Fast and Accurate Repeated Decision Making

Nicolò Cesa-Bianchi, Tommaso R Cesari, Yishay Mansour, Vanney Perchet

Fast and Accurate Repeated Decision Making

Nicolò Cesa-Bianchi
Università degli Studi di Milano & DSRC

Tommaso Cesari
Toulouse School of Economics (TSE) & Artificial and Natural Intelligence Toulouse Institute (ANITI)

Yishay Mansour
Tel Aviv University & Google research

Vanney Perchet
CREST, ENSAE & Criteo AI Lab, Paris

February 15, 2021

Abstract

We consider the scenario in which a company faces a sequence of decisions to make, such as whether or not to implement certain updates on their systems. Each such decision is usually made based on a so-called *A/B test*, estimating the potential benefits of the update (for simplicity, we focus on a single metric, such as the total benefits, the gross sales, etc.). A crucial feature is that the longer an A/B-test is run, the more accurate its prediction. On the other hand, shorter runs ensure that more updates can be tested out in the same amount of time.

The key questions are then for how long a test should run and how we make a decision at the end, and we introduce and study a model to answer these questions. Specifically, the n-th decision in the sequence is associated to a pair $(\mathcal{D}_n, \mu_n)$ of latent parameters, where $\mathcal{D}_n$ is a distribution over $\mathbb{R}$ and μ_n is the expected gain (or loss) of rolling the update out (say, the percentage change in the considered metric). The company can draw arbitrarily many i.i.d. samples from $\mathcal{D}_n$ by running the A/B test long enough, but the expectation μ_n is never revealed. At the end of the A/B test, the company can either accept the update, gaining μ_n as a reward, or reject it and get zero reward instead. A distinguishing feature of our model is that the performance is measured as the expected cumulative reward divided by the expected cumulative number of drawn samples (similarly to a global ROI over the test sequence). The objective is to get as close as possible to the performance of the optimal policy in a class of appropriately chosen policies. We design an online algorithm with data-dependent theoretical guarantees for finite sets of policies, and analyze its extension to infinite classes of policies. A key technical aspect of this setting, which sets it aside from stochastic bandits, is the impossibility of obtaining unbiased estimates of the policy's performance objective.

1 Introduction

Agents, either companies or individuals, repeatedly make decisions, where finding out the best option takes time and money. Often, we are better off making more decisions quickly and inexpensively, provided these decisions do not have a negative impact. We investigate the question of how an agent should design its decision policy to optimally balance the reward vs. the cost (i.e., the return on investment).

The original motivation. Consider companies, for instance in the technology sector, that keep on testing out innovations in order to increase their favorite metric (whether it is the benefits, the gross revenue, the revenue excluding the traffic acquisition cost, or others). Before deploying them, the company wants to figure out whether the innovations are actually more profitable —with respect to the chosen metric– than the technologies that are currently in place. As long as the “audience” (i.e., the set of users) can be split at random, the company can perform randomized tests and make statistically sound decisions[1]. These tests can not run forever, and decisions must be made at some point. The reason is that only a small number of independent tests can actually be run simultaneously (otherwise it is very difficult, if not impossible, to guarantee the soundness of the empirical results). Moreover, different teams want to try their new ideas concurrently, so test lengths usually have a fixed cap (say, two weeks if enough data are typically gathered in this time window).

More importantly, the sooner a decision is made, the better. Indeed, it is not really the cost of running a single A/B-test that is problematic, but rather keeping a live (negative, neutral, or even borderline positive) test prevents another team to try their ideas that might lead to a much larger increase in the metric. Stated otherwise, it is crucial to learn when to drop barely positive innovations and favor truly positive ones, so to increase the flow of positive improvement over time (i.e., the ROI of the tests).

More generally, our model describes problems where an agent is facing a sequence of tasks, each of which can be accepted or not. Before making each decision, some resources can be invested in order to reduce the uncertainty on the reward of the specific task. As before, the global objective is to maximize the ratio between the expected sum of rewards and the total expected cost (other global objectives can actually be considered, as discussed in Section 6).

The model. Let us index the sequence of tasks (the innovations in the motivating example) over $n \in \mathbb{N}$; a task n is then associated with a pair $(\mathcal{D}_n, \mu_n)$, where $\mathcal{D}_n$ is some unknown probability distribution over $\mathbb{R}$ of expectation μ_n; the latter represents the true *reward* of the n-th task. As a consequence, this number can be either positive or negative (depending on the quality of the innovation), but neither its absolute value nor its sign can be observed directly. Instead, the agent can gather i.i.d. samples from $\mathcal{D}_n$ in order to estimate μ_n, and use this estimate to decide whether the task is worth accepting. Stated otherwise, samples drawn from $\mathcal{D}_n$ represent noisy observations of the true value of the n-th task: getting more samples improves the estimate of μ_n, but also makes the assessment process run longer, with a higher cost.

The agent performance is measured as the total amount of value accumulated by accepting innovations divided by the total amount of requested samples over the sequence of tasks. The setting is made rigorous in Section 3. The choice of the performance measure is discussed in depth in Section 6.

I.I.D. assumption. We assume that the pair $(\mathcal{D}_n, \mu_n)$ associated with the value of the n-th innovation is drawn i.i.d. from an unknown but fixed distribution[2]. This assumption is meaningful if past decisions do not influence future innovations whose global quality remains stable over time. It particularly applies whenever innovations can progress along many orthogonal directions, each yielding a similar added value (for instance, when different teams of the same company test improvements on their own perimeter).

Admittedly, both the state of the agent's system and the one of the environment might evolve over time, but if the ratio of good versus bad innovations remains essentially the same, then this i.i.d. assumption is justified. In other words, it is not necessarily the absolute quality of innovations that remain stationary, but their relative added value given the current state of the system. In practice, this case is fairly frequent, especially when a system is close to its technological limit. Last but not least, algorithms designed under stochastic assumptions often performs surprisingly well in practice, even if i.i.d. assumptions are not fully satisfied or simply hard to check.

A baseline strategy and policy classes. A natural, yet suboptimal, approach for deciding if an innovation is worth accepting is to gather samples sequentially, stopping as soon as the absolute value of their running average surpasses a threshold, and then accepting the innovation if and only if the average is positive. The

[1] This is a bit of an over-simplification, as creating two independent populations is sometimes harder than just making a random split. But in this work we will nevertheless assume that the company manages to get independent feedback.

