2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)

Pandora's Box with Correlations: Learning and Approximation

Shuchi Chawla*, Evangelia Gergatsouli†, Yifeng Teng‡, Christos Tzamos§,
Ruimin Zhang¶
Department of Computer Sciences, University of Wisconsin - Madison
Madison, WI, USA
*Email: *shuchi@cs.wisc.edu, †evagerg@cs.wisc.edu, ‡yifengt@cs.wisc.edu, §tzamos@wisc.edu,*
¶rzhang274@wisc.edu

***Abstract*—The Pandora's Box problem and its extensions capture optimization problems with stochastic input where the algorithm can obtain instantiations of input random variables at some cost. To our knowledge, all previous work on this class of problems assumes that different random variables in the input are distributed independently. As such it does not capture many real-world settings. In this paper, we provide the first approximation algorithms for Pandora's Box-type problems with correlations. We assume that the algorithm has access to samples drawn from the joint distribution on input.**

Algorithms for these problems must determine an order in which to probe random variables, as well as when to stop and return the best solution found so far. In general, an optimal algorithm may make both decisions adaptively based on instantiations observed previously. Such *fully adaptive* (FA) strategies cannot be efficiently approximated to within any sub-linear factor with sample access. We therefore focus on the simpler objective of approximating *partially adaptive* (PA) strategies that probe random variables in a fixed predetermined order but decide when to stop based on the instantiations observed. We consider a number of different feasibility constraints and provide simple PA strategies that are approximately optimal with respect to the best PA strategy for each case. All of our algorithms have polynomial sample complexity. We further show that our results are tight within constant factors: better factors cannot be achieved even using the full power of FA strategies.

***Keywords*-Pandora's Box; correlation; online algorithms; approximation algorithms.**

I. Introduction

In many optimization settings involving uncertainty in the input, information about the input can be obtained at extra monetary or computational overhead; paying this overhead can allow the optimizer to improve its performance. Determining the optimal manner for acquiring information then becomes an online decision-making problem: each piece of information obtained by the algorithm can affect whether and which piece to acquire next. A classical example is the Pandora's Box problem due to Weitzman [2]. The online algorithm is presented with n boxes, each containing an unknown stochastic reward. The algorithm can open boxes in any order at a fixed overhead each; observes the rewards contained in the open boxes; and terminates upon selecting any one of the rewards observed. The goal is to maximize the reward selected minus the total overhead of opening boxes. Weitzman showed that a particularly simple policy is optimal for the Pandora's Box problem: the algorithm computes an index for each box based on its reward distribution and opens boxes in decreasing order of these indices until it finds a reward that exceeds all of the remaining indices. There is a long literature of generalizations of this problem, in many different settings [3]–[13].

A crucial assumption underlying Weitzman's optimality result is that the rewards in different boxes are independent. This does not always bear out in practice. Suppose, for example, that you want to buy an item online and look for a website that offers a cheap price. Your goal is to minimize the price you pay for the item plus the time it takes to search for a good deal. Since the websites are competing sellers, it is likely that prices on different sites are correlated. For another example, consider a route planning service that wants to determine the fastest route between two destinations from among a set of potential routes. The driving time for each route is stochastic and depends on traffic, but the route planning service can obtain its exact value at some cost. The service wants to minimize the driving time of the route selected plus the cost spent on querying routes. Once again, because of network effects, driving times along different routes may be correlated. How do we design an online search algorithm for these settings?

In this paper, we provide the first competitive algorithms for Pandora's Box-type problems with correlations. We begin our investigation with the simplest minimization variant of the problem, formalizing the examples described above: there are n alternatives with unknown costs that are drawn from some joint distribution. A search algorithm examines these alternatives one at a time, learning their costs, and after a few steps stops and selects one of the alternatives. Given sample access to the distribution of costs, our goal is to develop a search algorithm that minimizes the sum of the expected cost of the chosen alternative and the number of steps to find it. We call this the online stochastic search problem. Henceforth we will refer to the alternatives as boxes and different instantiations of costs in boxes as

Full version of this paper is available at arxiv.org [1]. This work was supported in part by NSF grants CCF-1617505 and SHF-1704117.

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00116

scenarios.

The optimal solution for online stochastic search is a fully-adaptive (FA) strategy that chooses which box to query each time based on all the costs that have been observed so far. While these are the best strategies one could hope for, they are impossible to find or approximate with samples. For example, it could be the case that the cost in the first few boxes encode the location of a box of cost 0 while every other box has infinite cost. While the best option can be identified with just few queries, any reasonable approximation to the optimal cost would need to accurately learn this mapping. Learning such an arbitrary mapping however is impossible through samples, unless there is significant probability of seeing the exact same combination of costs[1].

Competing against partially adaptive strategies.: Is there any hope for finding a good strategy for correlated costs? We show that positive results can be obtained if we target a simpler benchmark. We consider partially-adaptive (PA) strategies that have a fixed order in which they query the boxes but may have arbitrarily complex rules on when to stop.

Our main positive result is a constant-approximation to the optimal PA strategy with polynomial time and sample complexity in the number of boxes.

Our result directly generalizes the positive results for the special case of Pandora's box studied in prior work where costs are drawn independently. This is because optimal strategies for these settings are known to be partially adaptive intuitively because information about costs of opened boxes does not imply anything about future boxes.

In targeting the benchmark of PA strategies, we also give limited power to our algorithms. All of our approximations are achieved via simple PA strategies that can be described succinctly. This enables us to learn these strategies efficiently from data. While one might hope to achieve better results using the full power of FA strategies, perhaps even surpassing the performance of PA strategies entirely, we show that a constant factor loss is necessary for computational reasons.

Our inspiration for using the optimal PA strategy as a benchmark comes from other contexts where the optimal solution is impossible to approximate well. One example is prior-free mechanism design where for some objectives such as revenue, in the absence of stochastic information about input values, no finite approximation to the optimum can be achieved [14]. Hartline and Roughgarden [15] proposed a template whereby one characterizes the class of solutions that are optimal under the assumption that values are drawn i.i.d. from some unknown distribution. The goal then is to compete against the best mechanism from this class. Another example is the concept of static optimality in dynamic data structures. Consider, for example, the problem of maintaining a binary search tree with the goal of minimizing search cost over an online sequence of requests. When the requests are drawn from a fixed distribution, a static search tree is optimal. In the worst case, however, the optimal in hindsight algorithm maintains a dynamic search tree, performing rotations between consecutive requests. Achieving constant-factor competitiveness against the optimal in hindsight solution, known as dynamic optimality, is a major open problem [16], [17]. Early work therefore focused on static optimality, or achieving competitiveness (via a dynamic data structure) against the optimal static tree. In each of these cases, an appropriate benchmark is defined by first considering a special case of the problem (e.g. i.i.d. input); characterizing optimal solutions for that special case; and then competing in the general setting against the best out of all such solutions. Applying this approach to the online stochastic search problem, we obtain a benchmark by considering the special case we know how to solve: namely when the costs in boxes are independently distributed. Weitzman's work and its generalizations show that in this case the optimal strategy is always a PA strategy.

[1]For explicitly given distributions, this is not an issue. However, this is beyond the scope of the paper as we aim to provide good strategies that generalize to rich distributions rather than overfitting to and memorizing the costs in the few scenarios given.

A. Results and Techniques

We now describe our results and techniques in more detail.

Learning a good strategy from data.: To give some intuition about why PA strategies are learnable from data, consider the special case where the costs are either 0 or ∞. Any PA strategy then probes boxes in a particular order until it finds one with 0 cost, and then terminates. In other words, there is only one relevant stopping rule and the space of relevant PA strategies is "small" ($n!$; corresponding to each possible probing order). This coupled with the boundedness of the objective implies that $\mathrm{poly}(n)$ samples are enough to find the optimal PA strategy.

