Learning in Combinatorial Optimization: What and How to Explore*

Sajad Modaresi
University of Pittsburgh

Denis Saure
University of Pittsburgh

Juan Pablo Vielma
MIT Sloan School of Management

June 20, 2013

Abstract

We study a family of sequential decision problems under model uncertainty in which, at each period, the decision maker faces a different instance of a combinatorial optimization problem. Instances differ in their objective coefficient vectors, which are unobserved prior to implementation. These vectors however are known to be random draws from a common and initially unknown distribution function. By implementing different solutions, the decision maker extracts information about the objective function, but at the same time experiences the cost associated with said solutions. We show that balancing the implied *exploration vs. exploitation* trade-off requires addressing critical challenges not present in previous studies: in addition to determining exploration frequencies, the decision maker faces the issue of *what* to explore and *how* to do so. Our work provides clear answers to both questions. In particular, we show that efficient data collection might be achieved by solving a new class of combinatorial problems, which we refer to as *Optimality Cover Problem* (OCP). We establish a fundamental limit on the performance of any admissible policy, which relates to the solution to OCP. Using the insight derived from such a bound, we develop a family of policies and establish theoretical guarantees for their performances, which we contrast against the fundamental bound. These policies admit asynchronous versions, ensuring implementability.

1 Introduction

Motivation. Traditional solution approaches to many operational problems are based on combinatorial optimization problems, and typically involve instantiating a deterministic mathematical

*The authors thank the three anonymous referees, the area editor, and the associate editor for their helpful feedback. This research was supported by the National Science Foundation [Grant CMMI-1233441]. Correspondence: sam213@pitt.edu, dsaure@pitt.edu, jvielma@mit.edu.

program: nevertheless, in practice, instances are not usually known in advance. When possible, parameters characterizing said instances are estimated *off-line*, either by using historical data or from direct observation of the (idle) system. Unfortunately, off-line estimation is not always possible as, for example, historical data (if available) might only provide partial information pertaining previously implemented solutions. Consider, for instance, shortest path problems in network applications: repeated implementation of a given path might reveal cost information about arcs on such a path, but might provide no further information about costs of other arcs in the graph. Similar settings arise in other network applications (e.g., tomography and connectivity) in which feedback about cost follows from instantiating and solving combinatorial problems such as spanning and Steiner trees.

The situation depicted above is not exclusive to network applications: some assortment planning problems in the fast fashion industry, in which initial demand uncertainty is common, might also be casted as combinatorial problems (e.g., single-sale revenue maximization with display constraints and independent demands might be casted as a knapsack problem). Furthermore, in some application areas one might formulate ad-hoc combinatorial problems: e.g., clinical trials of antiretroviral therapies (ART) involve testing the (initially unknown) combined effect of multiple choices of antiretroviral drug cocktails. However, not all drugs are compatible, and clinical guidelines further restrict the set of admissible cocktails, thus resulting in a non-trivial set of feasible ART.

In the settings above, parameter estimation might be conducted *on-line* using feedback associated with implemented solutions, and revisited as more information about the system's primitives becomes available. In doing this, one must consider the interplay between the performance of a solution and the feedback generated by its implementation: some parameters might only be reconstructed by implementing solutions that perform poorly (relative to the optimal solution). This point is particularly relevant for applications in which there is a combinatorial structure in the solution set: there might be multiple ways of estimating a given set of parameters, each using feedback from a different set of solutions, thus experiencing different performances. Moreover, optimality of a solution to a combinatorial optimization problem might be invariant to changes in certain parameters. Hence, not all parameters might need to be estimated to find such a solution and prove its optimality. Therefore, the question of *what* information to collect and *how* to collect it is central in settings with model uncertainty and combinatorial nature.

In addition to the above, the temporal dimension plays a crucial role in incorporating feedback effectively: in assortment planning, for example, feedback from customers affects the demand estimates, which in turn affect the assortment decision; however, computing new assortment decisions might take non-trivial time as it might require solving an underlying combinatorial formulation. In the meantime, customers continue to arrive, new feedback is collected, and new estimates are computed (which calls for resolving the underlying model). Thus, in general, the question of how

frequently to solve the underlying formulation is crucial to performance and implementability.

In this work our focus is on settings in which an agent must implement solutions to a series of arriving instances of a given combinatorial problem, and there is initial uncertainty about said instances. In particular, we are interested in settings in which each instance is unknown prior to the implementation of a solution, but is known to be the realization of a (independent) random vector, drawn from unknown distribution common to all instances. We analyze settings in which implementing a solution provides *partial* feedback that depends on the solution implemented, and the agent's objective is to optimize cumulative performance. The main salient features of our work, which distinguish it from previous studies, are: (*i*) the combinatorial structure shared by the (large) set of implementable solutions; and (*ii*) the emphasis on the implementability of the proposed policies.

Main objectives and assumptions. The main focus in this work is on efficient information collection for expected cumulative performance optimization. By implementing various solutions, one is likely to experience suboptimal performance; however, the associated feedback should help in reconstructing the underlying instance distribution, which, if used efficiently, should improve future performance. Conversely, implementing solutions thought to be optimal should contribute to optimize cumulative performance, but at the expense of limiting information collection, which might compromise future performance.

The above is an instance of the classical *exploration vs. exploitation* trade-off, and as such it could be approached through the multi-armed bandit paradigm. Unfortunately, a naive implementation of such an approach would require collecting information on all solutions available to the agent, which in our combinatorial setting might be prohibitively large. Thus, one might expect classical bandit algorithms to perform poorly (although, at this point, it is not clear what the appropriate benchmark is). However, a thorough examination of the arguments in the bandit setting reveals that their basic principles are still applicable. Thus, our objective in this paper can be seen as interpreting said principles and adapting them to our setting with the goal of developing efficient policies.

Without loss of generality, we assume that the underlying combinatorial problem is that of cost minimization, thus the agent strives to minimize cumulative cost. In addition, we assume that instance uncertainty is restricted to cost-coefficients in the objective function. Hence, the feasible region is the same for each instance and known upfront by the agent. In this regard, we assume that cost-coefficients vary among instances, but they are drawn from a common time-invariant distribution function.

Main results. In this paper we develop policies that both efficiently collect information and are implementable in practice. These policies are obtained by establishing and studying a fundamental limit on the performance of any admissible policy. Through this limit, we establish that

any policy that is not specially tailored to a particular cost distribution (i.e., those that perform consistently well) must incur an additional cost –relative to that incurred by a clairvoyant agent with prior knowledge on the cost distribution– that grows with the logarithm of the total number of instances. More importantly, such an additional cost is proportional to the solution to a new class of combinatorial problems, which we denote as the *Optimality Cover Problem* (OCP). For a combinatorial optimization problem, OCP finds a set of feasible solutions with minimum expected "opportunity" cost so that an optimal solution remains optimal even when coefficients not "covered" by OCP are assumed to be the *lowest* possible. In other words, a feasible solution to OCP identifies an efficient exploration set: feedback coming from implementing such a set of solutions suffices to guarantee the optimality of a solution to the underlying combinatorial optimization problem, all of this while incurring minimum opportunity cost. Intuitively speaking, the fundamental limit result suggests that any admissible policy must incur an additional cost (relative to a clairvoyant agent) proportional to that associated with providing an efficient optimality guarantee for the optimal solution to the underlying combinatorial optimization problem. Our implementable policy adaptively approximates the solution to OCP and restricts exploration efforts accordingly. In addition, the proposed policy enforces the "right" frequency of exploration on such solutions. In particular, our policy constructs and solves a proxy for OCP at a decreasing frequency, and collects information in between successive decision epochs by implementing the solutions prescribed by such a proxy. Our numerical experiments show that our policy outperforms the relevant benchmarks in the long-term, and remains competitive in the short-term.

The remainder of the paper. Section 2 reviews related work, and Section 3 formulates the problem. In Section 4 we present a simple policy and compare its performance to that achieved under the classical multi-armed bandit paradigm. Section 5 presents a limit on achievable performance, and Section 6 presents a policy built on the insight gained from such a limit. In Section 7 we discuss computational issues, and Section 8 illustrates the results in the paper by means of numerical experiments. Finally, Section 9 presents extensions and concluding remarks. Proofs are relegated to Appendices A and B.

2 Literature Review

Classical bandit settings. Introduced in Thompson (1933) and Robbins (1952), the multi-armed bandit setting is a classical framework for sequential decision making under model uncertainty. Extensively studied in many fields (see Gittins (1989), Cesa-Bianchi and Lugosi (2006) and references therein), in its basic formulation a gambler aims to maximize cumulative reward by selecting and pulling arms of a slot machine sequentially over time when limited prior information on reward distributions is available. The gambler faces the classical exploration vs. exploitation trade-off: on the one hand, by pulling exclusively the arm thought to be the "best", the gambler might fail to

identify the arm with the highest mean reward. On the other hand, by pulling all arms repeatedly, the gambler might identify the best arm, but at the cost of sacrificing cumulative reward. Our setting can be seen as a multi-armed bandit via the following analogy: each solution corresponds to an arm, and implementing such a solution corresponds to pulling said arm.

The seminal work of Gittins (1979) shows that, for the case of independent and discounted arm rewards, and infinite horizon, the optimal policy is of the index type. Unfortunately, index-based policies are not always optimal (see Berry and Fristedt (1985), and Whittle (1982)) and research has focused on developing approximate policies. Even when optimal or used heuristically, most *Gittins* indices can not be computed in closed form, as they involve solving underlying optimal stopping problems. In this regard, the celebrated UCB1 policy (Auer et al. 2002) provides a tractable index-based policy while prioritizing practical implementability. In our bandit analogy rewards are correlated: only a few papers address this case (see e.g., Ryzhov and Powell (2009) and Ryzhov et al. (2012)). Unlike such research, which is conducted under the Bayesian paradigm, ours focuses on deriving asymptotically efficient policies from a frequentist perspective.

In their seminal work, Lai and Robbins (1985) study asymptotically efficient policies for the classical multi-armed bandit setting. Using a change of measure argument, they provide a fundamental limit on the performance of any policy relative to that of an oracle with prior knowledge on reward distributions. Their argument is based on the idea that any arm could be the "best" under an alternative parameter configuration, and thus *must* be pulled at a precise frequency. In the present paper we adapt this idea to establish similar performance bounds: see the discussion in Section 5. (See Kulkarni and Lugosi (1997) for a finite-sample minimax version of the asymptotic lower bound of Lai and Robbins (1985).).

Anantharam et al. (1987) study a version of the multi-armed bandit problem where a gambler must pull a fixed number of arms simultaneously. They extend the fundamental bound of Lai and Robbins (1985) to a setting with multiple plays and propose efficient allocations rules. Similarly, Agrawal et al. (1990) study multi-armed bandit problems with multiple plays and switching costs. Our setting can be seen as a multi-armed bandit with multiple plays via the following analogy: each ground element in the underlying formulation corresponds to an arm, thus implementing a solution corresponds to pulling the associated arms simultaneously. In this regard, our setting imposes a special structure on the set of feasible simultaneous plays, which prevents us from applying known results for settings with multiple plays.

Bandit problems with large set of arms. Bandit settings with large number of arms have received significant attention in the last decade. Similar to our setting, in such work arm rewards are typically endowed with some known structure that is exploited to improve upon the performance of regular bandit algorithms.

A first strain of literature considers settings with a continuous set of arms, so exploring all arms

is not possible. Agrawal (1995) studies a multi-armed bandit problem where arms represent points in the real line and their expected rewards are given by a continuous function of the arm. In a more general setting, Kleinberg et al. (2008) consider the case where the set of arms forms a metric space, and the mean reward function satisfies a Lipschitz condition. (See Bubeck et al. (2011) for a review of work in *continuum* bandits.). In a different setting Mersereau et al. (2009) study a bandit problem with a large (possibly infinite) collection of arms where expected rewards depend on an initially unknown scalar. Similarly, Rusmevichientong and Tsitsiklis (2010) consider settings where expected rewards are given by a linear function of a random vector.

To the best of our knowledge, only a few papers have considered bandit problems with combinatorial structure. In the context of assortment planning, Rusmevichientong et al. (2010) study a setting where the gambler may pull any subset of arms with cardinality below some fixed capacity. Unlike ours, their objective function is not additive on the selected arms and is driven by a discrete choice model (see Caro and Gallien (2007) for a similar formulation with additive objective function). Saure and Zeevi (2013) extend the performance bound in Lai and Robbins (1985) to such a setting: the simple policy we present in Section 4 is based on ideas in the latter paper (in particular, its cycling structure) .

In our setting, Gai et al. (2012) extend the well-known UCB1 policy for traditional multi-armed bandits in Auer et al. (2002) and provide what is *essentially* an order $|A|^4 \ln N$ performance bound, where $|A|$ denotes the size of the underlying ground set from which arms are formed, and N denotes the total number of pulls. They do not provide a fundamental bound on achievable performance, but propose a policy that is applicable to our setting, thus we use it as a benchmark later in the numerical experiments in Section 8. (Note that the performance bound for the simple policy in Section 4 is of order $|A| \ln N$.)

Considering an adversarial setting, Cesa-Bianchi and Lugosi (2012) study a bandit problem where arms belong to a given finite set in $\mathbb{R}^d$ (see Auer et al. (2003) for a description of the adversarial bandit setting). They introduce a randomized forecasting strategy and provide an order $\sqrt{Nd \ln |\mathcal{S}|}$ upper bound on its performance, where N denotes the total number of pulls and $|\mathcal{S}|$ denotes the total number of arms. Our focus instead is on *stochastic* (non-adversarial) settings. In this regard, our work leverages the additional structure imposed in the stochastic setting to establish limits on attainable performance and develop policies whose performance bounds exhibit the "right" dependence on A, $\mathcal{S}$, and N. It is worth mentioning that Cesa-Bianchi and Lugosi (2012) draw on ideas from the literature of prediction with expert advice (see e.g., Cesa-Bianchi and Lugosi (2006)). Note, however, that the focus of such literature is on settings with full-feedback, thus associated methods are not directly applicable to our setting.

Online subset selection. Broadly speaking, our work contributes to the literature of online learning with combinatorial number of alternatives. There are several studies that focus on similar

online learning problems, from the raking and selection perspective. Ryzhov and Powell (2011) study information collection in settings where the decision maker selects individual arcs from a directed graph, and Ryzhov and Powell (2012) consider a more general setting where selection is made from a polyhedron (in Section 7 we specialize some of our results to such a setting, from a bandit perspective). See also Ryzhov et al. (2012), and the references within.

3 Problem Formulation

Model primitives and basic assumptions. We consider the problem of an agent who must select and implement solutions to a series of instances of a given combinatorial optimization problem. Without loss of generality, we assume that such a problem is that of cost minimization. Instances are presented sequentially through time, and we use n to index them according to their arrival times, so $n = 1$ corresponds to the first instance, and $n = N$ to the last, where N denotes the (possibly unknown) total number of instances. Each instance is uniquely characterized by a set of cost-coefficients, i.e., instance $n \in \mathbb{N}$ is associated with cost-coefficients $B_n := (b_{a,n} : a \in A) \in \mathbb{R}^{|A|}$, and the full instance is defined as $f(B_n)$, where

$$f(B) : z^*(B) := \min \left\{ \sum_{a \in S} b_a : S \in \mathcal{S} \right\} \quad B \in \mathbb{R}^{|A|}, \tag{1}$$

$\mathcal{S}$ is a family of subsets of elements of a given ground set A, and b_a is the *cost* associated with a ground element $a \in A$. We let $\mathcal{S}^*(B)$ be the set of optimal solutions to (1) and $z^*(B)$ be its optimal objective value (cost).

We assume that each element $b_{a,n} \in B_n$ is a random variable, independent and identically distributed across instances, and independent of other components in B_n. We let $F(\cdot)$ denote the common distribution of B_n for $n \in \mathbb{N}$, which we assume is initially *unknown*. We assume, however, that upper and lower bounds on its range are known upfront. That is, it is known that $l_a \leq b_{a,n} \leq u_a$ a.s., for all $a \in A$ and $n \in \mathbb{N}$.

We assume that, in addition to not knowing $F(\cdot)$, the agent does not observe B_n prior to implementing a solution. Instead, we assume that B_n is only revealed *partially* and *after* a solution is implemented. More specifically, we assume that if solution $S_n \in \mathcal{S}$ is implemented, only cost-coefficients associated with ground elements in S_n (i.e., $\{b_{a,n} : a \in S_n\}$) are observed by the agent and after the associated cost is incurred. Finally, we assume that the agent is interested in minimizing the expected cumulative cost associated with implementing a sequence of solutions.

Full information problem and regret. Consider the case of a clairvoyant agent with prior knowledge about $F(\cdot)$. Such an agent, while still not capable of anticipating the value of B_n, can solve for the solution that minimizes the expected cumulative cost: for instance $n \in \mathbb{N}$ (by the

linearity of the objective function), it is optimal to implement $S_n \in \mathcal{S}^*(\mathbb{E}_F\{B_n\})$, where $\mathbb{E}_F\{\cdot\}$ denotes expectation with respect to F. Note that such a solution set is independent of n.

In practice, the agent does not know F upfront, and must learn about it through the feedback collected from implementing different solutions. Let $\mathcal{F} := \{\mathcal{F}_n : n \in \mathbb{N}\}$ denote the filtration generated by such a feedback (i.e., $\mathcal{F}_n = \sigma(\{b_{a,m} : a \in S_m\,,\, m < n\})$). We restrict our attention to non-anticipative (i.e., $\mathcal{F}$-adapted) policies. Let $\pi := (S_n)_{n=1}^{\infty}$ denote an admissible policy, where $S_n : \mathcal{F}_n \to \mathcal{S}$ maps the available "history" to a solution in $\mathcal{S}$. For any given F and N, the expected cumulative cost incurred by policy π is given by

$$J^{\pi}(F,N) := \sum_{n=1}^{N} \mathbb{E}_F\left\{\sum_{a\in S_n} b_{a,n}\right\}.$$

In practice, no admissible policy can achieve the expected cumulative cost of a clairvoyant agent, and hence we measure the performance of a policy in terms of its *regret*: for a given policy π, F, and N, the regret is defined as

$$R^{\pi}(F,N) := J^{\pi}(F,N) - N\ z^*\left(\mathbb{E}_F\{B_n\}\right).$$

The regret represents the additional expected cumulative cost incurred by policy π relative to that incurred by a clairvoyant agent that knows F upfront (note that regret is always non-negative).

Remark 3.1. Although the regret also depends on the combinatorial optimization problem through A and $\mathcal{S}$, we omit this dependence to simplify the notation.

Our exposition benefits from connecting the regret to the number of instances in which suboptimal solutions are implemented. To make this connection explicit, consider an alternative representation of the regret. For $S \in \mathcal{S}$, let Δ_S^F denote the expected optimality gap associated with implementing S, when costs are distributed according to F. That is,

$$\Delta_S^F := \sum_{a\in S} \mathbb{E}_F\{b_{a,n}\} - z^*\left(\mathbb{E}_F\{B_n\}\right).$$

(Note that the expected optimality gap associated with $S^* \in \mathcal{S}^*\left(\mathbb{E}_F\{B_n\}\right)$ is zero.) For $S \in \mathcal{S}$, let

$$T_n(S) := |\{m < n\ :\ S_m = S\}|$$

denote the number of times that the agent has implemented solution $S_m = S$ prior to instance n. Similarly, for $a \in A$, let

$$T_n(a) := |\{m < n\ :\ a \in S_m\}|$$

denote the number of times that the agent has selected element a prior to instance n (henceforth,

we say ground element $a \in A$ is selected or tried at instance n if $a \in S_n$). Note that $T_n(a)$ and $T_n(S)$ are $\mathcal{F}_n$-adapted for all $a \in A$, $S \in \mathcal{S}$, and $n \in \mathbb{N}$. Using this notation we have that

$$R^\pi(F, N) = \sum_{S \in \mathcal{S}} \Delta_S^F \; \mathbb{E}_F \left\{ T_{N+1}(S) \right\}. \tag{2}$$

Known results for the non-combinatorial case. Traditional multi-armed bandits correspond to settings where $\mathcal{S}$ is formed by singleton subsets of A (i.e., $\mathcal{S} = \{\{a\} : a \in A\}$), thus the combinatorial structure is absent. In their seminal work, Lai and Robbins (1985) (henceforth, LR) establish an asymptotic lower bound on the regret attainable by any "admissible" policy in the traditional setting, and provide policies achieving asymptotic efficiency. To avoid considering policies that perform well in a particular setting at the expense of performing poorly in others, LR restrict attention to policies that perform *consistently well* for any reward distribution F within a certain class. Formally, a policy π is said to be *consistent* if

$$R^\pi(F, N) = o(N^\alpha) \quad \forall \alpha > 0, \tag{3}$$

for every F on a class of *regular* distributions[1]. LR show that consistent policies must explore (pull) each element (arm) in A at least order $\ln N$ times, hence, by (2), their regret must also be of at least order $\ln N$.

Theorem 3.2 (**Lai and Robbins (1985)**). *Let $\mathcal{S} = \{\{a\} : a \in A\}$, then for any consistent policy π and for any $a \in A$ we have*

$$\liminf_{N \to \infty} \mathbb{P}_F \left\{ \frac{T_{N+1}(a)}{\ln N} \geq K_a \right\} = 1, \tag{4}$$

where K_a is a positive finite constant depending on F. In addition, we have

$$\liminf_{N \to \infty} \frac{R^\pi(F, N)}{\ln N} \geq \sum_{a \in A} \Delta_{\{a\}}^F K_a, \tag{5}$$

for any consistent policy π and regular distribution F.