[2] To be more precise, only μ_n are assumed to be i.i.d.; given μ_n, the actual distribution $\mathcal{D}_n$ is not really important, as long as it is sub-Gaussian or, say, bounded.

major drawback of this approach is that the value μ_n of an innovation n could be arbitrarily close to zero. In this case, the number of samples needed to reliably determine its sign (which is of order $1/\mu_n^2$) becomes arbitrarily and prohibitively large. A very long time would then be invested to assess an innovation whose return is negligible at best. In hindsight, it would have been better to reject the innovation early and move on to the next one. For this reason, testing processes in practice need hard termination rules of the form: if after drawing a certain number of samples no confident decision can be taken, then terminate the testing process and the reject the innovation. Denote by τ this capped early stopping rule and by accept the accept/reject decision rule that comes with it. We say that the pair $\pi = (\tau, \mathrm{accept})$ is a *policy*. Policies defined by capped early stopping rules (see (4) for a formal definition) are of great practical importance [13, 15]. However, policies can be defined more generally by any reasonable pair of *duration* and *decision* functions (formally defined in Section 3). Given a (possibly infinite) set of such policies, and assuming that $\mu_1, \mu_2, \ldots$ are drawn i.i.d. from some unknown but fixed distribution, the goal is to learn efficiently, at the smallest cost, the best policy $\pi_\star$ in the set with respect to a sensible metric. Competing against fixed policy classes is a common modeling choice that allows to express the intrinsic constraints that are imposed by the nature of the decision-making problem. For example, even if some policies outside of the class could theoretically yield a better performance, they might not be implementable because of time, budget, fairness, or technology constraints.

Challenges. One of the biggest challenges arising is that running a decision-making policy generates a collection of samples that —in general— cannot be used to form an unbiased estimate of the policy reward (see the impossibility result in Section 7). The presence of this bias is a significant departure from settings like multiarmed and firing bandits [2, 11], in which an unbiased sample of the target quantity is revealed at the end of each round (see the next section for additional details). Moreover, contrary to standard online learning problems, the relevant performance measure is neither additive in the number of innovations, nor in the number of samples per innovation. Therefore, algorithms have to be analyzed globally, and bandit-like techniques —in which the regret is additive over rounds— cannot be directly applied. We argue that these technical difficulties should not be ignored when defining a plausible setting, applicable to real-life scenarios.

Main contributions. For finite policy sets, we present an algorithm called Capped Policy Elimination (Algorithm 1, CAPE). The algorithm maintains a set of potentially optimal policies and keeps refining it until a single policy is left, or a certain number of innovations have been tested. After that, it uses the best policy in the set to test out all remaining innovations. The need for a cap on the number of policy elimination steps arise from the fact that, in order to gather usable estimates for the performance of our policies, we draw (and pay) extra samples. This use of limited oversampling is a key aspect of our algorithm. We prove high-probability distribution-dependent and distribution-free bounds (Theorem 1) for the performance of CAPE against finite classes of policies. We then show that, if an appropriate preprocessing step (Algorithm 2, ESC), is run before CAPE, the resulting algorithm ESC-CAPE is competitive against infinite sets of policies (Theorem 2).

2 Related Work

Although our formalization of the sequential decision problem is novel, it shares some similarities with prophets/Pandora problems, stochastic bandits, and repeated A/B testing. In this section, we review the relevant literature regarding these two settings and stress the differences with ours.

Relations and differences with prophet inequalities and Pandora's box. The motivating sequential problem described above shares many similarities with other online problems known as prophet inequalities [17, 6, 1] and Pandora's box [25, 14, 7]. In the former, an agent observes sequentially (usually non-negative) random variables $Z_1, \ldots, Z_n$ and decides to stop at some time τ; the reward is then Z_τ (in their original form, prophet inequalities are just stopping time problems). Variants include the possibility of choosing k as opposed to just one random variable (in which case the reward is some function of the selected random variables) and the possibility to partially go back in time. The Pandora's box problem is slightly different;

in its original formulation, the agent has to pay a cost $c_j \geq 0$ to observe the value Z_j (and the order can be chosen if the cost c_j are different, otherwise the order is just random). When the agent stops exploring boxes, her final utility is the maximum of the observed Z_j minus the cumulative cost (or, in other variants, some functions of these numbers).

Similarly to the (general) prophet inequality, the agent in our sequential problem faces random variables ($Z_j = \mu_j$ in our notation) and sequentially selects some of them (as many as she wants, as they can be negative) without the possibility to change her mind and go back in time. However, the difference is that the actual value of μ_j is not observed. This is the connection with Pandora's box; this value can actually be observed but at some price (that roughly scales as $1/\varepsilon^2$ where ε is the required precision). However, the global reward is the cumulative sums (as in prophets) and not the maximum (as in Pandora's box) of the selected variables, normalized by the total cost (as in Pandora's box, but our normalization is multiplicative instead of additive, as it represents a ROI).

Relations and differences with bandits. If the set of all policies (defined rigorously in Section 3) used by the agent to determine whether or not to accept an innovation are thought of as arms, our setting becomes somewhat reminiscent of multi-armed bandits [22, 4, 20]. However, the two problems are significantly different. At the end of each round of a stochastic bandit problem, the agent gets to see an unbiased estimate of the expected reward of the arm played. As this does not happen with innovations (see the impossibility result in Section 7), an off-the-shelf bandit algorithm cannot be directly run to solve our problem. In addition, bandits are typically analyzed under an additive notion of regret, whereas the relevant criterion with innovations —see definition (2)— is not additive. Thus, it is unclear how formal guarantees for bandit algorithms would translate.

Firing bandits [11] could also be seen as a variant of our problem, where μ_n belongs to $[0, 1]$, $\mathcal{D}_n$ are Bernoulli distribution with parameter μ_n, and policies have a very specific form that allows to easily define unbiased estimates of their rewards (which, we remind again, it is not possible in our setting). Furthermore, in firing bandits it is possible to go back and forth in time, sampling from any of the past $\mathcal{D}_n$ and gathering any number of samples from it.

This is a reasonable assumption for the original motivations of firing bandits, as μ_n is thought of as the value of a project in a crowdfunding platform, and drawing samples from $\mathcal{D}_n$ corresponds to displaying projects on web pages. However, when μ_n represents theoretical increment (or decrement) of a company's profit by means of an innovation, it is very unlikely that a company would show new interest in investing into a technology that has been tested before and did not prove to be useful (a killed project is almost never re-launched). Hence, when the sampling of $\mathcal{D}_n$ stops, an irrevocable decision is made. After that, no more samples of can be drawn in the future. Finally, as in multi-armed bandits, the performance criterion is the standard regret, and it is unclear how they control other criteria (like the global ROI).

Relations and differences with repeated A/B testing. As mentioned before, our problem can also be viewed as a framework for repeated A/B testing, in which assessing the value of an innovation corresponds to performing an A/B test. Performing repeated randomized trials for comparing statistical hypotheses dates back to the 1950's [23]. With the advent of internet companies, decision-making algorithms adhering to this paradigm witnessed a new wave of interest, and several variants of this problem have been introduced in recent years [9, 8, 10, 12, 3, 16, 21]. The use of such data-driven sequential decisions processes has been successfully used by companies like Amazon, Bing, Criteo, Facebook, Google, and Uber [3].

