(1)$ -approximation for this problem. In its stochastic version that we consider, each edge $e \in E$ is *active* for the purpose of dominating one end-point by the other, only with some independent probability p_e . The status of an edge is only known when we “probe” one of its end points. The goal is to find an adaptive strategy for probing a sequence of at most k vertices that are connected in the (deterministic) graph G , so as to maximize the expected number of dominated vertices. This models, for eg., a sensor network application in the stochastic setting (see Section 6). Using Corollary 1.1 and the deterministic algorithm from [31], we obtain:

THEOREM 1.4. *There is an $O(1)$ -approximation algorithm for stochastic connected dominating set.*

1.1.1 Our Techniques The proofs for both Theorems 1.1 and 1.2 rely on working directly with an optimal adaptive decision tree \mathcal{T}^* . The non-adaptive strategy in both cases has the useful property that it is a

random root-leaf path in \mathcal{T}^* drawn according to the natural distribution induced by \mathcal{T}^* . It seems important to explicitly reason about tree \mathcal{T}^* —see Appendix B.2 for a plausible approach that attempts to bypass working with \mathcal{T}^* and an example where it fails.

We now present the main ideas in proving Theorem 1.1 in the *special case of $k = 1$ matroid and $\{0, 1\}$ values*. Here, the distribution of each element e is just given by $p_e = \Pr[V_e = 1]$.

For each leaf node ℓ in \mathcal{T}^* (corresponding to probed elements P_ℓ and chosen elements $S_\ell \subseteq P_\ell$), we construct a “witness” set $W_\ell \subseteq P_\ell$ such that the corresponding “witness” vector

$$\mathbf{w}_\ell(e) := \begin{cases} p_e & \text{if } e \in W_\ell \\ 0 & \text{otherwise} \end{cases}, \quad \forall e \in U$$

satisfies *all* matroid rank constraints up to a factor $\rho = O(\log n)$. These witnesses are meant to capture the expected value obtainable by the respective non-adaptive strategies. Formally, using *contention resolution* [17] (or *correlation gap* [4]) of matroids, one can show that if $\frac{1}{\rho} \cdot \mathbf{w}_\ell$ lies in the matroid polytope then the expected rank of the non-adaptive strategy corresponding to leaf ℓ is $\Omega(\frac{1}{\rho}) \cdot \langle \mathbf{1}, \mathbf{w}_\ell \rangle$. So the expected non-adaptive value, taking expectation over (a) the random choice of the leaf ℓ (which determines the non-adaptive strategy) and (b) the random outcomes of elements P_ℓ probed by the non-adaptive strategy, is $\Omega(1/\rho) \cdot \mathbb{E}_\ell[\langle \mathbf{1}, \mathbf{w}_\ell \rangle]$.

The witness sets W_ℓ are constructed in a natural and consistent manner across all leaves: starting from the root of \mathcal{T}^* , set W_ℓ contains the maximum prefix of elements on the decision path to leaf ℓ that satisfies the ρ -scaled rank constraints. (Formally, we use a suitably modified version of \mathcal{T}^* in defining the witnesses.)

The final and crucial step is to show that $\mathbb{E}_\ell[\langle \mathbf{1}, \mathbf{w}_\ell \rangle]$, the expected value of the witness, is close to that of the best adaptive strategy. To show this, we prove that if the decision tree \mathcal{T}^* is truncated at the first node where *any* rank constraint is violated by more than factor ρ then the expected value of \mathcal{T}^* reduces only by a constant factor. This uses a martingale tail bound and a careful union bound to handle all 2^n matroid rank constraints.

Handling weights in Theorem 1.1 (i.e. when V_e are arbitrary random variables) requires more ideas. Indeed, if we were satisfied with losing another $O(\log n)$ factor then we could group values in powers of two and reduce to the $\{0, 1\}$ -valued case by focusing on a single group. Instead, we observe that any weighted matroid rank function can be reduced (at a constant-factor loss) to the sum of multiple *unweighted* matroid rank functions. Then we combine this with the “linearity prop-

erty” in the proof of Theorem 1.1 for unweighted matroids, i.e., the non-adaptive strategy is just a *random* root-leaf path in \mathcal{T}^* . One final step is to implement the non-adaptive strategy in an *online* fashion: this is because the witnesses only correspond to the value of an offline decision-making strategy (that selects elements after observing outcomes of all probes). This conversion from offline to online strategies is done by a direct application of matroid “prophet inequalities” [32] which incurs an extra factor of two.

To handle intersections of multiple matroids in Theorem 1.1 we just need the appropriate results on (i) approximation ratio of the greedy algorithm, (ii) contention resolution schemes, (iii) reducing weighted ranks to unweighted, and (iv) prophet inequalities. Each of these steps incurs an extra factor k loss.

The improved bound for partition-matroids in Theorem 1.2 also relies on the idea of witness sets described above. However, using a more nuanced “truncation” operation, we can bound the loss ρ by a $O(1)$ for any constant number of partition matroids, instead of the $O(\log n)$ loss in the weighted case. This truncation relies on the partition matroid structure and does not extend to general matroids.

1.2 Related Work Our results add to the body of work on stochastic packing problems. Various combinatorial optimization problems have been studied in this setting, e.g., knapsack [20, 10, 35], packing integer programs [19], budgeted multi-armed bandits [22, 25] and orienteering [23, 26]. Except for orienteering, all the other stochastic problems rely on having good LP-based approximation algorithms and adaptivity gaps.

When both outer and inner constraints of our stochastic probing model are b -matchings in a graph, and all values are Bernoulli random variables, this is the stochastic matching problem formalized by [18] with applications to online dating and kidney exchange. An important notion in this problem is the *query-commit* restriction: if a probed element is present (i.e., has positive value) then it must irrevocably be added to the solution S . The notion of “online decision-making” in our model is different since we can choose to reject an element after observing its value.³ Many recent papers have studied stochastic matchings [1, 7, 9, 2]. Again, the use of good LP relaxations is crucial to these approximation results. More generally, there is a vast body of work on algorithms for kidney exchange problems, see, e.g. [36, 5].

The general stochastic probing problem under

Bernoulli values was introduced in [27], where the authors gave an approximation algorithm and adaptivity gap for inner/outer constraints given by intersections of matroids and knapsacks. [3] improved the adaptive approximation algorithms for this problem under matroid intersection constraints, and also handled submodular objectives. Both results use LP relaxations; in the submodular case a non-linear relaxation [12] is used. This approach fails in our setting if good polyhedral relaxations for the outer constraints are missing.

A different stochastic probing model was considered in [21, 24] for minimization problems. In that setting, the “outer” constraint is very simple (probe at most some number/cost of elements) but the inner optimization problem is complex (e.g., scheduling, clustering, Steiner tree problems). Moreover, unprobed elements could be used in the solution to the inner optimization problem.⁴ Many results in this setting also bounded the adaptivity gap of probing. However, due to the complex outer constraints in our model, these previous results do not seem applicable.

A stochastic version of orienteering was introduced by [26] where the rewards v_i are deterministic, but each task requires a random amount of “processing” time before the reward can be collected, and the travel time *plus the total processing time* is constrained to be at most some budget B . [26] gave an $O(\log \log B)$ approximation and adaptivity gap for stochastic orienteering, and [8] showed that the adaptivity gap is $\Omega(\sqrt{\log \log B})$. This problem is very different from stochastic probing on metrics that we consider here: (i) there is no inner constraint in [26] and (ii) the stochasticity in [26] is in the constraint as opposed to the objective. There is still a high-level similarity in the approach since both papers directly deal with the optimal decision tree.

2 Definitions and Notation

A *stochastic probing problem* instance is given by $\mathcal{J} = (U, \mathcal{F}_{out}, \mathcal{F}_{in}, \{V_e\}_{e \in U})$, where U is a set of n elements. For each element $e \in U$, the *value* is a random variable V_e (whose probability distribution is given as input). The values of different elements are independent. There are two constraints:

- **Outer constraints:** this is a collection \mathcal{F}_{out} of *sequences* of distinct elements from U , which specifies which ordered set of elements can be probed. We make the minimal assumption that the outer constraint \mathcal{F}_{out} is *prefix-closed*, i.e., for every sequence $\sigma \in \mathcal{F}_{out}$, any prefix of σ is also in \mathcal{F}_{out} .
- **Inner constraints:** this is a collection $\mathcal{F}_{in} \subseteq 2^U$ specifying which *subsets* of elements S can be

³However, query-commit and online decision-making coincide in the case of Bernoulli values.

⁴We can also handle a variant of our model where unprobed elements may be chosen: details deferred to a full version.

output.

A *strategy* now corresponds to a decision tree \mathcal{T} as follows. Each node u of \mathcal{T} is either (i) a leaf node indicating the probing has ended, or (ii) an internal node corresponding to probing some element $\text{elt}(u) \in U$ and for each outcome of the random variable $V_{\text{elt}(u)}$, there is an outgoing arc to a child node. Elements cannot be repeated on root-leaf paths. Each leaf node ℓ is associated with:

- (i) The sequence of nodes P_ℓ on the path from root to leaf ℓ . We denote by $\text{elt}(P_\ell)$ the sequence of elements probed in P_ℓ . The outer constraints enforce that $\text{elt}(P_\ell) \in \mathcal{F}_{out}$.
- (ii) The set S_ℓ of elements chosen when this path P_ℓ is traced. Note that $S_\ell \subseteq \text{elt}(P_\ell)$ and $S_\ell \in \mathcal{F}_{in}$.

The strategy \mathcal{T} is called an *online strategy* if, once an element e is probed and its value V_e known, we make an irrevocable decision about whether to include it in the solution S or not. Formally, a strategy \mathcal{T} is online if for any pair of root-leaf paths P_ℓ, P_k in \mathcal{T} , we have $\text{elt}(u) \in S_\ell \iff \text{elt}(u) \in S_k$ for every node $u \in P_\ell \cap P_k$. On the other hand, an *offline strategy* may choose the solution S after observing the values of *all* the probed elements. Clearly offline strategies obtain at least as much value as online strategies. Our results will bound the gap between *online non-adaptive* and *offline adaptive* strategies.