In the above, K_a is the inverse of Kullback-Leibler distance (see e.g., Cover and Thomas (2006)) between the original distribution F and a distribution than makes a optimal.

Different policies have been shown to attain the logarithmic behavior in (5), and in general, there is a trade-off between computational complexity and larger leading constants. For instance, the index-based UCB1 algorithm introduced by Auer et al. (2002) is simple to compute and provides a finite-time theoretical performance guarantee. For $a \in A$, define $\tilde{K}_a := 8/(\Delta_{\{a\}}^F)^2$.

[1] These are distributions satisfying certain *continuity* and *indistinguishability* conditions: see proof of Proposition 5.2.

Theorem 3.3 (**Auer et al. (2002)**). *The expected regret of policy UCB1 after N plays is such that*

$$\frac{R^{\pi}(F,N)}{\ln N} \leq \sum_{a \in A} \Delta^{F}_{\{a\}} \tilde{K}_a + o(1). \tag{6}$$

The left-hand sides of (5) and (6) admit asymptotic lower and upper bounds of the form $C_F\,|A|$, respectively, where C_F is a finite constant depending on F. We informally refer to this behavior as the regret *being proportional* to $|A| \ln N$. In this context, UCB1 has the following favorable properties over other policies: (i) it collects information efficiently in that, for all periods, the growth rate of its regret is comparable to the best asymptotic rate (fundamental lower bound (5) says that the regret of any consistent policy should be asymptotically proportional to $|A| \ln N$); and (ii) it is effectively *implementable* in practice.

The main objective of this paper is to develop policies that share the two favorable properties of UCB1 in the context of combinatorial problems. The first step for this is to understand the fundamental asymptotic limits in this context. Unfortunately, the lower bound in (4) does not apply to the combinatorial setting. Furthermore, while classical multi-armed bandit policies can be easily adapted to our setting, their performance deteriorates significantly. To see this consider the following analogy: a combinatorial bandit is equivalent to a classical (non-combinatorial) multi-armed bandit problem with $|\mathcal{S}|$ arms, each corresponding to a different solution $S \in \mathcal{S}$. Implementing off-the-shelf bandit policies to such a multi-armed bandit setting will result in a regret proportional to $|\mathcal{S}| \ln N$. Unfortunately, for most problems of interest, $|\mathcal{S}|$ is exponential in $|A|$ and thus, when applied to combinatorial problems, traditional multi-armed bandit algorithms will exhibit regrets scaling rather unfavorably with the size of A. We will show that one can still achieve a regret proportional to $|A| \ln N$, but first we need a more detailed analysis of the differences between combinatorial and traditional bandit problems. This analysis eventually leads us to a policy with efficient information collection that, while computationally challenging, is still effectively implementable.

Differences between combinatorial and traditional bandits. In the classical multi-armed bandit, arms are ex-ante identical, thus any arm might turn out to be optimal. In other words, for any $a \in A$, there always exists a distribution F_a such that $a \in \mathcal{S}^*\left(\mathbb{E}_{F_a}\{B_n\}\right)$. Moreover, while executing a policy, it is always possible that a seemingly suboptimal arm might be in fact optimal. Because of this, consistent policies cannot "afford" exploring any given arm for a finite number of times (independent of N), and must explore all arms periodically. Thus, broadly speaking, balancing the exploration vs. exploitation trade-off narrows down to answering *when* (or how frequently) to explore each element $a \in A$ (the answer to this question is given by LR's $\ln N/N$ exploration frequency).

In the combinatorial setting achieving the aforementioned balance requires answering additional questions. First note that, if one is to take advantage of the structural properties of the rewards, information collection should be conducted on the elements of A rather than on those of $\mathcal{S}$. However,

information about $a \in A$ can be collected by implementing different solutions, and thus by incurring potentially different costs. Even if, as in the traditional setting, one is to collect information on all elements in A, there are many possibilities for doing so (e.g., one might implement all solutions in $\mathcal{S}$). Thus, efficient exploration in our setting also involves answering the question of *how* to collect the required information. Second, in the combinatorial setting, ground elements are not upfront identical due to bounds on the range of F. Moreover, even in the case where elements are a-priori identical (e.g., $l_a = c$ for all $a \in A$, for some $c \in \mathbb{R}$), feasible solutions combine elements so that solutions might not be identical upfront. In Section 5 we show that, thanks to this property, not all solutions need to be implemented periodically. More importantly, we show that not all elements of A need to be probed periodically. Thus, the question of *what* to explore proves to be key in limiting the exploration efforts, and therefore the regret.

4 How to Explore: Efficient Covers

Consider applying algorithm UCB1 in Auer et al. (2002) to the combinatorial setting via envisioning all $S \in \mathcal{S}$ as separate arms. After an initialization phase on which each element of $\mathcal{S}$ is implemented once to obtain an initial estimate of its cost, UCB1 implements solution

$$S_n \in \operatorname*{argmin}_{S \in \mathcal{S}} \left\{ \overline{b}_{S,n} - \sqrt{2 \ln(n-1)/T_n(S)} \right\}$$

on instance n, where $\overline{b}_{S,n}$ denotes the average observed cost of solution S prior to instance n. That is,

$$\overline{b}_{S,n} := \frac{1}{T_n(S)} \sum_{m<n \,:\, S_m = S} \sum_{a \in S} b_{a,m}.$$

As mentioned earlier, in many situations of interest, the regret of this algorithm scales exponentially with $|A|$, as illustrated by the following example.

Example 4.1. *Consider the digraph $G = (V, A)$ for $V = \{v_{i,j} : i, j \in \{1, \ldots, k+1\},\ i \leq j\}$ and $A = \{e_i\}_{i=1}^{k} \cup \{p_{i,j} : i \leq j \leq k\} \cup \{q_{i,j} : i \leq j \leq k\}$ where $e_i = (v_{i,i}, v_{i+1,i+1})$, $p_{i,j} = (v_{i,j}, v_{i,j+1})$, and $q_{i,j} = (v_{i,j}, v_{i+1,j})$. This digraph is depicted in Figure 1 for $k = 3$. Let $\mathcal{S}$ be composed of all paths from node $s := v_{1,1}$ to node $t := v_{k+1,k+1}$.*

Set $l_a = 0$ and $u_a = \infty$ for every arc $a \in A$, and let F be such that $\mathbb{E}_F\{b_{e_i,n}\} = 0.03$, and $\mathbb{E}_F\{b_{p_{i,j},n}\} = \mathbb{E}_F\{b_{q_{i,j},n}\} = 0.1$, for all $i \in \{1, \ldots, k\}$ and $i \leq j \leq k$, $n \in \mathbb{N}$. The shortest (expected) path is $S^*(\mathbb{E}_F\{B_n\}) = (e_1, e_2, \ldots, e_k)$ with expected length (cost) $z^*(\mathbb{E}_F\{B_n\}) = 0.03\ k$, and $|\mathcal{S}|$ corresponds to the number of $s - t$ paths, which is equal to $\frac{1}{k+2}\binom{2(k+1)}{(k+1)} \sim \frac{4^{k+1}}{(k+1)^{3/2}\sqrt{\pi}}$ (Stanley 1999).

For this example, the regret of UCB1 is proportional to $\frac{4^{k+1}}{(k+1)^{3/2}\sqrt{\pi}} \ln N$, which can be quite large even for moderate values of k. Broadly speaking, this algorithm picks the solution with the best cost estimate $\overline{b}_{S,n}$ while making sure that cost estimates are updated with LR's prescribed frequency $\ln N/N$. Because UCB1 is not aware of the combinatorial structure of the problem, it estimates the cost of solution $S \in \mathcal{S}$ via implementing such a solution. However, in practice, the cost estimate of $S \in \mathcal{S}$ can be derived from those of $a \in S$ (see Section 8 for tweaked versions of UCB1 incorporating this fact). Next, we present a different approach aiming to ensure the exploration frequency prescribed in LR on a rather more direct fashion.

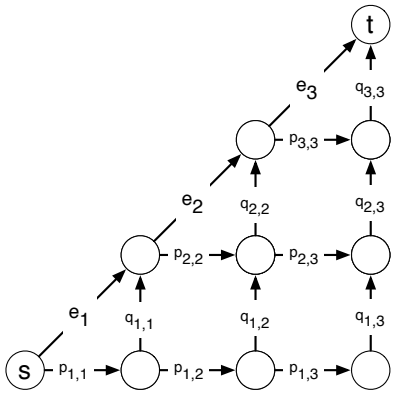

Figure 1: Graph for Example 4.1.

A simple policy. Let $\overline{B}_n := \left(\overline{b}_{a,n}, a \in A\right)$ be the average observed cost of ground elements before implementing a solution to instance n, where

$$\overline{b}_{a,n} := \frac{1}{T_n(a)} \sum_{m<n\,:\,a\in S_m} b_{a,m}. \tag{7}$$

Let $\mathcal{E}$ be a cover of A, i.e., $\mathcal{E} \subseteq \mathcal{S}$ such that each $a \in A$ belongs to at least one $S \in \mathcal{E}$. Note that implementing all $S \in \mathcal{E}$ provides feedback on the cost of every $a \in A$. Since N might not be known upfront, to induce the exploration frequency $\ln N/N$, we propose a policy that divides the time horizon into cycles with exponentially growing lengths. Each cycle consists of an exploration phase followed by an exploitation phase. Within the exploration phase of each cycle, the simple policy implements each solution $S \in \mathcal{E}$ *at most* once. After the exploration phase of each cycle, and if there is enough time to do so, the algorithm trusts the cost estimates and exploits any solution with minimum estimate cost until the end of that cycle. To formally describe this policy, for $i \in \mathbb{N}$ and $i \geq 2$, define the starting point of cycle i as

$$n_i := \max\left\{\lfloor e^{i/H} \rfloor, n_{i-1} + 1\right\},$$

where $n_1 = 1$ and the tuning parameter H is a positive finite constant that regulates the frequency of exploration. Define $\Phi := \{n_i : i \in \mathbb{N}\}$ to be the set of starting points of all cycles. The proposed simple policy, which we denote as $\pi_s(\mathcal{E})$, is summarized in Algorithm 1.

Remark 4.2. Note that reversing the order of the update of the exploitation set and the exploration phase in Algorithm 1 does not affect the performance bound in Theorem 4.3 (below). In practice, however, this leads to marginal improvements in performance.

Note that even if we take $\mathcal{E} = \mathcal{S}$, the regret associated with $\pi_s(\mathcal{S})$ should improve upon that of

Algorithm 1 Simple policy $\pi_s(\mathcal{E})$

Set $i = 0$, and $\mathcal{E}$ a minimal cover of A
for $n = 1$ to N **do**
  if $n \in \Phi$ **then**
    Set $i = i + 1$
    Set $S^* \in \mathcal{S}^*\left(\overline{B}_n\right)$ [Update exploitation set]
  end if
  if $T_n(a) < i$ for some $a \in S$, for some solution $S \in \mathcal{E}$ **then**
    Implement such a solution, i.e., set $S_n = S$ [Exploration]
  else
    Implement $S_n = S^*$ [Exploitation]
  end if
end for

traditional bandit algorithms: because cost estimation is conducted on elements of A, the simple policy implements at most $|A|$ solutions during the exploration phase of each cycle. Our first result provides a performance guarantee for $\pi_s(\mathcal{E})$ for any cover $\mathcal{E}$.

Theorem 4.3. *There exists a positive finite constant H_s such that for any $H \geq H_s$, $N > 0$, and cover $\mathcal{E}$, the regret of $\pi_s(\mathcal{E})$ admits the following bound*

$$\frac{R^{\pi_s(\mathcal{E})}(F, N)}{\ln N} \leq \min\left\{|\mathcal{E}|, |A|\right\} \Delta^F_{max} H + o(1),$$

where $\Delta^F_{max} := \max\left\{\Delta^F_S \,:\, S \in \mathcal{S}\right\}$.

Theorem 4.3 implies that policy $\pi_s(\mathcal{E})$ effectively conducts exploration on a cover contained in $\mathcal{E}$ that is minimal with respect to inclusion. In this regard, one should not consider any cover $\mathcal{E}$ with more than $|A|$ elements (e.g., one can select $\mathcal{E} = \{S_a\}_{a \in A}$ where $S_a \in \mathcal{S}$ is any solution such that $a \in S_a$)[2]. Thus, selecting small covers should result on a regret of Algorithm 1 proportional to at most $|A| \ln N$ in the worst case, and potentially much smaller in practice. For instance, in Example 4.1, $|A| = (k+2)(k+3)/2$, but we can easily construct a cover $\mathcal{E}$ of size $k+1$. Then, the regret of Algorithm 1 is at most order $k \ln N$, while the best estimate of the regret growth of, e.g., UCB1, is $\frac{4^{k+1}}{(k+1)^{3/2}\sqrt{\pi}} \ln N$.

Regarding the issue of when and what to explore, policy π_s (as well as most multi-armed bandit algorithms) applies LR's frequency $\ln N/N$, and explores every ground element in A. However, while classical bandit algorithms answer the question of how to explore by implementing all solutions in $\mathcal{S}$, Algorithm 1 implements a cover that is minimal with respect to inclusion. Ultimately, answers to the questions above are inconclusive unless contrasted to a fundamental lower bound on performance for the combinatorial bandit setting. We develop such a bound next.

[2]In fact, one could refine this policy to *adaptively* select $\mathcal{E}$ as the "cheapest" cover of A: see the discussion in Section 9.

5 What to Explore: A Limit on Achievable Performance

In this section we show that a consistent policy might not need to collect information on all elements of A. To illustrate this fact consider the following example.

Example 5.1. *Let $G = (V, A)$ be the digraph depicted in Figure 2 and let $\mathcal{S}$ be composed of all paths from node s to node t. Set $l_a = 0$ and $u_a = \infty$ for every arc $a \in A$, and F to be such that $\mathbb{E}_F\{b_{e,n}\} = c$, $\mathbb{E}_F\{b_{g,n}\} = 0$, $\mathbb{E}_F\{b_{f,n}\} = \mathbb{E}_F\{b_{h,n}\} = \frac{c+\varepsilon}{2}$, and $\mathbb{E}_F\{b_{p_i,n}\} = \mathbb{E}_F\{b_{q_i,n}\} = M$ for $n \in \mathbb{N}$ and for all $i \in \{1, \ldots, k\}$ where $0 < \varepsilon \ll c \ll M$. The shortest (expected) path in this digraph is (e).*

In Example 5.1, $|\mathcal{S}| = (k+2)$ and the smallest possible cover of A is $\mathcal{E} = \mathcal{S}$. Then both Algorithm 1 and UCB1 have regrets proportional to $k \ln N$ and hence Algorithm 1 does not seem to improve upon traditional algorithms. However, a careful analysis of this instance reveals that we do not need to collect information on all elements of A to identify the optimal solution (e), thus $\mathcal{E}$ in Algorithm 1 might not need to be a cover after all. Indeed, all paths except optimal path (e) share arcs $\{f, h\}$, whose expected costs suffice to guarantee the optimality of the shortest path. Then a possible answer to the issue of what to explore is just arcs f, h, and optimal arc/path e. This observation suggests that setting $\mathcal{E}$ to be path (e) plus any other path that contains both f and h will result on Algorithm 1 having a regret proportional to at most $M \ln N$ for every $k \in \mathbb{N}$ (note that path (e) does not add to the regret). Moreover, the "cheapest" way to explore f and h is by implementing path (f, g, h). Thus, setting $\mathcal{E} = \{(f, g, h), (e)\}$ in Algorithm 1 induces a regret proportional to at most $\varepsilon \ln N$ while UCB1 and Algorithm 1 with a cover have regrets proportional to $(kM + \varepsilon) \ln N$.

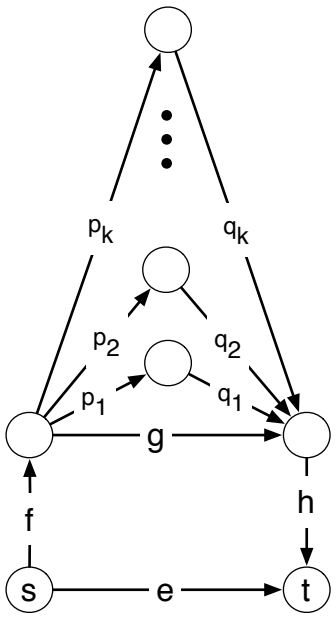

Figure 2: Graph for Example 5.1.

The analysis above shows that we may not need to explore every element of A. To understand exactly *what* needs to be explored, we extend the fundamental performance limit of LR for traditional multi-armed bandits to the combinatorial setting.

Following the argument in the traditional bandit setting, consistent policies must explore those subsets of ground elements that have a chance to be part of an optimal solution. We can formally define such subsets as follows, where for simplicity of exposition, we assume $\mathcal{S}^*(\mathbb{E}_F\{B_n\})$ is a singleton containing $S^*(\mathbb{E}_F\{B_n\})$. Let $\mathcal{D}$ contain all subsets D of suboptimal ground elements

such that they become part of every optimal solution if their costs are the lowest possible. That is,

$$\mathcal{D} := \left\{ D \subseteq A \setminus S^* \left(\mathbb{E}_F \left\{ B_n \right\} \right) : D \subseteq \bigcap_{S \in \mathcal{S}^*(B'_D)} S \right\}, \tag{8}$$

where $B'_D := (b'_a : a \in A)$ and

$$b'_a := \begin{cases} l_a & a \in D \\ \mathbb{E}_F \left\{ b_{a,n} \right\} & a \notin D. \end{cases}$$

By construction, any $D \in \mathcal{D}$ is such that there is an alternative distribution F_D under which all the elements of D are part of every optimal solution. The following proposition proves that for any $D \in \mathcal{D}$, the number of times that consistent policies implement solutions containing at least one ground element in D admits an asymptotic lower bound of the form $K_D \ln N$, where K_D corresponds to the inverse of Kullback-Leibler distance between the current (F) and the alternative (F_D) distributions.

Proposition 5.2. *For any consistent policy π, regular F, and $D \in \mathcal{D}$ we have that*

$$\lim_{N \to \infty} \mathbb{P}_F \left\{ \frac{\max \left\{ T_{N+1}(a) : a \in D \right\}}{\ln N} \geq K_D \right\} = 1, \tag{9}$$

where K_D is a positive finite constant depending on F.

While Theorem 3.2 imposes lower bounds on the number of times that a solution is implemented, Proposition 5.2 imposes similar bounds, but on the number of times that certain subsets of A are tried. To understand the difference, assume for now that $K_D = 1$ for every $D \in \mathcal{D}$, and that $K_a = 1$ for every $a \in A$: in this setting Theorem 3.2 says that every element in A needs to be explored with frequency $\ln N/N$; Proposition 5.2, on the other hand, says that the same holds, but for a set of ground elements that belong to $\mathcal{C} := \{C \subseteq A : \forall D \in \mathcal{D}, \exists\, a \in C \text{ s.t. } a \in D\}$. We refer to this family of subsets as the *Critical Subsets*. Considering the general case, the K_D's just change the precise frequency at which the elements of a critical subset need to be explored.

Proposition 5.2 gives an answer to *what* needs to be explored: a critical subset. However, such an answer is not conclusive as there might be multiple critical subsets. Moreover, transforming Proposition 5.2 into a limit on performance ultimately depends not only on the selection of the critical subset, but also on a cover of said selection. Nevertheless, choosing this cover and the critical subset can be done concurrently. For instance, assuming that the lower bounds of Proposition 5.2 are valid in the case of a finite N, we can find the minimum regret exploration set that satisfies the exploration conditions (9) by solving the following optimization problem, which we denote as

Integer Lower Bound Problem (ILBP).

$$
\begin{aligned}
ILBP: \quad z(F,N) := \min \quad & \sum_{S\in\mathcal{S}} \Delta_S^F \; y_S && \text{(10a)}\\
s.t. \quad & \max\{x_a : a \in D\} \geq K_D \ln N, \quad D \in \mathcal{D} && \text{(10b)}\\
& x_a \leq \sum_{S\in\mathcal{S}:a\in S} y_S, \quad a \in A && \text{(10c)}\\
& x_a,\, y_S \in \mathbb{N}_0, \quad a \in A, S \in \mathcal{S}. && \text{(10d)}
\end{aligned}
$$

In this formulation, y_S and x_a are meant to represent $T_{N+1}(S)$ and $T_{N+1}(a)$, respectively. Here, the a's with non-zero x_a correspond to the critical subset and the S's with non-zero y_S correspond to the cover of the critical subset. Indeed, constraints (10b) enforce exploration conditions (9) on the critical subset and constraints (10c) enforce the cover of the critical subset.

Intuitively, we expect the regret of any consistent policy to grow at least as fast as $z(F,N)$. To transform this intuition into a fundamental performance limit, we consider the following Lower Bound Problem (LBP), which characterizes the rate of growth of $z(F,N)/\ln N$.

$$
\begin{aligned}
LBP: \quad \kappa(F) := \min \quad & \sum_{S\in\mathcal{S}} \Delta_S^F \; y_S && \text{(11a)}\\
s.t. \quad & \max\{x_a : a \in D\} \geq K_D, \quad D \in \mathcal{D} && \text{(11b)}\\
& x_a \leq \sum_{S\in\mathcal{S}:a\in S} y_S, \quad a \in A && \text{(11c)}\\
& x_a,\, y_S \in \mathbb{R}_+, \quad a \in A, S \in \mathcal{S}. && \text{(11d)}
\end{aligned}
$$

Note that $\kappa(F)$ does not change if we refine $\mathcal{D}$ in (11b) to include only subsets that are minimal with respect to inclusion. One can show that

$$\lim_{N\to\infty} \frac{z(F,N)}{\ln N} = \kappa(F).$$

To see this, note that the continuous relaxation of formulation ILBP is homogeneous in N in the sense that $\tilde{z}(F,N) = \kappa(F)\ln N$, where $\tilde{z}(F,N)$ denotes the optimal objective value of continuous relaxation of formulation ILBP. In addition, note that $z(F,N) - \tilde{z}(F,N)$ is bounded above by a finite constant, which is independent of N. Thus

$$z(F,N) = \tilde{z}(F,N) + O(1) \quad N \in \mathbb{N}.$$

For any consistent policy π, define $\zeta^\pi(F,N) := \sum_{S\in\mathcal{S}} \Delta_S^F \; T_{N+1}(S)$ to be the total additional cost (relative to an oracle agent) associated with that policy. Note that $\mathbb{E}_F\{\zeta^\pi(F,N)\} = R^\pi(F,N)$. The next proposition combines the results above to establish an asymptotic result on the additional

cost incurred by any consistent policy.