A popular metric to optimize sequential A/B tests is the so-called *false discovery rate* (FDR) —see [18, 26] and references therein. Roughly speaking, the FDR is the ratio of accepted μ_n that are negative over the total number of accepted μ_n (or more generally, the number of incorrectly accepted tests over the total number if the metric used at each test changes with time). This unfortunately disregards the relative values of tests μ_n that must be taken into account when optimizing a single metric [5, 19]. Indeed, the effect of many even slightly negative accepted tests could be overcome by a few largely positive ones. For instance, assume that the samples X_t of any distribution $\mathcal{D}_n$ belong to $\{-1, 1\}$, and that their expected value μ_n is uniformly distributed on $\{-\varepsilon, \varepsilon\}$. To control the FDR, each A/B test should be run for approximately $1/\varepsilon^2$ times, yielding a ratio of the average value of an accepted test to the number of samples of order ε^3. A better strategy, using just one sample from each A/B test, is simply to accept μ_n if and only if the first sample is positive. Direct computations shows that this policy, that fits the innovation setting, achieves a significantly better performance of order ε.

Some other A/B testing settings are more closely related, but always at the cost of strong additional assumptions or preliminary knowledge: for example, smoothness assumptions can be made on both both $\mathcal{D}_n$ and the distributions of μ_n [3], or the distribution of μ_n is known, and the distribution of samples belongs to a single parameter exponential family, also known beforehand [21].

3 Preliminaries and Definitions

In this section, we formally introduce the repeated decision-making protocol for an agent that is facing a sequence of decision tasks to be solved back to back as quickly as possible. The goal in each of them is to determine whether an innovation is worth accepting. To achieve this, during each task the agent sequentially observes samples[3] $x_i \in [-1, 1]$ representing realizations of stochastic observations of the current innovation value. A map $\tau \colon [-1, 1]^{\mathbb{N}} \to \mathbb{N}$ is a *duration* (of a decision task) if for all $\boldsymbol{x} \in [-1, 1]^{\mathbb{N}}$, its value $d = \tau(\boldsymbol{x}) \in \mathbb{N}$ at $\boldsymbol{x}$ depends only on the first d components $x_1, x_2, \ldots, x_d$ of $\boldsymbol{x} = (x_1, x_2, \ldots)$; mathematically speaking, τ is a stopping time with respect to the filtration generated by $\boldsymbol{x}$. This definition reflects the fact that the components $x_1, x_2, \ldots$ of the sequence $\boldsymbol{x} = (x_1, x_2, \ldots)$ are generated sequentially, and the decision to stop testing an innovation depends only on what occurred so far. A concrete example of a duration function is the one, mentioned in the introduction and formalized in (4), that keeps drawing samples until the empirical average of the observed values x_i surpasses/falls below a certain threshold, or a maximum number of samples have been drawn.

When a task is concluded, the agent has to make a decision: either accepting or rejecting the current innovation. Formally, we say that a function $\mathrm{accept} \colon \mathbb{N} \times [-1, 1]^{\mathbb{N}} \to \{0, 1\}$ is a *decision* (to accept) if for all $k \in \mathbb{N}$ and $\boldsymbol{x} \in [-1, 1]^{\mathbb{N}}$, its value $\mathrm{accept}(k, \boldsymbol{x}) \in \{0, 1\}$ at $(k, \boldsymbol{x})$ depends only on the first k components $x_1, \ldots, x_k$ of $\boldsymbol{x} = (x_1, x_2, \ldots)$. Again, this definition reflects the fact that the decision $\mathrm{accept}(k, \boldsymbol{x})$ to either accept ($\mathrm{accept}(k, \boldsymbol{x}) = 1$) or reject ($\mathrm{accept}(k, \boldsymbol{x}) = 0$) the current innovation after observing the first k values $x_1, \ldots, x_k$ of $\boldsymbol{x} = (x_1, x_2, \ldots)$ is oblivious to all future observations $x_{k+1}, x_{k+2}, \ldots$. Following up on the concrete example above, the decision function is accepting the current innovation if and only if the the empirical average of the observed values x_i surpasses a certain threshold.[4]

Thus, the two choices that an agent makes in a decision task are when to stop drawing new samples, and whether or not to accept the current innovation. In other words, the behavior of the agent during each task is fully characterized by the choice of a pair $\pi = (\tau, \mathrm{accept})$ that we call a *policy*, where τ is a duration and accept is a decision.

An instance of such a repeated-decision problem is therefore determined by a set of admissible policies $\Pi = \{\pi_k\}_{k \in \mathcal{K}} = \big\{(\tau_k, \mathrm{accept})\big\}_{k \in \mathcal{K}}$ (with $\mathcal{K}$ either finite or countable) and a distribution[5] μ on $[-1, 1]$. Naturally, the former is known beforehand but the latter is unknown and can/should be learned on the fly.

For a fixed choice of such Π and μ, the protocol is formally described below.

For each decision task $n = 1, 2, \ldots$

[3] We assume that samples are supported in $[-1, 1]$ for the sake of simplicity. Our setting as well as all of our results can be extended in a straightforward manner if samples come from (shifted) subgaussian distributions.

[4] Note that, even for decision functions that only look at the mean of the first k values, our definition is significantly more general than simple threshold functions of the form $\mathbb{I}\{\mathrm{mean} \ge \varepsilon_k\}$, as it also includes all decisions of the form $\mathbb{I}\{\mathrm{mean} \in A_k\}$, for all measurable $A_k \subset \mathbb{R}$.

[5] Once again, we assume that μ is supported in $[-1, 1]$ for the sake of simplicity. Our setting as well as all of our results can be extended in a straightforward manner if μ is any (shifted) subgaussian distribution.

1. A sample μ_n (unknown to the agent), that we call *value*,[6] is drawn i.i.d. according to μ and is associated to some unknown probability distribution $\mathcal{D}_n$ of expectation μ_n. It will be used to generate i.i.d. samples $X_{n,i}$, for $i \in \mathbb{N}$.
2. The agent picks $k_n \in \mathcal{K}$ or, equivalently, a policy $\pi_{k_n} = (\tau_{k_n}, \mathrm{accept}) \in \Pi$.
3. The agent draws the first $d_n = \tau_{k_n}(\boldsymbol{X}_n)$ *samples*[7] of the i.i.d. (given μ_n) sequence of random variables $\boldsymbol{X}_n = (X_{n,1}, X_{n,2}, \dots)$.
4. The agent makes the final decision $\mathrm{accept}\big(d_n, \boldsymbol{X}_n\big)$.