Let ω denote the outcome of all the random variables $\{V_e\}_{e \in U}$. For any strategy \mathcal{T} , the random outcome naturally defines a probability distribution over root-leaf paths; we denote this distribution on paths by $\pi_{\mathcal{T}}$. For any particular outcome ω let $\ell(\omega)$ denote the unique leaf-node of \mathcal{T} at which the strategy ends. (Occasionally we use $\ell \leftarrow \pi_{\mathcal{T}}$ to denote this random choice of leaf.) The offline adaptive value of a strategy \mathcal{T} is denoted

$$(2.1) \quad \text{adapt}(\mathcal{T}) = \mathbb{E}_\omega \left[\max_{S \subseteq \text{elt}(P_{\ell(\omega)}): S \in \mathcal{F}_{in}} \sum_{e \in S} V_e(\omega) \right].$$

The *adaptive optimum* is the strategy \mathcal{T}^* that maximizes Eq.(2.1); let $\text{opt}_{ad} := \text{adapt}(\mathcal{T}^*)$ be the value of this offline adaptive optimum.⁵

We define a *non-adaptive strategy* as one where all nodes at any particular level of the decision tree probe the same element: the probing decisions can be specified by a single sequence. Of course, depending on different outcomes of each element, we may take different decisions about including elements into the solution S , so the sets labeling the leaves may still be

different. An important advantage of a non-adaptive strategy is that it can be specified using much less space—it is just an *a priori* path P with $\text{elt}(P) \in \mathcal{F}_{out}$. For an *offline* non-adaptive strategy given by path P , its objective is

$$(2.2) \quad \text{pathval}(P) := \mathbb{E}_\omega \left[\max_{S \subseteq \text{elt}(P): S \in \mathcal{F}_{in}} \sum_{e \in S} V_e(\omega) \right].$$

Recall that an offline strategy can look at all the outcomes before choosing S . While online non-adaptive strategies might give less value, we can bound this gap via “prophet inequalities” (see, e.g., [34, 32]).

A family $\mathcal{F} \subseteq 2^U$ of subsets of groundset U is said to admit a β -approximate prophet inequality if there is an efficient online algorithm that for any sequence $W_1, \dots, W_{|U|}$ of random variables with distributions known upfront, selects a subset $S \in \mathcal{F}$ of elements by deciding on each element e right after observing W_e , such that

$$\mathbb{E}[\sum_{e \in S} W_e] \geq \frac{1}{\beta} \cdot \mathbb{E}[\max_{T \in \mathcal{F}} \sum_{e \in T} W_e].$$

The value in the right-hand-side above corresponds to a set T that is chosen after observing all the random variables.

PROPOSITION 2.1. *If the inner constraint \mathcal{F}_{in} admits a β -approximate prophet inequality, then for any path P , there is an online non-adaptive strategy (that inspects elements in the order given by P) having expected value at least $\beta \cdot \text{pathval}(P)$.*

It is known that $\beta = \Omega(1/k)$ when the inner constraints are intersections of k matroids [32]. (For the simple case of a single matroid of rank 1, which corresponds to picking the maximum, the optimal value is $\beta = 1/2$ [34].) The advantage of Proposition 2.1 is that we can use the expression in Eq.(2.2) as a surrogate for the value of the online non-adaptive optimum (which is difficult to work with directly).

Let P^* be the optimal *online non-adaptive* strategy for a given instance, with value opt_{na} (that lies between $\beta \text{pathval}(P^*)$ and $\text{pathval}(P^*)$). The *adaptivity gap* of a stochastic probing problem is the maximum (over all instances) of the ratio $\frac{\text{opt}_{ad}}{\text{opt}_{na}}$; which compares to the *offline adaptive* strategy.

3 The Adaptivity Gap for Unweighted Instances

In this section we prove our main adaptivity gap result, Theorem 1.1, but in the “*unweighted*” case where each element e is either *active* with probability p_e and has value $V_e = 1$, or is *inactive* with probability $1 - p_e$ and

⁵We also consider randomized strategies, where some nodes correspond to making a random decision instead of probing a node. All definitions above extend to this case with minor changes.

has value $V_e = 0$. In Section 5 we extend this adaptivity gap result to the general weighted case, at the loss of an additional $O(k)$ factor. The central theorem of this section is the following.

THEOREM 3.1. (UNWEIGHTED ADAPTIVITY GAP)
Consider an unweighted stochastic probing instance with k -matroid intersection inner constraints and any prefix-closed outer constraints. For any strategy \mathcal{T} ,

$$\mathbb{E}_{P \leftarrow \pi_{\mathcal{T}}} [\text{pathval}(P)] \geq \frac{1}{O(k^2 \cdot \log(kn))} \cdot \left(\text{adapt}(\mathcal{T}) - \frac{1}{n^2} \max_{e \in U} p_e \right).$$

Note that $\max_{e \in U} p_e \leq \text{opt}_{ad}$ by Fact A.1. Choosing \mathcal{T} to be the optimal adaptive strategy \mathcal{T}^* and combining with Proposition 2.1, it follows that a random root-leaf path in \mathcal{T}^* (drawn from its natural probability measure) gives an online non-adaptive strategy with expected value within a factor of $O(k^3 \log(kn))$ of the optimum adaptive value. (The extra factor of k arises due to the prophet inequality, in going from the offline to the online strategy.) Using the probabilistic method, this immediately bounds the adaptivity gap, and proves Theorem 1.1 for the unweighted case. For the special case of a *uniform matroid*, we give a simpler self-contained proof in Section 7.1: that proof has the same high-level structure as the general proof below, but each step is simpler and it yields an $O(1)$ adaptivity gap.

In the rest of this section, we prove Theorem 3.1. Suppose the matroids defining the inner constraints are $\{\mathcal{M}_i\}_{i=1}^k$, with rank functions $r_i : 2^U \rightarrow \mathbb{Z}_{\geq 0}$. For any strategy \mathcal{T} with root node s , recall that \mathcal{T} is represented as a directed tree, where each internal node u corresponds to probing some element $\text{elt}(u) \in U$. For any node $u \in \mathcal{T}$, let $P_u = \langle s = v_0, v_1, \dots, v_t = u \rangle$ be the path connecting the root to u , so $\text{elt}(P_u) = \langle \text{elt}(v_0), \text{elt}(v_1), \dots, \text{elt}(v_t) \rangle$. A leaf node u does not probe any element, so define $\text{elt}(P_u) = \text{elt}(P_{\hat{u}})$ where \hat{u} is u 's parent.

Preprocessing. For each node $u \in \mathcal{T}$ let I_u be a greedy independent set in $\bigcap_{i=1}^k \mathcal{M}_i$ among the *active* elements probed strictly before u ; hence set I_u does not include $\text{elt}(u)$. We modify strategy \mathcal{T} by ignoring the value from every node u for which $\text{elt}(u) \cup I_u$ is not independent, i.e., by treating such nodes u as a random choice made by the strategy (with no additional value). Let R denote the set of nodes in \mathcal{T} which are not ignored—call them *relevant* nodes. Henceforth we restrict \mathcal{T} to the value obtained from nodes in R . This incurs a loss of at most factor k in the adaptive value since the greedy algorithm is a k -approximation for the maximum independent set in

k -matroid intersection [30]. Note that now the set of active elements seen along any path in \mathcal{T} is independent, so the expected “greedy” value of strategy \mathcal{T} is now:

$$(3.3) \quad \text{greedy}(\mathcal{T}) := \mathbb{E}_{\ell \leftarrow \pi_{\mathcal{T}}} \left[\sum_{e \in \text{elt}(R \cap P_{\ell})} p_e \right] \\ = \mathbb{E}_{\omega} \left[\sum_{e \in \text{elt}(R \cap P_{\ell(\omega)})} \mathbf{1}(e \text{ active in } \omega) \right].$$

Here ℓ denotes a leaf node in \mathcal{T} drawn according to the natural distribution $\pi_{\mathcal{T}}$ and P_{ℓ} is the root-to- ℓ path. The first expectation in Eq.(3.3) has the total probability on R -nodes of a random path and the second expectation has the number of active instantiations on R -nodes of a random path; these are equal since the decision to probe an element is independent of its outcome. The above discussion shows that $\text{greedy}(\mathcal{T}) \geq \frac{1}{k} \cdot \text{adapt}(\mathcal{T})$.

Now, for each node $u \in \mathcal{T}$ define a “witness” vector $x^{(u)} \in \mathbb{R}^U$ as follows:

$$(3.4) \quad x_e^{(u)} = \begin{cases} p_e & \text{if } \exists v \in R \text{ on path } P_u: \text{elt}(v) = e \\ 0 & \text{otherwise} \end{cases}$$

For any $i \in [k]$, a node u is called ρ -valid for matroid i if $x^{(u)}$ satisfies *all* rank constraints of \mathcal{M}_i up to a ρ -factor,

$$\text{i.e., } \sum_{e \in A} x_e^{(u)} \leq \rho \cdot r_i(A), \quad \forall A \subseteq U.$$

Otherwise, the node u is called ρ -invalid for matroid i . Since $x^{(u)} \geq x^{(v)}$ when u is a descendent of v , the ρ -valid nodes form a connected subtree from the root.

If $\rho = O(\log n + \log k)$, we will show how to truncate the tree (i.e., reduce the set R of relevant nodes) so that every node becomes ρ -valid for *one* matroid \mathcal{M}_i , and the value of the resulting tree (following the definition in Eq.(3.3)) reduces by at most $\frac{1}{n^2 k^3} \max_{e \in U} p_e$. By repeating this for each of the k matroids, we would obtain a tree $\tilde{\mathcal{T}}$ where each node is ρ -valid for *every* matroid $\{\mathcal{M}_i\}_{i=1}^k$ and has loss at most $\frac{1}{n^2 k^2} \max_{e \in U} p_e$.

Truncation for single matroid \mathcal{M} . Let \mathcal{T}' denote an arbitrary decision tree with relevant nodes R and vectors $x^{(u)}$ defined as above, where we consider only the value from nodes of R . Let \mathcal{M} be any matroid with rank function r . We now describe a “truncation” procedure that reduces the set R of relevant nodes so that *all* nodes in the resulting tree \mathcal{T}'' are ρ -valid for \mathcal{M} . Define *frontier nodes* to be the set F of ρ -invalid nodes that are closest to the root. The truncated tree \mathcal{T}'' is obtained as follows:

For each node $u \in F$ do: *remove* from R all nodes of the subtree of \mathcal{T}' below u (i.e., ignore the value obtained from all such nodes).