Proposition 5.3. *For any consistent policy π and any regular F we have that*

$$\lim_{N\to\infty} \mathbb{P}_F\Big(\zeta^\pi(F,N) \geq \kappa(F)\ln N\Big) = 1.$$

Note that the result above essentially indicates convergence in probability (hence it can be used to bound $\zeta^\pi(F,N)$, rather than just its expectation, which is the regret). The next result, whose proof we omit since follows from a direct application of Proposition 5.3 and Markov's inequality, shows that the regret of any consistent policy in the combinatorial setting is proportional to $\kappa(F)\ln N$.

Theorem 5.4. *The regret of any consistent policy π is such that for any regular F we have*

$$\liminf_{N\to\infty} \frac{R^\pi(F,N)}{\ln N} \geq \kappa(F),$$

where $\kappa(F)$ is the optimal objective value of formulation LBP in (11)*.*

In the next section we will use the intuition behind Theorem 5.4 to develop an algorithm that successfully answers all three fundamental questions regarding the exploration vs. exploitation trade-off (when, what, and how to explore) in the combinatorial setting.

6 How to Explore Revisited: The Optimality Cover Problem

In this section we propose a policy that incorporates the insight gained from the results in the previous sections. Such a policy builds on the structure of the simple policy $\pi_s(\cdot)$ and improves its performance by adapting the exploration set $\mathcal{E}$ dynamically over time. In particular, exploration is focused on the solution to a proxy for formulation (11), which is reconstructed and solved at the frequency prescribed by both Theorems 3.2 and 5.4. We begin by describing the approximation to LBP we use; then, we spell out the details of the proposed policy and its associated performance guarantee.

Optimality cover problem. As pointed out in the discussion following Proposition 5.2, the constant K_D in (11b) has the effect of adjusting the precise frequency at which the elements of the critical subset need to be explored. Following the spirit of the simple policy, we focus on an approximation to LBP that imposes the same rate of exploration across all relevant solutions[3]. This approximation focuses on the idea of recovering essentially optimal policies for the traditional bandit setting, which is aligned with the focus of this paper on improving upon the performance of traditional algorithms.

[3]One could refine our policy by adjusting exploration frequencies over time. See the discussion in Section 9.

Consider a special case of formulation (11) where $K_D = H$ for all $D \in \mathcal{D}$: one can show that $\kappa(F)$ is homogeneous in H. Thus, without loss of generality, one can take $H = 1$ and interpret LBP as the problem of finding a set of solutions with minimum regret that covers at least one critical subset $C \in \mathcal{C}$. For a given cost-coefficient vector B, such a formulation, which we denote as the *Optimality Cover Problem* (henceforth, OCP), solves for a minimum additional cost solution set whose feedback suffices to guarantee the optimality of $\mathcal{S}^*(B)$.

$$
\begin{aligned}
OCP(B): \quad \min \quad & \sum_{S \in \mathcal{S}} \Delta_S^F(B)\, y_S && \text{(12a)}\\
s.t. \quad & x_a \leq \sum_{S \in \mathcal{S}: a \in S} y_S, \quad a \in A && \text{(12b)}\\
& \sum_{a \in S} \left(l_a(1 - x_a) + b_a x_a\right) \geq z^*(B), \quad S \in \mathcal{S} && \text{(12c)}\\
& x_a,\, y_S \in \{0, 1\}, \quad a \in A, S \in \mathcal{S}, && \text{(12d)}
\end{aligned}
$$

where with a slight abuse of notation, we make the dependence of Δ_S^F on B explicit. By construction, a feasible solution (x, y) to this problem corresponds to incidence vectors of a set $C \subseteq A$ and a cover $\mathcal{E}$ of such a set[4]. In what follows we refer to a solution (x, y) to OCP and the induced pair of sets $(C, \mathcal{E})$ interchangeably.

Constraints (12c) guarantee the optimality of $\mathcal{S}^*(B)$ even if costs of elements outside C are set to their lowest possible values (i.e., $b_a = l_a$ for all $a \notin C$), and constraints (12b) guarantee that $\mathcal{E}$ covers C (i.e., $a \in S$ for some $S \in \mathcal{E}$, for all $a \in C$). One can show that the former constraints imply that C is in fact a critical subset, i.e., $C \in \mathcal{C}$. On the other hand, while the converse does not hold in general (i.e., not all incidence vectors of critical subsets satisfy (12c)), all critical subsets covering optimal elements of A do satisfy (12c). In addition, note that when solving (11), one can impose $y_S = 1$ for all $S \in \mathcal{S}^*(\mathbb{E}_F\{B_n\})$ without affecting the objective function, thus one can restrict attention to critical subsets that cover optimal elements of A.

The above suggests a close connection between solutions to OCP and a restricted class of solutions to LBP. In particular, it suggests that optimal solutions to OCP are also optimal to LBP. The next Lemma formalizes this result.

Lemma 6.1. *Let $K_D = 1$ for all $D \in \mathcal{D}$ in formulation LBP. An optimal solution (x, y) to $OCP(\mathbb{E}_F\{B_n\})$ is also optimal to LBP.*

The connection between these formulations goes beyond that indicated in Lemma 6.1. When $K_D = 1$ for all $D \in \mathcal{D}$, one can always select integral optimal solutions to (11), so that they can be mapped into feasible solutions to OCP (via proper augmentation), and that the opposite holds true as well. Thus, one can argue that these formulations are essentially equivalent. The key

[4]That is, $(x, y) := \left(x^C, y^{\mathcal{E}}\right)$ where $x_a^C = 1$ if $a \in C$ and zero otherwise and $y_S^{\mathcal{E}} = 1$ if $S \in \mathcal{E}$ and zero otherwise.

difference between LBP and OCP is that optimal solutions to OCP *must* cover all optimal ground elements. It turns out that this property is key to implementing the policy we propose below. To see why, consider an optimal solution to OCP: based solely on the feedback from implementing such a solution, one can guarantee that said solution is indeed feasible to OCP; however, such information might not suffice to guarantee the optimality of the solution. In practical terms, this implies that a policy that focuses on minimizing exploration costs by redirecting exploration efforts according to OCP might never be sure that such an exploration cost is in fact minimal. However, such an approach will still provide sufficient information to guarantee optimality during exploitation.

An adaptive policy. The proposed adaptive policy follows the structure of the simple policy: time horizon is divided into cycles with exponentially growing lengths, each composed of an exploration phased followed by an exploitation phase. However, unlike the simple policy, the exploration set $\mathcal{E}$ is recomputed at the beginning of each cycle i and equaled to the solution to $OCP(\bar{B}_{n_i})$. Regarding exploration frequencies, note that we should aim to collect information on the implied critical subset, rather than on every element of A. Thus, our policy enforces the $\ln N/N$ exploration frequency on elements $a \in C$ for a feasible solution $(C, \mathcal{E})$ to $OCP(\bar{B}_{n_i})$ that is minimal with respect to inclusion (this choice will become apparent after analyzing the performance of the algorithm). In particular, we aim to have elements in C tried at least i times before "exploiting" in cycle i.

To formalize the above, recall the definition of cycles: for $i \in \mathbb{N}$ and $i \geq 2$, define the starting point of cycle i as

$$n_i := \max\left\{\lfloor e^{i/H} \rfloor, n_{i-1} + 1\right\},$$

where $n_1 = 1$ and H is a positive finite constant that regulates the frequency of exploration. As before, let $\Phi := \{n_i : i \in \mathbb{N}\}$ be the set of starting points of all cycles. Also, let $\Gamma(B)$ denote the set of feasible solutions $(C, \mathcal{E})$ to OCP such that both C and $\mathcal{E}$ are minimal with respect to inclusion, and let $\Gamma^*(B)$ denote the set of optimal solutions to $OCP(B)$. The proposed adaptive policy, which we denote as π_a is summarized in Algorithm 2.

The following result provides a theoretical upper bound on the performance of policy π_a.

Theorem 6.2. *There exists a positive finite constant H_a such that for any $H \geq H_a$ and $N > 0$, the regret of π_a admits the following bound*

$$\frac{R^{\pi_a}(F, N)}{\ln N} \leq G\ \Delta_{max}^{F} H + o(1),$$

where $G := \max\{|\mathcal{E}| : (C, \mathcal{E}) \in \Gamma(\mathbb{E}_F\{B_n\})\}$.

Note that $G \leq \max\{|\mathcal{E}| : \mathcal{E} \text{ is a minimal cover of A}\} \leq |A|$, so the performance bound above is at least as good as that in Theorem 4.3. However, as illustrated in Example 5.1, G can be arbitrarily smaller than even the smallest cover (in Example 5.1, the smallest cover has size $k + 2$,

Algorithm 2 Adaptive policy π_a

Set $i = 0$, $C = A$, and $\mathcal{E}$ a minimal cover of A
for $n = 1$ to N **do**
  if $n \in \Phi$ **then**
    Set $i = i + 1$
    Set $S^* \in \mathcal{S}^*\left(\overline{B}_n\right)$ [Update exploitation set]
    if $(C, \mathcal{E}) \notin \Gamma(\overline{B}_n)$ **then**
      Set $(C, \mathcal{E}) \in \Gamma^*(\overline{B}_n)$ [Update exploration set]
    end if
  end if
  if $T_n(a) < i$ for some $a \in C$ **then**
    Try such an element, i.e., set $S_n = S$ with $S \in \mathcal{E}$ such that $a \in S$ [Exploration]
  else
    Implement $S_n = S^*$ [Exploitation]
  end if
end for

but $G = 2$).

Comparing the bound above with the lower bound in Theorem 5.4, one notes a gap in performance. Such a mismatch emanates from various sources: some pertain our lower bound, and arise from using formulation (11) to construct a valid lower bound; others pertain the upper bound, and arise from: (i) adopting OCP as a valid proxy for formulation (11); and (ii) the inability of the proposed policy to consistently reconstruct the optimal solution to OCP. In Section 9.2 we provide a further analysis of the gap between the bounds in Theorems 5.4 and 6.2, and discuss means to tighten the alluded mismatch.

7 Computational Issues

Policy $\pi_s(\cdot)$ as well as π_a need to solve the underlying combinatorial optimization problem $f(B)$ repeatedly, for many inputs B. Thus, tractability of said problems is essential for practical implementation. For this reason, we now consider some aspects concerning the potential implementability of the proposed policies. In particular, we provide strong evidence suggesting that, at least for a large class of combinatorial problems, the proposed policies scale reasonably well and should be implementable for real-size instances.

The optimization problem $f(B)$ has a generic combinatorial structure, and thus it could be extremely hard to solve. For this reason, we concentrate on two broad classes of combinatorial optimization problems that have reasonably effective solution algorithms. The first class corresponds to theoretically tractable problems that have polynomial time algorithms. Among this class, we are especially interested in those that have Linear Programming (LP) formulations such as shortest path, network flow, matching, and spanning tree problems (Schrijver 2003). The second

class corresponds to those that are not expected to have polynomial time algorithms, but can frequently be solved effectively in practice. Among this class, we are particularly interested in medium sized instances of NP-complete problems (Cook et al. 1998) with effective Integer Programming (IP) formulations. These problems can usually be effectively solved by specialized or general purpose IP solvers and include the traveling salesman problem (Applegate et al. 2011), Steiner tree (Magnanti and Wolsey 1995, Koch and Martin 1998, Carvajal et al. 2013), and set cover problems (Etcheberry 1977, Hoffman and Padberg 1993, Balas and Carrera 1996).

Now, in addition to solving $f(B)$, implementing Algorithm 2 requires solving $OCP(B)$, also for many inputs B. Unfortunately, even if $f(B)$ is tractable, it is not clear whether $OCP(B)$ is tractable or not. Indeed, even if $f(B)$ is in P, the class of polynomially solvable problems, the number of variables and constraints of $OCP(B)$ might be exponential in $|A|$ [5] (that is the case in Example 4.1, where $f(\cdot)$ corresponds to a shortest path formulation, which is in P), thus $OCP(B)$ could even fail to be in NP, the class of non-deterministic polynomially solvable problems (see e.g., Cook et al. (1998)). Fortunately, by using the fact that $\Gamma^*(B) \subseteq \Gamma(B)$ (i.e., every optimal solution to OCP is minimal) and that solutions in $\Gamma(B)$ have polynomially bounded sizes, we can show the following complexity result.

Theorem 7.1. *If $f(B)$ is in P, then $OCP(B)$ is in NP.*

Determining the precise complexity relation between $f(B)$ and $OCP(B)$ is beyond the scope of this paper. However, we do have evidence suggesting that for some specific problems there could be a non-trivial jump in complexity (e.g., for $f(B)$ in P, $OCP(B)$ could be NP-complete). For this reason, we now consider the practical implementability of policies $\pi_s(\cdot)$ and π_a. In particular, we explore two characteristics of these policies that show they should be implementable in practice for a wide range of combinatorial optimization problems. The first characteristic is the existence of IP formulations for $OCP(B)$, including polynomial sized formulations, for any combinatorial optimization with an LP formulation for $f(B)$. These formulations should be effectively tackled by a state of the art Branch-and-Cut algorithm (Nemhauser and Wolsey 1988, Cook 2010, Mitchell 2011). The second characteristic is the fact that the frequency at which $f(B)$ and $OCP(B)$ are solved by these policies decreases exponentially.

[5] In most cases the natural size of $f(B)$ is $O(|A| + \sum_{a \in A:|l_a|<\infty} \ln_2 l_a + \sum_{a \in A:|u_a|<\infty} \ln_2 u_a)$, and l_a and u_a are usually bounded. For instance, in the class of shortest path problems in Example 4.1, the natural size of the considered graph is the number of arcs $|A| = O(k^2)$ (not the number of paths) and the bounds of all arc lengths are constant. If we instead had non-constant finite upper bounds u_a (with $u_a > 0.03$ for the analysis to remain valid), we would also include the number of bits needed to encode them, which is equal to $\sum_{a \in A} \ln_2 u_a$. We refer the interested reader to Schrijver (2003) for more information on the input sizes of combinatorial optimization problems.

7.1 Integer Programming Formulations for OCP

Independent of the theoretical complexity of OCP, formulation (12) of OCP is not amenable to practical solutions; it is an IP with an exponential number of variables and constraints that would likely need to be solved through an intricate Branch-and-Cut-and-Price procedure. However, using simple IP techniques, we can reformulate (12) to obtain a formulation with a polynomial number of variables. While this formulation still has an exponential number of constraints, such problems are regularly solved in practice using effective Branch-and-Cut procedures (Applegate et al. 2011, Carvajal et al. 2013). If $f(B)$ has an LP formulation we can further reduce the size to a polynomial number of variables and constraints. This version can then be solved directly by a general purpose state of the art IP solver such as CPLEX (IBM ILOG n.d.). Finally, by exploiting the specific structure of $f(B)$, we can construct even more effective formulations for OCP. To describe these formulations it is convenient to use the following notation.

Definition 7.2. *Let I be an arbitrary finite set and $x \in \{0,1\}^{|I|}$. We let the support of x be* $\operatorname{supp}(x) := \{i \in I : x_i = 1\}$.

The following proposition introduces our first IP formulation for OCP that can be applied to any combinatorial optimization problem with an IP formulation.

Proposition 7.3. *Let y^S be the incidence vector of $S \in \mathcal{S}$, $M \in \mathbb{R}^{m \times |A|}$, and $d \in \mathbb{R}^m$ be such that $\{y^S\}_{S \in \mathcal{S}} = \left\{y \in \{0,1\}^{|A|} : My \leq d\right\}$. Then an IP formulation of $OCP(B)$ is given by*

$$\min \quad \sum_{i=1}^{|A|} \left(\sum_{a \in A} b_a y_a^i - z^*(B) \right) \tag{13a}$$

$$s.t. \qquad x_a \leq \sum_{i=1}^{|A|} y_a^i, \quad a \in A \tag{13b}$$

$$My^i \leq d, \qquad i \in \{1, \ldots, |A|\} \tag{13c}$$

$$\sum_{a \in S} (l_a(1 - x_a) + b_a x_a) \geq z^*(B), \quad S \in \mathcal{S} \tag{13d}$$

$$x_a,\, y_a^i \in \{0,1\}, \quad a \in A, i \in \{1, \ldots, |A|\}. \tag{13e}$$

That is, if (x, y) is an optimal solution to (13)*, then $(C, \mathcal{E})$ is an optimal solution to $OCP(B)$, where $C = \operatorname{supp}(x)$ and $\mathcal{E} = \left\{\operatorname{supp}\left(y^i\right)\right\}_{i=1}^{|A|}$. Furthermore, for a given $x \in \{0,1\}^{|A|}$, finding a violated inequality* (13d) *or showing that it satisfies all these inequalities can be done by solving $f(B')$ for $b'_a = l_a(1 - x_a) + b_a x_a$.*

Proof. For any feasible solution (x, y) to (13) we have that x is the incidence vector of a critical subset from (13d). Similarly, we have that any y^i is the incidence vector of some $S \in \mathcal{S}$ because of

(13c) and the assumptions on M and d. Finally, (13b) ensures that $\mathcal{E} = \left\{\text{supp}\left(y^i\right)\right\}_{i=1}^{|A|}$ covers the critical subset. This covering can always be achieved by including a sufficient number of variables y^i. However, it is likely that fewer elements of $\mathcal{S}$ are sufficient to cover the critical subsets, which suggests that formulation (13) could lead to inefficient covers. Fortunately, if less than $|A|$ elements are needed for the cover, the optimization can pick the additional y^i to be the incidence vector of an optimal solution to $f(B)$ so that they do not increase the objective function value.

The result concerning (13d) follows from the fact that $x \in \{0,1\}^{|A|}$ satisfies this constraint if and only if $\min_{S\in\mathcal{S}} \left\{\sum_{a\in S} \left(l_a(1-x_a) + b_a x_a\right)\right\} \geq z^*(B)$, which is equivalent to the optimal value of $f(B')$ being greater than $z^*(B)$.

□

Formulation (13) has a polynomial number of variables, but the number of constraints described by (13d) is in general exponential. However, the computational burden of separating these constraints is the same as solving $f(B)$. Hence, if we can solve $f(B)$ sufficiently fast we should be able to effectively solve (13) with a Branch-and-Cut algorithm that dynamically adds constraints (13d) as needed . The most effective Branch-and-Cut algorithms are obtained when finding a violated inequality can be done in polynomial time, which is a common assumption in most applications of Branch-and-Cut. However, an effective algorithm can still be obtained if finding a violated inequality is NP-hard, but can be obtained sufficiently fast (e.g., using a state of the art IP solver).

While having $f(B)$ be polynomially solvable should improve the performance of a Branch-and-Cut algorithm, if $f(B)$ additionally has an LP formulation, we can actually construct an IP formulation of $OCP(B)$ with a polynomial number of variables and constraints as follows.

Proposition 7.4. *Let y^S be the incidence vector of $S \in \mathcal{S}$, $M \in \mathbb{R}^{m\times|A|}$, and $d \in \mathbb{R}^m$ be such that $\left\{y^S\right\}_{S\in\mathcal{S}} = \left\{y \in \{0,1\}^{|A|} : My \leq d\right\}$ and $\text{conv}\left(\left\{y^S\right\}_{S\in\mathcal{S}}\right) = \left\{y \in [0,1]^{|A|} : My \leq d\right\}$. Then an IP formulation of $OCP(B)$ is given by*

$$\min \quad \sum_{i=1}^{|A|}\left(\sum_{a\in A} b_a y_a^i - z^*(B)\right) \tag{14a}$$

$$s.t. \qquad x_a \leq \sum_{i=1}^{|A|} y_a^i, \qquad a \in A \tag{14b}$$

$$My^i \leq d, \qquad i \in \{1,\ldots,|A|\} \tag{14c}$$

$$M^T w \leq \text{diag}(l)(\mathbf{1} - x) + \text{diag}(b)x \tag{14d}$$

$$d^T w \geq z^*(B) \tag{14e}$$

$$x_a,\, y_a^i \in \{0,1\},\, w \in \mathbb{R}^m, \qquad a \in A, i \in \{1,\ldots,|A|\}, \tag{14f}$$

where for $v \in \mathbb{R}^r$, $\text{diag}(v)$ is the $r \times r$ diagonal matrix with v as its diagonal.