We will say that the agent *runs a policy* $\pi_{k_n} = (\tau_{k_n}, \mathrm{accept})$ (on a value μ_n) when steps 2–4 occur. We also say that they accept (resp., rejects) μ_n if their decision at step 4 is equal to 1 (resp., 0). Moreover, we say that the *reward* obtained and the *cost* payed by running a policy $\pi_k = (\tau_k, \mathrm{accept})$ on a value μ_n are, respectively,

$$\mathrm{reward}(\pi_k, \mu_n) = \mu_n \,\mathrm{accept}\big(\tau_k(\boldsymbol{X}_n), \boldsymbol{X}_n\big) \qquad \text{and} \qquad \mathrm{cost}(\pi_k, \mu_n) = \tau_k(\boldsymbol{X}_n) \tag{1}$$

The objective of the agent is to minimize the (cost-normalized) regret R_N after N consecutive tasks, defined as

$$R_N = \sup_{k_0 \in \mathcal{K}} \frac{\mathbb{E}\big[\mathrm{reward}(\pi_{k_0}, \mu_0)\big]}{\mathbb{E}\big[\mathrm{cost}(\pi_{k_0}, \mu_0)\big]} - \frac{\sum_{n=1}^{N} \mathbb{E}\big[\mathrm{reward}(\pi_{k_n}, \mu_n)\big]}{\sum_{m=1}^{N} \mathbb{E}\big[\mathrm{cost}(\pi_{k_m}, \mu_m)\big]} \tag{2}$$

where μ_0 is drawn i.i.d. according to the distribution μ, the random quantities $\boldsymbol{X}_0$, $\mathrm{reward}(\pi_{k_0}, \mu_0)$, and $\mathrm{cost}(\pi_{k_0}, \mu_n)$ are defined as in bullet points 1, 3, and equation (1) respectively (with $n = 0$ and $k = k_0$), and the expectations are taken with respect to μ_n, $\boldsymbol{X}_n$, and (possibly) the random choices of k_n (for $n \geq 0$).

To further lighten notations, we denote the expected rewards and costs of policies π by

$$\mathrm{reward}(\pi) = \mathbb{E}\big[\mathrm{reward}(\pi, \mu_0)\big] \qquad \text{and} \qquad \mathrm{cost}(\pi) = \mathbb{E}\big[\mathrm{cost}(\pi, \mu_0)\big] \tag{3}$$

respectively and we say that $\pi_{k^\star}$ is an *optimal policy* if $k^\star \in \arg\max_{k \in \mathcal{K}} \big(\mathrm{reward}(\pi_k)/\mathrm{cost}(\pi_k)\big)$.

For each policy $(\tau, \mathrm{accept}) \in \Pi$ and all tasks n, we let the agent reject any value regardless of the outcome of the sampling. Formally, the agent can always run the policy $(\tau, 0)$, where the second component of the pair is the decision identically equal to zero.

We also let the agent draw arbitrarily many extra samples in addition to the number $\tau(\boldsymbol{X}_n)$ that they would otherwise draw when running a policy $(\tau, \mathrm{accept}) \in \Pi$ on a value μ_n, provided that these additional samples are not taken into account in their decision to either accept or reject μ_n. Formally, the agent can always draw $\tau(\boldsymbol{X}_n) + k$ many samples (for any $k \in \mathbb{N}$) before making the decision $\mathrm{accept}\big(\tau(\boldsymbol{X}_n), \boldsymbol{X}_n\big)$, where we stress that the first argument of the decision function accept is $\tau(\boldsymbol{X}_n)$ and not $\tau(\boldsymbol{X}_n) + k$.

Note that invoking the power to reject a value μ_n after observing $\tau(\boldsymbol{X}_n)$ samples increases the cost of sampling in the denominator of (2) by $\mathbb{E}\big[\tau(\boldsymbol{X}_n)\big]$ while adding no reward to the numerator. Similarly, drawing k extra samples without using them to make the decision has no effect on the numerator but increases the cost in the denominator by k. For these reasons, doing any of these might seem utterly counterproductive. It will become apparent later that rejecting some of the values is indeed mostly a theoretical technique useful to get a cleaner analysis. However, we will show that a carefully designed use of oversampling is crucial for building unbiased estimates of the rewards of our policies, a task which is impossible without oversampling (for more details, see Section 7).

4 Competing Against the Best Policy (CAPE)

As described in the introduction, the duration of a decision task is usually defined by a capped early-stopping rule —e.g., drawing samples until 0 falls outside of a confidence interval around the empirical average, or a

[6] Since it represents of the value of the current innovation being tested.

[7] Note that given μ_n, the random variable d_n is a stopping time with respect to the natural filtration associated to the stochastic process $\boldsymbol{X}_n$ (by definition of duration).

maximum number of draws has been reached. More precisely, if N tasks have to be performed, one could consider the natural policy class $\big\{(\tau_k, \text{accept})\big\}_{k\in\{1,\dots,K\}}$ given by

$$\tau_k(\boldsymbol{x}) = \min\left(k,\ \inf\left\{n\in\mathbb{N} : |\overline{x}_n| \ge c\sqrt{\frac{\ln\frac{KN}{\delta}}{n}}\right\}\right), \qquad \text{accept}(n,\boldsymbol{x}) = \mathbb{I}\left\{\overline{x}_n \ge c\sqrt{\frac{\ln\frac{KN}{\delta}}{n}}\right\} \tag{4}$$

for some $c > 0$ and $\delta \in (0, 1)$, where $\overline{x}_n = ({}^1\!/{}_n)\sum_{i=1}^n x_i$ is the average of the first n elements of the sequence $\boldsymbol{x} = (x_1, x_2, \dots)$.

In this section we generalize this notion and we present an algorithm with provable regret guarantees against finite families of policies. Formally, we focus on set of policies $\Pi = \{\pi_k\}_{k\in\{1,\dots,K\}} = \big\{(\tau_k, \text{accept})\big\}_{k\in\{1,\dots,K\}}$ where accept is an arbitrary decision and $\tau_1, \dots, \tau_K$ is any sequence of durations which, for the sake of convenience, we assume to be sorted by index ($\tau_k \le \tau_h$ if $k \le h$) and bounded ($\tau_k \le D_k$ for all k, with $D_k \le D_h$ if $k \le h$). We now present a simple and efficient algorithm (Algorithm 1, CAPE) that achieves vanishing regret (with high probability) against finite families of policies. We will later discuss how to extend the analysis even further, including countable families of polices.

Our algorithm performs policy elimination (lines 1–5) for a certain number of tasks (line 1) or until a single policy is left (line 6). After that, it runs the best policy left in the set (line 7) for all remaining tasks. During each policy elimination step, the algorithm oversamples (line 2) by drawing twice as many samples as it would suffice to take its decision $\text{accept}\big(\tau_{\max(C_n)}(\boldsymbol{X}_n), \boldsymbol{X}_n\big)$ (at line 3). These extra samples are used to compute rough estimates of rewards and costs of all potentially optimal policies and more specifically to build *unbiased* estimates of these rewards (which, we recall, we would not otherwise have access to). The test at line 4 has the only purpose of ensuring that the denominators $\widehat{c}_n^{\,-}(k)$ at line 5 are bounded away from zero, so that all quantities are well-defined.