The value of the new tree \mathcal{T}'' is calculated by applying Eq.(3.3) on the modified set R . The vectors $x^{(u)}$ on nodes of \mathcal{T}'' are also updated according to Eq.(3.4) with the new set R . Due to truncation, the value of \mathcal{T}'' is now less than that of \mathcal{T} . The following key lemma helps us control this loss.

LEMMA 3.1. *The value of the truncated tree is $\text{greedy}(\mathcal{T}'') \geq \text{greedy}(\mathcal{T}') - \frac{1}{n^2 k^3} \max_{e \in U} p_e$.*

Proof. [Proof of Lemma 3.1] Recall that $u \in F$ (frontier) if it is the highest node on P_u such that for some set $A \subseteq U$, we have $\sum_{e \in A} x_e^{(u)} > \rho \cdot r(A)$. Let P denote a random path in \mathcal{T}' drawn according to distribution $\pi_{\mathcal{T}'}$. For any $A \subseteq U$, let \mathcal{E}_A be the event that P hits a node of F violated for set A . Clearly, the event of reaching some frontier node, $\{P \cap F \neq \emptyset\} = \bigcup_{A \subseteq U} \mathcal{E}_A$.

Using the first expectation in the definition of **greedy** from Eq.(3.3), the loss in truncation below any node $u \in F$ is at most $\Pr[P \ni u] \cdot n \cdot \max_e p_e$. So the total loss, $\text{greedy}(\mathcal{T}') - \text{greedy}(\mathcal{T}'') \leq n \cdot (\max_e p_e) \cdot \Pr[P \cap F \neq \emptyset]$. Next, we show $\Pr[P \cap F \neq \emptyset] \leq \frac{1}{(nk)^3}$ which would prove the lemma. We need the following crucial claim.

CLAIM 4. *If $\rho = \Omega(\log(nk))$, then for any set $A \subseteq U$, $\Pr[\mathcal{E}_A] \leq \frac{1}{2^{nr(A)} n^3 k^3}$.*

The proof, which is deferred for now, uses the fact that the set I_u of observed active elements is always independent and the $x^{(u)}$ variables (roughly) capture the expected behaviour of I_u . So the probability that these two deviate from each other falls exponentially. The formal proof is via a large-deviations bound for martingales.

Since there are an exponential number of sets A , we cannot directly use Claim 4 and union bound to get a handle on $\Pr[P \cap F \neq \emptyset]$. Still, we show how this can be done using the matroid structure. Call a set A *critical* of rank q if it is a maximal set of rank q , i.e., $r(A) = q$ and $r(A') > r(A)$ for all $A' \supsetneq A$. Since each critical set is the span of some independent set of rank q , there are at most $\binom{n}{q}$ critical sets of rank q for matroid \mathcal{M} . Moreover, by maximality of critical sets, if $x(A) \leq \rho r(A)$ for every critical set A , then $x(Y) \leq \rho r(Y)$ for all $Y \subseteq U$. Hence it suffices to focus on the events \mathcal{E}_A over critical sets A . So,

$$\begin{aligned} \Pr[P \cap F \neq \emptyset] &\leq \sum_{A \text{ critical}} \Pr[\mathcal{E}_A] \\ &\leq \frac{1}{2} \sum_{q=1}^n \binom{n}{q} \frac{n^{-q}}{(nk)^3} \leq \frac{1}{(nk)^3}. \end{aligned}$$

The second inequality uses Claim 4. This completes the proof of Lemma 3.1.

The final truncated tree $\tilde{\mathcal{T}}$. We now take the tree \mathcal{T} and perform the above (single matroid) truncation sequentially for each matroid $\{\mathcal{M}_i\}_{i=1}^k$. Repeated application of Lemma 3.1 yields a decision tree $\tilde{\mathcal{T}}$ where each node is ρ -valid for *every* matroid $\{\mathcal{M}_i\}_{i=1}^k$ and has value:

$$\begin{aligned} (4.5) \quad \text{greedy}(\tilde{\mathcal{T}}) &\geq \text{greedy}(\mathcal{T}) - \frac{1}{(nk)^2} \max_{e \in U} p_e \\ &\geq \frac{1}{k} \cdot \text{adapt}(\mathcal{T}) - \frac{1}{(nk)^2} \max_{e \in U} p_e \end{aligned}$$

The second inequality uses the fact that $\text{greedy}(\mathcal{T}) \geq \frac{1}{k} \cdot \text{adapt}(\mathcal{T})$ from the preprocessing on \mathcal{T} . Note that $\text{greedy}(\tilde{\mathcal{T}}) = \mathbb{E}_{\ell \leftarrow \pi_{\tilde{\mathcal{T}}}} \left[\sum_{e \in U} x_e^{(\ell)} \right]$ by the first expectation in Eq.(3.3). Thus to complete the proof, it suffices to show:

LEMMA 4.1. *For any root-leaf path $P_\ell \in \tilde{\mathcal{T}}$ with leaf ℓ , $\text{pathval}(P_\ell) \geq \frac{1}{(k+1)\rho} \cdot \sum_{e \in U} x_e^{(\ell)}$.*

Proof. For brevity, denote $y_e := \frac{1}{\rho} x_e^{(\ell)}$. Since all nodes in $\tilde{\mathcal{T}}$ (including ℓ) are ρ -valid for all k matroids, \mathbf{y} is a point in the matroid intersection polytope of $\bigcap_{i=1}^k \mathcal{M}_i$. Recall that $\text{pathval}(P_\ell)$ is precisely the expected maximum independent (in $\bigcap_{i=1}^k \mathcal{M}_i$) subset of a random subset of P_ℓ that contains each element e independently with probability p_e . Using ‘‘contention resolution schemes’’ [17] or the ‘‘correlation gap’’ [37] for k -matroid intersection, it follows that $\text{pathval}(P_\ell) \geq \frac{1}{k+1} \sum_{e \in U} y_e$.

Putting together all the pieces, the expected **pathval** of a random path in \mathcal{T} is at least that of a random path in $\tilde{\mathcal{T}}$ since we only truncated the tree. By Lemma 4.1 this is at least a $\frac{1}{(k+1)\rho}$ factor of $\sum_{e \in U} x_e^{(\ell)}$. Taking expectation over ℓ and using Eq.(3.3) and Eq.(3.4), the latter summation equals $\text{greedy}(\tilde{\mathcal{T}})$; finally using Eq.(4.5) we obtain Theorem 3.1.

It only remains to prove Claim 4 which we had assumed in the above analysis.

Proof. [Proof of Claim 4] The idea here is to use the following concentration inequality:

THEOREM 4.1. (THEOREM 1 IN [38]) *Let N_1, N_2, \dots be a sequence of possibly dependent random variables: for each $t \geq 1$ variable N_t depends only on N_{t-1}, \dots, N_1 . Consider also a sequence of random values $\xi_t(N_1, \dots, N_t)$ that lie in $[0, 1]$. Let $\mathbb{E}_{N_t}[\xi_t(N_1, \dots, N_t)]$ denote the expectation of ξ_t with respect to N_t , conditional on N_1, \dots, N_{t-1} . Furthermore, let τ denote any stopping time. Then we have $\forall \delta \geq 0$:*

$$\Pr \left[\sum_{t=1}^{\tau} \mathbb{E}_{N_t}[\xi_t(N_1, \dots, N_t)] \geq \right]$$

$$\frac{e}{e-1} \cdot \left(\sum_{t=1}^{\tau} \xi_t(N_1, \dots, N_t) + \delta \right) \leq \exp(-\delta)$$

We make use of this result by setting N_t to be the t^{th} node seen by a random path P in tree \mathcal{T}' and

$$\xi_t(N_1, \dots, N_t) = \begin{cases} \mathbf{1}(\text{elt}(N_t) \text{ is active}) & \text{if } \text{elt}(N_t) \in A \\ 0 & \text{otherwise} \end{cases}$$

The stopping time τ corresponds to path P reaching either a frontier node violated for set A (which captures event \mathcal{E}_A) or the end of tree \mathcal{T}' . By the preprocessing of \mathcal{T}' and definition of the relevant nodes R , it follows that:

$$\begin{aligned} & \sum_{t=1}^{\tau} \xi_t(N_1, \dots, N_t) \\ &= \sum_{t=1}^{\tau} \mathbf{1}(\text{elt}(N_t) \in A) \cdot \mathbf{1}(\text{elt}(N_t) \text{ active}) \\ &= \sum_{e \in \text{elt}(R \cap P) \cap A} \mathbf{1}(e \text{ is active}) \leq r(A) \end{aligned}$$

The inequality is because the set of active elements before any node in \mathcal{T}' is independent. On the other hand,

$$\begin{aligned} \sum_{t=1}^{\tau} \mathbb{E}_{N_t}[\xi_t(N_1, \dots, N_t)] &= \sum_{t=1}^{\tau} \mathbf{1}(\text{elt}(N_t) \in A) \cdot p_{\text{elt}(N_t)} \\ &= \sum_{e \in A} x_e^{(N_\tau)} > \rho \cdot r(A), \end{aligned}$$

where we used the definition of the “expected” active set $x^{(u)}$ from Eq.(3.4) and the fact that τ corresponds to nodes where the rank of A is violated by factor more than ρ .

Applying Theorem 4.1 with $\delta = 4 \log(nk) \cdot r(A)$ and assuming $\rho > 8 \log(nk)$, it follows that $\Pr[\mathcal{E}_A]$ is at most $\exp(-4 \log(nk) \cdot r(A)) \leq \frac{1}{2 n^{r(A)} n^3 k^3}$ (for $n \geq 2$) as desired.

5 Weighted Stochastic Probing

We now extend the proof of Theorem 3.1 that gave an adaptivity gap for unweighted rank functions to the weighted case, and hence prove our main adaptivity gap in Theorem 1.1.

The first ingredient is a lemma showing that weighted rank functions of intersections of k matroids can be written as sums of unweighted rank functions of intersections of some other related matroids, up to a loss of a factor of $O(k)$. Consider k matroids $\{\mathcal{M}_i\}_{i=1}^k$ defined on ground set U with element weights $\{w_e \geq 0\}_{e \in U}$. The k -matroid weighted rank function

$w : 2^U \rightarrow \mathbb{R}_+$ is defined as

$$w(S) = \max \left\{ \sum_{e \in I} w_e : I \subseteq S, I \text{ independent in } \bigcap_{i=1}^k \mathcal{M}_i \right\}$$

for each $S \subseteq U$. A k -matroid *unweighted* rank function corresponds to the special case of unit weights.