Proof. The only difference between formulations (14) and (13) is that instead of explicitly enforcing criticality through (13d), formulation (14) does so through strong duality. Indeed, (14d) ensures that w is a dual feasible solution to the LP formulation of $f(B)$ and (14e) forces the objective of this solution to be greater than $z^*(B)$. With this, we have that the optimal solution to $f(B')$ for $B' = \text{diag}(l)(\mathbf{1} - x) + \text{diag}(b)x$ is greater than or equal to $z^*(B)$. □

Formulation (14) has $O(|A|^2)$ variables and $O(m|A|)$ constraints. If m is polynomial on the size of the input of $f(B)$, then we should be able to solve (14) directly with a state of the art IP solver. If m is exponential, but the constraints in the LP formulation can be separated effectively, we should still be able to solve (14) with a Branch-and-Cut algorithm. A standard example of this class of problems is the spanning tree problem. However, in the case of spanning trees, we can additionally use a known polynomial sized extended formulation of the form $\left\{x \in \{0,1\}^{|A|} : \exists y \in \mathbb{R}^p, \quad Cx + Dy \leq d\right\}$ (Martin 1991). Similar techniques can be used to construct polynomial sized formulations for other problems and to further improve formulation (14). We now show how such techniques can be used to construct a linear sized formulation of $OCP(B)$ for shortest path problems. For more information on advanced IP formulation techniques, we refer the interested reader to Vielma (2012).

Proposition 7.5. *Let $f(B)$ correspond to a shortest $s-t$ path problem in a digraph $G = (V, A)$. Define $\hat{A} = A \cup \{(t,s)\}$ and let $\hat{\delta}_{out}$ and $\hat{\delta}_{in}$ denote the outbound and inbound arcs in digraph $\hat{G} = (V, \hat{A})$. An optimal solution (x, p, w) to*

$$\min \quad \left(\sum_{a \in A} b_a p_a\right) - z^*(B)\ p_{(t,s)} \tag{15a}$$

$$s.t. \qquad x_a \leq p_a, \qquad a \in A \tag{15b}$$

$$\sum_{a \in \hat{\delta}_{out}(v)} p_a - \sum_{a \in \hat{\delta}_{in}(v)} p_a = 0, \qquad v \in V \tag{15c}$$

$$w_u - w_v \leq l_{(u,v)}(1 - x_{(u,v)}) + b_{(u,v)}x_{(u,v)}, \quad (u,v) \in A \tag{15d}$$

$$w_s - w_t \geq z^*(B) \tag{15e}$$

$$p_a \in \mathbb{Z}_+, \qquad a \in \hat{A} \tag{15f}$$

$$x_a \in \{0,1\},\ w_v \in \mathbb{R}, \qquad a \in A,\ v \in V, \tag{15g}$$

is such that $(C, \mathcal{E})$ is an optimal solution to $OCP(B)$, where $C = \{a \in A : x_a = 1\}$ and $\mathcal{E} \subseteq \mathcal{S}$ is a set of paths for which $p_a = |\{S \in \mathcal{E} : a \in S\}|$. Such a set $\mathcal{E}$ can be constructed from p in time $O(|A||V|)$.

Proof. The first difference between formulations (15) and (14) is the specialization of the LP duality constraints to the shortest path setting. The second one is the fact that the paths in cover $\mathcal{E}$

are aggregated into an integer circulation in augmented graph $\hat{G}$, which is encoded in variables p. Indeed, using known properties of circulations (Schrijver 2003, pp. 170-171), we have that $p = \sum_{S \in \mathcal{E}} y^{\hat{S}}$, where $y^{\hat{S}}$ is the incidence vector of the circulation obtained by adding (t, s) to path S. Furthermore, given a feasible p we can recover the paths in $\mathcal{E}$ in time $O(|A||V|)$. □

It is possible to construct similar formulations for other problems with the well known integer decomposition property (Schrijver 2003).

7.2 Practical Implementability and Solution Frequency

Propositions 7.3, 7.4, and 7.5 show that we should be able to solve $OCP(B)$ for problems $f(B)$ with effective IP or LP formulations. However, the solution times for $OCP(B)$, or even $f(B)$, could still be longer than the time available in between successive solution implementations. Fortunately, a key feature of the proposed policies is that the frequency at which either of these problems need to be solved decreases exponentially. Indeed, the proposed policies solve $f(B)$ and $OCP(B)$ only at the beginning of each cycle and the length of cycle n is $\Theta(\exp(n/H))$. Hence, the cycle length will grow exponentially and thus provide enough time to solve both $f(B)$ and $OCP(B)$. As described in Algorithms 1 and 2, the policies cannot proceed until the corresponding problems are solved, but they can easily be modified to a more asynchronous setting where they begin solving $f(B)$ and/or $OCP(B)$ at the beginning of a cycle, but continue to implement solutions while these problems are solved. If n is large enough, both problems will be solved by the end of the cycle. Similarly, we can easily modify the algorithms to implement alternate exploitation and exploration solutions in the transient period when the cycle lengths are not long enough. Algorithm 3 presents one possible time-constrained asynchronous modification that can be implemented in real time.

Algorithm 3 essentially applies the simple policy in the transient period and eventually implements the adaptive policy with one cycle delay (once the cycles are long enough, the policy essentially implements the exploration and exploitation solutions that the adaptive policy would have implemented in the previous cycle). This one cycle delay does not affect the asymptotic analysis of the policy and hence the performance guaranty of the adaptive policy is preserved. In addition, the short-term performance of this policy should be similar to that of the simple policy.

8 Numerical Experiments

In this section we illustrate our results via simple numerical experiments. The focus here is on illustrating the ability of the proposed policies to leverage the combinatorial structure of our setting to improve upon the performance of relevant benchmarks. The numerical experiments are divided into two general settings: long-term and short-term experiments. We compare the performance

Algorithm 3 Basic Time-Constrained Asynchronous Policy

Set $i = 0$, $C = A$, and $\mathcal{E}$ a minimal cover of A
Let $S^* \in \mathcal{S}$ be an arbitrary solution and $B_f = B_{OCP}$ be an initial cost estimate
Asynchronously begin solving $f(B_f)$ and $OCP(B_{OCP})$
for $n = 1$ to N **do**
  if $n \in \Phi$ **then**
    Set $i = i + 1$
    if Asynchronous solution to $f(B_f)$ has finished **then**
      Set $S^* \in \mathcal{S}^*(B_f)$ [Update exploitation set]
      Set $B_f = \overline{B}_n$
      Asynchronously begin solving $f(B_f)$
    end if
    if Asynchronous solution to $OCP(B_{OCP})$ has finished **then**
      if $(C, \mathcal{E}) \notin \Gamma(B_{OCP})$ **then**
        Set $(C, \mathcal{E}) \in \Gamma^*(B_{OCP})$ [Update exploration set]
      end if
      Set $B_{OCP} = \overline{B}_n$
      Asynchronously begin solving $OCP(B_{OCP})$
    end if
  end if
  if $T_n(a) < i$ for some $a \in C$ **then**
    Try such an element, i.e., set $S_n = S$ with $S \in \mathcal{E}$ such that $a \in S$ [Exploration]
  else
    Implement $S_n = S^*$ [Exploitation]
  end if
end for

of the proposed policies with those of the relevant benchmarks in each of such settings. We first discuss the long-term experiments by describing the benchmark policies and illustrating the results. Later on, we discuss the short-term experiments. In both cases we consider shortest path, set cover, Steiner tree, and knapsack problems.

8.1 Long-term Experiments

The long-term experiments consist of the Examples 4.1 and 5.1 together with representative settings of the shortest path, set cover, Steiner tree, and knapsack problems. We discuss the settings and results in section 8.1.3.

8.1.1 Benchmark Policies

Our benchmark policies are several versions of UCB1, adapted to provide an improved performance on our specific settings. Remember that UCB1 implements a solution

$$S_n \in \operatorname*{argmin}_{S \in \mathcal{S}} \{b(S,n) - k(S,n)\}$$

for instance n, where $b(S,n) = \bar{b}_{S,n}$ and $k(S,n) = \sqrt{2\ln(n-1)/T_n(S)}$. Estimate $b(S,n)$ is meant to be the "best" estimate of the expected cost of S and $k(S,n)$ is a correction factor that penalizes the lack of periodic refreshment of the expected cost estimate. With this interpretation, there are a few straightforward improvements to UCB1 for the combinatorial setting.

A first improvement comes from using a more recent expected cost estimate ($\bar{b}_{S,n}$ only uses information obtained when S is implemented): one can use the estimate

$$b(S,n) = \sum_{a \in S} \bar{b}_{a,n},$$

with $\bar{b}_{a,n}$ defined in (7). A second improvement is to adjust the penalty factor to reflect the "right" amount of information available, that is, one can use $k(S,n) = \sqrt{2\ln(n-1)/(\min_{a \in S}\{T_n(a)\})}$. A third improvement follows from realizing that costs cannot be below their lower bounds. In particular, the cost of a solution, as well as its index, should not go below the implied bound. A policy combining all the above implements

$$S_n \in \operatorname*{argmin}_{S \in \mathcal{S}} \left\{ \max \left\{ \sum_{a \in S} \bar{b}_{a,n} - \sqrt{2\ln(n-1)/(\min_{a \in S}\{T_n(a)\})}, \sum_{a \in S} l_a \right\} \right\}$$

for instance n. We denote this policy as UCB1+.

In a similar setting, Gai et al. (2012) propose another adaptation of UCB1: an improved version of such a policy implements

$$S_n \in \operatorname*{argmin}_{S \in \mathcal{S}} \left\{ \sum_{a \in S} \max \left\{ \bar{b}_{a,n} - \sqrt{(L+1)\ln(n-1)/T_n(a)}, l_a \right\} \right\}$$

for instance n, for some positive finite constant L. We denote this policy as Extended UCB1+.

While the theoretical performance guarantees of UCB1 and the policy in Gai et al. (2012) compare unfavorably to those of π_s and π_a, UCB1+ and Extended UCB1+ could perform closer to π_s or π_a in practice. In particular, UCB1+ incorporates some ideas used in crafting $\pi_s(\cdot)$ and hence could have a comparable performance.

8.1.2 Implementation Details

We considered several cost distributions and observed consistent performances. Here we report the results associated with exponential distributions.

Tuning parameters. We initialize every policy by implementing the solutions in a minimum size cover of A once (this might be seen as an initial improvement to the benchmark policies). The same cover is used to initialize different policies. Policies $\pi_s(\cdot)$ and π_a receive a tuning parameter H, which must be above some threshold to guarantee the performances in Theorems 4.3 and 6.2. In the preliminary tests we experimented with $H \in \{5, 10, 20\}$ and observed that all consistently resulted on logarithmic regrets. Here we report the results using $H = 10$.

Policy $\pi_s(\cdot)$ requires an additional input, a cover $\mathcal{E}$ of A: we report the results based on arbitrarily selected minimum size covers. We also improve the performance of our adaptive policy by letting it update the exploration set on each cycle.

Finally, we set $L = 1$ in Extended UCB1+, as this selection outperformed the recommendation in Gai et al. (2012), and also is the natural choice for extending the UCB1 policy.

Computational Setting. Our results track the performance of each policy on each setting considering $N = 2000$. We report on the average behavior over 100 replications. On each of the figures in this section, dotted lines represent 95% confidence intervals on our results. All policies were implemented in MATLAB R2011b. Shortest path problems were solved using Dijkstra's algorithm except when implementing UCB1+ (note that because of the index computation, $f(\cdot)$ must be solved by enumeration). Set cover, Steiner tree, and knapsack problems were solved by their integer programs using GUROBI 5.0 Optimizer. The adaptive policy solves OCP using GUROBI 5.0 Optimizer using the formulation (12). We ran all experiments on a machine with Intel(R) Xeon(R) 2.80GHz CPU and 16GB Memory. The average running time for a single replication is less than 4 seconds for simple policy and around 50 seconds for the others.[6]

8.1.3 Settings and Results

We begin by considering the shortest path problems in Examples 4.1 and 5.1 for $k = 3$ and $k = 20$, respectively. These problems were initially introduced to illustrate different aspects of efficient information collection and hence are somewhat biased in favor of the adaptive policy. For this reason, we also test our policies on randomly generated settings whose design is not meant to favor any particular policy. We considered four classes of combinatorial optimization problems: shortest path, set cover, Steiner tree, and knapsack. For each class of problems, we generated several settings, but did not find a setting where the benchmarks outperform the adaptive policy. Here we only show one representative from each class. All of these representatives are complementary

[6]Note, however, that while the running times of simple and adaptive policies grow (roughly) logarithmically with the horizon, those of UCB1+ and Extended UCB1+ grow linearly.

to Examples 4.1 and 5.1 in that the optimal critical subsets are large and hence the adaptive policy does not have an immediate advantage. Finally, in order for all policies to be consistent, we normalize the mean costs of the ground elements so that the maximum solution cost is at most one (see the consistency argument before Theorem 3.2).

Examples 4.1 and 5.1. Figures 3-(a) and 3-(b) depict the average performance of four different policies on Examples 4.1 and 5.1, respectively.

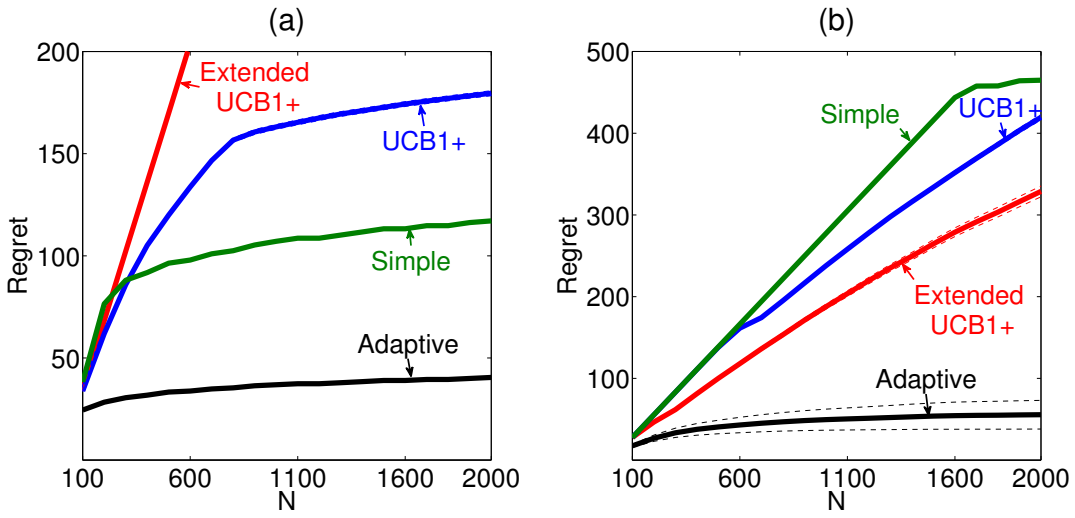

Figure 3: Graphs (a) and (b) depict the average performance of different policies on Examples 4.1 and 5.1, respectively.

On Example 4.1, we see that Extended UCB1+ has the worst performance followed by UCB1+. The simple policy performs better than both these policies, aided in part by the fact that it restricts exploration to a minimum size cover, which for this setting is only of size 4. Finally, the adaptive policy performs significantly better than the other policies, as it successfully limits exploration to a minimum regret exploration set (i.e., the implied subset of solutions from an optimal solution to $OCP(\mathbb{E}_F\{B_n\})$).

On Example 5.1, Extended UCB1+ outperforms the UCB1+ as well as the simple policy. The performance of the simple policy is likely hindered by the fact that, in this setting, the minimum size cover is equal to $\mathcal{S}$, which has size 22. In contrast, minimum regret exploration set of the adaptive policy is only of size 2, which helps it achieve the best performance.[7]

In terms of efficient information collection, one can divide the set of ground elements (arcs) into

[7]In our experiments, on both Examples 4.1 and 5.1, the adaptive policy selects a minimum regret exploration set practically on every replication in the long run.

three classes: the ones that are part of the optimal solution, the ones that can provide efficient exploration (i.e., the ones that belong to at lease one optimal solution to $OCP(\mathbb{E}_F\{B_n\})$), and the remaining ones which we denote *uninformative* as they do not need to be explored. Table 1 shows the average number of trials of an arc in each of such classes up to horizon $N = 2000$ over different policies. We note that the adaptive policy spends significantly less time exploring uninformative

	Example 4.1				Example 5.1			
	Adaptive	Simple	UCB1+	Extended UCB1+	Adaptive	Simple	UCB1+	Extended UCB1+
Optimal Arcs	1920.67	1770.37	1647.83	665.81	1787.50	395.49	472.84	867.70
Exploration Arcs	860.91	846.19	872.48	897.41	737.49	1201.50	1175.75	1044.10
Uninformative Arcs	26.45	95.69	118.65	465.09	9.36	76.25	69.77	53.91

Table 1: Average number of trials of different arcs up to horizon $N = 2000$ over different policies on Examples 4.1 and 5.1.

arcs.

Shortest path problem. We consider a shortest path problem on a randomly generated layered graph similar to that considered in Ryzhov and Powell (2011). This graph consists of a source node, a destination node, and 5 layers in between, each containing 4 nodes. In each layer, every node (but those in the last layer) is connected to 3 randomly chosen nodes in the next layer. The source node is connected to every node in the first layer and every node in the last layer is connected to the destination node. With this, the resulting graph is such that $|A| = 56$ and $|\mathcal{S}| = 324$. The expected arc costs are selected uniformly randomly from the set $\{0.1, 0.2, \ldots, 1\}$ and then normalized. An interesting property of the representative setting is that, while the minimum size cover of A is of size 13, the minimum regret exploration set is of size 16 even though the implied critical subset has 40 arcs. Figure 4 depicts the average performance of four different policies on this setting.

The adaptive policy outperforms the others. Moreover, the benchmark policies are outperformed by the simple policy.

Set cover problem. In the set cover problem we are given a universe set and a family A of subsets of the universe set. The solution set $\mathcal{S}$ consists of families of subsets of the universe set from A whose union contains all the elements in the universe set. The representative setting is such that the universe set is of size 6 and A has 25 subsets. The mean cost of any subset $a \in A$ is selected randomly and proportional to the size of that set and $|\mathcal{S}| = 285$. The minimum size cover is of size 1 and the minimum regret exploration set is of size 13 with an implied critical subset of size 23. Figure 5 depicts the average performance of four different policies on the representative from the set cover setting. In ranking the performance of different policies, the relative order is essentially the same as in the case of the previous setting.

Minimum Steiner tree problem. We consider a generalized version of the Steiner tree problem

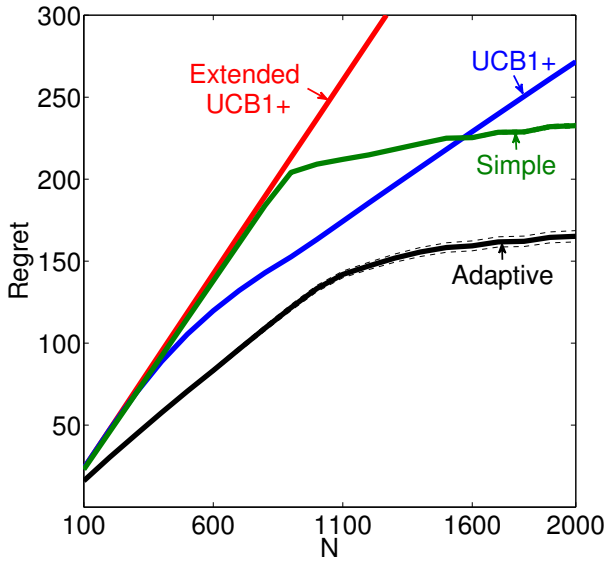

Figure 4: Average performance of different policies on the representative from the shortest path setting.

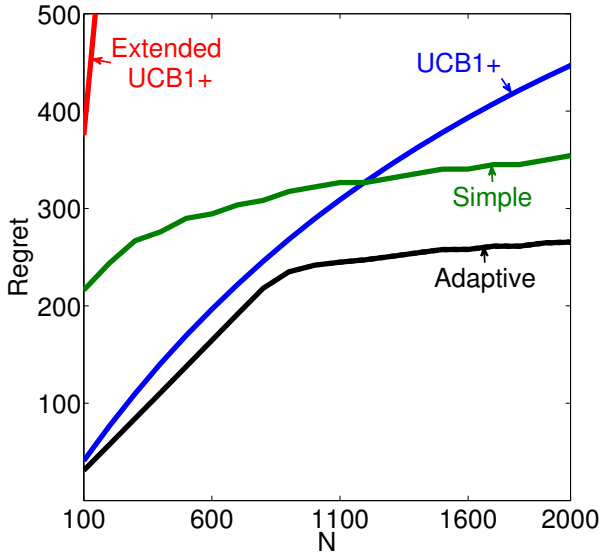

Figure 5: Average performance of different policies on the representative from the set cover setting.

(Williamson and Shmoys 2011), where we are given an undirected graph with non-negative edge costs, and a set of pairs of vertices. Our objective is to find a minimum cost subset of edges such that every given pair is connected in the set of selected edges.

For this problem, we generate a random graph and select a set of pairs of vertices randomly. The ground element (edge) mean costs are selected uniformly randomly from the set $\{0.1, 0.2, \ldots, 1\}$ and then normalized. The representative setting is such that $|A| = 18$, $|\mathcal{S}| = 2490$, the minimum size

cover is of size 1, and the minimum regret exploration set is of size 7 with an implied critical subset of size 17. Figure 6 depicts the average performance of four different policies on the representative from the Steiner tree setting.