As usual in online learning, the *gap* in performances between optimal and sub-optimal policies serves as a complexity parameter. We define it as $\Delta = \min_{k\ne k^\star} \frac{\text{reward}(\pi_{k^\star})}{\text{cost}(\pi_{k^\star})} - \frac{\text{reward}(\pi_k)}{\text{cost}(\pi_k)}$ where we recall that $k^\star \in \arg\max_k\big(\text{reward}(\pi_k)/\text{cost}(\pi_k)\big)$ is the index of an optimal policy. Conventionally, we set $1/\Delta = \infty$ if $\Delta = 0$.

Theorem 1. *If Π is finite and durations are uniformly bounded by some $D \in \mathbb{N}$, then Algorithm 1, run with $N_{\text{ex}} = \lceil N^{2/3}\rceil$ and $\delta \in (0, 1)$ has a cost-normalized regret satisfying, with probability at least $1-\delta$,*

$$R_N = \widetilde{\mathcal{O}}\left(\min\left(\frac{D^3}{\Delta^2\,N}, \frac{D}{N^{1/3}}\right)\right) \tag{8}$$

as soon as $N \ge D^3$ (where the $\widetilde{\mathcal{O}}$ notation hides only logarithmic terms, including a $\log(1/\delta)$ term).

Note that the smaller the D, the smaller the regret bound. This is not surprising. Indeed, a small D limits the effective number of policies, which in turn worsens the benchmark in the definition of regret (2). In the extreme case $D = 1$, all policies become optimal, because they all collapse into a unique policy $\pi_1 = (1, \text{accept})$, that collects exactly one sample and accepts accordingly. We now sketch the analysis of CAPE (the missing technical details are deferred to Section 4.1).

Proof sketch. This theorem relies on four technical lemmas (Lemmas 1-4) whose proofs are deferred to the next section. Note that durations are uniformly bounded by D_K, and D_K is the smallest uniform bound on all these durations. Thus, without loss of generality, we prove the result for $D = D_K$.

With a concentration argument (Lemma 1), one can leverage the definitions of $\widehat{r}_n^{\,\pm}(k), \widehat{c}_n^{\,\pm}(k)$ and the i.i.d. assumptions on the samples $X_{n,i}$ to show that, with probability at least $1-\delta$, the event

$$\widehat{r}_n^{\,-}(k) \le \text{reward}(\pi_k) \le \widehat{r}_n^{\,+}(k) \qquad \text{and} \qquad \widehat{c}_n^{\,-}(k) \le \text{cost}(\pi_k) \le \widehat{c}_n^{\,+}(k) \tag{9}$$

Algorithm 1: Capped Policy Elimination (CAPE)

```
Input: finite policy set $\Pi$, number of tasks $N$, confidence $\delta$, exploration cap $N_{\mathrm{ex}}$
Initialization: let $C_1 \leftarrow \{1, \dots, K\}$ be the set of indices of all currently optimal candidates
for task $n = 1, \dots, N_{\mathrm{ex}}$ do
    draw the first $2D_{\max(C_n)}$ samples $X_{n,1}, \dots, X_{n,2D_{\max(C_n)}}$ of $\boldsymbol{X}_n$
    make the decision $\mathrm{accept}\big(\tau_{\max(C_n)}(\boldsymbol{X}_n), \boldsymbol{X}_n\big)$
    if $n \ge 2D_K^2 \ln(4KN_{\mathrm{ex}}/\delta)$ then
        let $C_{n+1} \leftarrow C_n \setminus C'_n$, where
```

$$C'_n = \Bigg\{ k \in C_n \;:\; \left(\widehat{r}_n^{\,+}(k) \ge 0 \text{ and } \frac{\widehat{r}_n^{\,+}(k)}{\widehat{c}_n^{\,-}(k)} < \frac{\widehat{r}_n^{\,-}(j)}{\widehat{c}_n^{\,+}(j)}, \text{ for some } j \in C_n \right)$$
$$\text{or } \left(\widehat{r}_n^{\,+}(k) < 0 \text{ and } \frac{\widehat{r}_n^{\,+}(k)}{\widehat{c}_n^{\,+}(k)} < \frac{\widehat{r}_n^{\,-}(j)}{\widehat{c}_n^{\,-}(j)}, \text{ for some } j \in C_n \right) \Bigg\}$$

$$\widehat{r}_n^{\,\pm}(k) = \frac{1}{n} \sum_{m=1}^{n} \sum_{i=1}^{D_{\max(C_m)}} \frac{X_{m, D_{\max(C_m)}+i}}{D_{\max(C_m)}} \, \mathrm{accept}\big(\tau_k(\boldsymbol{X}_m), \boldsymbol{X}_m\big) \pm \sqrt{\frac{2}{n} \ln \frac{4KN_{\mathrm{ex}}}{\delta}} \tag{5}$$

$$\widehat{c}_n^{\,\pm}(k) = \frac{1}{n} \sum_{m=1}^{n} \tau_k(\boldsymbol{X}_m) \pm (D_k - 1) \sqrt{\frac{1}{2n} \ln \frac{4KN_{\mathrm{ex}}}{\delta}} \tag{6}$$

```
    if $|C_{n+1}| = 1$ then let $\widehat{r}_{N_{\mathrm{ex}}}^{\,\pm}(k) \leftarrow \widehat{r}_n^{\,\pm}(k)$, $\widehat{c}_{N_{\mathrm{ex}}}^{\,\pm}(k) \leftarrow \widehat{c}_n^{\,\pm}(k)$, $C_{N_{\mathrm{ex}}+1} \leftarrow C_{n+1}$, break
run policy $\pi_{k'}$ for all remaining tasks, where
```

$$k' \in \begin{cases} \operatorname*{arg\,max}_{k \in C_{N_{\mathrm{ex}}+1}} \big(\widehat{r}_{N_{\mathrm{ex}}}^{\,+}(k) / \widehat{c}_{N_{\mathrm{ex}}}^{\,-}(k) \big) & \text{if } \widehat{r}_{N_{\mathrm{ex}}}^{\,+}(k) \ge 0 \text{ for some } k \in C_{N_{\mathrm{ex}}+1} \\[2ex] \operatorname*{arg\,max}_{k \in C_{N_{\mathrm{ex}}+1}} \big(\widehat{r}_{N_{\mathrm{ex}}}^{\,+}(k) / \widehat{c}_{N_{\mathrm{ex}}}^{\,+}(k) \big) & \text{if } \widehat{r}_{N_{\mathrm{ex}}}^{\,+}(k) < 0 \text{ for all } k \in C_{N_{\mathrm{ex}}+1} \end{cases} \tag{7}$$

occurs simultaneously for all $n \le N_{\mathrm{ex}}$ and all $k \le \max(C_n)$. In order to avoid repetitions, from here on out we assume that all subsequent statements hold over the common high-probability event (9),

If $\Delta > 0$ (i.e., if there is a unique optimal policy), we then obtain (Lemma 2) that suboptimal policies are eliminated after at most N'_{ex} tasks, where $N'_{\mathrm{ex}} \le 288\, D_K^2 \ln(4KN_{\mathrm{ex}}/\delta)/\Delta^2 + 1$. To prove it we upper bound the length of the confidence interval for $\mathrm{reward}(\pi_k)/\mathrm{cost}(\pi_k)$:

$$\left[\frac{\widehat{r}_n^-(k)}{\widehat{c}_n^+(k)}\mathbb{I}\bigl\{\widehat{r}_n^+(k) \ge 0\bigr\} + \frac{\widehat{r}_n^-(k)}{\widehat{c}_n^-(k)}\mathbb{I}\bigl\{\widehat{r}_n^+(k) < 0\bigr\},\ \frac{\widehat{r}_n^+(k)}{\widehat{c}_n^-(k)}\mathbb{I}\bigl\{\widehat{r}_n^+(k) \ge 0\bigr\} + \frac{\widehat{r}_n^+(k)}{\widehat{c}_n^+(k)}\mathbb{I}\bigl\{\widehat{r}_n^+(k) < 0\bigr\}\right]$$

and we compute an N'_{ex} such that this upped bound is smaller than $\Delta/2$.