The case of general costs is trickier as it is unclear when a low cost option has been identified. In particular, the class of all PA strategies can be quite large and complex because the stopping rule can depend in a complex manner on the costs observed in the boxes. One of our main technical contributions is to show that once we have determined an order in which to query boxes, it becomes easy to find an *approximately optimal* stopping rule at the loss of a small constant factor (Lemma III.1). This technical lemma is based on an extension of the ski-rental online algorithm [18] and is presented in Section III. This allows us to focus on finding good *scenario-aware* strategies – that is, an ordering of the boxes that performs well assuming that we know when to stop. The implication is that the space of "interesting" PA strategies is small, characterized by the $n!$ different orderings over boxes, and therefore approximately optimal PA strategies can be identified from polynomially many samples.

Finding approximately optimal PA strategies.: As a warm-up, we first develop PA strategies that are competitive against completely non-adaptive (NA) strategies. NA strategies simply select a fixed set of boxes to probe and pick the box with the cheapest cost among these. Despite their simplicity, optimizing over NA strategies using NA strategies is intractable: it captures the hitting set formulation of set cover and is therefore hard to approximate better than a logarithmic factor in the number of scenarios. It is also intractable from the viewpoint of learning: if there is a tiny probability scenario that has infinite cost on all boxes but one, the expected cost of the algorithm would be infinite if the algorithm does not sample that scenario or query all boxes.

Our first result shows that it is possible to efficiently compute a scenario-aware PA strategy that beats any NA strategy entirely (Corollary III.2). Combining this with our approximately-optimal stopping rule gives a PA strategy that achieves a constant factor approximation (1.58) to the optimal NA strategy. While a better constant factor approximation might be possible through a more direct argument, we show that it is NP-hard to approximate the optimal NA strategy beyond some constant (1.278) even if one is allowed to use FA strategies. Our lower-bound is based on the logarithmic lower-bound for set-cover [19] which restricts how many scenarios can be covered within the first few time steps (Lemma IV.4).

Our main result extends the above constant factor approximation guarantees even against PA strategies. We again restrict our attention to scenario-aware strategies and seek to find an ordering that approximates the optimal PA strategy. We solve the resulting problem by formulating a linear programming relaxation to identify for each scenario a set of "good" boxes with suitably low values. This allows us to reduce the problem at a cost of a constant factor to finding an ordering of boxes so that the expected time until a scenario visits one of its "good" boxes is minimized. This problem is known as the min-sum set cover problem and is known to be approximable within a factor of 4 [20]. The resulting approximation factor we obtain is 9.22.

Further extensions.: Beyond the problem of identifying a single option with low cost, we also consider several extensions. One extension is the case where k options must be identified so that the sum of their costs is minimized. A further generalization is the case where the set of options must form a base of rank k in a given underlying matroid. This allows expressing many combinatorial problems in this framework such as the minimum spanning tree problem. For the first extension where any k options are feasible (corresponding to a uniform matroid) we obtain a constant factor approximation. For general matroids however, the approximation factor decays to $O(\log k)$. We show that this is necessary even for the much weaker objective of approximating NA strategies with arbitrary FA strategies, and even for very simple matroids such as the partition matroid. We obtain the upper-bounds by modifying the techniques developed for extensions of min-sum set cover – the generalized min-sum set cover and the submodular ranking problem. The following table shows a summary of the results obtained.

While all of the settings above assume that every box takes the same amount of time to probe (one step), we show in Section VII that our results extend easily to settings where different boxes have different probing times. We assume that probing times lie in the range $[1, P]$. Both the running time and sample complexity of our algorithms depend linearly on P and are efficient when P is polynomially large. This dependence on P for the sample complexity is necessary to observe scenarios that happen with probability $O(1/P)$ but contribute a significant amount to the objective.

Finally in Section VIII we consider a modification of the framework to maximization instead of minimization problems where the goal is to maximize the value of the chosen alternative minus the time it takes to find it (as in the Pandora's Box problem). In contrast to the minimization version, we show that in this setting even the simplest possible benchmark – the optimal NA strategy – cannot be efficiently approximated within any constant factor using the full power of FA algorithms.

B. Related Work

Our framework is inspired by the Pandora's box model which has its roots in the Economics literature. Since Weitzman's seminal work on this problem, there has been a large line of research studying the price of information [3]–[6] and the structure of approximately optimal rules for several combinatorial problems [7]–[13], [21]. Another variant of Pandora's Box problem is studied by Beyhaghi and Kleinberg [22] where they also define a *commiting* policy, which resembles our Partially-adaptive one.

Our work also advances a recent line of research on the foundations of data-driven algorithm design. The seminal work of Gupta and Roughgarden [23] introduced the problem of algorithm selection in a distributional learning setting focusing on the number of samples required to learn an approximately optimal algorithm. A long line of recent research extends this framework to efficient sample-based optimization over parameterized classes of algorithms [23]–[31] In contrast to these results our work studies optimization over larger, non-parametric classes of algorithms, indeed any polynomial time (partially-adaptive) algorithm. Beyond this line of research, there has also been a lot of work in the context of improving algorithms using data that combines machine learning predictions to improve traditional worst case guarantees of online algorithms [32]–[35].

Finally our work can also be seen as a generalization of the min-sum set cover problem (MSSC). Indeed MSSC corresponds to the special case where costs are either 0

	Single Option	k Options	Matroid of rank k
PA vs PA (Upper-bound)	9.22 **[Theorem V.2]**	$O(1)$ **[Theorem VI.2]**	$O(\log k)$ **[Theorem VI.4]**
FA vs NA (Lower-bound)	1.27 **[Theorem IV.3]**	1.27 **[Theorem IV.3]**	$\Omega(\log k)$ **[Theorem VI.5]**

Table I

THE MAIN RESULTS SHOWN IN THIS WORK. THE ENTRIES CORRESPOND TO THE ACHIEVED COMPETITIVE RATIO FOR DIFFERENT SETTINGS. THE UPPER-BOUNDS ARE SHOWN FOR ANY PA ALGORITHM COMPARED WITH THE OPTIMAL PA. THE LOWER-BOUNDS ARE SHOWN IN THE MUCH WEAKER SETTING OF FA VS NA.

or ∞. Some of our LP-rounding techniques are similar to those developed for MSSC [20] and its generalizations [36]–[40]. Our algorithms for the setting of general probing times generalize results for the MSSC to settings with arbitrary "lengths" for elements.

II. MODEL

In the optimal search problem, we are given a set $\mathcal{B}$ of n boxes with unknown costs and a distribution $\mathcal{D}$ over a set of possible scenarios that determine these costs. Nature chooses a scenario s from the distribution, which then instantiates the cost of each box. We use c_{is} to denote the cost of box i when scenario s is instantiated.

The goal of the online algorithm is to choose a box of small cost while spending as little time as possible gathering information. The algorithm cannot directly observe the scenario that is instantiated, however, is allowed to "probe" boxes one at a time. Upon probing a box, the algorithm gets to observe the cost of the box. Formally let $\mathcal{P}_s$ be the random variable denoting the set of probed boxes when scenario s is instantiated and let $i_s \in \mathcal{P}_s$ be the (random) index of the box chosen by the algorithm. We require $i_s \in \mathcal{P}_s$, that is, the algorithm must probe a box to choose it. Note that the randomness in the choice of $\mathcal{P}_s$ and i_s arises both from the random instantiation of scenarios as well as from any coins the algorithm itself may flip. Our goal then is to minimize the total probing time plus the cost of the chosen box:

$$\mathbf{E}_s\left[\min_{i\in\mathcal{P}_s} c_{is} + |\mathcal{P}_s|\right].$$

Any online algorithm can be described by the pair (σ, τ), where σ is a permutation of the boxes representing the order in which they get probed, and τ is a stopping rule – the time at which the algorithm stops probing and returns the minimum cost it has seen so far. Observe that in its full generality, an algorithm may choose the i'th box to probe, $\sigma(i)$, as a function of the identities and costs of the first $i-1$ boxes, $\{\sigma(1), \cdots, \sigma(i-1)\}$ and $\{c_{\sigma(1)s}, \cdots, c_{\sigma(i-1)s}\}$[2]. Likewise, the decision of setting $\tau = i$ for $i \in [n]$ may depend on $\{\sigma(1), \cdots, \sigma(i)\}$ and $\{c_{\sigma(1)s}, \cdots, c_{\sigma(i)s}\}$. Optimizing over the class of all such algorithms is intractable. So we will consider simpler classes of strategies, as formalized in the following definition.