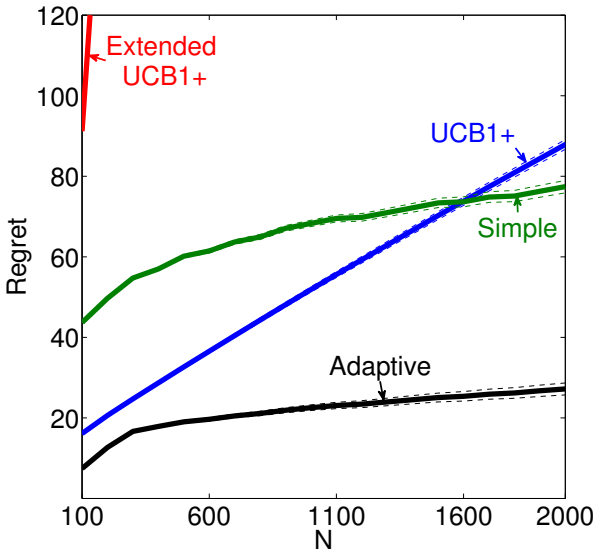

Figure 6: Average performance of different policies on the representative from the Steiner tree setting.

In ranking the performance of different policies, the relative order is essentially the same as in the case of the previous two settings.

Knapsack problem. In the knapsack problem we are given a set A of items to put in a knapsack. The solution set $\mathcal{S}$ consists of the subsets of items whose total weights do not exceed the knapsack weight limit. We generate a set of items with random utilities and weights. A random weight limit is also selected for the knapsack. The representative setting is such that $|A| = 20$, $|\mathcal{S}| = 9078$, the minimum size cover is of size 4, and the minimum regret exploration set is of size 6 with an implied critical subset of size 16. Figure 7 depicts the average performance of four different policies on the representative form the knapsack setting. The adaptive policy outperforms the others. Moreover, the benchmarks are outperformed by the simple policy.

8.2 Short-term Experiments

The purpose of this section is to evaluate the performance of different policies in the short-term. Although the adaptive policy is not designed for such a setting, our numerical experiments show that it provides a competitive performance in several cases. We first describe different benchmark policies. After briefly discussing some implementation details, we close this section presenting our

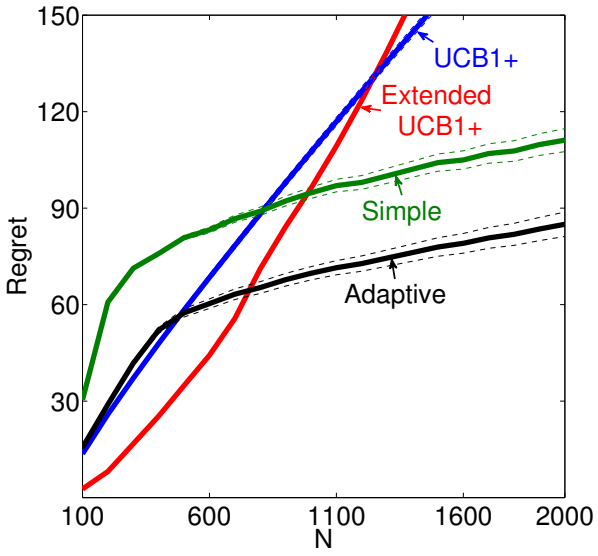

Figure 7: Average performance of different policies on the representative from the knapsack setting.

numerical results.

8.2.1 Benchmark Policies

Our benchmark policies are based on the Knowledge-Gradient (KG) policy in Ryzhov et al. (2012) and the Gittins index approximation in Lai (1987). These policies are not designed for our setting and therefore, we implement adapted versions of them, which we detail below.

While our policies are non-parametric and are designed to have any-time performance guarantees, the KG algorithm requires specifying a prior distribution for the cost and associated hyper-parameters, and prior knowledge of the time horizon N. In our experiments we allow KG access to such prior information. In particular, since we consider exponentially distributed costs, we implement KG using the Exponential-Gamma conjugate prior for each ground element.

The details of our implementation are the following. The KG algorithm assumes that $b_{a,n}$ follows an exponential distribution with rate μ_a, where this rate itself is random, and initially distributed according to a Gamma distribution with parameters $\alpha_{a,0}$ and $\beta_{a,0}$. At time n, the posterior distribution of μ_a is a Gamma with parameters

$$\alpha_{a,n} = \alpha_{a,0} + T_n(a), \quad \beta_{a,n} = \beta_{a,0} + \sum_{m<n:a\in S_m} b_{a,m}, \quad a \in A.$$

Thus at time n, the KG algorithm implements solution S_n^{KG}, where

$$S_n^{KG} \in \underset{S\in\mathcal{S}}{\operatorname{argmin}} \left\{ \sum_{a\in S} \frac{\beta_{a,n}}{\alpha_{a,n}-1} - (N-n)\mathbb{E}_S^n \left\{ \min_{S'\in\mathcal{S}} \left\{ \sum_{a\in S'} \frac{\beta_{a,n}}{\alpha_{a,n}-1} \right\} - \min_{S'\in\mathcal{S}} \left\{ \sum_{a\in S'} \frac{\beta_{a,n+1}}{\alpha_{a,n+1}-1} \right\} \right\} \right\},$$

where the expectation is taken with respect to $\{b_{a,n} : a \in S\}$ and N denotes the total number of instances, which KG assumes as known. The expectation above corresponds to the knowledge gradient $v_S^{KG,n}$ in the notation of Ryzhov et al. (2012). Unlike in that paper, there is no closed form expression for $v_S^{KG,n}$ in our setting. Our plain vanilla implementation of the KG algorithm computes such a term via Monte Carlo simulation, and performs the outer minimization via enumeration. The high computational complexity of this implementation limited the size of the settings we could test.

The second benchmark is an approximation based on the Gittins index rule which in the finite-horizon undiscounted settings takes the form of an *average productivity* index (see Niño-Mora (2011)), and although it is not optimal in general outside the discounted infinite horizon setting, it is still applied heuristically. We adjusted the Gittins index policy to the combinatorial setting by associating an index with each ground element separately, and computing the index of a solution as the sum of the indices of the ground elements included in such a solution.

The Gittins policy also requires prior knowledge of the time horizon N and a parametric representation of the uncertainty. In our experiments we provided Gittins with the additional information of the time horizon N. To mimic a setting where the functional form of reward distributions is unknown, we consider the approximation in Lai (1987) based on normally distributed rewards and use Normal/Normal-Gamma conjugate priors (this is motivated by a central limit argument): in our approximation, the index of a ground element $a \in A$ is given by

$$g_{n,N}^a(\mu_{a,n}, \lambda_{a,n}, \alpha_{a,n}, \beta_{a,n}) = \left(\mu_{a,n} - \sqrt{\frac{\beta_{a,n}}{(\alpha_{a,n}-1)\lambda_{a,n}}}\; h\left(\frac{\lambda_{a,n}-\lambda_{a,0}}{N-n+1+\lambda_{a,n}-\lambda_{a,0}} \right) \right)^+,$$

where $\mu_{a,n}$ and $\lambda_{a,n}$ are the mean and variance of the normal posterior, respectively, $\alpha_{a,n}$ and $\beta_{a,n}$ are the hyper parameters of the Gamma posterior, respectively, and $h(\cdot)$ approximates the boundary of an underlying optimal stopping problem. Thus at time n, the Gittins policy implements solution S_n^{Gitt}, where

$$S_n^{Gitt} \in \underset{S\in\mathcal{S}}{\operatorname{argmin}} \left\{ \sum_{a\in S} g_{n,N}^a(\mu_{a,n}, \lambda_{a,n}, \alpha_{a,n}, \beta_{a,n}) \right\}.$$

Note that since the benchmarks take $N = n$ as an input, several runs of the benchmark policies are necessary to construct their cumulative regret curves.

8.2.2 Implementation Details

The implementation details are essentially the same as in the case of the long-term experiments, except the fact that here we use $H = 2.5$ for the adaptive policy.[8] We report on the average behavior of different policies over 100 replications. On each of the figures in this section, the vertical lines represent 95% confidence intervals on our results. The average running time for a single replication ranged from less than a second for the adaptive policy to around 5 seconds for Gittins to several minutes for KG.

Finally, we exclude the results for simple, UCB1+, and Extended UCB1+ policies, since they were all outperformed by the adaptive policy.

8.2.3 Settings and Results

We use the same randomization scheme as in the case of the long-term experiments. Our results show that in the short-term experiments, the adaptive policy did not always outperform the benchmarks, but was still competitive.

Shortest path problem. Figure 8 depicts the average performance of three different policies for a shortest path problem in a layered graph with 5 layers, each with 4 nodes, and 2 connections between each inner layer. The representative setting is such that $|A| = 40$, $|\mathcal{S}| = 64$, the minimum size cover is of size 9, and the minimum regret exploration set is of size 10 with an implied critical subset of size 23. In the very short-term, the benchmark policies provide a better performance compared to the adaptive policy, but adaptive eventually surpasses the benchmarks for moderate values of N. Furthermore, Gittins provides a better performance than KG.

Set cover problem. Figure 9 depicts the average performances on a representative from the set cover setting. The representative setting is such that $|A| = 15$, $|\mathcal{S}| = 50$, the minimum size cover is of size 1, and the minimum regret exploration set is of size 7 with an implied critical subset of size 12. In ranking the performance of different policies, the relative order is essentially the same as the previous case, with the small difference that in the very short-term, KG provides the best performance.

Minimum Steiner tree problem. Figure 10 depicts the average performances on a representative from the Steiner tree setting. The representative setting is such that $|A| = 12$, $|\mathcal{S}| = 50$, the minimum size cover is of size 1, and the minimum regret exploration set is of size 7 with an implied critical subset of size 12. KG provides a better performance than the adaptive policy in the very short-term, but is outperformed eventually for moderate values of N. Gittins performs poorly in

[8] This selection favored the short-term performance by limiting the exploration, and resulted in logarithmic grow of the regret.

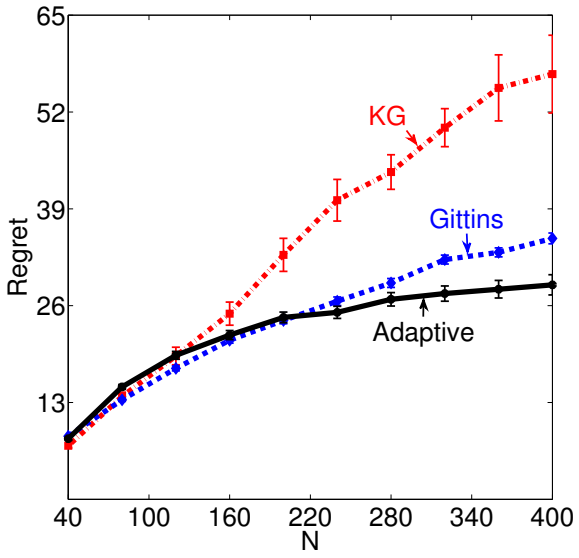

Figure 8: Average performance of different policies on the representative from the shortest path setting.

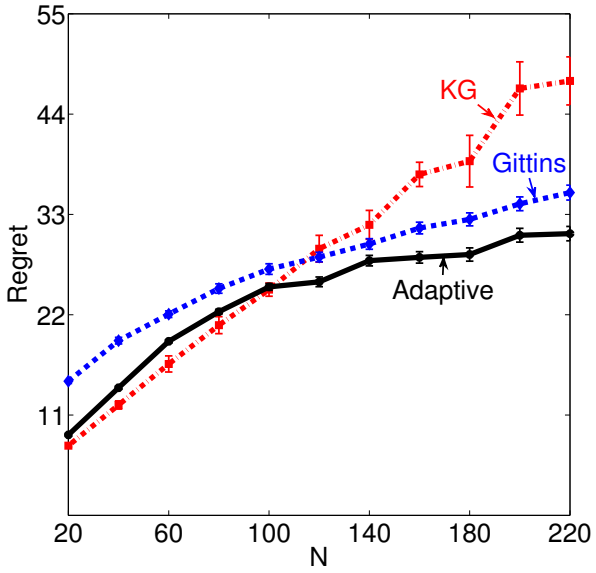

Figure 9: Average performance of different policies on the representative from the set cover setting.

this setting.

Knapsack problem. Figure 11 depicts the average performances on a representative from the knapsack setting[9]. The representative setting is such that $|A| = 11$, $|\mathcal{S}| = 50$, the minimum size cover is of size 5, and the minimum regret exploration set is of size 5 with an implied critical subset

[9]Here we report on the average behavior over 500 replications so that the confidence intervals do not cross.

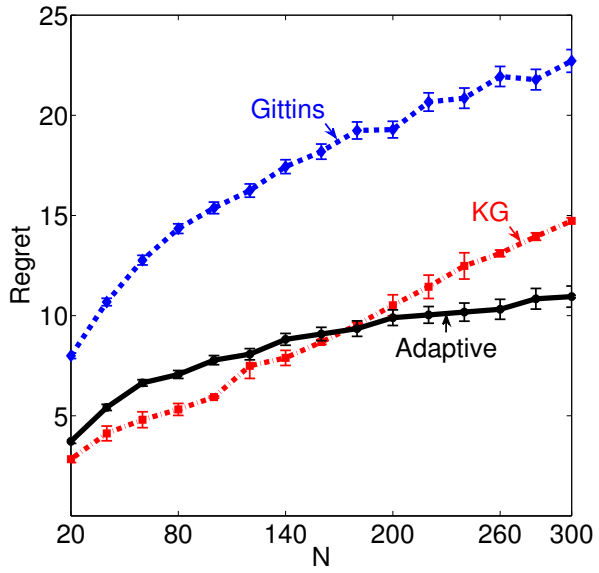

Figure 10: Average performance of different policies on the representative from the Steiner tree setting.

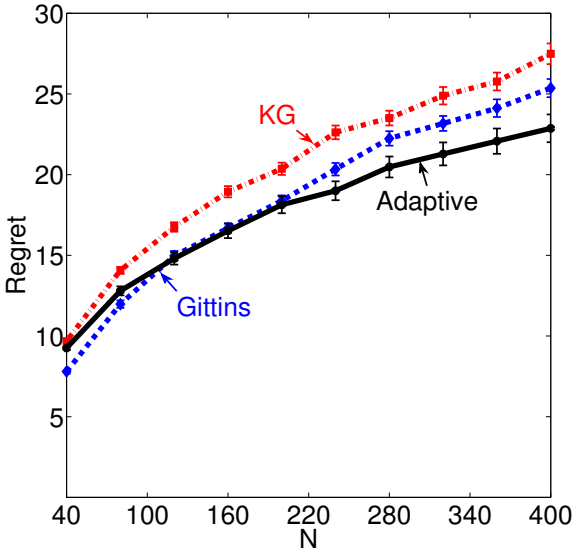

Figure 11: Average performance of different policies on the representative from the knapsack setting.

of size 8. Gittins provides a better performance than the adaptive policy in the very short-term, but is outperformed eventually, while KG performs poorly compared to the others.

9 Extensions and Future Work

9.1 Extension to Alternative Feedback Settings

Our problem formulation assumes that B_n is revealed *partially* after a solution is implemented. In particular, we assume that the decision maker observes the cost realization on each ground element $a \in S_n$. Depending on the application, however, it is plausible that the decision maker only has access to the *total* cost incurred by implementing solution S_n. That is, after implementing S_n, the decision maker might have access to $\sum_{a \in S_n} b_{a,n}$ as opposed to $\{b_{a,n} : a \in S_n\}$. Next, we indicate how to extend the proposed policies to this setting, and how such modifications affect their performance guarantees.

For a set of ground elements $S \subseteq A$, let $I_S \in \{0,1\}^{|A|}$ denote the incidence vector over the ground set (so that $S = \text{supp}\,(I_S)$). We say a solution set $\mathcal{E}$ *recovers* a set $E \subseteq A$ if for each $a \in E$, there exists a vector $\gamma(a) := (\gamma_S(a), S \in \mathcal{E})$ such that

$$\sum_{S \in \mathcal{E}} \gamma_S(a) I_S = I_{\{a\}}. \tag{16}$$

Without loss of generality, one can assume that each ground element is recovered by at least one solution set[10]. Let $\mathcal{E}$ be a solution set that recovers A, and let $\gamma := (\gamma(a), a \in A)$ be such that $\sum_{S \in \mathcal{E}} \gamma_S(a) I_S = I_{\{a\}}$, $a \in A$. One can implement Algorithm 1 with $\mathcal{E}$ playing the role of a cover while replacing the estimate in (7) with

$$\bar{b}_{a,n} := \sum_{S \in \mathcal{E}} \frac{\gamma_S(a)}{T_n(S)} \sum_{m<n : S_m = S} \sum_{a \in S} b_{a,m} \quad a \in A. \tag{17}$$

The estimate above reconstructs the expected cost of each solution in $\mathcal{E}$ and uses (16) to translate such estimates to the ground-element level. Implementing this modification requires precomputing a solution set $\mathcal{E}$ recovering A. Such a set can be selected so that $|\mathcal{E}| \leq |A|$, and computed by solving $O(|A|)$ instances of $f(\cdot)$ (see e.g., the algorithm in Awerbuch and Kleinberg (2004)). A close inspection to the proof of Theorem 4.3 reveals that its performance guarantee would remain valid (modulo changes to constant H_s) after incorporating the new estimation procedure.

The idea above can also be used to extend the adaptive policy to this new setting. In particular, Algorithm 2 would consider the estimates in (17) and $(C, \mathcal{E})$ to be solution to an alternative version

[10] If this is not the case, then it must be that a appears in a solution if and only if that solution also includes some ground element a'. Thus, one can (w.l.o.g.) combine such ground elements into a single element. Alternatively, one can assign costs only to one of such elements, so as to not modify the combinatorial structure of the solution set.

of OCP where in addition to (12b)-(12d), one imposes that $\mathcal{E}$ recovers C, that is

$$
\begin{aligned}
OCP'(B): \quad \min \quad & \sum_{S\in\mathcal{S}} \Delta_S^F(B)\; y_S && \text{(18a)}\\
s.t. \quad & \sum_{S\in\mathcal{S}} \gamma_S(a) I_S = x_a I_{\{a\}}, \quad a \in A && \text{(18b)}\\
& \gamma_S(a) \le Q\; y_S, \quad S \in \mathcal{S},\, a \in A && \text{(18c)}\\
& -\gamma_S(a) \le Q\; y_S, \quad S \in \mathcal{S},\, a \in A && \text{(18d)}\\
& \sum_{a\in S} (l_a(1-x_a) + b_a x_a) \ge z^*(B), \quad S \in \mathcal{S} && \text{(18e)}\\
& x_a,\, y_S \in \{0,1\},\, \gamma_S(a) \in \mathbb{R}, \quad a \in A, S \in \mathcal{S}, && \text{(18f)}
\end{aligned}
$$

where Q is an instance-dependent constant, whose size is polynomial on the size of the instance. The additional constraints(18b)-(18d) in OCP' ensure that the solution set $\mathcal{E}$ recovers the critical subset C. Like OCP, the formulation above can be specialized to accommodate the combinatorial structure of $f(\cdot)$ (as shown in Section 7.1). The performance guarantee in Theorem 6.2 would remain valid with the constant G referring to the size of a minimal solution to OCP'. We anticipate that the challenge of solving OCP' effectively is comparable to that of solving OCP, which is the subject of current research.

From the discussion above, we claim that our results can be extended to further feedback settings provided that two conditions hold: first, the feedback provided by implementing a solution should admit a consistent estimator (moreover, it should admit a concentration inequality, such as those used in Appendix A); second, one should be able to reconstruct the expected cost of a ground element based on the feedback from some set of solutions. Note, however, that while the simple policy can be extended by redefining the concept of a solution cover, extending the adaptive policy calls for reformulating OCP. This task might be non-trivial if, for example, the feedback is not linear.

9.2 Improving Performance Bounds

The mismatch between the leading constants in the lower and upper performance bounds presented in this paper suggests that there is room for improvement. We see two possible directions for narrowing said gap.

First, our policy aims to match the cost of exploration incurred by a solution to OCP, while that in Theorem 5.4 is driven by the solution to Formulation (11). Our policy approximates the latter formulation by imposing equal exploration frequencies on each element of its solution (i.e., $K_D = H$ for all $D \in \mathcal{D}$): this allows us to keep tractability of the resulting formulation. In this regard, our analysis favors tractability and implementability, in the same way that UCB1 does for

the traditional multi-armed bandit. As in the multi-armed bandit setting, further improvements might be attained by approximating Formulation (11) more closely; this would involve changing exploration frequencies adaptively over time. We expect, however, that the improvements attained by such a modification might be shadowed by the increase in computational complexity from approximating the coefficients K_D (which depend on the distribution of B_n) and from solving the resulting proxy formulation (Formulation (11) can be transformed into a linear IP formulation, but constraint (11b) is notoriously difficult to handle (Toriello and Vielma 2012)).

Second, a critical question is whether the fundamental bound in Theorem 5.4 can be achieved in practice, as is the case in traditional multi-armed bandits. The answer is unclear: the lower bound follows from the solution to Formulation (11), which finds the minimum regret among all "exploration sets" that comply with the consistency requirements in Proposition 5.2. Note that such a formulation assumes full knowledge of the B; however, the feedback obtained by implementing solutions solving such a formulation does not, in general, suffice to guarantee the optimality of such a solution, only its feasibility. In practical terms, this implies that a policy that limits exploration to a solution to (11) might fail to find the minimum regret exploration set in settings that generate the same feedback, but under which such a solution is only suboptimal. Instead of coping with this lack of robustness, Formulation (11) assumes advance knowledge of the input. This is impractical and suggests that the bound might be perfectible.

The issue above also arises when implementing the proposed policies: while one can confirm that a solution $(C, \mathcal{E})$ to OCP is feasible and minimal based on the feedback provided by its implementation, its optimality might depend on information that is not collected. Just like our adaptive policy was informed by the insight developed in Theorem 5.4, we expect that development of tighter fundamental performance bounds should guide the development of new policies with matching performance guarantees.