Let us partition the ground set based on the weights as follows:

$$U_j := \{e \in U : 2^{j-1} \leq w_e < 2^j\} \quad \forall j \in \mathbb{Z}.$$

For each j , let $u_j(\cdot)$ denote the k -matroid *unweighted* rank function restricted to elements in U_j . We claim that the desired positive linear combination is $w'(\cdot) = \sum_{j \in \mathbb{Z}} 2^j \cdot u_j(\cdot)$.

LEMMA 5.1. *For any k -matroid weighted rank function $w : 2^U \rightarrow \mathbb{R}_+$, the positive linear combination w' of k -matroid unweighted rank functions satisfies $w(S) \leq w'(S) \leq 4k w(S), \forall S \subseteq U$.*

Proof. The first inequality is easy: $w'(S) \geq w(S)$ for any $S \subseteq U$. Indeed, if $I \subseteq S$ is the max-weight independent set then each $I \cap U_j$ is independent in the restriction to U_j , and so $w'(S) \geq \sum_j 2^j \cdot |I \cap U_j| \geq \sum_{e \in I} w_e$.

For the other inequality, fix some $S \subseteq U$. For each j , let I_j be a maximum cardinality independent set in $S \cap U_j$, so $u_j(S) = |I_j|$. If $\bigcup_j I_j$ were independent then we would have $w(S) \geq w'(S)/2$. But this is not necessarily independent, which loses an extra k factor as follows.

Consider running the greedy algorithm on $\bigcup_j I_j$ to select an independent set in the k -matroid intersection, where elements are inspected in decreasing order of the I_j s. Let $G_j \subseteq I_j$ denote the elements chosen into the greedy independent set. We use the shorthand $G_{\geq j} := \bigcup_{l \geq j} G_l$; set $I_{\geq j}$ is defined similarly. Notice that both I_j and $G_{\geq j}$ are independent subsets of $I_{\geq j}$: the properties of the greedy algorithm on intersections of k matroids implies $k \cdot |G_{\geq j}| \geq |I_j|$. So

$$k \sum_{j \in \mathbb{Z}} 2^j \cdot |G_j| \geq \sum_{j \in \mathbb{Z}} 2^j \cdot |I_j| - k \sum_{j \in \mathbb{Z}} 2^j \cdot |G_{\geq j+1}|.$$

By rearranging, we have

$$k \sum_{j \in \mathbb{Z}} |G_j| (2^j + 2^{j-1} + \dots) \geq \sum_{j \in \mathbb{Z}} 2^j \cdot |I_j|.$$

Hence $2k \sum_j 2^j \cdot |G_j| \geq \sum_j 2^j \cdot |I_j|$. Finally, since $G := \bigcup_j G_j$ is an independent set in the k -matroid intersection,

$$w(S) \geq \sum_{e \in G} w_e \geq \sum_{j \in \mathbb{Z}} 2^{j-1} \cdot |G_j|$$

$$\geq \frac{1}{4k} \sum_{j \in \mathbb{Z}} 2^j \cdot |I_j| = \frac{1}{4k} w'(S).$$

This completes the proof of Lemma 5.1.

Next we combine Lemma 5.1 with Theorem 3.1 for unweighted probing (where elements randomly take on values in $\{0, 1\}$ with some probabilities) to prove Theorem 1.1 for weighted probing (where element values are drawn from general distributions over $\mathbb{R}_{\geq 0}$).

Proof. [Proof of Theorem 1.1 (weighted case)] Consider a stochastic probing instance \mathcal{J} with k -matroid intersection inner constraints and any prefix-closed outer constraints. Let \mathcal{T} be any adaptive strategy. By discretising values to powers of two, we may assume (at the loss of a constant factor) that each $V_e \in \{2^j \mid j \in \mathbb{Z}\}$. Let $p_{e,j}$ be the probability that V_e takes on value 2^j , where $\sum_j p_{e,j} = 1$ for all e . Define the mean value $\mu_e := \mathbb{E}[V_e] = \sum_j 2^j \cdot p_{e,j}$.

We create a new *unweighted* stochastic probing instance \mathcal{J}_j for each integer $j \in \mathbb{Z}$. The ground set U , inner and outer constraints are the same; however, the values $\{V_e^j\}_{e \in U}$ are restricted to $\{0, 1\}$. In particular, $V_e^j = 1$ with probability $p_{e,j}$ and $V_e^j = 0$ with probability $1 - p_{e,j}$. Note that \mathcal{T} corresponds to a *randomized* adaptive strategy for instance \mathcal{J}_j as follows: at a node containing element e :

- if $V_e^j = 1$ then take the branch $V_e = 2^j$, and
- if $V_e^j = 0$, randomly take one of the other branches, where branch corresponding to $V_e = 2^h$ (for $h \neq j$) is taken with probability proportional to $p_{e,h}$.

Since Theorem 3.1 also holds for randomized strategies \mathcal{T} , setting $\alpha = \Omega(1/k^2 \log(kn))$ we have:

$$\begin{aligned} \mathbb{E}_{P \leftarrow \pi_{\mathcal{T}}} [\text{pathval}_j(P)] &\geq \alpha \cdot \left(\text{adapt}_j(\mathcal{T}) - \frac{1}{n^2} \max_{e \in U} p_{e,j} \right) \\ &\geq \alpha \cdot \left(\text{adapt}_j(\mathcal{T}) - \frac{1}{n^2} \sum_{e \in U} p_{e,j} \right). \end{aligned}$$

Here we used the notation pathval_j and adapt_j to emphasize that these correspond to instance \mathcal{J}_j . Clearly, $\sum_j 2^j \cdot \text{adapt}_j(\mathcal{T}) \geq \text{adapt}(\mathcal{T})$ since this holds for every outcome of the values. Hence

$$\begin{aligned} &\mathbb{E}_{P \leftarrow \pi_{\mathcal{T}}} \left[\sum_j 2^j \cdot \text{pathval}_j(P) \right] \\ &\geq \alpha \cdot \left(\text{adapt}(\mathcal{T}) - \frac{1}{n^2} \sum_j 2^j \cdot \sum_{e \in U} p_{e,j} \right) \\ &\geq \alpha \cdot \left(\text{adapt}(\mathcal{T}) - \frac{1}{n} \max_{e \in U} \mathbb{E}[V_e] \right). \end{aligned}$$

Fix any root-leaf path P in \mathcal{T} . Notice that $\text{pathval}(P)$ is the expected *weighted* rank of the inner k -matroid intersection. Whereas, $\sum_j 2^j \cdot \text{pathval}_j(P)$ is a positive linear combination of *unweighted* rank functions, whose definition matches that in Lemma 5.1. Now conditioning on all the outcomes ω , $\text{pathval}(P)$ is just the weighted rank of P , whereas $\text{pathval}_j(P)$ is the unweighted rank restricted to the elements in U_j (under the outcome ω). Applying Lemma 5.1, we get that $\sum_j 2^j \cdot \text{pathval}_j(P) \leq 4k \cdot \text{pathval}(P)$ for each fixed ω . Taking expectations over ω and using linearity of expectations, we have

$$\mathbb{E}_{P \leftarrow \pi_{\mathcal{T}}} [\text{pathval}(P)] \geq \frac{\alpha}{4k} \cdot (\text{adapt}(\mathcal{T}) - \frac{1}{n} \max_{e \in U} \mathbb{E}[V_e]).$$

Now choosing \mathcal{T} to be an optimal adaptive strategy \mathcal{T}^* for the given (weighted) instance and using Fact A.1, it follows that a random path P in the optimal adaptive strategy is a non-adaptive strategy with *offline* value $\text{pathval}(P) \geq \text{opt}_{ad}/O(k^3 \log(kn))$. Finally, using prophet inequalities (Proposition 2.1) we obtain an *online* non-adaptive strategy with value at least $\text{opt}_{ad}/O(k^4 \log(kn))$.

6 Applications

6.1 Stochastic Probing on Metrics In the stochastic probing problem on metrics, recall that the elements are in a metric space (U, d) , and outer constraints enforce that the probed elements lie on a path of length at most B .

To begin, consider the simpler case where the inner constraint is a single matroid \mathcal{M} , with rank function $r(\cdot)$. Moreover, suppose each element e is active and has value 1 with probability p_e , or is inactive (has value zero) with probability $1 - p_e$. For the outcome ω , let A_ω be the active elements for this outcome. The non-adaptive problem is now to find a path Q of total length B , to maximize the quantity $g(Q) := \mathbb{E}_\omega[r(Q \cap A_\omega)]$. Since the rank function $r(\cdot)$ is a submodular function, so is $g(\cdot)$, and we have a problem of finding a path Q of length at most B to maximize the submodular function $g(Q)$. This is precisely the submodular orienteering problem for which Chekuri and Pál [16] gave an $O(\log n)$ approximation in quasi-polynomial time.

The case of multiple matroid constraints is only slightly more involved. For the intersection of k matroids, define the “rank” $r(S)$ to be the size of the largest common independent set contained in S . The function $r(\cdot)$ is no longer submodular, so we cannot use the same arguments as above. Instead we rework the proof of Chekuri and Pál, with an additional loss of a factor of k^2 . First, we put the stochasticity aside, and try to maximize the cardinality of a common independent set in the intersection of k matroids.

PROPOSITION 6.1. *There is a quasi-polynomial time $(k \log_2(2n))$ -approximate algorithm to maximize the rank function of the intersection of k matroids, subject to an orienteering constraint.*

Proof. (Sketch) We assume the reader is familiar with the elegant Savitch-like idea from [16]. By suitable guessing, we want to find the best s - t path of length B with ℓ hops, where the “half-way” point is v , the length of the s - v portion is B_1 and the rest is B_2 . Let P_1^* and P_2^* be the s - v and v - t portions of the optimal path P^* , and let I^* be the optimal independent set. Since we are in the unweighted case, the optimum solution value is $|I^*| = |I^* \cap P_1^*| + |I^* \cap P_2^*|$. We recursively find a $1/(k \log_2 \ell)$ -approximate solution I_1 to the s - v problem (with $\ell/2$ hops and budget B_1):

$$|I_1| \geq \frac{1}{k \log_2 \ell} |I^* \cap P_1^*|.$$

Then we contract the elements of I_1 and find an approximate solution I_2 for the v - t problem (with $\ell/2$ hops and budget $B - B_1$). Since contracting each element of I_1 can only reduce the rank of the problem by k , we get

$$|I_2| \geq \frac{1}{k \log_2 \ell} \left(|I^* \cap P_2^*| - k|I_1| \right).$$

Combining, we have

$$\begin{aligned} |I| &= |I_1| + |I_2| \\ &\geq \frac{1}{k \log_2 \ell} \left(|I^* \cap P_1^*| + |I^* \cap P_2^*| - k|I_1| \right) \\ \Rightarrow |I| &\geq \frac{|I^*|}{k(1 + \log_2 \ell)} = \frac{|I^*|}{k \log_2(2\ell)}. \end{aligned}$$

For the base case $\ell = 1$ the only option is to go from s to t , and get a $1 \geq \frac{1}{k \log_2 2}$ -approximation.