[2]For some realized scenario $s \in \mathcal{S}$.

Definition II.1 (Adaptivity of Strategies)**.** In a *Fully-Adaptive (FA)* strategy, both σ and τ can depend on any costs seen in a previous time step, as described above.

In a *Partially-Adaptive (PA)* strategy, the sequence σ is independent of the costs observed in probed boxes. The sequence is determined before any boxes are probed. However, the stopping rule τ can depend on the identities and costs of boxes probed previously.

In a *Non-Adaptive (NA)* strategy, both σ and τ are fixed before any costs are revealed to the algorithm. In particular, the algorithm probes a fixed subset of the boxes, $I \subseteq [n]$, and returns the minimum cost $\min_{i\in I} c_{is}$. The algorithm's expected total cost is then $\mathbf{E}_s[\min_{i\in I} c_{is} + |I|]$.

General feasibility constraints.: In Section VI we study extensions of the search problem where our goal is to pick multiple boxes satisfying a given feasibility constraint. Let $\mathcal{F} \subseteq 2^{\mathcal{B}}$ denote the feasibility constraint. Our goal is to probe boxes in some order and select a subset of the probed boxes that is feasible. Once again we can describe an algorithm using the pair (σ, τ) where σ denotes the probing order, and τ denotes the stopping time at which the algorithm stops and returns the cheapest feasible set found so far. The total cost of the algorithm then is the cost of the feasible set returned plus the stopping time. We emphasize that the algorithm faces the same feasibility constraint in every scenario. We consider two different kinds of feasibility constraints. In the first, the algorithm is required to select exactly k boxes for some $k \geq 1$. In the second, the algorithm is required to select a basis of a given matroid.

III. A REDUCTION TO SCENARIO-AWARE STRATEGIES AND ITS IMPLICATIONS TO LEARNING

Recall that designing a PA strategy involves determining a non-adaptive probing order, and a good stopping rule for that probing order. We do not place any bounds on the number of different scenarios, m, or the support size and range of the boxes' costs. These numbers can be exponential or even unbounded. As a result, the optimal stopping rule can be very complicated and it appears to be challenging to characterize the set of all possible PA strategies. We simplify the optimization problem by providing *extra power* to the algorithm and then removing this power at a small loss in approximation factor.

In particular, we define a *Scenario-Aware Partially-Adaptive (SPA)* strategy as one where the probing order σ is independent of the costs observed in probed boxes, however, the stopping time τ is a function of the instantiated scenario s. In other words, the algorithm fixes a probing order, then learns of the scenario instantiated, and then determines a stopping rule for the chosen probing order based on the revealed scenario.

Observe that once a probing order and instantiated scenario are fixed, it is trivial to determine an optimal stopping time in a scenario aware manner. The problem therefore boils down to determining a good probing order. The space of all possible SPA strategies is also likewise much smaller and simpler than the space of all possible PA strategies. We can therefore argue that in order to learn a good SPA strategy, it suffices to optimize over a small sample of scenarios drawn randomly from the underlying distribution. We denote the cost of an SPA strategy with probing order σ by $\mathrm{cost}(\sigma)$.

On the other hand, we argue that scenario-awareness does not buy much power for the algorithm. In particular, given any fixed probing order, we can construct a stopping time that depends only on the observed costs, but that achieves a constant factor approximation to the optimal scenario-aware stopping time for that probing order.

The rest of this section is organized as follows. In Section III-A we exhibit a connection between our problem and a generalized version of the ski rental problem to show that PA strategies are competitive against SPA strategies. In Section III-B we show that optimizing for SPA strategies over a small sample of scenarios suffices to obtain a good approximation. In Section III-C we develop LP relaxations for the optimal NA and SPA strategies. Then in the remainder of the paper we focus on finding approximately-optimal SPA strategies over a small set of scenarios.

A. Ski Rental with varying buy costs

We now define a generalized version of the ski rental problem which is closely related to SPA strategies. The input to the generalized version is a sequence of non-increasing buy costs, $a_1 \geq a_2 \geq a_3 \geq \ldots$. These costs are presented one at a time to the algorithm. At each step t, the algorithm decides to either rent skis at a cost of 1, or buy skis at a cost of a_t. If the algorithm decides to buy, then it incurs no further costs for the remainder of the process. Observe that an offline algorithm that knows the entire cost sequences $a_1, a_2, \ldots$ can pay $\min_{t\geq 1}(t-1+a_t)$. We call this problem *ski rental with time-varying buy costs*. The original ski rental problem is the special case where $a_t = B$ or 0 from the time we stop skiing and on.

We first provide a simple randomized algorithm for ski rental with time-varying costs that achieves a competitive ratio of $e/(e-1)$. Then we extend this to general p_t in Corollary VII.1. Our algorithm uses the randomized algorithm of [18] for ski rental as a building block, essentially by starting a new instance of ski rental every time the cost of the offline optimum changes.

Lemma III.1 (Ski Rental with time-varying buy costs)**.** *Consider any sequence $a_1 \geq a_2 \geq \ldots$. There exists an online algorithm that chooses a stopping time t so that*

$$t - 1 + a_t \leq \frac{e}{e-1} \min_j \{j - 1 + a_j\}.$$

The next corollary connects scenario-aware partially-adaptive strategies with partially-adaptive strategy through our competitive algorithm for ski-rental with time-varying costs. Specifically, given an SPA strategy, we construct an instance of the ski-rental problem, where the buy cost a_t at any step is equal to the cost of the best feasible solution seen so far by the SPA strategy. The rent cost of the ski rental instance reflects the probing time of the search algorithm, whereas the buy cost reflects the cost of the boxes chosen by the algorithm. Our algorithm for ski rental chooses a stopping time as a function of the costs observed in the past and without knowing the (scenario-dependent) costs to be revealed in the future, and therefore gives us a PA strategy for the search problem. This result is formalized below.

Corollary III.2. *Given any* scenario-aware partially-adaptive *strategy σ, we can efficiently construct a stopping time τ, such that the cost of the* partially-adaptive *strategy (σ, τ) is no more than a factor of $e/(e-1)$ times the cost of σ.*

B. Learning a good probing order

Henceforth, we focus on designing good scenario-aware partially adaptive strategies for the search problem. As noted previously, once we fix a probing order, determining the optimal scenario-aware stopping time is easy. We will now show that in order to optimize over all possible probing orders, it suffices to optimize with respect to a small set of scenarios drawn randomly from the underlying distribution.

Formally, let $\mathcal{D}$ denote the distribution over scenarios and let $\mathcal{S}$ be a collection of m scenarios drawn independently from $\mathcal{D}$, with m being a large enough polynomial in n. Then, we claim that with high probability, for *every* probing order σ, $\mathrm{cost}_{\mathcal{D}}(\sigma)$ is close to $\mathrm{cost}_{\mathcal{S}}(\sigma)$, where $\mathrm{cost}_{\mathcal{D}}(\sigma)$ denotes the total expected cost of the SPA strategy σ over the scenario distribution $\mathcal{D}$, and $\mathrm{cost}_{\mathcal{S}}(\sigma)$ denotes its cost over the uniform distribution over the sample $\mathcal{S}$. The implication is that it suffices for us to optimize for SPA strategies over scenario distributions with finite small support.