9.3 Final Remarks

In this paper we have studied a class of sequential decision making problems where the underlying single-period decision problem is a combinatorial optimization problem, and there is initial uncertainty about its objective coefficients. By framing the problem as a *combinatorial* multi-armed bandit, we have developed a family of asymptotically efficient policies and showed that their performance is *essentially* the best possible. In doing so, we have extended results developed in the classical multi-armed bandit setting and highlighted key differences between the two settings. In particular, we have shown that, in addition to answering the question of *when* (with what frequency) to explore, which is key in the traditional setting, in the combinatorial setting one must also answer the questions of *what* and *how* to explore. In this regard, by establishing a fundamental limit on the performance of all admissible policies, we have shown that answering these new ques-

tions efficiently involves solving a new class of combinatorial problems that aim to find the cheapest optimality guarantee for the optimal solution to the underlying combinatorial problem. Our policies are based on the insight provided by this fundamental bound. We have shown evidence that the proposed policies are scalable and perform reasonably well relative to the relevant benchmark algorithms, both in the short and long-term.

The complexity of OCP is crucial for implementing the proposed policies, especially in settings where instances arrive rather frequently. In this regard, Section 7 sheds light on the potential for implementing our adaptive policy for real-size instances of problems with effective LP and IP formulations. However, complexity of OCP is not known in general. Proving that OCP is in P or is NP-hard is a first step in understanding the scalability of our policies. Further studies on OCP might provide us with some insight for solving this problem efficiently. Formulation (15), for example, provides a clear path for developing efficient custom formulations for network flow problems with strong duality properties.

References

Agrawal, R. (1995), 'The continuum-armed bandit problem', *SIAM J. Control Optim.* **33**(6), 1926–1951.

Agrawal, R., Hegde, M. and Teneketzis, D. (1990), 'Multi-armed bandit problems with multiple plays and switching cost', *Stochastics: An International Journal of Probability and Stochastic Processes* **29**(4), 437–459.

Anantharam, V., Varaiya, P. and Walrand, J. (1987), 'Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part I: IID rewards', *Automatic Control, IEEE Transactions on* **32**(11), 968–976.

Applegate, D., Bixby, R., Chvátal, V. and Cook, W. (2011), *The Traveling Salesman Problem: A Computational Study*, Princeton Series in Applied Mathematics, Princeton University Press.

Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002), 'Finite-time Analysis of the Multiarmed Bandit Problem', *Machine Learning* **47**(2-3), 235–256.

Auer, P., Cesa-bianchi, N., Freund, Y. and Schapire, R. E. (2003), 'The non-stochastic multi-armed bandit problem', *SIAM Journal on Computing* **32**, 48–77.

Awerbuch, B. and Kleinberg, R. D. (2004), Adaptive routing with end-to-end feedback: distributed learning and geometric approaches, *in* 'Proceedings of the thirty-sixth annual ACM symposium on Theory of computing', STOC '04, ACM, New York, NY, USA, pp. 45–53.

Balas, E. and Carrera, M. C. (1996), 'A dynamic subgradient-based branch-and-bound procedure for set covering', *Operations Research* **44**, 875–890.

Berry, D. and Fristedt, B. (1985), *Bandit Problems*, Chapman and Hall, London, UK.

Bubeck, S., Munos, R., Stoltz, G. and Szepesvári, C. (2011), 'X-armed bandits', *Journal of Machine Learning Research* **12**, 1655–1695.

Caro, F. and Gallien, J. (2007), 'Dynamic assortment with demand learning for seasonal consumer goods', *Management Science* **53**, 276–292.

Carvajal, R., Constantino, M., Goycoolea, M., Vielma, J. P. and Weintraub, A. (2013), 'Imposing connectivity constraints in forest planning models', *to Appear in Operations Research* .

Cesa-Bianchi, N. and Lugosi, G. (2006), *Prediction, Learning, and Games*, Cambridge University Press.

Cesa-Bianchi, N. and Lugosi, G. (2012), 'Combinatorial bandits', *Journal of Computer and System Sciences* .

Cook, W. (2010), Fifty-plus years of combinatorial integer programming, *in* '50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art', Springer-Verlag, New York, chapter 12, pp. 387–430.

Cook, W. J., Cunningham, W. H., Pulleyblank, W. R. and Schrijver, A. (1998), *Combinatorial optimization*, John Wiley & Sons, Inc., New York, NY, USA.

Cover, T. and Thomas, J. (2006), *Elements of Information theory*, John Wiley & Sons, Inc., Hoboken, NJ.

Etcheberry, J. (1977), 'The set-covering problem: A new implicit enumeration algorithm', *Operations research* **25**, 760–772.

Gai, Y., Krishnamachari, B. and Jain, R. (2012), 'Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations', *IEEE/ACM Transactions on Networking (TON)* **20**(5), 1466–1478.

Gittins, J. (1979), 'Bandit processes and dynamic allocation rules', *Journal of the Royal Statistical Society* **41**, 148–177.

Gittins, J. (1989), *Multi-armed bandit allocation indices*, Wiley.

Hoffman, K. L. and Padberg, M. (1993), 'Solving airline crew scheduling problems by branch-and-cut', *Management Science* **39**, 657–682.

IBM ILOG (n.d.), 'CPLEX High-performance mathematical programming engine'. `http://www.ibm.com/software/integration/optimization/cplex/`.

Kleinberg, R., Slivkins, A. and Upfal, E. (2008), 'Multi-armed bandits in metric spaces', *CoRR* **abs/0809.4882**.

Koch, T. and Martin, A. (1998), 'Solving steiner tree problems in graphs to optimality', *Networks* **32**(3), 207–232.

Kulkarni, S. and Lugosi, G. (1997), Minimax lower bounds for the two-armed bandit problem, *in* 'Decision and Control, 1997., Proceedings of the 36th IEEE Conference on', Vol. 3, IEEE, pp. 2293–2297.

Lai, T. L. (1987), 'Adaptive treatment allocation and the multi-armed bandit problem', *The Annals of Statistics* pp. 1091–1114.

Lai, T. L. and Robbins, H. (1985), 'Asymptotically efficient adaptive allocation rules', *Advances in Applied Mathematics* **6**(1), 4–22.

Magnanti, T. L. and Wolsey, L. A. (1995), *Optimal trees*, Vol. 7 of *Handbooks in Operational Research and Management Science*, North-Holland, Amsterdam, pp. 503–615.

Martin, R. K. (1991), 'Using separation algorithms to generate mixed integer model reformulations', *Operations Research Letters* **10**, 119–128.

Mersereau, A., Rusmevichientong, P. and Tsitsiklis, J. (2009), 'A structured multiarmed bandit problem and the greedy policy', *IEEE Transactions on Automatic Control* **54**(12), 2787–2802.

Mitchell, J. E. (2011), Branch and cut, *in* 'Wiley Encyclopedia of Operations Research and Management Science', John Wiley & Sons.

Nemhauser, G. L. and Wolsey, L. A. (1988), *Integer and combinatorial optimization*, Wiley-Interscience.

Niño-Mora, J. (2011), 'Computing a classic index for finite-horizon bandits', *INFORMS Journal on Computing* **23**(2), 254–267.

Robbins, H. (1952), 'Some aspects of the sequential design of experiments', *Bulletin of the American Mathematical Society* **58**, 527–535.

Rusmevichientong, P., Shen, Z. and Shmoys, D. (2010), 'Dynamic assortment optimization with a multinomial logit choice model and capacity constraint', *Operations Research* **58**(6), 1666–1680.

Rusmevichientong, P. and Tsitsiklis, J. (2010), 'Linearly parameterized bandits', *Mathematics of Operations Research* **35**(2), 395–411.

Ryzhov, I. O. and Powell, W. B. (2009), The knowledge gradient algorithm for online subset selection, *in* 'Proceedings of the 2009 IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning', pp. 137–144.

Ryzhov, I. O. and Powell, W. B. (2011), 'Information collection on a graph', *Operations Research* **59**(1), 188–201.

Ryzhov, I. O. and Powell, W. B. (2012), Information collection in linear programs with uncertain objective coefficients. To appear in SIAM Journal on Optimization.

Ryzhov, I. O., Powell, W. B. and Frazier, P. I. (2012), 'The knowledge gradient algorithm for a general class of online learning problems', *Operations Research* **60**(1), 180–195.

Saure, D. and Zeevi, A. (2013), 'Optimal dynamic assortment planning with demand learning', *Forthcomming, MSOM* .

Schrijver, A. (2003), *Combinatorial Optimization - Polyhedra and Efficiency*, Springer.

Stanley, R. (1999), *Enumerative combinatorics, Volume 2*, Cambridge studies in advanced mathematics, Cambridge University Press.

Thompson, W. R. (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples', *Biometrika* **25**, 285–294.

Toriello, A. and Vielma, J. P. (2012), 'Fitting piecewise linear continuous functions', *European Journal of Operational Research* **219**, 86 – 95.

Vielma, J. P. (2012), 'Mixed integer linear programming formulation techniques', *Optimization Online* .
URL: *http://www.optimization-online.org/DB_HTML/2012/07/3539.html*

Whittle, P. (1982), *Optimization over time: Vol I*, John Wiley and Sons Ltd.

Williamson, D. P. and Shmoys, D. B. (2011), *The Design of Approximation Algorithms*, Cambridge University Press.

A Proof of main results

Proof of Theorem 4.3. The regret of the simple policy $\pi_s(\mathcal{E})$ stems from two sources: exploration and errors during exploitation. That is,

$$R^{\pi_s(\mathcal{E})}(F,N) = \sum_{S\in\mathcal{S}} \Delta_S^F \; \mathbb{E}_F\left\{T_{N+1}(S)\right\} = R_1^{\pi_s(\mathcal{E})}(F,N) + R_2^{\pi_s(\mathcal{E})}(F,N), \tag{A-1}$$

where $R_1^{\pi_s(\mathcal{E})}(F,N)$ is the exploration-based regret, i.e., that incurred while $T_n(a) < i$ for some $a \in A$ at instance n in cycle i, and $R_2^{\pi_s(\mathcal{E})}(F,N)$ is the exploitation-based regret, i.e., that incurred when $T_n(a) \geq i$ for all $a \in A$. We prove the result by bounding each term above separately.

In the remainder of this proof, $\mathbb{E}$ and $\mathbb{P}$ denote expectation and probability when costs are distributed according to F and policy $\pi_s(\mathcal{E})$ is implemented.

Step 1 (Exploration-based regret). By construction, $\pi_s(\mathcal{E})$ implements each solution $S \in \mathcal{E}$ at most $\lceil H \ln N \rceil$ while exploring. Moreover, no more than $|A|$ of these are implemented within each exploration phase. Therefore

$$R_1^{\pi_s(\mathcal{E})}(F,N) \leq \min\left\{|A|, |\mathcal{E}|\right\} \Delta_{max}^F (H \ln N + 1). \tag{A-2}$$

Step 2 (Exploitation-based regret). Exploitation-based regret during cycle i is due to implementing suboptimal solutions when all elements in A have been tried at least on i instances.

Let $i' := \inf\left\{i \in \mathbb{N}, i \geq 2 : n_i \geq (i-1)|\mathcal{E}|, n_{i+1} - n_i > |\mathcal{E}|\right\}$ denote the first cycle in which one is sure to exploit on at least one instance. Note that i' does not depend on N, thus neither does the exploitation-based regret prior to cycle i'.

Fix $i \geq i'$. For $n \in [n_i, n_{i+1} - 1]$, let $\overline{S}_n \in \mathcal{S}^*(\overline{B}_n)$ be any solution with minimum average cost at time n. We have that

$$\begin{aligned} R_2^{\pi_s(\mathcal{E})}(F,N) &\leq n_{i'}\Delta_{max}^F + \mathbb{E}\left\{\sum_{i=i'}^{\lceil H\ln N\rceil}\sum_{n=n_i}^{n_{i+1}-1} \mathbb{P}\left\{\overline{S}_n \notin \mathcal{S}^*\left(\mathbb{E}\left\{B_n\right\}\right), T_n(a) \geq i, \forall\, a \in A\right\} \Delta_{\overline{S}_n}^F\right\} \\ &\leq n_{i'}\Delta_{max}^F + \sum_{i=i'}^{\infty}\sum_{n=n_i}^{n_{i+1}-1} \mathbb{P}\left\{\overline{S}_n \notin \mathcal{S}^*(\mathbb{E}\left\{B_n\right\}), T_n(a) \geq i, \forall\, a \in A\right\} \Delta_{max}^F. \end{aligned} \tag{A-3}$$

Next we find an upper bound for the probability inside the sum in (A-3). For this, note that

$$\left\{\overline{S}_n \notin \mathcal{S}^*(\mathbb{E}\left\{B_n\right\})\right\} \subseteq \left\{|\overline{z}_n^* - \mathbb{E}\left\{\overline{z}_n^*\right\}| \geq \frac{\Delta_{min}^F}{2}\right\} \cup \left\{|\overline{z}_n - \mathbb{E}\left\{\overline{z}_n\right\}| \geq \frac{\Delta_{min}^F}{2}\right\}, \tag{A-4}$$

where $\overline{z}_n := \sum_{a \in \overline{S}_n} \overline{b}_{a,n}$, $\overline{z}^*_n := \sum_{a \in S^*} \overline{b}_{a,n}$ for some $S^* \in \mathcal{S}^*(\mathbb{E}\{B_n\})$, and

$$\Delta^F_{min} := \min\left\{\Delta^F_S \,:\, \Delta^F_S > 0\,,\, S \in \mathcal{S}\right\},$$

is the minimum optimality gap. Indeed, note that $\left\{|\overline{z}^*_n - \mathbb{E}\{\overline{z}^*_n\}| < \frac{\Delta^F_{min}}{2}\right\}$ and $\left\{|\overline{z}_n - \mathbb{E}\{\overline{z}_n\}| < \frac{\Delta^F_{min}}{2}\right\}$ implies that $\overline{z}_n > \overline{z}^*_n$.

The next proposition, whose proof can be found in Appendix B, allows us to bound (A-3) using the observation above.

Proposition A.1. *For any fixed $S \subseteq A$, $n \in \mathbb{N}$, $i \in \mathbb{N}$, and any constant $\epsilon > 0$ we have*

$$\mathbb{P}\left\{\left|\sum_{a \in S}\left(\overline{b}_{a,n} - \mathbb{E}\{b_{a,n}\}\right)\right| \geq \epsilon, T_n(a) \geq i\,,\, \forall a \in S\right\} \leq 2\; K(\epsilon)\,|S| \exp\left\{-\frac{2\epsilon^2 i}{|S|^2\, L^2}\right\},$$

where $L := \max\{u_a - l_a : a \in A\}$, and $K(\epsilon)$ is a positive finite constant that only depends on ϵ.

Using the above, one has that

$$\begin{aligned}\mathbb{P}\left\{|\overline{z}^*_n - \mathbb{E}\{\overline{z}^*_n\}| \geq \frac{\Delta^F_{min}}{2}\,,\, T_n(a) \geq i\,,\, \forall\, a \in A\right\} &\leq \sum_{S \in \mathcal{S}} \mathbb{P}\left\{\left|\sum_{a \in S}\left(\overline{b}_{a,n} - \mathbb{E}\{b_{a,n}\}\right)\right| \geq \frac{\Delta^F_{min}}{2}\,,\, T_n(a) \geq i\,,\, \forall\, a \in S\right\} \\ &\overset{(a)}{\leq} 2\; K\,|\mathcal{S}|\,|A| \exp\left\{-\frac{{\Delta^F_{min}}^2 i}{2|A|^2 L^2}\right\},\end{aligned}$$

where K is a positive finite constant, and (a) follows from noting that $|S| \leq |A|$ for all $S \in \mathcal{S}$. Consider (A-4): applying Proposition A.1 and the above to the first and second term on its right-hand side, respectively, one obtains

$$\mathbb{P}\left\{\overline{S}_n \notin \mathcal{S}^*(\mathbb{E}\{B_n\})\,,\, T_n(a) \geq i\,,\, \forall\, a \in A\right\} \leq 4\; K\,|\mathcal{S}|\,|A| \exp\{-C_1 i\}\,, \tag{A-5}$$

where $C_1 := {\Delta^F_{min}}^2/2|A|^2L^2$. Note that this final bound does not depend on n but rather on i. Now, for $i \geq i'$, one has that $n_{i+1} \leq e^{(i+1)/H}$ and $n_i \geq e^{(i-1)/H}$, hence

$$n_{i+1} - n_i \leq C_2\; e^{\frac{i}{H}}, \quad i \geq i',$$

where $C_2 := e^{1/H} - e^{-1/H}$. Using this latter fact, (A-3) and (A-5) we conclude that

$$R_2^{\pi_s(\mathcal{E})}(F, N) \leq n_{i'}\; \Delta^F_{max} + \sum_{i=i'}^{\infty} C_3 \exp\left\{i\left(\frac{1}{H} - C_1\right)\right\},$$

where $C_3 := 4\ K\,|\mathcal{S}|\,|A|\Delta_{max}^F C_2$. Choosing H so that $1/H < C_1$ we have

$$R_2^{\pi_s(\mathcal{E})}(F,N) \leq C_4, \tag{A-6}$$

where C_4 is a positive finite constant. Combining (A-1), (A-2) and (A-6) we conclude that

$$R^{\pi_s(\mathcal{E})}(F,N) \leq \min\left\{|A|\,,|\mathcal{E}|\right\}\Delta_{max}^F H \ln N + C_5,$$

where C_5 is a positive finite constant. This proves the result. □

Proof of Proposition 5.2. We begin by imposing some structure on F.

Preliminaries. We assume F_a, the *common* distribution of $b_{a,n}$, $n \in \mathbb{N}$, is uniformly continuous with respect to Lebesgue measure in $\mathbb{R}$ and let f_a denote its density function. To simplify the notation we assume that these functions accept parametric representations, and let θ_a denote the "true" parameter for $f_a, a \in A$. Finally, we let Θ_a denote the set of feasible parameters for $f_a, a \in A$. These *mild* conditions are fulfilled by most commonly used distribution functions.

For $\lambda_a \in \Theta_a$, let $I_a(\theta_a,\lambda_a)$ denote the Kullback-Leibler distance between $F_a(\cdot;\theta_a)$ and $F_a(\cdot;\lambda_a)$, i.e.,

$$I_a(\theta_a,\lambda_a) = \int_{-\infty}^{\infty}[\ln(f_a(x_a;\theta_a)/f_a(x_a;\lambda_a))]f_a(x_a;\theta_a)\ dx_a.$$

Define $b_a(\lambda_a) := E_{F(\cdot:\lambda_a)}\{b_{a,n}\}$, $n \in \mathbb{N}$. In addition to the conditions above, we assume the following hold as well.

- **Indistinguishability.** $0 < I_a(\theta_a,\lambda_a) < \infty$ whenever $b_a(\theta_a) > b_a(\lambda_a)$.
- **Continuity.** For all $\epsilon > 0$ and $\lambda_a \in \Theta_a$ such that $b_a(\theta_a) > b_a(\lambda_a) > l_a$, there exists a $\delta > 0$ for which $|I_a(\theta_a,\lambda_a) - I_a(\theta_a,\lambda_a')| < \epsilon$ whenever $b_a(\lambda_a) \geq b_a(\lambda_a') \geq b_a(\lambda_a) - \delta$.

The first condition implies that distributions with different mean costs are not distinguishable based on a finite sample. The second condition essentially says that distributions with similar mean costs should be "close" to each other.

Finally, define $\theta = (\theta_a : a \in A)$ and let $\mathbb{E}_\lambda$ and P_λ denote the expectation and probability induced when F receives the parameter $\lambda := (\lambda_a : a \in A) \in \mathbb{R}^{|A|}$. Also, define $\mathcal{S}_\lambda^* := \mathcal{S}^*\left(\mathbb{E}_\lambda\{B_n\}\right)$.

Proof of the result. Consider $D \in \mathcal{D}$ as defined in (8), and take $\lambda \in \mathbb{R}^{|A|}$ so that $\lambda_a = \theta_a$ for $a \notin D$, and that $D \subseteq S^*$ for all $S^* \in \mathcal{S}_\lambda^*$. By the consistency of π, one has that

$$\mathbb{E}_\lambda\left\{N - \sum_{S^*\in\mathcal{S}_\lambda^*} T_{N+1}(S)\right\} = o(N^\alpha),$$

for any $\alpha > 0$. By construction, each optimal solution under λ tries each $a \in D$ when implemented.