Afterwards, we analyze separately the case in which the test at line 6 is true for some task $N'_{\mathrm{ex}} \le N_{\mathrm{ex}}$ and its complement (i.e., when the test is always false).

In the first case, by (9) there exists a unique optimal policy, i.e., we have that $\Delta > 0$. We can therefore apply the bound above on N'_{ex}, obtaining a deterministic upper bound N''_{ex} on the number N'_{ex} of tasks needed to identify the optimal policy. Using this upper bound, writing the definition of regret, and further upper bounding (Lemma 3) yields

$$R_N \le \min\left(\frac{(2D_K+1)N_{\mathrm{ex}}}{N},\ \frac{(2D_K+1)\bigl(288\,(D_K/\Delta)^2\ln(4KN_{\mathrm{ex}}/\delta)+1\bigr)}{N}\right) \tag{10}$$

Finally, we consider the case in which the test at line 6 is false for all tasks $n \le N_{\mathrm{ex}}$, and line 7 is executed with $C_{N_{\mathrm{ex}}+1}$ containing two or more policies. The key idea here is to use the definition of k' in Equation (7) to lower-bound $\mathrm{reward}(\pi_{k'})$ in terms of $\mathrm{reward}(\pi_{k^\star})/\mathrm{cost}(\pi_{k^\star})$. This, together with some additional technical estimations (Lemma 4) leads to the result. □

4.1 Technical Lemmas

In this section, we give formal proofs of all results needed to prove Theorem 1.

Lemma 1. *Under the assumptions of Theorem 1, the event*

$$\widehat{r}_n^-(k) \le \mathrm{reward}(\pi_k) \le \widehat{r}_n^+(k) \qquad \textit{and} \qquad \widehat{c}_n^-(k) \le \mathrm{cost}(\pi_k) \le \widehat{c}_n^+(k) \tag{11}$$

occurs simultaneously for all $n = 1, \dots, N_{\mathrm{ex}}$ and all $k = 1, \dots, \max(C_n)$ with probability at least $1-\delta$.

Proof. Let, for all n, k,

$$\varepsilon_n = \sqrt{\frac{\ln(4KN_{\mathrm{ex}}/\delta)}{2n}}, \qquad \overline{r}_n(k) = \widehat{r}_n^+(k) - 2\varepsilon_n, \qquad \overline{c}_n(k) = \widehat{c}_n^+(k) - (D_k - 1)\varepsilon_n \tag{12}$$

Note that $\overline{c}_n(k)$ is the empirical average of n i.i.d. samples of $\mathrm{cost}(\pi_k)$ for all n, k by definitions (12), (6), (1), (3), and point 3 in the formal definition of our protocol (Section 3). We show now that $\overline{r}_n(k)$ is the empirical average of n i.i.d. samples of $\mathrm{reward}(\pi_k)$ for all n, k; then claim (9) follows by Hoeffding's inequality. Indeed, by the conditional independence of the samples and being $\mathrm{accept}(k, \boldsymbol{x})$ independent of the variables $(x_{k+1}, x_{k+2}, \dots)$ by definition, for all tasks n, all policies $k \in C_n$, and all $i > D_{\max(C_n)}$ ($\ge D_k$ by monotonicity of $k \mapsto D_k$),

$$\begin{aligned}
\mathbb{E}\left[X_{n,i}\,\mathrm{accept}\bigl(\tau_k(\boldsymbol{X}_n), \boldsymbol{X}_n\bigr) \,\Big|\, \mu_n\right] &= \mathbb{E}\left[X_{n,i} \mid \mu_n\right]\mathbb{E}\Big[\mathrm{accept}\bigl(\tau_k(\boldsymbol{X}_n), \boldsymbol{X}_n\bigr) \,\Big|\, \mu_n\Big] \\
&= \mu_n\,\mathbb{E}\Big[\mathrm{accept}\bigl(\tau_k(\boldsymbol{X}_n), \boldsymbol{X}_n\bigr) \,\Big|\, \mu_n\Big] \\
&= \mathbb{E}\Big[\mu_n\,\mathrm{accept}\bigl(\tau_k(\boldsymbol{X}_n), \boldsymbol{X}_n\bigr) \,\Big|\, \mu_n\Big]
\end{aligned}$$

Taking expectations with respect to μ_n on both sides of the above, and recalling definitions (12), (5), (1), (3), (3) proves the claim. Thus, Hoeffding's inequality implies, for all fixed n, k,

$$\mathbb{P}\big(\widehat{r}_n^-(k) \le \mathrm{reward}(\pi_k) \le \widehat{r}_n^+(k)\big) = \mathbb{P}\Big(\big|\overline{r}_n(k) - \mathrm{reward}(\pi_k)\big| \le 2\varepsilon_n\Big) \ge 1 - \frac{\delta}{2KN_{\mathrm{ex}}}$$
$$\mathbb{P}\big(\widehat{c}_n^-(k) \le \mathrm{cost}(\pi_k) \le \widehat{c}_n^+(k)\big) = \mathbb{P}\Big(\big|\overline{c}_n(k) - \mathrm{cost}(\pi_k)\big| \le (D_K - 1)\varepsilon_n\Big) \ge 1 - \frac{\delta}{2KN_{\mathrm{ex}}}$$

Applying a union bound shows that event (9) occurs simultaneously for all $n \in \{1, \dots, N_{\mathrm{ex}}\}$ and $k \in \{1, \dots, \max(C_n)\}$ with probability at least $1 - \delta$. □

Lemma 2. *Under the assumptions of Theorem 1, if the event* (11) *occurs simultaneously for all* $n = 1, \dots, N_{\mathrm{ex}}$ *and all* $k = 1, \dots, \max(C_n)$*, and* $\Delta > 0$*, (i.e., if there is a unique optimal policy), then all suboptimal policies are eliminated after at most* N'_{ex} *tasks, where*

$$N'_{\mathrm{ex}} \le \frac{288\, D_K^2 \ln(4KN_{\mathrm{ex}}/\delta)}{\Delta^2} + 1 \tag{13}$$