Lemma III.3. *Let $\epsilon, \delta > 0$ be given parameters. Let $\mathcal{S}$ be a set of m scenarios chosen independently at random from $\mathcal{D}$ with $m = \mathrm{poly}(n, 1/\epsilon, \log(1/\delta))$. Then, with probability at least $1-\delta$, for all permutations $\pi : [n] \to [n]$, we have*

$$\mathrm{cost}_{\mathcal{S}}(\pi) \in (1 \pm \epsilon)\, \mathrm{cost}_{\mathcal{D}}(\pi).$$

Proof: Fix a permutation π. For scenario s, let $\mathrm{cost}_s(\pi) = \min_i\{i + c_{\pi(i)s}\}$ denote the total cost incurred by SPA strategy π in scenario s. Observe that for any π and any s, we have $\mathrm{cost}_s(\pi) \in [1 + \min_i c_{is}, n + \min_i c_{is}]$. Furthermore, $\mathrm{cost}_{\mathcal{D}}(\pi) = \mathbf{E}_{s\sim\mathcal{D}}[\mathrm{cost}_s(\pi)]$, and $\mathrm{cost}_{\mathcal{S}}(\pi) = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} cost_s(\pi)$. The lemma now follows by using the Hoeffding inequality and applying the union bound over all possible permutations π. ■

Combining Corollary III.2 and Lemma III.3 yields the following theorem.

Theorem III.4. *Suppose there exists an algorithm for the optimal search problem that runs in time polynomial in the number of boxes n and the number of scenarios m, and returns an SPA strategy achieving an α-approximation. Then, for any $\epsilon > 0$, there exists an algorithm that runs in time polynomial in n and $1/\epsilon$ and returns a PA strategy with competitive ratio $\frac{e}{e-1}(1+\epsilon)\alpha$, where $n = |\mathcal{B}|$.*

C. LP formulations

We will now construct an LP relaxation for the optimal scenario-aware partially adaptive strategy. Following Theorem III.4 we focus on the setting where the scenario distribution is uniform over a small support set $\mathcal{S}$.

The program (LP-SPA) is given below and is similar to the one used for the generalized min-sum set cover problem in [37] and [38]. Denote by $\mathcal{T}$ to set of time steps. Let x_{it} be an indicator variable for whether box i is opened at time t. Constraints (1) and (2) model matching constraints between boxes and time slots. The variable z_{ist} indicates whether box i is selected for scenario s at time t. Constraints (3) ensure that we only select opened boxes. Constraints (4) ensure that for every scenario we have selected exactly one box. The cost of the box assigned to scenario s is given by $\sum_{i,t} z_{ist}c_{is}$. Furthermore, for any scenario s and time t, the sum $\sum_i z_{ist}$ indicates whether the scenario is covered at time t, and therefore, the probing time for the scenario is given by $\sum_t\sum_i t z_{ist}$.

As a warm-up for our main result, we approximate the optimal NA strategy by a PA strategy. The relaxation (LP-NA) for the optimal NA strategy is simpler. Here x_i is an indicator variable for whether box i is opened and z_{is} indicates whether box i is assigned to scenario s.

IV. Competing with the non-adaptive benchmark

As a warm-up to our main result, in this section we consider competing against the optimal non-adaptive strategy. Recall that a non-adaptive strategy probes a fixed subset of boxes, and then picks a probed box of minimum cost. Is it possible to efficiently find an adaptive strategy that performs just as well? We show two results. On the one hand, in Section IV-A we show that we can efficiently find an SPA strategy that beats the performance of the optimal NA strategy. This along with Theorem III.4 implies that we can efficiently find an $e/(e-1) \approx 1.582$-competitive PA strategy. On the other hand, in Section IV-B we show that it is NP-hard to obtain a competitive ratio better than 1.278 against the optimal NA strategy even using the full power of FA strategies.

A. An upper bound via PA strategies

Our main result of this section is as follows.

Lemma IV.1. *We can efficiently compute a scenario-aware partially-adaptive strategy with competitive ratio 1 against the optimal non-adaptive strategy.*

Putting this together with Theorem III.4 we get the following theorem.

Theorem IV.2. *We can efficiently find a partially-adaptive strategy with total expected cost at most $e/(e-1)$ times the total cost of the optimal non-adaptive strategy.*

Proof of Lemma IV.1: We use the LP relaxation (LP-NA) from Section III-C. Given an optimal fractional solution $(\boldsymbol{x},\boldsymbol{z})$, we denote by $\mathrm{OPT}_{c,s} = \sum_i c_{is}z_{is}$ the cost for scenario s in this solution, and by $\mathrm{OPT}_c = \frac{1}{|\mathcal{S}|}\sum_s \mathrm{OPT}_{c,s}$ the cost for all scenarios. Let $\mathrm{OPT}_t = \sum_{i\in\mathcal{B}} x_i$ denote the probing time for the fractional solution. Similarly, we define ALG_t, ALG_c and $\mathrm{ALG}_{c,s}$ to be the algorithm's query time, cost for all scenarios and cost for scenario s respectively.

Algorithm 1 rounds $(\boldsymbol{x},\boldsymbol{z})$ to an SPA strategy. Note that the probing order σ in the rounded solution is independent of the instantiated scenario, but the stopping time τ_s depends on the scenario specific variables z_{is}. τ_s is not necessarily the optimal stopping time for the constructed probing order, but its definition allows us to relate the cost of our solution to the fractional cost OPT_c.

Algorithm 1: SPA vs NA

Data: Solution $\boldsymbol{x},\boldsymbol{z}$ to program (LP-NA); scenario s
1. $\sigma :=$ For $t \geq 1$, select and open box i with probability $\frac{x_i}{\sum_{i\in\mathcal{B}} x_i}$.
2. $\tau_s :=$ If box i is opened at step t, select the box and stop with probability $\frac{z_{is}}{x_i}$.

Notice that for each step t, the probability of stopping is

$$\begin{aligned}\mathbf{Pr}[\text{stop at step } t] &= \sum_{i\in\mathcal{B}} \frac{x_i}{\sum_{i\in\mathcal{B}} x_i}\frac{z_{is}}{x_i}\\ &= \frac{\sum_{i\in\mathcal{B}} z_{is}}{\sum_{i\in\mathcal{B}} x_i} = \frac{1}{\mathrm{OPT}_t},\end{aligned}$$

where we used the first set of LP constraints (5) and the definition of OPT_t. Observe that the probability is independent of the step t and therefore $\mathbf{E}[\mathrm{ALG}_t] = \mathrm{OPT}_t$. Denote by P_t the random variable indicating whether we stop at step

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{B},s\in\mathcal{S},t\in\mathcal{T}} t z_{ist} + \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{B},s\in\mathcal{S},t\in\mathcal{T}} c_{is} z_{ist} && && \text{(LP-SPA)}\\
\text{subject to} \quad & \sum_{i\in\mathcal{B}} x_{it} = 1, && \forall t\in\mathcal{T} && (1)\\
& \sum_{t\in\mathcal{T}} x_{it} \le 1, && \forall i\in\mathcal{B} && (2)\\
& z_{ist} \le x_{it}, && \forall s\in\mathcal{S}, i\in\mathcal{B}, t\in\mathcal{T} && (3)\\
& \sum_{t'\in\mathcal{T},i\in\mathcal{B}} z_{ist'} = 1, && \forall s\in\mathcal{S} && (4)\\
& x_{it}, z_{ist} \in [0,1] && \forall s\in\mathcal{S}, i\in\mathcal{B}, t\in\mathcal{T} &&
\end{aligned}$$

Figure 1. LP formulation for scenario-aware partially adaptive strategies.