Thus, one has that $\sum_{S^* \in \mathcal{S}^*_\lambda} T_{N+1}(S) \leq \max\{T_{N+1}(a) : a \in D\}$, and therefore

$$\mathbb{E}_\lambda\{N - \max\{T_{N+1}(a) : a \in D\}\} \leq \mathbb{E}_\lambda\left\{N - \sum_{S^* \in \mathcal{S}^*_\lambda} T_{N+1}(S)\right\} = o(N^\alpha). \tag{A-7}$$

Take $\epsilon > \alpha$. Define $I(D,\lambda) := |D| \max\{I_a(\theta_a, \lambda_a) : a \in D\}$, $D \in \mathcal{D}$. Applying Markov's inequality to $N - \max\{T_{N+1}(a) : a \in D\}$ and using (A-7), one has that

$$(N - O(\ln N))\ \mathbb{P}_\lambda\left\{\max\{T_{N+1}(a) : a \in D\} < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\} = o(N^\alpha).$$

The above can be re-written as

$$\mathbb{P}_\lambda\left\{\max\{T_{N+1}(a) : a \in D\} < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\} = o(N^{\alpha-1}). \tag{A-8}$$

For $a \in D$ and $n \in \mathbb{N}$ define

$$L_n(a) := \sum_{k=1}^{n} \ln\left(f_a(b_a^k;\theta_a)/f_a(b_a^k;\lambda_a)\right),$$

where b_a^k denotes the k-th cost observation for $a \in D$ when policy π is implemented. Also, define the event

$$\Xi(N) := \left\{L_{T_{N+1}(a)}(a) \leq \frac{(1-\alpha)\ln N}{|D|} \text{ for all } a \in D\,,\ \max\{T_{N+1}(a) : a \in D\} < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\},$$

and note that

$$\mathbb{P}_\lambda\{\Xi(N)\} \leq \mathbb{P}_\lambda\left\{\max\{T_{N+1}(a) : a \in D\} < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\}.$$

Next, we relate the probability of the event $\Xi(N)$ under the two parameter configurations.

$$\begin{aligned}
\mathbb{P}_\lambda\{\Xi(N)\} &= \int_{\omega\in\Xi(N)} d\mathbb{P}_\lambda(\omega) \\
&\overset{(a)}{=} \int_{\omega\in\Xi(N)} \prod_{a\in D} \exp(-L_{T_{N+1}(a)}(a))\ d\mathbb{P}_\theta(\omega) \\
&\overset{(b)}{\geq} \int_{\omega\in\Xi(N)} \exp(-(1-\alpha)\ln N)\ d\mathbb{P}_\theta(\omega) \\
&= N^{\alpha-1}\mathbb{P}_\theta\{\Xi(N)\},
\end{aligned}$$

where (a) follows from noting that probabilities under λ and θ differ only in that cost observations

in D have different probabilities under λ and θ, and (b) follows from noting that $L_{T_{N+1}(a)}(a) \leq (1-\alpha)\ln N/|D|$ for all $\omega \in \Xi(N)$.

From above and (A-8) we have that

$$\lim_{N\to\infty} \mathbb{P}_\theta\{\Xi(N)\} \leq \lim_{N\to\infty} N^{1-\alpha}\, \mathbb{P}_\lambda\{\Xi(N)\} = 0. \tag{A-9}$$

Now, fix $a \in D$. By the Strong Law of Large Numbers we have

$$\lim_{n\to\infty} \max_{m\leq n} L_m(a)/n = I_a(\theta_a, \lambda_a), \quad \text{a.s.}[\mathbb{P}_\theta], \quad \forall a \in D.$$

Since, $\alpha < \epsilon$, we have

$$\lim_{N\to\infty} \mathbb{P}_\theta\left\{L_m(a) > \frac{(1-\alpha)\ln N}{|D|} \text{ for some } m < \frac{(1-\epsilon)\ln N}{|D|\, I_a(\theta_a,\lambda_a)}\right\} = 0 \quad \forall a \in D,$$

and since $I(D,\lambda) \geq |D|\ I_a(\theta_a,\lambda_a)$, we further have

$$\lim_{N\to\infty} \mathbb{P}_\theta\left\{L_m(a) > \frac{(1-\alpha)\ln N}{|D|} \text{ for some } m < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\} = 0 \quad \forall a \in D.$$

Then, in particular by taking $m = T_{N+1}(a)$ we have

$$\lim_{N\to\infty} \mathbb{P}_\theta\left\{L_{T_{N+1}(a)}(a) > \frac{(1-\alpha)\ln N}{|D|}, \quad T_{N+1}(a) < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\} = 0 \quad \forall a \in D,$$

which in turn implies

$$\lim_{N\to\infty} \mathbb{P}_\theta\left\{L_{T_{N+1}(a)}(a) > \frac{(1-\alpha)\ln N}{|D|}, \quad \max\{T_{N+1}(a) : a \in D\} < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\} = 0 \quad \forall a \in D.$$

Finally, by taking the union of events over $a \in D$ we have

$$\lim_{N\to\infty} \mathbb{P}_\theta\left\{L_{T_{N+1}(a)}(a) > \frac{(1-\alpha)\ln N}{|D|} \text{ for some } a \in D\,,\, \max\{T_{N+1}(a) : a \in D\} < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\} = 0. \tag{A-10}$$

Combining (A-9) and (A-10) we have that

$$\lim_{N\to\infty} \mathbb{P}_\theta\left\{\max\{T_{N+1}(a) : a \in D\} < \frac{(1-\epsilon)\ln N}{I(D,\lambda)}\right\} = 0.$$

The result follows from letting ϵ approach zero, and taking $K_D := I(D,\lambda)^{-1}$. □

Proof of Proposition 5.3. Define $\Upsilon_N := \bigcap_{D\in\mathcal{D}} \{\max\{T_{N+1}(a) : a \in D\} \geq K_D \ln N\}$ and note that $\zeta^\pi(F,N) \geq z(F,N) \geq \tilde{z}(F,N) = \kappa(F)\ln N$ when Υ_N occurs, because $(x_a = T_{N+1}(a), a \in A)$

and $(y_S = T_{N+1}(S), S \in \mathcal{S})$ are feasible to ILBP, thus

$$\begin{aligned}\mathbb{P}_F\left\{\frac{\zeta^\pi(F,N)}{\ln N} < \kappa(F)\right\} &= \mathbb{P}_F\left\{\frac{\zeta^\pi(F,N)}{\ln N} < \kappa(F)\,,\,\Upsilon_N\right\} + \mathbb{P}_F\left\{\frac{\zeta^\pi(F,N)}{\ln N} < \kappa(F)\,,\,\Upsilon_N^c\right\} \\ &\leq \mathbb{P}_F\left\{\Upsilon_N^c\right\}. \end{aligned} \tag{A-11}$$

However, by Proposition 5.2, we have that

$$\lim_{N\to\infty} \mathbb{P}_F\left\{\Upsilon_N^c\right\} \leq \sum_{D\in\mathcal{D}} \lim_{N\to\infty} \mathbb{P}_F\left\{\max\left\{T_{N+1}(a) : a \in D\right\} < K_D \ln N\right\} = 0,$$

since $|\mathcal{D}| < \infty$. Thus, taking the limit in (A-11) we have that

$$\lim_{N\to\infty} \mathbb{P}_F\left\{\zeta^\pi(F,N) < \kappa(F)\ln N\right\} = 0.$$

This concludes the proof. □

Proof of Lemma 6.1. We prove the result by contradiction. Let (x, y) be a feasible solution to OCP, and suppose that there is a $D \in \mathcal{D}$ such that $\max\{x_a : a \in D\} = 0$. By the definition of $\mathcal{D}$, one has that $z^*(B'_D) < z^*(\mathbb{E}_F\{B_n\})$, thus

$$\begin{aligned} z^*(B'_D) &= \sum_{a\in S^*\setminus D} \mathbb{E}_F\{b_{a,n}\} + \sum_{a\in D} l_a \\ &\overset{(a)}{\geq} \sum_{a\in S^*} \left(l_a(1-x_a) + \mathbb{E}_F\{b_{a,n}\}\,x_a\right) \\ &\overset{(b)}{\geq} z^*(\mathbb{E}_F\{B_n\}), \end{aligned}$$

for $S^* \in \mathcal{S}^*(\mathbb{E}_F\{B'_D\})$, where (a) follows from the fact that $l_a = (l_a(1-x_a) + \mathbb{E}_F\{b_{a,n}\}\,x_a)$, for $a \in D$, and $\mathbb{E}_F\{b_{a,n}\} \geq (l_a(1-x_a) + \mathbb{E}_F\{b_{a,n}\}\,x_a)$, for $a \notin D$, and (b) follows from the fact that (x, y) satisfies constraints (12c) (since it is feasible to OCP). The last inequality above contradicts $z^*(B'_D) < z^*(\mathbb{E}_F\{B_n\})$, thus we have that $\max\{x_a : a \in D\} = 1$ for all $D \in \mathcal{D}$, therefore (x, y) is feasible to (11).

Now, let (x, y) be a feasible solution to (11) such that $x_a \in \{0, 1\}$ for all $a \in A$, and that $x_a = 1$ and $y_{S^*} = 1$ for $a \in S^*$ and $S^* \in \mathcal{S}^*(\mathbb{E}_F\{B_n\})$ (one can always select integral solutions to (11), and $\Delta^F_{S^*} = 0$ for all $S^* \in \mathcal{S}^*(\mathbb{E}_F\{B_n\})$, this extra requirement does not affect the solution to (11)). Suppose (x, y) is not feasible to OCP, i.e., there exists some $S \in \mathcal{S}$ such that

$$\sum_{a\in S} \left(l_a(1-x_a) + \mathbb{E}_F\{b_{a,n}\}\,x_a\right) < z^*(\mathbb{E}_F\{B_n\}). \tag{A-12}$$

Let S_0 be one such S that additionally minimizes the left-hand side in (A-12) (in case of ties we pick

any minimizing solution S_0 with smallest value of $|\{a \in S_0 : x_a = 0\}|$). Then $D = \{a \in S_0 : x_a = 0\} \in \mathcal{D}$, which contradicts the feasibility of (x, y) to (11), since if (x, y) is feasible to (11), then we must have $\max\{x_a : a \in D\} \geq 1$ for all $D \in \mathcal{D}$. Thus, we conclude that (x, y) is feasible to OCP.

Summarizing, feasible solutions to OCP are feasible to (11), and integral feasible solutions to (11) that cover all optimal elements in A are feasible to OCP. The result follows from noting that there is always an integral optimal solution to (11) such that $x_a = 1$ and $y_{S^*} = 1$ for $a \in S^*$ for all $S^* \in \mathcal{S}^*(\mathbb{E}_F\{B_n\})$. □

Proof of Theorem 6.2. Similar to the case of the simple policy, regret of the adaptive policy π_a stems from two sources: exploration and errors during exploitation. That is,

$$R^{\pi_a}(F, N) = \sum_{S \in \mathcal{S}} \Delta_S^F \ \mathbb{E}_F\{T_{N+1}(S)\} = R_1^{\pi_a}(F, N) + R_2^{\pi_a}(F, N), \tag{A-13}$$

where $R_1^{\pi_a}(F, N)$ is the exploration-based regret and $R_2^{\pi_a}(F, N)$ is the exploitation-based regret. We prove the result by bounding each term above separately. As in the proof of Theorem 4.3, we drop the dependence of F and π_a on $\mathbb{E}_F$ and $\mathbb{P}_F$.

Step 1 (Exploration-based regret). We begin by setting up some notation. Let $(C_i, \mathcal{E}_i)$ denote the critical subset and exploration set used during cycle i, and for $S \in \mathcal{S}$ define $\Delta T_i(S) := T_{n_{i+1}}(S) - T_{n_i}(S)$. Also, define $i'' := \max\{|A| + 2, i' + 1\}$ where i' is the first cycle in which one is sure to exploit[11], and $U_i := \{C_i \subseteq C_{i-1}\}$, for $i \geq i''$. Using these definitions, one has that

$$\begin{aligned} \frac{R_1^{\pi_a}(F, N)}{\Delta_{max}^F} &\leq n_{i''} + \sum_{i=i''}^{\lceil H \ln N \rceil} \mathbb{E}\left\{\sum_{S \in \mathcal{E}_i} \Delta T_i(S)\right\} \\ &= n_{i''} + \sum_{i=i''}^{\lceil H \ln N \rceil} \left(\mathbb{E}\left\{\sum_{S \in \mathcal{E}_i} \Delta T_i(S) \mathbf{1}\{U_i\}\right\} + \mathbb{E}\left\{\sum_{S \in \mathcal{E}_i} \Delta T_i(S) \mathbf{1}\{U_i^c\}\right\}\right). \end{aligned} \tag{A-14}$$

We bound the exploration-based regret in two steps by bounding the second and third term in (A-14).

Step 1-(a). First, we bound the second term in (A-14). For $i \geq i''$, define the event $\tilde{U}_i := \{(C_i, \mathcal{E}_i) \in \Gamma(\mathbb{E}\{B_n\})\}$. We have that

$$\mathbb{E}\left\{\sum_{S \in \mathcal{E}_i} \Delta T_i(S) \mathbf{1}\{U_i\}\right\} = \mathbb{E}\left\{\sum_{S \in \mathcal{E}_i} \Delta T_i(S) \mathbf{1}\left\{U_i \cap \tilde{U}_i\right\}\right\} + \mathbb{E}\left\{\sum_{S \in \mathcal{E}_i} \Delta T_i(S) \mathbf{1}\left\{U_i \cap \tilde{U}_i^c\right\}\right\}.$$

[11] For instance, we can take $i' := \inf\left\{i \in \mathbb{N}, i \geq 2 : \lfloor e^{i/H} \rfloor > n_{i-1} + i\,|A|\right\}$.

Note that event U_i implies that $\Delta T_i(S) \leq 1$ for all $S \in \mathcal{E}_i$. In addition, the event $\tilde{U}_i$ implies that $|\mathcal{E}_i| \leq G$, where G is the constant in Theorem 6.2. Thus, we conclude that

$$\mathbb{E}\left\{\sum_{S\in\mathcal{E}_i} \Delta T_i(S)\mathbf{1}\left\{U_i \cap \tilde{U}_i\right\}\right\} \leq G.$$

The following proposition, whose proof can be found in Appendix B, bounds the probability of $\tilde{U}_i^c$.

Proposition A.2. *If $i > |A|$, then*

$$\mathbb{P}\left((C_i, \mathcal{E}_i) \notin \Gamma(\mathbb{E}\{B_n\})\right) \leq \tilde{K} \exp\left\{-\tilde{C}(i - |A|)\right\},$$

where $\tilde{C}$ and $\tilde{K}$ are some positive finite constants.

Using the above, one has that

$$\mathbb{E}\left\{\sum_{S\in\mathcal{E}_i} \Delta T_i(S)\mathbf{1}\left\{U_i \cap \tilde{U}_i^c\right\}\right\} \overset{(a)}{\leq} |A|\ \mathbb{P}\left(\tilde{U}_i^c\right) \overset{(b)}{\leq} \tilde{K}\,|A| \exp\left\{-\tilde{C}(i - |A|)\right\},$$

where (a) follows from noting that $\Delta T_i(S) \leq 1$ under U_i and $|\mathcal{E}_i| \leq |A|$, and (b) follows from Proposition A.2.

With the above, one can bound the second term in (A-14) as follows

$$\sum_{i=i''}^{\lceil H \ln N\rceil} \mathbb{E}\left\{\sum_{S\in\mathcal{E}_i} \Delta T_i(S)\mathbf{1}\{U_i\}\right\} \leq G\ (H \ln N + 1) + \tilde{K}\,|A| \sum_{i=i''}^{\infty} \exp\left\{-\tilde{C}(i - |A|)\right\}. \tag{A-15}$$

Note that the second term above is finite.

Step 1-(b). To bound the third term in (A-14), note that $\mathbb{E}\{\Delta T_i(S)\} \leq i$ for all $S \in \mathcal{E}_i$, and that $|\mathcal{E}_i| \leq |A|$, hence

$$\mathbb{E}\left\{\sum_{S\in\mathcal{E}_i} \Delta T_i(S)\mathbf{1}\{U_i^c\}\right\} \leq i\,|A|\,\mathbb{P}\left(U_i^c\right).$$

The following proposition, whose proof can be found in Appendix B, bounds the probability of U_i^c.

Proposition A.3. *If $i \geq i''$, then*

$$\mathbb{P}\{C_i \not\subseteq C_{i-1}\} \leq C' \exp\left\{-\tilde{C}(i - |A| - 1)\right\},$$

where C' is a positive finite constant and $\tilde{C}$ is as in Proposition A.2.

Using the result above and Proposition A.3 we have

$$\sum_{i=i''}^{\lceil H \ln N \rceil} \mathbb{E}\left\{ \sum_{S \in \mathcal{E}_i} \Delta T_i(S) \mathbf{1}\left\{U_i^c\right\} \right\} \leq \sum_{i=i''}^{\infty} i\,|A|\,C' \exp\left\{-\tilde{C}(i-|A|-1)\right\}. \tag{A-16}$$

Thus, combining (A-14), (A-15) and (A-16) we have that

$$\begin{aligned} R_1^{\pi_a}(F,N) &\leq n_{i''}\,\Delta_{max}^F + G\,\Delta_{max}^F\,(H\ln N+1) + \Delta_{max}^F \sum_{i=i''}^{\infty} |A| \left(C'i + \tilde{K}\right) \exp\left\{-\tilde{C}(i-|A|-1)\right\} \\ &= G\,\Delta_{max}^F H \ln N + C_6, \end{aligned} \tag{A-17}$$

where C_6 is a positive finite constant.

Step 2 (Exploitation-based regret). From Proposition A.2 one has that

$$\begin{aligned} R_2^{\pi_a}(F,N) \leq\ & n_{i''}\,\Delta_{max}^F + \sum_{i=i''}^{\infty} (n_{i+1}-n_i)\,\mathbb{P}\left\{\overline{S}_{n_i} \notin \mathcal{S}^*(\mathbb{E}\left\{B_n\right\}),\, \tilde{U}_{i-1}\right\} \Delta_{max}^F \\ &+ \sum_{i=i''}^{\infty} (n_{i+1}-n_i)\,\tilde{K} \exp\left\{-\tilde{C}(i-|A|-1)\right\} \Delta_{max}^F, \end{aligned}$$

where $\overline{S}_{n_i} \in \mathcal{S}^*(\overline{B}_{n_i})$ is any solution with minimum average cost at time n_i. We use the following proposition to bound the second term in the right-hand side of the inequality above.

Proposition A.4. *If $i \geq i''$, then*

$$\mathbb{P}\left\{\overline{S}_{n_i} \notin \mathcal{S}^*(\mathbb{E}\left\{B_n\right\}),\, \tilde{U}_{i-1}\right\} \leq C'' \exp\left\{-\tilde{C}(i-1)\right\},$$

where C'' is a positive finite constant and $\tilde{C}$ is as in Proposition A.2.

From proof of Theorem 4.3, we know that $n_{i+1} - n_i \leq C_2\ e^{\frac{i}{H}}$, for $i \geq i''$, where C_2 is as in the proof of Theorem 4.3. Thus, using the results above and Proposition A.4 one has that

$$R_2^{\pi_a}(F,N) \leq n_{i''}\Delta_{max}^F + C_2\left(\tilde{K}+C''\right)\Delta_{max}^F \sum_{i=i''}^{\infty} \exp\left\{-\tilde{C}(i-|A|-1)+i/H\right\}. \tag{A-18}$$

Therefore, when $1/H < \tilde{C}$, one has that $R_2^{\pi_a}(F,N) \leq C_7$ for some positive finite constant C_7, independent of N.

Finally, combining (A-13), (A-17) and (A-18) results in the following bound

$$R^{\pi_a}(F,N) \leq G\,\Delta_{max}^F H \ln N + C_8,$$

for a positive finite constant C_8, when $1/H < \tilde{C}$. □

Proof of Theorem 7.1. We have that $\Gamma^*(B) \subseteq \Gamma(B)$ (i.e., every optimal solution to OCP is minimal) and the sizes of solutions in $\Gamma(B)$ are $O(|A|)$. Hence, optimal solutions to OCP have sizes that are polynomial in $|A|$ and their objective function can be evaluated in polynomial time. Checking the feasibility of these solutions can be achieved in polynomial time, since checking (12c) can be achieved by solving $f(B_x)$ where $B_x := (b_{a,x} : a \in A)$ for $b_{a,x} := l_a(1 - x_a) + b_a x_a$. This problem is polynomially solvable by assumption. □

B Proof of Auxiliary Results.

Proof of Proposition A.1. Consider $S \subseteq A$ and note that

$$\left\{ \left| \sum_{a \in S} \left(\overline{b}_{a,n} - \mathbb{E}\left\{ b_{a,n} \right\} \right) \right| \geq \epsilon \right\} \subseteq \bigcup_{a \in S} \left\{ \left| \overline{b}_{a,n} - \mathbb{E}\left\{ b_{a,n} \right\} \right| \geq \frac{\epsilon}{|S|} \right\},$$

hence using the union bound one has that

$$\mathbb{P}\left\{ \left| \sum_{a \in S} \left(\overline{b}_{a,n} - \mathbb{E}\left\{ b_{a,n} \right\} \right) \right| \geq \epsilon \,,\, T_n(a) \geq i \,,\, \forall a \in S \right\} \leq \sum_{a \in S} \mathbb{P}\left\{ \left| \overline{b}_{a,n} - \mathbb{E}\left\{ b_{a,n} \right\} \right| \geq \frac{\epsilon}{|S|} \,,\, T_n(a) \geq i \right\}. \tag{B-19}$$

For $m \in \mathbb{N}$ define $t_m(a) := \inf\{ n \in \mathbb{N} : T_n(a) = m \}$. Indexed by m, one has that $b_{a,t_m(a)} - \mathbb{E}\left\{ b_{a,n} \right\} = b_{a,t_m(a)} - \mathbb{E}\left\{ b_{a,t_m(a)} \right\}$ is a bounded martingale difference sequence, thus one has that

$$\begin{aligned}
\mathbb{P}\left\{ \left| \overline{b}_{a,n} - \mathbb{E}\left\{ b_{a,n} \right\} \right| \geq \frac{\epsilon}{|S|} \,,\, T_n(a) \geq i \right\} &= \mathbb{P}\left\{ \left| \sum_{m=1}^{T_n(a)} \left(b_{a,t_m(a)} - \mathbb{E}\left\{ b_{a,n} \right\} \right) \right| \geq \frac{\epsilon \, T_n(a)}{|S|} \,,\, T_n(a) \geq i \right\} \\
&\leq \sum_{k=i}^{\infty} \mathbb{P}\left\{ \left| \sum_{m=1}^{k} \left(b_{a,t_m(a)} - \mathbb{E}\left\{ b_{a,n} \right\} \right) \right| \geq \frac{\epsilon \, k}{|S|} \right\} \\
&\overset{(a)}{\leq} 2 \, \exp\left\{ \frac{-2 \, i\epsilon^2}{|S|^2 \, L^2} \right\} \sum_{k=0}^{\infty} \exp\left\{ \frac{-2 \, k\epsilon^2}{|S|^2 \, L^2} \right\} \\
&\leq 2 \, K \exp\left\{ \frac{-2 \, i\epsilon^2}{|S|^2 \, L^2} \right\},
\end{aligned}$$

where (a) follows from the Hoeffding-Azuma Inequality (see, for example, Cesa-Bianchi and Lugosi (2006, Lemma A.7)), and K is a positive finite constant. Combining the above with (B-19) one has that

$$\mathbb{P}\left\{ \left| \sum_{a \in S} \left(\overline{b}_{a,n} - \mathbb{E}\left\{ b_{a,n} \right\} \right) \right| \geq \epsilon \,,\, T_n(a) \geq i \,,\, \forall a \in S \right\} \leq 2 \, K \, |S| \exp\left\{ \frac{-2 \, i\epsilon^2}{|S|^2 \, L^2} \right\},$$

which is the desired result. □

Proof of Proposition A.2. Remember that $(C_i, \mathcal{E}_i)$ denotes the solution to $OCP(\overline{B}_{n_i})$, that

$\tilde{U}_i = \{(C_i, \mathcal{E}_i) \in \Gamma(\mathbb{E}\{B_n\})\}$, and that $U_i = \{C_i \subseteq C_{i-1}\}$. Note that

$$\mathbb{P}\left\{\tilde{U}_i^c\right\} \leq \sum_{j=i-|A|}^{i} \mathbb{P}\left\{\tilde{U}_i^c\,,\, U_j\right\} + \mathbb{P}\left\{\tilde{U}_i^c\,,\, U_j^c\,,\, j \in \{i-|A|\,,\ldots,i\}\right\}. \tag{B-20}$$

We prove the result in two steps by bounding the first and second terms in the right-hand side of (B-20).