Proof. Note first that (11) implies, for all $n \ge 2D_K^2 \ln(4KN_{\mathrm{ex}}/\delta)$ (guaranteed by line 5) and all $k \in C_n$

$$\frac{\widehat{r}_n^-(k)}{\widehat{c}_n^+(k)} \le \frac{\mathrm{reward}(\pi_k)}{\mathrm{cost}(\pi_k)} \le \frac{\widehat{r}_n^+(k)}{\widehat{c}_n^-(k)} \qquad \text{if } \widehat{r}_n^+(k) \ge 0$$
$$\frac{\widehat{r}_n^-(k)}{\widehat{c}_n^-(k)} \le \frac{\mathrm{reward}(\pi_k)}{\mathrm{cost}(\pi_k)} \le \frac{\widehat{r}_n^+(k)}{\widehat{c}_n^+(k)} \qquad \text{if } \widehat{r}_n^+(k) < 0$$

In other words, the interval

$$\left[\frac{\widehat{r}_n^-(k)}{\widehat{c}_n^+(k)}\mathbb{I}\big\{\widehat{r}_n^+(k) \ge 0\big\} + \frac{\widehat{r}_n^-(k)}{\widehat{c}_n^-(k)}\mathbb{I}\big\{\widehat{r}_n^+(k) < 0\big\},\ \frac{\widehat{r}_n^+(k)}{\widehat{c}_n^-(k)}\mathbb{I}\big\{\widehat{r}_n^+(k) \ge 0\big\} + \frac{\widehat{r}_n^+(k)}{\widehat{c}_n^+(k)}\mathbb{I}\big\{\widehat{r}_n^+(k) < 0\big\}\right]$$

is a confidence interval for the value $\mathrm{reward}(\pi_k)/\mathrm{cost}(\pi_k)$ that measures the performance of π_k. Let, for all n, k,

$$\varepsilon_n = \sqrt{\frac{\ln(4KN_{\mathrm{ex}}/\delta)}{2n}}, \qquad \overline{r}_n(k) = \widehat{r}_n^+(k) - 2\varepsilon_n, \qquad \overline{c}_n(k) = \widehat{c}_n^+(k) - (D_k - 1)\varepsilon_n \tag{14}$$

If $\widehat{r}_n^+(k) \ge 0$, by the definitions in (14), the length of this confidence interval is

$$\frac{\overline{r}_n(k) + 2\varepsilon_n}{\overline{c}_n(k) - (D_k - 1)\varepsilon_n} - \frac{\overline{r}_n(k) - 2\varepsilon_n}{\overline{c}_n(k) + (D_k - 1)\varepsilon_n} = \frac{2\varepsilon_n\big(2\,\overline{c}_n(k) + (D_k - 1)\,\overline{r}_n(k)\big)}{\overline{c}_n(k)^2 - (D_k - 1)^2\,\varepsilon_n^2} \le 12\, D_K \varepsilon_n$$

where for the numerator we used the fact that $\overline{c}_n(k)$ (resp., $\overline{r}_n(k)$) is an average of random variables all upper bounded by D_k (resp., 1) and the denominator is lower bounded by 1/2 because $\overline{c}_n(k)^2 \ge 1$, $(D_k^2 - 1)\,\varepsilon_n^2 \le 1/2$ by $n \ge 2D_K^2 \ln(4KN_{\mathrm{ex}}/\delta)$ (line 4), and $D_k/D_K \le 1$ (by monotonicity of $k \mapsto D_k$). Similarly, if $\widehat{r}_n^+(k) < 0$, the length of the confidence interval is

$$\frac{\overline{r}_n(k) + 2\varepsilon_n}{\overline{c}_n(k) + (D_k - 1)\varepsilon_n} - \frac{\overline{r}_n(k) - 2\varepsilon_n}{\overline{c}_n(k) - (D_k - 1)\varepsilon_n} = \frac{2\varepsilon_n\big(2\,\overline{c}_n(k) - (D_k - 1)\,\overline{r}_n(k)\big)}{\overline{c}_n(k)^2 - (D_k - 1)^2\,\varepsilon_n^2} \le 12\, D_K \varepsilon_n$$

where, in addition to the considerations above, we used $0 < -\widehat{r}_n^+(k) < -\overline{r}_n(k) \le 1$. Hence, as soon as the upper bound $12\, D_K \varepsilon_n$ on the length of each of the confidence interval above falls below $\Delta/2$, all such intervals are guaranteed to be disjoint and by definition of C_n (line 5) all suboptimal policies are guaranteed to have left C_{n+1}. In formulas, this happens at the latest during task n, where $n \ge 2D_K^2 \ln(4KN_{\mathrm{ex}}/\delta)$ satisfies

$$12\, D_K \varepsilon_n < \frac{\Delta}{2} \iff n > 288\,(D_K/\Delta)^2 \ln(4KN_{\mathrm{ex}}/\delta)$$

This proves the result. □

Lemma 3. *Under the assumptions of Theorem 1, if the event* (11) *occurs simultaneously for all* $n = 1, \dots, N_{\text{ex}}$ *and all* $k = 1, \dots, \max(C_n)$*, and the test at line 6 is true for some* $N'_{\text{ex}} \le N_{\text{ex}}$*, then*

$$R_N \le \min\left(\frac{(2D_K + 1)N_{\text{ex}}}{N}, \; \frac{(2D_K+1)\big(288\,(D_K/\Delta)^2 \ln(4KN_{\text{ex}}/\delta) + 1\big)}{N} \right) \tag{15}$$

Proof. Note that if the test at line 6 is true, than by (11) there exists a unique optimal policy, i.e., we have $\Delta > 0$. We can therefore apply Lemma 2, obtaining a deterministic upper bound N''_{ex} on the number N'_{ex} of tasks needed to identify the optimal policy, where

$$N''_{\text{ex}} = \min\left(N_{\text{ex}}, \; \frac{128\, D_K^2 \ln(4KN_{\text{ex}}/\delta)}{\Delta^2} + 1 \right)$$

The total expected reward of Algorithm 1 divided by its total expected cost is lower bounded by

$$\xi = \frac{\mathbb{E}\left[-N'_{\text{ex}} + \sum_{n=N'_{\text{ex}}+1}^{N} \text{reward}(\pi_{k^\star}, \mu_n) \right]}{\mathbb{E}\left[2\sum_{m=1}^{N'_{\text{ex}}} D_{\max(C_m)} + \sum_{n=N'_{\text{ex}}+1}^{N} \text{cost}(\pi_{k^\star}, \mu_n) \right]}$$