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in\mathcal{B}} x_i + \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{B},s\in\mathcal{S}} c_{is} z_{is} && && \text{(LP-NA)}\\
\text{subject to} \quad & \sum_{i\in\mathcal{B}} z_{is} = 1, && \forall s\in\mathcal{S} && (5)\\
& z_{is} \le x_i, && \forall i\in\mathcal{B}, s\in\mathcal{S} &&\\
& x_i, z_{is} \in [0,1] && \forall i\in\mathcal{B}, s\in\mathcal{S} &&
\end{aligned}$$

Figure 2. LP formulation for the non-adaptive strategies.

t. The expected cost of the algorithm is

$$\begin{aligned}
\mathbf{E}\left[\mathrm{ALG}_{c,s}\right] &= \sum_{i\in\mathcal{B},t} \mathbf{Pr}\left[\text{select } i \text{ at } t \mid P_t\right]\mathbf{Pr}\left[P_t\right] c_{is}\\
&\le \sum_{i\in\mathcal{B},t} \frac{z_{is}}{\sum_{i\in\mathcal{B}} z_{is}} \mathbf{Pr}\left[P_t\right] c_{is}\\
&= \sum_{i\in\mathcal{B}} z_{is} c_{is} = \mathrm{OPT}_{c,s}
\end{aligned}$$

Taking expectation over all scenarios we get $\mathbf{E}\left[\mathrm{ALG}_c\right] \le \mathrm{OPT}_c$, and the lemma follows. ∎

B. A lower bound for FA strategies

We now show that we cannot achieve a competitive ratio of 1 against the optimal NA strategy even if we use the full power of fully adaptive strategies.

Theorem IV.3. *Assuming P≠NP, no computationally efficient fully-adaptive algorithm can approximate the optimal non-adaptive strategy within a factor smaller than* 1.278.

Our lower bound is based on the hardness of approximating Set Cover. We use the following lemma which rules out bicriteria results for Set Cover.

Lemma IV.4. *Unless P=NP, for any constant* $\epsilon > 0$, *there is no algorithm that for every instance of Set Cover finds* k *sets that cover at least a fraction* $1-\left(1-\frac{1+\varepsilon}{OPT}\right)^k$ *of the elements for some integer* $k \in \left[1, \frac{\log n}{1+\varepsilon} OPT\right]$.

Proof of Theorem IV.3: Let $H > 0$ and $p \in [0,1]$ be appropriate constants, to be determined later. We will define a family of instances of the optimal search problem based on set cover. Let $\mathcal{SC} = ([m], \{S_1, \dots, S_n\})$ be a set cover instance with m elements and n sets. Denote its optimal value by OPT_{SC}. To transform this into an instance of the search problem, every element $e_j \in [m]$ corresponds to a scenario j, and every set S_i to a box i. We set $c_{ij} = 0$ iff $e_j \in S_i$, otherwise $c_{ij} = H$. We also add a new scenario X with $v_{Xi} = H$, $\forall i \in [n]$. Scenario X occurs with probability p and all the other m scenarios happen with probability $(1-p)/m$ each.

In this instance, the total cost of optimal non-adaptive strategy is $\mathrm{OPT}_{NA} \le pH + \mathrm{OPT}_{SC}$, since we may pay the set-cover cost to cover all scenarios other than X, and pay an additional cost H to cover X.

Consider any computationally efficient algorithm $\mathcal{A}$ that returns a fully adaptive strategy for such an instance. Since the costs of the boxes are 0 or H, we may assume without loss of generality that any FA strategy stops probing as soon as it observes a box with cost 0 and chooses that box. We say that the strategy covers a scenario when it finds a box of cost 0 in that scenario. Furthermore, prior to finding a box with cost 0, the FA strategy learns no information about which scenario it is in other than that the scenario is as yet uncovered. Consequently, the strategy follows a fixed probing order that is independent of the scenario that is instantiated. We can now convert such a strategy into a bicriteria approximation for the underlying set cover instance. In particular, for $k \in [n]$, let r_k denote the number of scenarios that are covered by the first k probed boxes. Then, we obtain a solution to $\mathcal{SC}$ with k sets covering r_k elements. By Lemma IV.4 then, for every $\varepsilon > 0$, there must exist an instance of set cover, $\mathcal{SC}$, and by extension an instance of optimal search, on which $\mathcal{A}$ satisfies $r_k \leq 1 - \left(1 - \frac{1+\varepsilon}{\text{OPT}_{SC}}\right)^{k-1}$ for all $k \leq \frac{\log n}{(1+\varepsilon)}\text{OPT}_{SC}$.

For the rest of the argument, we focus on that hard instance for $\mathcal{A}$. Let N denote the maximum number of boxes $\mathcal{A}$ probes before stopping to return a box of cost H,[3] and R_k the event that FA reaches step k. Then the expected query time of the strategy is at least

$$\begin{aligned}
&\mathbf{Pr}\left[s = X\right] \cdot N + \mathbf{Pr}\left[s \neq X\right] \sum_{k=1}^{N} \mathbf{Pr}\left[R_k | s \neq X\right] \\
&\geq pN + (1-p) \sum_{k=1}^{N} \left(1 - \frac{1+\varepsilon}{\text{OPT}_{SC}}\right)^{k-1} \\
&= pN + (1-p) \left(1 - \left(1 - \frac{1+\varepsilon}{\text{OPT}_{SC}}\right)^{N}\right) \frac{\text{OPT}_{SC}}{1+\varepsilon}.
\end{aligned}$$

On the other hand, the expected cost of the FA strategy is at least

$$\begin{aligned}
&H \cdot (\mathbf{Pr}\left[s = X\right] + \\
&\quad \mathbf{Pr}\left[s \neq X \wedge \text{FA didn't find 0 in first } N \text{ steps }\right]) \\
&\quad \geq pH + (1-p) H \left(1 - \frac{1+\varepsilon}{\text{OPT}_{SC}}\right)^{N}.
\end{aligned}$$

Thus the total cost of such fully-adaptive strategy is lower bounded by

$$\begin{aligned}
\text{ALG}_{FA} \geq pH &\\
&+ (1-p) H \left(1 - \frac{1+\varepsilon}{\text{OPT}_{SC}}\right)^{N} + pN \\
&+ (1-p) \left(1 - \left(1 - \frac{1+\varepsilon}{\text{OPT}_{SC}}\right)^{N}\right) \frac{\text{OPT}_{SC}}{1+\varepsilon}.
\end{aligned}$$

Let x be defined so that $(1-(1+\varepsilon)/\text{OPT}_{SC})^N = e^{-x}$. Then, $N = -x/\ln(1-(1+\varepsilon)/\text{OPT}_{SC}) \geq x(\text{OPT}_{SC}/(1+\varepsilon)-1)$. Substituting these expressions in the above equation we get

$$\begin{aligned}
\text{ALG}_{FA} \geq& pH + (1-p) H e^{-x} \\
&+ p \cdot x \left(\frac{\text{OPT}_{SC}}{1+\varepsilon} - 1\right) \\
&+ (1-p)(1-e^{-x}) \frac{\text{OPT}_{SC}}{1+\varepsilon}.
\end{aligned}$$

[3] We may safely assume that $N \leq \log n \text{OPT}_{SC}$.

The RHS is minimized at $x = \ln\left(\frac{(1-p)(H(1+\varepsilon)-\text{OPT}_{SC})}{p(\text{OPT}_{SC}-(1+\varepsilon))}\right)$. By setting $\epsilon \to 0$, $p = 0.22$ and $H = 4.59\text{OPT}_{SC}$, the competitive ratio becomes

$$\frac{\text{ALG}_{FA}}{\text{OPT}_{NA}} \geq 1.278$$

when $\text{OPT}_{SC} \to \infty$. ∎

V. Competing with the partially-adaptive benchmark

Moving on to our main result, in this section we compete against the optimal partially-adaptive strategy. Recall that the program (LP-SPA) is a relaxation for the optimal SPA strategy, and therefore, also bounds from below the cost of the optimal PA strategy. We round the optimal solution to this LP to obtain a constant-competitive SPA strategy.