Step 1. Here, we bound the first term in the right-hand side of (B-20). Consider $j \in \{i-|A|\,,\ldots,i\}$ and note that $T_{n_j}(a) \geq i-|A|$ for all $a \in C_j$ under U_j. One has that

$$\mathbb{P}\left\{\tilde{U}_i^c\,,\, U_j\right\} = \mathbb{P}\left\{\tilde{U}_i^c\,,\, (C_j,\mathcal{E}_j) \in \Gamma(\mathbb{E}\{B_n\})\,,\, U_j\right\} + \mathbb{P}\left\{\tilde{U}_i^c\,,\, (C_j,\mathcal{E}_j) \notin \Gamma(\mathbb{E}\{B_n\})\,,\, U_j\right\}. \tag{B-21}$$

We complete the Step 1 by bounding the two terms in the right-hand side of (B-21).

Step 1-(a). The first event in the right-hand side of (B-21) implies that $C_i \neq C_j$, since $(C_i,\mathcal{E}_i) \notin \Gamma(\mathbb{E}\{B_n\})$ under $\tilde{U}_i^c$ and $(C_j,\mathcal{E}_j) \in \Gamma(\mathbb{E}\{B_n\})$. Thus, there exists a $j' \in \{j+1,\ldots,i\}$ such that j' is the first cycle after cycle j at which $(C_{j'},\mathcal{E}_{j'}) \neq (C_j,\mathcal{E}_j)$. Then one concludes that $(C_j,\mathcal{E}_j) \notin \Gamma(\overline{B}_{n_{j'}})$, since at each cycle, the adaptive policy selects the previous exploration set if it is feasible (i.e., if $(C_{j'-1},\mathcal{E}_{j'-1}) \in \Gamma(\overline{B}_{n_{j'}})$). Also note that $T_{n_{j'}}(a) \geq i-|A|$ for all $a \in C_j$, since $i-|A| \leq j \leq j'$. We use the following lemma to bound the first term in the right-hand side of (B-21).

Lemma B.1. *For any fixed $(C,\mathcal{E}) \in \Gamma(\mathbb{E}\{B_n\})$ we have*

$$\left\{(C,\mathcal{E}) \notin \Gamma(\overline{B}_n)\right\} \subseteq \bigcup_{a\in C}\left\{\left|\overline{b}_{a,n} - \mathbb{E}\{b_{a,n}\}\right| \geq \Delta_{\mathcal{C}}/(2\,|A|)\right\},$$

where $\Delta_{\mathcal{C}}$ is a positive finite constant.

Using the discussion above and applying Lemma B.1 by letting $C = C_j \in \mathcal{C}$ and $n = n_{j'}$ we have

$$\begin{aligned}
\mathbb{P}\left\{\tilde{U}_i^c\,,\, (C_j,\mathcal{E}_j) \in \Gamma(\mathbb{E}\{B_n\})\,,\, U_j\right\} &\leq \mathbb{P}\left\{(C_j,\mathcal{E}_j) \notin \Gamma(\overline{B}_{n_{j'}})\,,\, T_{n_{j'}}(a) \geq i-|A|\;,\, a \in C_j\,,\, C_j \in \mathcal{C}\right\} \\
&\leq \sum_{j'=j+1}^{i}\sum_{C\in\mathcal{C}} \mathbb{P}\left\{\bigcup_{a\in C}\left\{\left|\overline{b}_{a,n_{j'}} - \mathbb{E}\{b_{a,n}\}\right| \geq \Delta_{\mathcal{C}}/(2\,|A|)\,,\, T_{n_{j'}}(a) \geq i-|A|\right\}\right\} \\
&\overset{(a)}{\leq} 2\;K'\,|\mathcal{C}|\,|A|^2 \exp\left\{-\tilde{C}_1(i-|A|)\right\},
\end{aligned}$$

where K' is a positive finite constant, $\tilde{C}_1 := (\Delta_{\mathcal{C}})^2/(2\,|A|^2\,L^2)$, and (a) follows from Proposition A.1 and also noting that $i-|A| \leq j \leq j' \leq i$ and $|C| \leq |A|$.

Step 1-(b). We use the following lemma to bound the second term in the right-hand side of (B-21)

Lemma B.2. *For any $(C,\mathcal{E})$ such that $\mathcal{E}$ covers C, and $a \in C$ for all $a \in \bar{S}_{n_j}$ we have*

$$\{(C,\mathcal{E}) \notin \Gamma(\mathbb{E}\{B_n\})\} \cap \left\{\left|\bar{b}_{a,n} - \mathbb{E}\{b_{a,n}\}\right| < \Delta'_{\mathcal{C}}/(2|A|)\,,\, a \in C\right\} \subseteq \left\{(C,\mathcal{E}) \notin \Gamma(\overline{B}_n)\right\},$$

where $\Delta'_{\mathcal{C}}$ is a positive finite constant.

Using Lemma B.2 by letting $C = C_j$ and $n = n_j$, we have

$$\begin{aligned}
\mathbb{P}\left\{\tilde{U}_i^c\,,\,(C_j,\mathcal{E}_j) \notin \Gamma(\mathbb{E}\{B_n\})\,,\,U_j\right\} &\leq \mathbb{P}\{(C_j,\mathcal{E}_j) \notin \Gamma(\mathbb{E}\{B_n\})\,,\,U_j\} \\
&\leq \mathbb{P}\left\{(C_j,\mathcal{E}_j) \notin \Gamma(\mathbb{E}\{B_n\})\,,\,(C_j,\mathcal{E}_j) \in \Gamma(\overline{B}_{n_j})\,,\,T_{n_j}(a) \geq i - |A|\ ,\ a \in C_j\right\} \\
&\leq \sum_{C \in \mathcal{P}^A} \mathbb{P}\left\{\bigcup_{a \in C}\left\{\left|\bar{b}_{a,n_j} - \mathbb{E}\{b_{a,n}\}\right| \geq \Delta'_{\mathcal{C}}/(2|A|)\,,\,T_{n_j}(a) \geq i - |A|\right\}\right\} \\
&\overset{(a)}{\leq} 2\ K''\,|A|\,2^{|A|}\exp\left\{-\tilde{C}_2(i-|A|)\right\},
\end{aligned}$$

where $\mathcal{P}^A$ denotes the power set of A, $\tilde{C}_2 := (\Delta'_{\mathcal{C}})^2/(2|A|^2 L^2)$, K'' is a positive finite constant, and (a) follows from Proposition A.1.

Combining the results in Step 1-(a) and 1-(b) one gets that

$$\sum_{j=i-|A|}^{i} \mathbb{P}\left\{\tilde{U}_i^c\,,\,U_j\right\} \leq 2|A|^2\left(K'|A|\,|\mathcal{C}| + K''2^{|A|}\right)\exp\left\{-\tilde{C}(i-|A|)\right\},$$

where $\tilde{C} := \min\left\{\tilde{C}_1, \tilde{C}_2\right\}$.

Step 2. Here, we bound the second term in the right-hand side of (B-20). Note that U_j^c for $j \in \{i-|A|\,,\ldots,i\}$ implies that there exists a $j'' \in \{i-|A|\,,\ldots,i\}$ such that $T_{n_{j''}}(a) \geq i - |A|$ for all $a \in C_{j''}$. With this observation, one can apply the arguments in Step 1 to show that

$$\mathbb{P}\left\{\tilde{U}_i^c\,,\,U_j^c\,,\,j \in \{i-|A|\,,\ldots,i\}\ ,\,(C_{j''},\mathcal{E}_{j''}) \in \Gamma(\mathbb{E}\{B_n\})\right\} \leq 2\ K'\,|A|^3\,|\mathcal{C}|\exp\left\{-\tilde{C}_1(i-|A|)\right\},$$

and

$$\mathbb{P}\left\{\tilde{U}_i^c\,,\,U_j^c\,,\,j \in \{i-|A|\,,\ldots,i\}\ ,\,(C_{j''},\mathcal{E}_{j''}) \notin \Gamma(\mathbb{E}\{B_n\})\right\} \leq 2\ K''\,|A|^2\,2^{|A|}\exp\left\{-\tilde{C}_2(i-|A|)\right\},$$

where the extra $|A|$ in the right-hand side of the two inequalities above (compared to that in Step 1) comes from the fact that we do not know the exact value of j''.

Combining the above one has that

$$\mathbb{P}\left\{\tilde{U}_i^c\,,\,U_j^c\,,\,j \in \{i-|A|\,,\ldots,i\}\right\} \leq 2|A|^2\left(K'|A|\,|\mathcal{C}| + K''2^{|A|}\right)\exp\left\{-\tilde{C}(i-|A|)\right\}.$$

Combining the results from Steps 1 and 2 one obtains

$$\mathbb{P}\left\{\tilde{U}_i^c\right\} \leq \tilde{K} \exp\left\{-\tilde{C}(i-|A|)\right\},$$

where $\tilde{K}$ is a finite positive constant. □

Proof of Proposition A.3. Remember that $U_i = \{C_i \subseteq C_{i-1}\}$. Then

$$\mathbb{P}\left(U_i{}^c\right) = \mathbb{P}\left(U_i{}^c \cap \tilde{U}_{i-1}\right) + \mathbb{P}\left(U_i{}^c \cap \tilde{U}_{i-1}^c\right) \overset{(a)}{\leq} \mathbb{P}\left((C_{i-1}, \mathcal{E}_{i-1}) \notin \Gamma(\overline{B}_{n_i}), \tilde{U}_{i-1}\right) + \mathbb{P}\left(\tilde{U}_{i-1}^c\right), \quad \text{(B-22)}$$

where (a) follows from the fact that at each cycle, the adaptive policy selects the previous exploration set if it is feasible (i.e., if $(C_{i-1}, \mathcal{E}_{i-1}) \in \Gamma(\overline{B}_{n_i})$). Also note that $T_{n_i}(a) \geq i-1$ for all $a \in C_{i-1}$. By Lemma B.1, one has that

$$\begin{aligned}
\mathbb{P}\left((C_{i-1}, \mathcal{E}_{i-1}) \notin \Gamma(\overline{B}_{n_i}), \tilde{U}_{i-1}\right) &\leq \mathbb{P}\left\{\bigcup_{a \in C_{i-1}} \left\{\left|\overline{b}_{a,n_i} - \mathbb{E}\left\{b_{a,n}\right\}\right| \geq \frac{\Delta_{\mathcal{C}}}{2\,|A|}\right\} \cap \tilde{U}_{i-1}\right\} \\
&\leq \sum_{C \in \mathcal{C}} \mathbb{P}\left\{\bigcup_{a \in C} \left\{\left|\overline{b}_{a,n_i} - \mathbb{E}\left\{b_{a,n}\right\}\right| \geq \frac{\Delta_{\mathcal{C}}}{2\,|A|}\right\}, T_{n_i}(a) \geq i-1\right\} \\
&\overset{(b)}{\leq} 2\; K'\,|\mathcal{C}|\,|A| \exp\left\{-\tilde{C}(i-1)\right\},
\end{aligned}$$

where $\tilde{C}$ is as in the proof of Proposition A.2, K' is a positive finite constant, and (b) follows from applying the union bound and Proposition A.1. Using this and Proposition A.2 to bound the second term in the right-hand side of (B-22) gives

$$\mathbb{P}\left(U_i{}^c\right) \leq \left(\tilde{K} + 2\; K'\,|\mathcal{C}|\,|A|\right) \exp\left\{-\tilde{C}(i-|A|-1)\right\},$$

where $\tilde{K}$ is as in the proof of Proposition A.2. □

Proof of Proposition A.4. When $\tilde{U}_{i-1}$ happens, then $C_{i-1} \in \mathcal{C}$. In particular, all $a \in S^*$ are included in C_{i-1}, for all $S^* \in \mathcal{S}^*(\mathbb{E}\left\{B_n\right\})$. Note that

$$\left\{\overline{S}_{n_i} \notin \mathcal{S}^*(\mathbb{E}\left\{B_n\right\})\right\} \subseteq \bigcup_{a \in C} \left\{\left|\overline{b}_{a,n_i} - \mathbb{E}\left\{b_{a,n}\right\}\right| \geq \Delta_{\mathcal{C}}/(2\,|A|)\right\},$$

for any $C \in \mathcal{C}$, where $\Delta_{\mathcal{C}}$ is as in the proof of Lemma B.1. To prove the statement above, assume that the complement of the right-hand side holds for some fixed $C \in \mathcal{C}$. Then for any $S \in \mathcal{S}$ we

have that

$$\begin{aligned}
\sum_{a\in S}\overline{b}_{a,n_i} &\geq \sum_{a\in C\cap S}\overline{b}_{a,n_i}+\sum_{a\in S\setminus C} l_a \\
&> \sum_{a\in C\cap S}\mathbb{E}\left\{b_{a,n}\right\}+\sum_{a\in S\setminus C} l_a-\Delta_{\mathcal{C}}/2 \\
&\overset{(a)}{\geq} \sum_{a\in S^*}\mathbb{E}\left\{b_{a,n}\right\}+\Delta_{\mathcal{C}}-\Delta_{\mathcal{C}}/2 \\
&> \sum_{a\in S^*}\overline{b}_{a,n_i}-\Delta_{\mathcal{C}}/2+\Delta_{\mathcal{C}}-\Delta_{\mathcal{C}}/2=\sum_{a\in S^*}\overline{b}_{a,n_i},
\end{aligned}$$

where (a) follows from the definition of $\Delta_{\mathcal{C}}$. The last inequality above implies that $\left\{\overline{S}_{n_i}\in\mathcal{S}^*(\mathbb{E}\left\{B_n\right\})\right\}$.

In addition, one has that $T_{n_i}(a)\geq i-1$ for all $a\in C_{i-1}$. Therefore

$$\begin{aligned}
\mathbb{P}\left\{\overline{S}_{n_i}\notin\mathcal{S}^*(\mathbb{E}\left\{B_n\right\}),\,\tilde{U}_{i-1}\right\} &= \mathbb{P}\left\{\overline{S}_{n_i}\notin\mathcal{S}^*(\mathbb{E}\left\{B_n\right\}),\,T_{n_i}(a)\geq i-1\,,\,\forall\, a\in C_{i-1}\,,\,\tilde{U}_{i-1}\right\} \\
&\leq \mathbb{P}\left\{\bigcup_{a\in C_{i-1}}\left\{\left|\overline{b}_{a,n_i}-\mathbb{E}\left\{b_{a,n}\right\}\right|\geq\Delta_{\mathcal{C}}/(2\left|A\right|),\,T_{n_i}(a)\geq i-1\right\}\cap\tilde{U}_{i-1}\right\} \\
&\leq \sum_{C\in\mathcal{C}}\mathbb{P}\left\{\bigcup_{a\in C}\left\{\left|\overline{b}_{a,n_i}-\mathbb{E}\left\{b_{a,n}\right\}\right|\geq\Delta_{\mathcal{C}}/(2\left|A\right|),\,T_{n_i}(a)\geq i-1\right\}\right\} \\
&\leq 2\;K'\left|\mathcal{C}\right|\left|A\right|\exp\left\{-\tilde{C}(i-1)\right\},
\end{aligned}$$

where $\tilde{C}$ is as in the proof of Proposition A.2. □

Proof of Lemma B.1. We prove this lemma by proving the complement. Define

$$\Delta_{\mathcal{C}}:=\inf\left\{\sum_{a\in C\cap S}\mathbb{E}\left\{b_{a,n}\right\}+\sum_{a\in S\setminus C}l_a-\sum_{a\in S^*}\mathbb{E}\left\{b_{a,n}\right\}\,:\,S\in\mathcal{S}\setminus\mathcal{S}^*(\mathbb{E}\left\{B_n\right\}),\,C\in\mathcal{C}\right\},$$

where $S^*\in\mathcal{S}^*(\mathbb{E}\left\{B_n\right\})$. We consider the case of $\Delta_{\mathcal{C}}>0$[12].

Assume that the complement of the right-hand side holds for any fixed $(C,\mathcal{E})\in\Gamma(\mathbb{E}\left\{B_n\right\})$. To prove that the complement of the left-hand side holds, we must show that constraints (12c) are

[12]This extra requirement over F implies that (12c) is never tight for suboptimal $S\in\mathcal{S}$, and can be relaxed provided that, for example, $\mathbb{P}\left\{b_{a,n}=l_a\right\}=0$ for all $a\in A$.

satisfied for all $S \in \mathcal{S}$ when $b_a = \overline{b}_{a.n}$. For any $S \in \mathcal{S}$ in constraints (12c) we have

$$\begin{aligned}\sum_{a\in C\cap S} \overline{b}_{a,n} + \sum_{a\in S\setminus C} l_a &> \sum_{a\in C\cap S} \mathbb{E}\{b_{a,n}\} + \sum_{a\in S\setminus C} l_a - \Delta_{\mathcal{C}}/2 \\ &\overset{(a)}{\geq} \sum_{a\in S^*} \mathbb{E}\{b_{a,n}\} + \Delta_{\mathcal{C}} - \Delta_{\mathcal{C}}/2 \\ &> \sum_{a\in S^*} \overline{b}_{a,n} - \Delta_{\mathcal{C}}/2 + \Delta_{\mathcal{C}} - \Delta_{\mathcal{C}}/2 = \sum_{a\in S^*} \overline{b}_{a,n},\end{aligned}$$

where (a) follows from the definition of $\Delta_{\mathcal{C}}$. The last inequality above concludes the proof. □

Proof of Lemma B.2. Define

$$\Delta'_{\mathcal{C}} := \min\left\{\max\left\{\sum_{a\in S^*} \mathbb{E}\{b_{a,n}\} - \sum_{a\in C\cap S} \mathbb{E}\{b_{a,n}\} - \sum_{a\in S\setminus C} l_a \,:\, S \in \mathcal{S}\right\} \,:\, C \notin \mathcal{C}\right\},$$

where $S^* \in \mathcal{S}^*(\mathbb{E}\{B_n\})$. Note that $\Delta'_{\mathcal{C}} > 0$.

Assume that the left-hand side holds. Since $(C, \mathcal{E}) \notin \Gamma(\mathbb{E}\{B_n\})$, therefore constraint (12c) must be violated for some $S \in \mathcal{S}$, that is there exists at least one $S \in \mathcal{S}$ such that

$$\sum_{a\in C\cap S} \mathbb{E}\{b_{a,n}\} + \sum_{a\in S\setminus C} l_a < \sum_{a\in S^*} \mathbb{E}\{b_{a,n}\},$$

where $S^* \in \mathcal{S}^*(\mathbb{E}\{B_n\})$. Let S' be one such S that additionally minimizes the left-hand side above. Then for such S' we have

$$\begin{aligned}\sum_{a\in C\cap S'} \overline{b}_{a,n} + \sum_{a\in S'\setminus C} l_a &< \sum_{a\in C\cap S'} \mathbb{E}\{b_{a,n}\} + \sum_{a\in S'\setminus C} l_a + \Delta'_{\mathcal{C}}/2 \\ &\overset{(a)}{\leq} \sum_{a\in S^*} \mathbb{E}\{b_{a,n}\} - \Delta'_{\mathcal{C}} + \Delta'_{\mathcal{C}}/2 \\ &< \sum_{a\in S^*} \overline{b}_{a,n} + \Delta'_{\mathcal{C}}/2 - \Delta'_{\mathcal{C}} + \Delta'_{\mathcal{C}}/2 = \sum_{a\in S^*} \overline{b}_{a,n},\end{aligned}$$

where (a) follows from the definition of $\Delta'_{\mathcal{C}}$. The last inequality above implies that constraint (12c) is not satisfied for $S' \in \mathcal{S}$ and thus $(C, \mathcal{E}) \notin \Gamma(\overline{B}_n)$. □