The entire analysis above is linear, and immediately extends to any positive linear combination of such unweighted rank functions, say those functions of the form $g(Q) := \mathbb{E}_\omega[r(Q \cap A_\omega)]$. Finally, using the reduction from weighted to unweighted rank functions (Lemma 5.1) we lose another factor of $O(k)$ and get:

THEOREM 6.1. *There is a quasi-polynomial time $O(k^2 \log n)$ -approximation algorithm for non-adaptive stochastic probing on metrics given intersections of k matroids.*

Finally, combining with Theorem 1.1 completes the proof of Theorem 1.3.

Stochastic Submodular Orienteering for Matroid Rank Functions

The above result also implies an approximation algorithm for the following *stochastic submodular orienteering* (StocSO) problem. Given a metric (U, d) with a root where each vertex $v \in U$ is active independently with probability p_v , a monotone submodular function $g : 2^U \rightarrow \mathbb{R}_+$ and length bound B , the goal is to find an adaptive path of length at most B that originates from the root and maximizes the expected function value on *active* elements. When function g is a *matroid rank function*, notice that StocSO is precisely the above stochastic probing problem on metrics with $k = 1$. Hence there is a quasi-polynomial time $O(\log^2 n)$ -approximation algorithm for StocSO under matroid rank functions. In fact, this result also holds for any positive linear combination g of matroid rank functions: this relies on the “linearity property” of the adaptivity gap result (Theorem 1.1) and the fact that such functions g are submodular (so the [16] algorithm can be used to find a non-adaptive strategy). We summarize this discussion below:

COROLLARY 6.1. *There is a quasi-polynomial time $O(\log^2 n)$ -approximation algorithm for stochastic submodular orienteering on positive linear combinations of matroid rank functions.*

This result will be useful for the next section.

6.2 Stochastic Minimum Latency Submodular Cover

The *minimum latency submodular cover* problem studied in [29] is as follows: given a metric (U, d) with root r and m monotone submodular functions $f_i : 2^U \rightarrow \mathbb{R}_+$ (for $i \in [m]$), the goal is to find a path originating from r that minimizes the total “cover time” of the m functions. Here, function f_i is said to be covered at time t if this is the minimum value such that

$$f_i(\{v \mid \text{vertex } v \text{ visited before time } t\}) = f_i(U).$$

In the *stochastic minimum latency submodular cover* (StocMLSC) problem, the input is the same as above and furthermore each vertex (element) $v \in U$ is *active* independently with probability p_v . The goal is to find an adaptive strategy to minimize the *expected* total cover time of the m functions. In the stochastic setting, function f_i is said to be covered at time t if this is the minimum value such that

$$f_i(\{v \mid v \text{ visited before time } t \text{ and active}\}) = f_i(U).$$

Due to the stochasticity, there may be situations where some functions f_i never reach the maximum value $f_i(U)$ —in such cases the cover time is just set to be the entire path length. We assume, without loss

of generality, that functions are normalized so that $f_i(U) = 1$ for $i \in [m]$.

Obtaining a poly-logarithmic approximation ratio for StocMLSC was left open in [29]. In this paper, we answer this question for the interesting case when each f_i is a matroid rank function.

THEOREM 6.2. *There is a quasi-polynomial time $O(\log^3 n)$ -approximation algorithm for stochastic minimum latency submodular cover when all the input functions are matroid rank functions.*

Proof. (Sketch) The main ingredient in this algorithm is an approximation algorithm for stochastic submodular orienteering (StocSO). Given an α -approximation algorithm for this problem, the algorithm and analysis for the *deterministic* minimum latency submodular cover problem in [29] apply directly to yield an $O(\alpha \cdot \log \frac{1}{\epsilon})$ -approximation algorithm for StocMLSC, where ϵ is the smallest positive marginal value of any function $\{f_i\}_{i=1}^m$. Moreover, the function g in the StocSO instances solved by this algorithm is always a non-negative linear combination of the f_i s. We defer additional details to the full version.

In this theorem, each function f_i is restricted to be the rank function of some matroid \mathcal{M}_i . Since g is a linear combination of the f_i s, Corollary 6.1 yields quasi-polynomial time $O(\log^2 n)$ -approximation algorithm for StocSO with objective g . Applying this within the algorithm from [29], we obtain a quasi-polynomial time $O(\log^3 n)$ -approximation algorithm for StocMLSC (note that $\epsilon \geq \frac{1}{n}$ for matroid rank functions).

As an example, consider the stochastic version of the *latency group Steiner* problem [28, 13]. The input consists of (i) metric (U, d) where each vertex $v \in U$ is *active* independently with probability p_v , (ii) root $r \in U$ and (iii) m groups $\{X_i \subseteq U\}_{i=1}^m$ of vertices. The goal is to find an adaptive path (starting from r) that minimized the expected cover-time of the m groups. Group X_i is covered at the earliest time when some *active* vertex from X_i is observed. A direct application of Theorem 6.2 yields an $O(\log^3 n)$ -approximation algorithm. In this special case however, we can use (a) Theorem 1.2 to bound the adaptivity gap by $O(1)$ and (b) the fact that $\epsilon = 1$, to obtain a quasi-polynomial time $O(\log n)$ -approximation algorithm.

6.3 Stochastic Connected Dominating Set The *budgeted connected dominating set* problem (BCDS) studied in [31] is the following. Given an undirected graph $G = (V, E)$ and budget k , the goal is to choose a set S of k vertices such that $G[S]$ is a connected subgraph, and the number of vertices dominated by S

are maximized. A vertex $v \in V$ is said to be dominated by set $S \subseteq V$ if $N(v) \cap S \neq \emptyset$, where $N(v)$ is the set of neighbors of v which includes v .

We consider the following stochastic version of BCDS. The elements are edges E of a graph, and each edge $e \in E$ is *active* for the purpose of dominating one end-point by the other only with some independent probability p_e . Furthermore, the status of an edge e is only known when we “probe” one of its end points. The (random) subset of active edges is denoted $A \subseteq E$. We want an adaptive strategy for probing a sequence of at most k vertices such that:

- the set P of probed vertices at any point in time is connected in the (deterministic) graph $G = (V, E)$, and
- the expected number of vertices dominated by P in the graph $G_p = (V, A)$ is maximized.

This models, for example, a sensor network application where vertices V denote potential coverage locations and edges E correspond to pairs of locations between which two sensors can communicate. A sensor placed at location u covers u , and additionally a (random) subset of u ’s neighbors between whom the sensing quality is good enough. *A priori* we only know the probabilities of having a good sensing quality; the exact status is known only when a sensor is placed at u or v . We want to set up a connected network using k sensors (possibly adaptively) so as to maximize the expected number of covered locations.

We formally cast the problem in our framework: The ground set contains the edge set E with probabilities $\{p_e\}_{e \in E}$. In our model we must probe elements/edges (and not vertices), but the generality of our framework helps capture this. The outer constraints require that there exists a subset $S \subseteq V$ of vertices such that (i) every probed edge has at least one end-point in S , (ii) S is connected in G , and (iii) $|S| \leq k$. It is easy to see that this can be captured by a prefix-closed collection of sequences of elements in E . There is no inner constraint, but the objective is the coverage function

$$f(A) := \sum_{v \in V} \mathbf{1}(A \text{ contains edge incident to } v),$$

where A is the set of probed edges that are *active*. By Corollary 1.1 the adaptivity gap of this stochastic probing problem is $O(1)$, so we can focus on the non-adaptive problem.

In the non-adaptive stochastic BCDS problem, we want a static set $S \subseteq V$ of at most k connected vertices in G that maximizes the function

$$g(S) := |S| + \sum_{v \in V \setminus S} \Pr[v \text{ dominated by } S]$$

$$= |S| + \sum_{v \in V \setminus S} \left(1 - \prod_{e \in E(S,v)} (1 - p_e) \right).$$

Above $E(S, v)$ denotes the set of edges incident to v whose other endpoint lies in S . Not only is g a submodular function on V , it is also a “special submodular” function as defined in [31], so the algorithm from [31] yields an $O(1)$ -approximation for non-adaptive stochastic BCDS, and hence proves Theorem 1.4.

6.4 Stochastic Precedence Constrained Scheduling Consider a set U of tasks with precedence constraints, where each task $e \in U$ has an independent random *value/reward* $V_e \in \mathbb{Z}_+$. The exact value of a task is known only when it is probed, and the order of probing tasks must satisfy the precedence. (I.e., a task can be probed only when its predecessors have all be probed.) There is a bound B on the total number of probes. Each task must be irrevocably accepted/rejected immediately after it is probed. Finally, we are allowed to accept at most k tasks. The objective is to maximize the expected total value of the selected tasks.

This can be modeled as a stochastic probing problem on the ground set U , with independent random values V_e for each $e \in U$. The outer constraints requires that we probe a set $S \subseteq U$ with $|S| \leq B$ that satisfies the precedence (i.e., it is a lower-ideal of the poset giving the precedence constraints). The inner constraint is a single uniform matroid of rank k . By Theorem 1.2 the adaptivity gap of this probing problem is $O(1)$, and hence it suffices to give an algorithm for the non-adaptive problem.