Given a solution to (LP-SPA), we identify for each scenario a subset of low cost boxes. Our goal is then to find a probing order, so that for each scenario we quickly find one of the low cost boxes. This problem of "covering" every scenario with a low cost box is identical to the min-sum set cover (MSSC) problem introduced by [20]. Employing this connection allows us to convert an approximation for MSSC into an SPA strategy at a slight loss in approximation factor. Our main result is as follows.

Lemma V.1. *There exists a scenario-aware partially-adaptive strategy with competitive ratio $3 + 2\sqrt{2}$ against the optimal partially-adaptive strategy.*

Combining this with Theorem III.4 we get the following theorem.

Theorem V.2. *We can efficiently find a partially-adaptive strategy that is $(3 + 2\sqrt{2})\frac{e}{e-1} = 9.22$-competitive against the optimal partially-adaptive strategy.*

VI. Extension to other feasibility constraints

In this section we extend the problem in cases where there is a feasibility constraint $\mathcal{F}$, that limits what or how many boxes we can choose. We consider the cases where we are required to select k distinct boxes, and k independent boxes from a matroid. In both cases we design SPA strategies that can be converted to PA. These two variants are described in more detail in subsections VI-A and VI-B that follow.

A. *Selecting k items*

In this section $\mathcal{F}$ requires that we pick k boxes to minimize the total cost and query time. As in Section V we aim to compete against the optimal partially-adaptive strategy. We design a PA strategy which achieves an $O(1)$-competitive ratio. If $c_{is} \in \{0, \infty\}$, the problem is the generalized min-sum set cover problem first introduced in [36]. [36] gave a $\log n$-approximation, which then was improved to a constant in [37] via an LP-rounding based algorithm. Our proof follows the latter proof in spirit, and generalize to the case where boxes have arbitrary values. Our main result is the following.

Lemma VI.1. *There exists a scenario-aware partially-adaptive $O(1)$-competitive algorithm to the optimal partially-adaptive algorithm for picking k boxes.*

Combining this with Theorem III.4 we get the following theorem.

Theorem VI.2. *We can efficiently find a partially-adaptive strategy for optimal search with k options that is $O(1)$-competitive against the optimal partially-adaptive strategy.*

B. Picking a matroid basis of rank k

In this section $\mathcal{F}$ requires us to select a basis of a given matroid. More specifically, assuming that boxes have an underlying matroid structure we seek to find a base of size k with the minimum cost and the minimum query time. We first design a scenario-aware partially-adaptive strategy in Lemma VI.3 that is $O(\log k)$-competitive against optimal partially-adaptive strategy. Then, in Theorem VI.5 we argue that such competitive ratio is asymptotically tight.

Lemma VI.3. *There exists a scenario-aware partially-adaptive $O(\log k)$-approximate algorithm to the optimal partially-adaptive algorithm for picking a matroid basis of rank k.*

Combining this lemma with Theorem III.4 we get the following theorem.

Theorem VI.4. *We can efficiently find a partially-adaptive strategy for optimal search over a matroid of rank k that is $O(\log k)$-competitive against the optimal partially-adaptive strategy.*

Now we argue that the $O(\log k)$-approximation we got is essentially tight. The following theorem implies that under common complexity assumption, no efficient fully-adaptive algorithm can get asymptotically better competitive ratio, even compared to optimal non-adaptive cost.

Theorem VI.5. *Assuming NP$\not\subseteq$RP, no computationally efficient fully-adaptive algorithm can approximate the optimal non-adaptive cost within a factor of $o(\log k)$.*

Proof: We provide an approximation-preserving reduction from Set Cover problem to finding good fully-adaptive strategy. Let $\mathcal{SC} = ([n], \{S_1, \dots, S_k\})$ be a Set Cover instance on a ground set of n elements, and k sets $S_1, \dots, S_k$. Denote by OPT_{SC} the optimal solution to this Set Cover instance. We construct an instance of partition matroid coverage, where the rank is k. Each segment of the partition consists of multiple copies of the sets $S_1, \dots, S_k$. Every scenario consists of one set from each segment, as seen in Table II.

	Segment 1	**Segment** 2	$\dots$	**Segment** k
Scenario 1	S_1	S_1	$\dots$	S_1
Scenario 2	S_1	S_1		S_2
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
Scenario k^k	S_k	S_k		S_k

Table II
INSTANCE OF PARTITION MATROID k-COVERAGE

A scenario is covered when k elements are selected, one for each of the k sets of every segment. This is an instance of the probing problem we study with cost for each box being 0 or ∞. Similarly, we say a segment is *covered* when we have chosen at least one element in every set it contains. Denote by ALG_{FA} and OPT_{NA} the solution of any fully-adaptive algorithm and the optimal non-adaptive solution respectively for this transformed instance. Any fully-adaptive algorithm will select elements, trying to cover all scenarios. Initially, observe that $\text{OPT}_{NA} \leq k\text{OPT}_{SC}$, since the non-adaptive will at most solve the Set Cover problem in the k different segments. We assume that we can approximate the non-adaptive strategy with competitive ratio α i.e. $\text{ALG}_{FA} = O(\alpha)\text{OPT}_{NA} = O(k\alpha)\text{OPT}_{SC}$.

Let ℓ be the number of elements ALG_{FA} has selected when exactly $k/2$ segments are covered and let s be a randomly chosen scenario. For each one of the $k/2$ uncovered segments, there are at least 1 uncovered set. Therefore

$$\mathbf{Pr}\left[s \text{ is uncovered}\right] \geq 1 - \left(1 - \frac{1}{k}\right)^{k/2} \approx 1 - \frac{1}{\sqrt{e}}.$$

This implies $\text{ALG}_{FA} \geq \ell\left(1 - \frac{1}{\sqrt{e}}\right)$ thus $\ell = O(k\alpha)\text{OPT}_{SC}$. Notice that there exists some segment that is covered using $\ell/(k/2) = O(\alpha)\text{OPT}_{SC}$ elements. Thus any efficient algorithm that provides $O(\alpha)$-approximation of non-adaptive strategy using fully-adaptive strategy can be transformed efficiently to an $O(\alpha)$-approximation algorithm for Set Cover.

Although above reduction from set cover has k^k scenarios that cannot be constructed in polynomial time, by Lemma III.3 $\text{poly}(n, \frac{1}{\epsilon}, \log\frac{1}{\delta})$ samples of all scenarios is sufficient to get accuracy within ϵ with probability $1 - \delta$ for any probing strategy. Let $\varepsilon = 1$ and $\delta = \frac{1}{3}$. The above reduction implies that if there is a poly-time algorithm that computes a probing strategy with cost $o(\log k)\text{OPT}_{NA}$, there exists a poly-time algorithm to solve Set Cover with competitive ratio $o(\log k)$ with probability $\frac{2}{3}$. By [19] such algorithm cannot exist assuming NP$\not\subseteq$RP. ■

VII. Boxes with General Probing Times: Revisiting the Main Results

In this section, we consider settings where different boxes require different amounts of time to probe. Let p_i denote the probing time required to probe box i. We assume $p_i \in [1, P]$ for some P that is polynomially large in n. The running time and sample complexity of our algorithms will depend linearly on P. Henceforth we will assume that the p_i's are integers: rounding up each probing time to the next integer only increases the total objective function value by a factor of at most 2.

A. Ski rental with general rent cost and learnability via sampling

We first investigate the learnability of the optimal search algorithm via sampling polynomially many scenarios, i.e. Theorem III.4. Recall that Theorem III.4 requires two building blocks: Corollary III.2 which describes a reduction from a general strategy to a scenario-aware strategy and Lemma III.3 that guarantees that a small sample over scenarios suffices to achieve a good approximation.

We proved Lemma III.3 by observing that for any probing order π, the cost of any scenario s is bounded in a polynomial range. This still holds since the total probing time is bounded by nP.

In order to show Corollary III.2 in our case, we need to solve a further generalization of the ski rental problem where we have arbitrary rent costs. Specifically, in the ski rental problem with general rent cost, the input is a sequence of non-increasing buy costs, $a_1 \geq a_2 \geq a_3 \geq \dots$ as well as an integral rent costs p_t for each time t. At each step t, the algorithm decides to either rent skis at a cost of p_t, or buy skis at a cost of a_t. We show that Lemma III.1 still holds beyond the unit-rental-cost case. Together with Lemma III.3 we recover Theorem III.4.

Lemma VII.1. *Consider any sequence of integral buy cost $a_1 \geq a_2 \geq \dots$ and integral rent cost $p_1, p_2, \cdots$. There exists an online algorithm that chooses a stopping time t so that*

$$\sum_{i=1}^{t-1} p_i + a_t \leq \frac{e}{e-1} \min_j \left\{ \sum_{i=1}^{j-1} p_i + a_j \right\}.$$

B. Linear program formulations

To get the linear program relaxation of the optimal Non-Adaptive strategy for selecting one box, we only need to change the objective of the linear program to the following function in (LP-NA):

$$\text{minimize} \sum_{i\in\mathcal{B}} x_i p_i + \frac{1}{|\mathcal{S}|} \sum_{i\in\mathcal{B}, s\in\mathcal{S}} c_{is} z_{is}.$$

For the LP of optimal SPA strategy for selecting one box, we need to account for the probing time of every box in the constraint. In order to do that, we will require that every box is being probed for p_i consecutive steps: $x_{it} = 1$ means that box i has been probed since time $t - p_i + 1$, and the probing of the box finishes at time t. Thus at each time step t, there are $\sum_{i\in\mathcal{B}} \sum_{t\leq t'\leq t+p_i-1} x_{it'}$ boxes under probing, and this should be upper bounded by 1. In other words, we replace constraint (1) in linear program (LP-SPA) by

$$\sum_{i\in\mathcal{B}} \sum_{t\leq t'\leq t+p_i-1} x_{it'} \leq 1, \forall t \in \mathcal{T}.$$

The rest of the program will be the same. Since the probing time of each box is polynomially bounded, such LP still has a polynomial size.

C. SPA vs NA: selecting a single item and beyond

We show that Algorithm 1 works for the general-probing-time case with approximation ratio only losing a factor of 2.

Lemma VII.2. *In general-probing-times case, we can efficiently compute a scenario-aware partially-adaptive strategy with competitive ratio 2 against the optimal non-adaptive strategy.*

Proof: The analysis of ALG_c remains the same, i.e. $\mathbf{E}[\text{ALG}_c] \leq \text{OPT}_c$. Now we consider ALG_t. Notice that each step of the algorithm for constructing the probing order is completely independent, with stopping probability $\frac{1}{\sum_{i\in\mathcal{B}} x_i}$ at each point. However, the "length" of each step depends on the probing time for the box picked for that step. Let τ denote the step at which we stop. We have $\mathbf{E}[\tau] = \sum_{i\in\mathcal{B}} x_i$. For any step $t < \tau$, the expected probing time for this step is

$$\begin{aligned} & \mathbf{E}[\text{probing time at step } t | t \text{ is not stopping time}] \\ < \; & \frac{\mathbf{E}[\text{probing time at step } t]}{\mathbf{Pr}[t \text{ is not stopping time}]} \\ = \; & \frac{\sum_{i\in\mathcal{B}} x_i p_i / \sum_{i\in\mathcal{B}} x_i}{1 - 1/\sum_{i\in\mathcal{B}} x_i}. \end{aligned}$$

Thus the expected total probing time is

$$\begin{aligned} \mathbf{E}[\text{ALG}_t] =& \mathbf{E}[\text{probing time at steps } < \tau] \\ & + \mathbf{E}[\text{probing time at step } \tau] \\ \leq & \frac{\sum_{i\in\mathcal{B}} x_i p_i / \sum_{i\in\mathcal{B}} x_i}{1 - 1/\sum_{i\in\mathcal{B}} x_i} (\mathbf{E}[\tau] - 1) + \sum_{i\in\mathcal{B}} x_i p_i \\ =& 2 \sum_{i\in\mathcal{B}} x_i p_i = 2\text{OPT}_t. \end{aligned}$$

Thus $\mathbf{E}[\text{ALG}] \leq 2\text{OPT}$. ■

Similar analysis implies that Lemma VI.1, Theorem VI.2, Lemma VI.3 and Theorem VI.4 still hold for k-coverage case and matroid base case when we have general probing times.

VIII. Inapproximability of the profit maximization variant

In this section we consider the profit maximization variant of the problem discussed above. The boxes now contain some prize value v_{is} for each box i in scenario s, and we want to maximize expected *profit*. Formally, let $\mathcal{P}_s$ be the set of probed boxes in scenario s, our objective is to maximize

$$\mathbf{E}_s\left[\max_{i\in\mathcal{P}_s} v_{is} - |\mathcal{P}_s|\right].$$

It turns out that, contrary to the minimization case, obtaining a constant approximation in this setting is impossible, as the following theorem shows.

Theorem VIII.1. *Assuming P≠NP, no computationally efficient fully-adaptive algorithm can approximate the optimal non-adaptive profit within a constant factor.*

References

[1] S. Chawla, E. Gergatsouli, Y. Teng, C. Tzamos, and R. Zhang, "Pandora's box with correlations: Learning and approximation," *arXiv preprint arXiv:1911.01632*, 2019. [Online]. Available: https://arxiv.org/abs/1911.01632

[2] M. L. Weitzman, "Optimal Search for the Best Alternative," *Econometrica*, vol. 47, no. 3, pp. 641–654, May 1979.

[3] M. Charikar, R. Fagin, V. Guruswami, J. M. Kleinberg, P. Raghavan, and A. Sahai, "Query strategies for priced information (extended abstract)," in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, 2000, pp. 582–591. [Online]. Available: https://doi.org/10.1145/335305.335382

[4] A. Gupta and A. Kumar, "Sorting and selection with structured costs," in *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, 2001, pp. 416–425. [Online]. Available: https://doi.org/10.1109/SFCS.2001.959916

[5] Y. Chen, S. Javdani, A. Karbasi, J. A. Bagnell, S. S. Srinivasa, and A. Krause, "Submodular surrogates for value of information," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 2015, pp. 3511–3518. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9841

[6] Y. Chen, S. H. Hassani, A. Karbasi, and A. Krause, "Sequential information maximization: When is greedy near-optimal?" in *Proceedings of The 28th Conference on Learning Theory*, 2015, pp. 338–363. [Online]. Available: http://proceedings.mlr.press/v40/Chen15b.html

[7] S. Singla, "The price of information in combinatorial optimization," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, 2018, pp. 2523–2532. [Online]. Available: https://doi.org/10.1137/1.9781611975031.161

[8] A. Goel, S. Guha, and K. Munagala, "Asking the right questions: model-driven optimization using probes," in *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 26-28, 2006, Chicago, Illinois, USA*, 2006, pp. 203–212. [Online]. Available: https://doi.org/10.1145/1142351.1142380

[9] A. Gupta and V. Nagarajan, "A stochastic probing problem with applications," in *Integer Programming and Combinatorial Optimization - 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings*, 2013, pp. 205–216. [Online]. Available: https://doi.org/10.1007/978-3-642-36694-9_18