The non-adaptive problem involves choosing a precedence-constrained set $S \subseteq U$ with $|S| \leq B$ that maximizes:

$$\text{value}(S) := \mathbb{E} \left[\max \left\{ \sum_{e \in I} V_e \mid I \subseteq S, |I| \leq k \right\} \right].$$

By Lemma 5.1 this value is approximated (within factor 4) by $\text{value}'(S)$ which is:

$$\begin{aligned} & \sum_{j \in \mathbb{Z}} 2^j \cdot \mathbb{E} [\max\{|I| : I \subseteq S \cap \{e \mid 2^{j-1} \leq V_e < 2^j\}, |I| \leq k\}] \\ &= \sum_{j \in \mathbb{Z}} 2^j \cdot \mathbb{E} [\min\{S \cap \{e \mid 2^{j-1} \leq V_e < 2^j\}, k\}]. \end{aligned}$$

This in turn is approximated within factor $\frac{e}{e-1}$ by

$$\text{value}''(S) = \sum_{j \in \mathbb{Z}} 2^j \cdot \min \left\{ \sum_{e \in S} \Pr[2^{j-1} \leq V_e < 2^j], k \right\}.$$

This follows from the fact that for independent $[0, 1]$ valued random variables X_1, \dots, X_n with $Y = \sum X_i$, it

holds that $\mathbb{E}[\min\{Y, 1\}] \geq (1 - 1/e) \cdot \min\{\mathbb{E}[Y], 1\}$ (see, e.g., [6, Theorem 4]). For each j , we use this result with $X_e = \frac{1}{k} \cdot \mathbf{1}(2^{j-1} \leq V_e < 2^j)$ and $Y = \sum_{e \in S} X_e$.

Define $p_{e,j} := \Pr[2^{j-1} \leq V_e < 2^j]$ for each $e \in U$ and $j \in \mathbb{Z}$. Using standard scaling arguments, we may assume that the random variable takes on values in the range $[1, \text{poly}(n)]$, that so we have $L = O(\log n)$ weight classes $j = 1, \dots, L$. Summarizing the above reductions, we know that with a constant factor loss, the non-adaptive problem reduces to finding a precedence-constrained set $S \subseteq U$ with $|S| \leq B$ that maximizes $V'(S) = \sum_{j=1}^L 2^j \cdot \min\{\sum_{e \in S} p_{e,j}, r\}$.

When $L = 1$, this reduces to the *partially ordered knapsack* problem for which an FPTAS is known when the precedence constraint is an “2-dimensional order” [33].⁶ The algorithm in [33] is a dynamic program which easily extends to give a quasi-polynomial $n^{O(L)}$ time approximation scheme for non-adaptive stochastic precedence constrained scheduling in this setting. Hence we get:

THEOREM 6.3. *There is a quasi-polynomial time constant-factor approximation algorithm for stochastic precedence constrained scheduling under 2-dimensional orders.*

It remains an interesting direction of research to extend this result to more general classes of matroid inner constraints, and also more general precedence constraints.

7 Extensions

7.1 Improved Bound for Uniform Matroids In this section we improve our bounds from Theorem 3.1 for the special case when the inner constraints are given by a single uniform matroid. This proof is similar structure to of Theorem 3.1, however, each step is simpler.

THEOREM 7.1. *Consider an unweighted stochastic probing instance where the inner constraint is a uniform matroid of rank R , and the outer constraint is prefix closed. Let \mathcal{T}^* be an optimal adaptive strategy and P be a random path according to $\pi_{\mathcal{T}^*}$. Then $\mathbb{E}_P[\text{pathval}(P)] \geq \frac{e-1}{2e-1} \cdot (1 - \frac{1}{e})^2 \cdot \text{adapt}(\mathcal{T}^*)$.*

Proof. Since the greedy algorithm is optimal for a single matroid, we may assume w.l.o.g. that \mathcal{T}^* selects every active element into its solution, and the tree is terminated after R elements are chosen. This (online) strategy also has expected value $\text{adapt}(\mathcal{T}^*)$.

⁶This class is strictly more general than series-parallel precedence constraints. For general precedence constraints, no approximation algorithm is known: this is at least as hard as Dense- k -subgraph.

Recall that the uniform matroid polytope $\mathcal{P} := \{x \in [0, 1]^U \mid \sum_{e \in U} x_e \leq R\}$.

Define the set \mathcal{F} of *frontier* nodes in \mathcal{T}^* to be those nodes u closest to the root that have $\sum_{e \in \text{elt}(P_u)} p_e \geq \frac{e}{e-1}R$. (Recall that P_u is the path in \mathcal{T}^* from the root to node u .) We truncate \mathcal{T}^* below all nodes of \mathcal{F} to obtain tree $\bar{\mathcal{T}}$. We have the following two useful properties:

1. The conditional expected value in \mathcal{T}^* below any node u is at most the adaptive optimum. This is because a feasible strategy is to deterministically follow the path in \mathcal{T}^* to u (without making any selections) and then implement \mathcal{T}^* 's strategy below u .
2. The probability that a random path P in \mathcal{T}^* crosses \mathcal{F} is at most $\frac{1}{e}$. This follows from Theorem 4.1, by setting N_t to be the t^{th} node seen by P in \mathcal{T}^* and

$$\xi_t(N_1, \dots, N_t) = \mathbf{1}(\text{elt}(N_t) \text{ is active}).$$

The stopping time τ corresponds to path P crossing a frontier node \mathcal{F} strictly before reaching a leaf in \mathcal{T}^* . By the preprocessing of \mathcal{T}^* it follows that:

$$\sum_{t=1}^{\tau} \xi_t(N_1, \dots, N_t) = \sum_{t=1}^{\tau} \mathbf{1}(\text{elt}(N_t) \text{ is active}) \leq R-1.$$

On the other hand, by definition of \mathcal{F} , we have:

$$\sum_{t=1}^{\tau} \mathbb{E}_{N_t}[\xi_t(N_1, \dots, N_t)] = \sum_{t=1}^{\tau} p_{\text{elt}(N_t)} > \frac{e}{e-1} \cdot R.$$

Applying Theorem 4.1 with $\delta = 1$, we have $\Pr[\text{cross } \mathcal{F}] \leq \frac{1}{e}$.

Combining the above two properties, the expected value of $\bar{\mathcal{T}}$ is at least $(1 - \frac{1}{e}) \cdot \text{adapt}(\mathcal{T}^*)$.

Consider now a random path P in $\bar{\mathcal{T}}$ and the probability vector:

$$x_e = \begin{cases} p_e & \text{if } e \in \text{elt}(P) \\ 0 & \text{otherwise} \end{cases},$$

By construction of $\bar{\mathcal{T}}$, with probability one, we have $\sum_{e \in U} x_e \leq 1 + \frac{eR}{e-1}$, since any frontier node contributes at most one and the nodes before it contribute at most $\frac{e}{e-1}R$. So $\frac{e-1}{2e-1}\mathbf{x} \in \mathcal{P}$. Moreover, by the truncation above the expected total probability $\mathbb{E}[\sum_{e \in U} x_e]$ equals the expected value of $\bar{\mathcal{T}}$ which is at least $(1 - \frac{1}{e}) \cdot \text{adapt}(\mathcal{T}^*)$.

For any vector $\mathbf{y} \in \mathcal{P}$, contention resolution schemes [4, 37, 17] imply that if we randomly and independently select each element e with probability y_e then the resulting set has expected rank at least $(1 - \frac{1}{e}) \cdot \sum_e y_e$. Using this result with $\mathbf{y} = \frac{e-1}{2e-1} \cdot \mathbf{x}$, the expected value from any fixed P is at least $\frac{e-1}{2e-1} \cdot (1 - \frac{1}{e})$ times $\sum_e x_e$, i.e. at least $\frac{e-1}{2e-1} (1 - \frac{1}{e})^2 \cdot \text{adapt}(\mathcal{T}^*)$.

7.2 Improved Bound for Intersections of Partition Matroids

Recall that a partition matroid is given by a partition $\{V_j\}$ of the elements U , and a subset $I \subseteq U$ is independent if $|I \cap V_j| \leq 1$ for all parts j . Here we show that the adaptivity gap when the inner constraints are intersections of k *partition matroids*, is bounded by $O(k^3 \log k)$. This improves upon the result for general matroids: in particular we get a constant-factor adaptivity gap for $k = O(1)$ partition matroids (instead of $O(\log n)$ for arbitrary matroids).

Specifically, we prove the following result for the unweighted case, which combined with the reduction in Section 5 from weighted to unweighted instances, implies Theorem 1.2.

THEOREM 7.2. *Consider an unweighted stochastic probing instance with inner constraint being the intersection of k partition matroids, and any prefix-closed outer constraint. For any strategy \mathcal{T} ,*

$$\mathbb{E}_{P \leftarrow \pi_{\mathcal{T}}}[\text{pathval}(P)] \geq \frac{1}{O(k^2 \cdot \log k)} \cdot \text{adapt}(\mathcal{T}).$$

For this proof, we assume familiarity with the proof of Theorem 3.1. We use the same notation and preprocessing step from Section 3. In particular, recall the definitions of the independent sets I_u , vectors $x^{(u)}$, relevant nodes R and ρ -valid nodes. Also recall that after preprocessing, the value of strategy \mathcal{T} is given by (3.3) and due to the greedy algorithm being a k -approximation, $\text{greedy}(\mathcal{T}) \geq \frac{1}{k} \cdot \text{adapt}(\mathcal{T})$. The key difference is in the truncation operation, which we now describe.

With $\rho = 1 + \ln k$, we show how to truncate the tree (i.e. reduce the set R of relevant nodes) so that every node is (roughly) ρ -valid for *one* partition matroid \mathcal{M}_i , and the value of the resulting tree (following the definition in (3.3)) is at least $1 - \frac{1}{ek}$ times the original. By repeating this for each of the k partition matroids, we would obtain a tree $\tilde{\mathcal{T}}$ where each node is ρ -valid for *every* matroid $\{\mathcal{M}_i\}_{i=1}^k$ and the value is at least $\Omega(1)$ times that of \mathcal{T} .

Truncation for single partition matroid \mathcal{M} .

Let \mathcal{T}' denote an arbitrary decision tree with relevant nodes R , and \mathcal{M} a partition matroid with parts $\{V_j\}$. A node u in \mathcal{T}' is called ρ -valid for part j of matroid \mathcal{M} if $\sum_{e \in V_j} x_e^{(u)} \leq \rho$; it is called ρ -invalid otherwise. Note that for any part j , the set of ρ -valid nodes forms a connected subtree from the root of \mathcal{T}' . Define the *frontier nodes of part j* to be the set F_j of ρ -invalid nodes (for j) that are closest to the root. We are now ready to define the truncated tree \mathcal{T}'' .

For each part j and node $u \in F_j$ do : *remove* from R all nodes in the subtree of \mathcal{T}' *strictly*

below u that contain an element of V_j (i.e., ignore the value obtained from such nodes).