[10] M. Adamczyk, M. Sviridenko, and J. Ward, "Submodular stochastic probing on matroids," *Math. Oper. Res.*, vol. 41, no. 3, pp. 1022–1038, 2016. [Online]. Available: https://doi.org/10.1287/moor.2015.0766

[11] A. Gupta, V. Nagarajan, and S. Singla, "Algorithms and adaptivity gaps for stochastic probing," in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, 2016, pp. 1731–1747. [Online]. Available: https://doi.org/10.1137/1.9781611974331.ch120

[12] ——, "Adaptivity gaps for stochastic probing: Submodular and XOS functions," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, 2017, pp. 1688–1702. [Online]. Available: https://doi.org/10.1137/1.9781611974782.111

[13] A. Gupta, H. Jiang, Z. Scully, and S. Singla, "The markovian price of information," in *Integer Programming and Combinatorial Optimization - 20th International Conference, IPCO 2019, Ann Arbor, MI, USA, May 22-24, 2019, Proceedings*, 2019, pp. 233–246. [Online]. Available: https://doi.org/10.1007/978-3-030-17953-3_18

[14] J. D. Hartline, *Mechanism design and approximation*, 2013, chapter 6: Prior-free Approximation. [Online]. Available: http://jasonhartline.com/MDnA/

[15] J. D. Hartline and T. Roughgarden, "Optimal mechanism design and money burning," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, 2008, pp. 75–84. [Online]. Available: https://doi.org/10.1145/1374376.1374390

[16] D. D. Sleator and R. E. Tarjan, "Self-adjusting binary search trees," *J. ACM*, vol. 32, no. 3, p. 652–686, Jul. 1985. [Online]. Available: https://doi.org/10.1145/3828.3835

[17] E. D. Demaine, D. Harmon, J. Iacono, and M. Patrascu, "Dynamic optimality—almost," *SIAM J. Comput.*, vol. 37, no. 1, p. 240–251, Apr. 2007. [Online]. Available: https://doi.org/10.1137/S0097539705447347

[18] A. R. Karlin, M. S. Manasse, L. A. McGeoch, and S. S. Owicki, "Competitive randomized algorithms for non-uniform problems," in *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California, USA.*, 1990, pp. 301–309. [Online]. Available: http://dl.acm.org/citation.cfm?id=320176.320216

[19] I. Dinur and D. Steurer, "Analytical approach to parallel repetition," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, 2014, pp. 624–633. [Online]. Available: https://doi.org/10.1145/2591796.2591884

[20] U. Feige, L. Lovász, and P. Tetali, "Approximating min-sum set cover," in *Approximation Algorithms for Combinatorial Optimization, 5th International Workshop, APPROX 2002, Rome, Italy, September 17-21, 2002, Proceedings*, 2002, pp. 94–107. [Online]. Available: https://doi.org/10.1007/3-540-45753-4_10

[21] H. Jiang, J. Li, D. Liu, and S. Singla, "Algorithms and adaptivity gaps for stochastic k-tsp," in *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, ser. LIPIcs, T. Vidick, Ed., vol. 151. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 45:1–45:25. [Online]. Available: https://doi.org/10.4230/LIPIcs.ITCS.2020.45

[22] H. Beyhaghi and R. Kleinberg, "Pandora's problem with nonobligatory inspection," in *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM, 2019, pp. 131–132. [Online]. Available: https://doi.org/10.1145/3328526.3329626

[23] R. Gupta and T. Roughgarden, "A PAC approach to application-specific algorithm selection," *SIAM J. Comput.*, vol. 46, no. 3, pp. 992–1017, 2017. [Online]. Available: https://doi.org/10.1137/15M1050276

[24] N. Ailon, B. Chazelle, S. Comandur, and D. Liu, "Self-improving algorithms," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, 2006, pp. 261–270. [Online]. Available: http://dl.acm.org/citation.cfm?id=1109557.1109587

[25] K. L. Clarkson, W. Mulzer, and C. Seshadhri, "Self-improving algorithms for convex hulls," in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, 2010, pp. 1546–1565. [Online]. Available: https://doi.org/10.1137/1.9781611973075.126

[26] M. Balcan, V. Nagarajan, E. Vitercik, and C. White, "Learning-theoretic foundations of algorithm configuration for combinatorial partitioning problems," in *Proceedings of the 30th Conference on Learning Theory*, 2017, pp. 213–274. [Online]. Available: http://proceedings.mlr.press/v65/balcan17a.html

[27] M. Balcan, T. Dick, T. Sandholm, and E. Vitercik, "Learning to branch," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 353–362. [Online]. Available: http://proceedings.mlr.press/v80/balcan18a.html

[28] M. Balcan, T. Dick, and E. Vitercik, "Dispersion for data-driven algorithm design, online learning, and private optimization," in *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, 2018, pp. 603–614. [Online]. Available: https://doi.org/10.1109/FOCS.2018.00064

[29] R. Kleinberg, K. Leyton-Brown, and B. Lucier, "Efficiency through procrastination: Approximately optimal algorithm configuration with runtime guarantees," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 2017, pp. 2023–2031. [Online]. Available: https://doi.org/10.24963/ijcai.2017/281

[30] G. Weisz, A. Gyorgy, and C. Szepesvari, "Leapsandbounds: A method for approximately optimal algorithm configuration," in *International Conference on Machine Learning*, 2018, pp. 5257–5265.

[31] D. Alabi, A. T. Kalai, K. Ligett, C. Musco, C. Tzamos, and E. Vitercik, "Learning to prune: Speeding up repeated computations," in *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, 2019, pp. 30–33. [Online]. Available: http://proceedings.mlr.press/v99/alabi19a.html

[32] T. Lykouris and S. Vassilvitskii, "Competitive caching with machine learned advice," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018, pp. 3302–3311. [Online]. Available: http://proceedings.mlr.press/v80/lykouris18a.html

[33] M. Purohit, Z. Svitkina, and R. Kumar, "Improving online algorithms via ML predictions," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, 2018, pp. 9684–9693. [Online]. Available: http://papers.nips.cc/paper/8174-improving-online-algorithms-via-ml-predictions

[34] C. Hsu, P. Indyk, D. Katabi, and A. Vakilian, "Learning-based frequency estimation algorithms," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. [Online]. Available: https://openreview.net/forum?id=r1lohoCqY7

[35] S. Gollapudi and D. Panigrahi, "Online algorithms for rent-or-buy with expert advice," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019, pp. 2319–2327. [Online]. Available: http://proceedings.mlr.press/v97/gollapudi19a.html

[36] Y. Azar, I. Gamzu, and X. Yin, "Multiple intents re-ranking," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, 2009, pp. 669–678. [Online]. Available: https://doi.org/10.1145/1536414.1536505

[37] N. Bansal, A. Gupta, and R. Krishnaswamy, "A constant factor approximation algorithm for generalized min-sum set cover," in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010, pp. 1539–1545. [Online]. Available: https://doi.org/10.1137/1.9781611973075.125

[38] M. Skutella and D. P. Williamson, "A note on the generalized min-sum set cover problem," *Oper. Res. Lett.*, vol. 39, no. 6, pp. 433–436, 2011. [Online]. Available: https://doi.org/10.1016/j.orl.2011.08.002

[39] S. Im, M. Sviridenko, and R. Van Der Zwaan, "Preemptive and non-preemptive generalized min sum set cover," *Mathematical Programming*, vol. 145, no. 1-2, pp. 377–401, 2014.

[40] Y. Azar and I. Gamzu, "Ranking with submodular valuations," in *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, 2011, pp. 1070–1079. [Online]. Available: https://doi.org/10.1137/1.9781611973082.81