Note that any node $v \in \mathcal{T}'$ may get removed from R only while processing the unique part V_j that contains $\text{elt}(v)$. The value of the new tree \mathcal{T}'' is calculated using (3.3) on the modified set R . The vectors $x^{(u)}$ on nodes of \mathcal{T}'' are also updated according to (3.4) with the new set R .

LEMMA 7.1. *The value of the truncated tree $\text{greedy}(\mathcal{T}'') \geq (1 - \frac{1}{ek}) \cdot \text{greedy}(\mathcal{T}')$.*

Proof. We consider the nodes removed while processing each part j separately. First, let us *condition* on the instantiation σ of all elements in $U \setminus V_j$ and let \mathcal{T}'_σ denote the conditional decision tree. Note that \mathcal{T}'_σ differs from \mathcal{T}' only in the transition probabilities out of nodes containing elements of $U \setminus V_j$: these transitions are deterministic in \mathcal{T}'_σ . In \mathcal{T}'_σ , consider the event of a random path crossing a frontier node for V_j *before* observing any active element from V_j . Note that there is at most one such path in \mathcal{T}'_σ , which is obtained by following the “inactive” outcome at each node containing a V_j -element. Let u denote the unique frontier node on this path, i.e., the first ρ -invalid node (if any). We assume that such a node u exists, otherwise the following analysis is trivial. Recall that P_u denotes the path in \mathcal{T}' (and also \mathcal{T}'_σ) from the root to u . So

$$\sum_{e \in V_j \cap \text{nelt}(R \cap P_u)} p_e = \sum_{e \in V_j} x_e^{(u)} > \rho = 1 + \ln k.$$

We now have (using “inst.” to mean “instantiation”):

$$\begin{aligned} & \Pr_{\mathcal{T}'}[\text{cross frontier node before an active inst. for } V_j \mid \sigma] \\ &= \Pr_{\mathcal{T}'_\sigma}[\text{cross } u \text{ before active inst. for } V_j \mid \sigma] \\ &= \prod_{e \in V_j \cap \text{nelt}(R \cap P_u)} (1 - p_e) \\ &\leq \exp\left(-\sum_{e \in V_j \cap \text{nelt}(R \cap P_u)} p_e\right) < e^{-\rho}. \end{aligned}$$

All nodes on path P_u appear in both \mathcal{T}' and \mathcal{T}'' . So, conditioned on σ , the random path traced in \mathcal{T}' agrees with that in \mathcal{T}'' until u . Moreover, if the paths traced by \mathcal{T}' and \mathcal{T}'' diverge from P_u before the truncation below node u then they must both contain an active instantiation for V_j . Hence,

$$\begin{aligned} & \Pr[\text{an active inst. from } V_j \text{ in } \mathcal{T}'' \mid \sigma] \\ &= \Pr_{\mathcal{T}'_\sigma}[\text{do not cross } u \text{ before active inst. for } V_j \mid \sigma] \\ &> 1 - e^{-\rho}, \end{aligned}$$

which in turn is clearly at least $(1 - e^{-\rho}) \cdot \Pr[\text{an active instantiation from } V_j \text{ in } \mathcal{T}' \mid \sigma]$.

Taking expectations over σ and removing its conditioning,

$$\text{greedy}_j(\mathcal{T}'') \geq (1 - e^{-\rho}) \cdot \text{greedy}_j(\mathcal{T}')$$

where we use greedy_j to denote the value in (3.3) *only from* nodes containing V_j elements. Since $\{V_j\}$ partitions U , we have $\text{greedy} = \sum_j \text{greedy}_j$.⁷ The lemma now follows by adding the above inequality over all parts j .

By iterating the truncation for each of the k partition matroids on \mathcal{T} , we obtain a decision tree $\tilde{\mathcal{T}}$ with $\text{greedy}(\tilde{\mathcal{T}}) \geq (1 - \frac{1}{ek})^k \cdot \text{greedy}(\mathcal{T}) \geq (1 - 1/e) \cdot \text{greedy}(\mathcal{T})$. The tree $\tilde{\mathcal{T}}$ has the property that every non-leaf node is ρ -valid for each matroid. We emphasize that the frontier nodes are ρ -invalid but included in $\tilde{\mathcal{T}}$.

If we ignore the leaf-nodes of $\tilde{\mathcal{T}}$ then exactly as in the proof for Theorem 3.1, we can use contention resolution to obtain that a random path P in \mathcal{T} has $\mathbb{E}[\text{pathval}(P)]$ at least $\frac{1}{\rho^{(k+1)}}$ times the total expected value in non-leaf nodes of $\tilde{\mathcal{T}}$. On the other hand, a random path P in \mathcal{T} clearly has $\mathbb{E}[\text{pathval}(P)]$ at least the total expected value in leaf nodes of $\tilde{\mathcal{T}}$. Balancing these two quantities, using the fact that $\text{greedy}(\tilde{\mathcal{T}}) \geq \frac{1}{k}(1 - 1/e) \cdot \text{adapt}(\mathcal{T})$, we obtain

$$\begin{aligned} & \mathbb{E}_{P \leftarrow \pi_{\mathcal{T}}}[\text{pathval}(P)] \\ &\geq \frac{1 - 1/e}{k \cdot (1 + (k + 1)(1 + \ln k))} \text{adapt}(\mathcal{T}). \end{aligned}$$

Remark: In the special case of $k = 1$ partition matroid, we can use a better $(1 - 1/e)$ -factor contention resolution scheme (instead of $1/2$) to obtain a slightly better bound of $\frac{1}{1 + (1 - 1/e)^{-2}}$ above.

7.3 Sums of Matroid Rank Functions, and Coverage Functions The adaptivity gaps in Sections 3 and 7.2 show that a random path P in any adaptive strategy \mathcal{T} (chosen from the distribution $\pi_{\mathcal{T}}$) has expected $\text{pathval}(P)$ which is a large fraction of $\text{adapt}(\mathcal{T})$. While this implies that there exists a good P , the fact that our claim is about the expected value of a random path P gives us more.

For example, consider the unweighted case with a single matroid \mathcal{M} with rank function $r(\cdot)$ as the inner

⁷The claim that $\text{greedy} = \sum_j \text{greedy}_j$ is the key point where this proof fails for a general matroid \mathcal{M} —if we were to perform a truncation for each (critical) rank constraint S then we can no longer lower bound $\text{greedy}(\mathcal{T}'')$ by $\sum_S \mathbb{E}[\#\text{ active instantiations from } S \text{ in } \mathcal{T}'']$.

constraint. Let A_ω be the active elements in outcome ω (i.e., those with non-zero—and hence unit—value), then by Theorem 3.1 we get that

$$\begin{aligned} \mathbb{E}_P [\mathbb{E}_\omega [r(\text{elt}(P) \cap A_\omega)]] &= \Omega\left(\frac{1}{\log n}\right) \cdot \text{adapt}(\mathcal{T}) \\ &\stackrel{\text{(by def)}}{=} \Omega\left(\frac{1}{\log n}\right) \cdot \mathbb{E}_\omega [r(\text{elt}(P_{\ell(\omega)}) \cap A_\omega)]. \end{aligned}$$

Hence, if we are given several matroids \mathcal{M}_i with rank functions r_i , and $\hat{r} = \sum w_i r_i$ with non-negative weights w_i , we get the same bound above for \hat{r} by linearity of expectations. For partition matroids (and hence for rank 1 uniform matroids), we can use Theorem 7.2 with $k = 1$ (see the remark after proof of Theorem 7.2) to show the improved bound:

$$\begin{aligned} \mathbb{E}_P [\mathbb{E}_\omega [r(\text{elt}(P) \cap A_\omega)]] &= \frac{1}{1 + (1 - 1/e)^{-2}} \cdot \text{adapt}(\mathcal{T}) \\ &= \frac{1}{1 + (1 - 1/e)^{-2}} \cdot \mathbb{E}_\omega [r(\text{elt}(P_{\ell(\omega)}) \cap A_\omega)]. \end{aligned}$$

Recall that a weighted coverage function $f : U \rightarrow \mathbb{R}_+$ is defined by a set system (U, \mathcal{S}) where each set $S_i \in \mathcal{S}$ has weight w_i , and where $f(H) := \sum_{S_i \in \mathcal{S}} w_i \cdot \mathbf{1}(S_i \cap H \neq \emptyset)$ for $H \subseteq U$. This function f is a sum of a collection of uniform matroids of rank 1 defined over (subsets of) U . The argument above implies that given any decision tree \mathcal{T} , we have

$$\begin{aligned} \mathbb{E}_P [\mathbb{E}_\omega [f(\text{elt}(P) \cap A_\omega)]] \\ = \frac{1}{1 + (1 - 1/e)^{-2}} \cdot \mathbb{E}_\omega [f(\text{elt}(P_{\ell(\omega)}) \cap A_\omega)]. \end{aligned}$$

This proves Corollary 1.1.

Acknowledgements We thank R. Ravi and Ravishankar Krishnaswamy for useful discussions.

References

- [1] M. Adamczyk. Improved analysis of the greedy algorithm for stochastic matching. *Inf. Process. Lett.*, 111(15):731–737, 2011.
- [2] M. Adamczyk, F. Grandoni, and J. Mukherjee. Improved approximation algorithms for stochastic matching. *CoRR*, abs/1505.01439, 2015.
- [3] M. Adamczyk, M. Sviridenko, and J. Ward. Submodular stochastic probing on matroids. In *STACS*, pages 29–40, 2014.
- [4] S. Agrawal, Y. Ding, A. Saberi, and Y. Ye. Correlation robust stochastic optimization. In *SODA*, pages 1087–1096, 2010.
- [5] I. Ashlagi and A. E. Roth. New challenges in multi-hospital kidney exchange. *American Economic Review*, 102(3):354–59, 2012.

- [6] Y. Azar, A. Madry, T. Moscibroda, D. Panigrahi, and A. Srinivasan. Maximum bipartite flow in networks with adaptive channel width. *Theor. Comput. Sci.*, 412(24):2577–2587, 2011.
- [7] 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.
- [8] N. Bansal and V. Nagarajan. On the adaptivity gap of stochastic orienteering. In *IPCO*, pages 114–125, 2014.
- [9] A. Baveja, A. Chavan, A. Nikiforov, A. Srinivasan, and P. Xu. Improved bounds in stochastic matching and optimization. In *APPROX (to appear)*.
- [10] A. Bhalgat, A. Goel, and S. Khanna. Improved approximation results for stochastic knapsack problems. In *SODA*, pages 1647–1665, 2011.
- [11] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward TSP. *SIAM J. Comput.*, 37(2):653–670 (electronic), 2007.
- [12] G. Călinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.
- [13] D. Chakrabarty and C. Swamy. Facility location with client latencies: Linear programming based techniques for minimum latency problems. In *15th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 92–103, 2011.
- [14] S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan. Multi-parameter mechanism design and sequential posted pricing. In *STOC*, pages 311–320, 2010.
- [15] C. Chekuri, N. Korula, and M. Pál. Improved algorithms for orienteering and related problems. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 661–670, New York, 2008. ACM.
- [16] C. Chekuri and M. Pál. A recursive greedy algorithm for walks in directed graphs. In *FOCS*, pages 245–253, 2005.
- [17] C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM J. Comput.*, 43(6):1831–1879, 2014.
- [18] N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra. Approximating Matches Made in Heaven. In *ICALP (1)*, pages 266–278, 2009.
- [19] B. C. Dean, M. X. Goemans, and J. Vondrák. Adaptivity and approximation for stochastic packing problems. In *SODA*, pages 395–404, 2005.
- [20] 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.
- [21] A. Goel, S. Guha, and K. Munagala. How to probe for an extreme value. *ACM Transactions on Algorithms*, 7(1):12, 2010.
- [22] S. Guha and K. Munagala. Approximation algorithms for budgeted learning problems. In *STOC*, pages 104–

113. 2007.

- [23] S. Guha and K. Munagala. Multi-armed bandits with metric switching costs. In *ICALP*, pages 496–507, 2009.
- [24] S. Guha and K. Munagala. Adaptive uncertainty resolution in bayesian combinatorial optimization problems. *ACM Transactions on Algorithms*, 8(1):1, 2012.
- [25] A. Gupta, R. Krishnaswamy, M. Molinaro, and R. Ravi. Approximation algorithms for correlated knapsacks and non-martingale bandits. In *FOCS*, pages 827–836, 2011.
- [26] A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi. Approximation algorithms for stochastic orienteering. In *SODA*, 2012.
- [27] A. Gupta and V. Nagarajan. A stochastic probing problem with applications. In *IPCO*, pages 205–216, 2013.
- [28] A. Gupta, V. Nagarajan, and R. Ravi. Approximation algorithms for optimal decision trees and adaptive tsp problems. In *ICALP (1)*, pages 690–701, 2010.
- [29] S. Im, V. Nagarajan, and R. van der Zwaan. Minimum latency submodular cover. In *ICALP*, pages 485–497, 2012.
- [30] T. Jenkyns. The efficiency of the “greedy” algorithm. In *7th South Eastern Conference on Combinatorics, Graph Theory and Computing*, pages 341–350, 1976.
- [31] S. Khuller, M. Purohit, and K. K. Sarpatwar. Analyzing the optimal neighborhood: Algorithms for budgeted and partial connected dominating set problems. In *SODA*, pages 1702–1713, 2014.
- [32] R. Kleinberg and S. M. Weinberg. Matroid prophet inequalities. In *STOC*, pages 123–136, 2012.
- [33] S. G. Kolliopoulos and G. Steiner. Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics*, 155(8):889–897, 2007.
- [34] U. Krengel and L. Sucheston. On semiamarts, amarts, and processes with finite value. *Adv. in Prob. Related Topics*, 4:197–266, 1978.
- [35] W. Ma. Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms: Extended abstract. In *SODA*, pages 1154–1163, 2014.
- [36] A. E. Roth, T. Sönmez, and M. U. Ünver. Pairwise kidney exchange. *J. Econom. Theory*, 125(2):151–188, 2005.
- [37] Q. Yan. Mechanism Design via Correlation Gap. In *SODA*, pages 710–719, 2011.
- [38] T. Zhang. Data dependent concentration bounds for sequential prediction algorithms. In *Conference on Learning Theory (COLT)*, pages 173–187, 2005.

A Miscellaneous Facts

FACT A.1. *Given a probing problem with n elements where the outer constraints are prefix-closed and the inner constraints are subset-closed, the strategy of probing the element*

$$e^* := \arg \max_{e \in U} \mathbb{E}[V_e]$$

is a non-adaptive strategy with value at least opt_{ad}/n , and at most opt_{ad} .

Proof. For any $e \in U$, we can assume that (a) the singleton e satisfies the inner constraints (i.e., set $\{e\} \in \mathcal{F}_{out}$), and (b) there is some sequence $\sigma \ni e$ satisfying the outer constraints (i.e., $e \in \sigma \in \mathcal{F}_{out}$). Indeed, if not, element e can be discarded, and its probe (if any) in the optimal strategy can be replaced by a random decision.

For the first part of the claim,

$$\text{opt}_{ad} \leq \sum_{e \in U} \mathbb{E}[V_e] \leq n \times \arg \max_{e \in U} \mathbb{E}[V_e].$$

For second part, one feasible strategy is to probe the sequence of elements such that e can be probed (ignoring their outcomes), probe e and output the singleton e . This is a valid non-adaptive strategy, and hence the best adaptive strategy must give at least as much value.

B Illustrative examples

In this section we give some examples that illustrate the challenges in bounding the adaptivity gap.

B.1 Adaptivity gap is at least $\frac{7}{6}$ The example below shows a constant adaptivity gap for a simple stochastic probing problem where the inner constraints form a partition matroid and the outer constraints form a uniform matroid.

Example 1: Consider three items $\{e_1, e_2, e_3\}$, each having a Bernoulli value distribution. V_{e_1} and V_{e_2} are identically distributed and have value 0 with probability $\frac{1}{2}$ and value 1 with probability $\frac{1}{2}$. V_{e_3} has value $\frac{1}{4}$ with probability 1. Suppose the outer constraints allow us to probe at most two elements and the inner constraints form a partition matroid with the partitions $(\{e_1, e_2\}, \{e_3\})$. Here the adaptive optimum probes element e_1 and depending on whether/not it has positive value, the strategy probes e_3 or e_2 respectively. This gives $\text{adapt} = \frac{7}{8}$. On the other hand, any non adaptive strategy can obtain at most $\frac{3}{4}$ in expectation. Hence the adaptivity gap is at least $\frac{7}{6}$.

B.2 Moving Beyond Decision Trees Here we discuss a different (seemingly promising, but ultimately futile) approach to proving Theorem 3.1 (and hence Theorem 1.1). This is motivated by the concept of correlation gaps and completely avoids arguing about the exponential-sized decision tree.

Recall that the *correlation gap* [4] of a matroid rank function $r : 2^U \rightarrow \mathbb{Z}_+$ is a statement about the expected rank under two different distributions over subsets. Formally, given any distribution $\{\lambda_i, S_i\}$ over subsets $S_i \subseteq U$ (where $\lambda \geq 0$ and $\sum_i \lambda_i = 1$) with

$p_e = \sum_{i:e \in S_i} \lambda_i$, we have

$$\begin{aligned} \mathbb{E}[\text{rank of set containing each } e \in U \text{ indep. w.p. } p_e] \\ \geq (1 - \frac{1}{e}) \cdot \sum_i \lambda_i \cdot r(S_i). \end{aligned}$$

In words, among all distributions over subsets having the same marginal probabilities, the independent distribution approximately maximizes the expected rank.

Given probabilities $\{p_e\}_{e \in U}$, recall the *multilinear extension* [12] of a matroid rank function is $R : 2^U \rightarrow \mathbb{R}_+$ where for each $S \subseteq U$, $R(S)$ equals:

$$\mathbb{E}[\text{rank of set containing each } e \in S \text{ indep. w.p. } p_e].$$

So the correlation gap says $R(U) \geq (1 - \frac{1}{e}) \cdot \sum_i \lambda_i \cdot r(S_i)$ for any distribution $\{\lambda_i, S_i\}$ with marginals $\{p_e\}$.

Consider now the following approach to proving Theorem 3.1 which would generalize the correlation gap. Let λ_i denote the probability that the optimum adaptive strategy (opt_{ad}) probes set $S_i \subseteq U$, and let $T_i \subseteq S_i$ denote the subset of active elements that are observed under this outcome. So $\text{opt}_{ad} = \sum_i \lambda_i \cdot r(T_i)$. On the other hand, the non-adaptive value (measured as pathval) of a random root-leaf path in opt_{ad} is precisely $\mathbb{E}[\text{pathval}] = \sum_i \lambda_i \cdot R(S_i)$. The expressions for opt_{ad} and $\mathbb{E}[\text{pathval}]$ correspond to the expected rank under two different distributions, having the same marginal probabilities:

$$(B.1) \quad \sum_{i:e \in T_i} \lambda_i = p_e \sum_{i:e \in S_i} \lambda_i, \quad \forall e \in U.$$

Bounding the adaptivity gap now reduces to the task of relating the expected rank under these two distributions; i.e., for some small value ρ ,

$$(B.2) \quad \text{Is } \sum_i \lambda_i \cdot r(T_i) \leq \rho \cdot \sum_i \lambda_i \cdot R(S_i) \quad ?$$

Observe that when all the subsets S_i s are identical, the question in (B.2) is precisely the correlation gap question, in which case the inequality holds with $\rho \leq \frac{e}{e-1}$. Unfortunately, as shown next, the ratio of these two expectations may be $\Omega(\sqrt{n})$ in general.

Example 2: Consider a stochastic probing instance on $U = [n]$. The inner matroid constraint is a uniform matroid of rank one: the rank function has value 1 if at least one element is picked, and is 0 otherwise. Each element is active independently with probability $p = \frac{\sqrt{n}-1}{n+\sqrt{n}-1}$. Let $S_0 := \{1, 2, \dots, n\}$, $T_0 := \emptyset$, and $\lambda_0 = \frac{1}{\sqrt{n}}$. For $1 \leq i \leq n$, let $S_i := \{i\}$, $T_i := \{i\}$, and $\lambda_i = \frac{1}{n} \left(1 - \frac{1}{\sqrt{n}}\right)$. It is easy to verify that $\sum_i \lambda_i r(T_i) = 1 - \frac{1}{\sqrt{n}}$ but $\sum_i \lambda_i R(S_i) = O(\frac{1}{\sqrt{n}})$.