If $\xi < 0$, we can further lower bound it by

$$\frac{(N - N''_{\text{ex}})\,\text{reward}(\pi_{k^\star}) - N''_{\text{ex}}}{(N - N''_{\text{ex}})\,\text{cost}(\pi_{k^\star}) + 2N''_{\text{ex}}} \ge \frac{\text{reward}(\pi_{k^\star})}{\text{cost}(\pi_{k^\star})} - \frac{3N''_{\text{ex}}}{N}$$

where the inequality follows by $(a-b)/(c+d) \ge a/c - (d+b)/(c+d)$ for all $a, b, c, d \in \mathbb{R}$ with $0 \ne c > -d$ and $a/c \le 1$, and then using $c + d \ge N$ which holds because $\text{cost}(\pi_{k^\star}) \ge 1$. Similarly, if $\xi \ge 0$, we can further lower bound it by

$$\frac{(N - N''_{\text{ex}})\,\text{reward}(\pi_{k^\star}) - N''_{\text{ex}}}{(N - N''_{\text{ex}})\,\text{cost}(\pi_{k^\star}) + 2D_K N''_{\text{ex}}} \ge \frac{\text{reward}(\pi_{k^\star})}{\text{cost}(\pi_{k^\star})} - \frac{(2D_K + 1)N''_{\text{ex}}}{N}$$

Thus, the result follows by $D_K \ge 1$ and the definition of N''_{ex}. □

Lemma 4. *Under the assumptions of Theorem 1, if the event* (11) *occurs simultaneously for all* $n = 1, \dots, N_{\text{ex}}$ *and all* $k = 1, \dots, \max(C_n)$*, and the test at line 6 is false for all tasks* $n \le N_{\text{ex}}$ *(i.e., if line 7 is executed with* $C_{N_{\text{ex}}+1}$ *containing two or more policies), then*

$$R_T \le (D_K + 1)\sqrt{\frac{8\ln(4KN_{\text{ex}}/\delta)}{N_{\text{ex}}}} + \frac{(2D_K + 1)N_{\text{ex}}}{N}$$

Proof. Note first that by (11) and the definition of C_n (line 5), all optimal policies belong to $C_{N_{\text{ex}}+1}$. Let, for all n, k,

$$\varepsilon_n = \sqrt{\frac{\ln(4KN_{\text{ex}}/\delta)}{2n}}, \qquad \overline{r}_n(k) = \widehat{r}_n^{\,+}(k) - 2\varepsilon_n, \qquad \overline{c}_n(k) = \widehat{c}_n^{\,+}(k) - (D_k - 1)\varepsilon_n \tag{16}$$

By (11) and the definitions of k', $\widehat{r}_n^{\,\pm}(k)$, and ε_n (line 7, (5), (5), and (16) respectively), for all optimal policies $\pi_{k^\star}$, if $\widehat{r}_{N_{\text{ex}}}^{\,+}(k^\star) \ge 0$, then also $\widehat{r}_{N_{\text{ex}}}^{\,+}(k') \ge 0$[8] and

$$\begin{aligned} \frac{\text{reward}(\pi_{k^\star})}{\text{cost}(\pi_{k^\star})} \le \frac{\widehat{r}_{N_{\text{ex}}}^{\,+}(k^\star)}{\widehat{c}_{N_{\text{ex}}}^{\,-}(k^\star)} \le \frac{\widehat{r}_{N_{\text{ex}}}^{\,+}(k')}{\widehat{c}_{N_{\text{ex}}}^{\,-}(k')} &\le \frac{\text{reward}(\pi_{k'}) + 4\varepsilon_n}{\text{cost}(\pi_{k'}) - 2(D_{k'} - 1)\varepsilon_n} \\ &\le \frac{\text{reward}(\pi_{k'})}{\text{cost}(\pi_{k'})} + \frac{2(D_{k'} + 1)\varepsilon_n}{\text{cost}(\pi_{k'}) - 2(D_{k'} - 1)\varepsilon_n} \end{aligned}$$

[8] Indeed, $k' \in \arg\max_{k \in C_{N_{\text{ex}}+1}} \big(\widehat{r}_{N_{\text{ex}}}^{\,+}(k) / \widehat{c}_{N_{\text{ex}}}^{\,-}(k) \big)$ in this case, and $\widehat{r}_{N_{\text{ex}}}^{\,+}(k') \ge 0$ follows by the two inequalities $\widehat{r}_{N_{\text{ex}}}^{\,+}(k') / \widehat{c}_{N_{\text{ex}}}^{\,-}(k') \ge \widehat{r}_{N_{\text{ex}}}^{\,+}(k^\star) / \widehat{c}_{N_{\text{ex}}}^{\,-}(k^\star) \ge 0$.

where all the denominators are positive because $N_{\mathrm{ex}} \ge 8(D_K - 1)^2 \ln(4KN_{\mathrm{ex}}/\delta)$ and the last inequality follows by $(a+b)/(c-d) \le a/c + (d+b)/(c-d)$ for all $a \le 1$, $b \in \mathbb{R}$, $c \ge 1$, and $d < c$; next, if $\widehat{r}^{+}_{N_{\mathrm{ex}}}(k^\star) < 0$ but $\widehat{r}^{+}_{N_{\mathrm{ex}}}(k') \ge 0$ the exact same chain of inequalities hold; finally, if both $\widehat{r}^{+}_{N_{\mathrm{ex}}}(k^\star) < 0$ and $\widehat{r}^{+}_{N_{\mathrm{ex}}}(k') < 0$, then $\widehat{r}^{+}_{N_{\mathrm{ex}}}(k) < 0$ for all $k \in C_{N_{\mathrm{ex}}+1}$[9], hence, by definition of k' and the same arguments used above

$$\begin{aligned}
\frac{\mathrm{reward}(\pi_{k^\star})}{\mathrm{cost}(\pi_{k^\star})} \le \frac{\widehat{r}^{+}_{N_{\mathrm{ex}}}(k^\star)}{\widehat{c}^{+}_{N_{\mathrm{ex}}}(k^\star)} \le \frac{\widehat{r}^{+}_{N_{\mathrm{ex}}}(k')}{\widehat{c}^{+}_{N_{\mathrm{ex}}}(k')} &\le \frac{\mathrm{reward}(\pi_{k'}) + 4\varepsilon_n}{\mathrm{cost}(\pi_{k'}) + 2(D_{k'} - 1)\varepsilon_n} \\
&\le \frac{\mathrm{reward}(\pi_{k'})}{\mathrm{cost}(\pi_{k'})} + \frac{2(D_{k'} + 1)\varepsilon_n}{\mathrm{cost}(\pi_{k'}) + 2(D_{k'} - 1)\varepsilon_n} \le \frac{\mathrm{reward}(\pi_{k'})}{\mathrm{cost}(\pi_{k'})} + \frac{2(D_{k'} + 1)\varepsilon_n}{\mathrm{cost}(\pi_{k'}) - 2(D_{k'} - 1)\varepsilon_n}
\end{aligned}$$

That is, for all optimal policies $\pi_{k^\star}$, the policy $\pi_{k'}$ run at line 7 satisfies

$$\begin{aligned}
\mathrm{reward}(\pi_{k'}) &\ge \mathrm{cost}(\pi_{k'}) \left(\frac{\mathrm{reward}(\pi_{k^\star})}{\mathrm{cost}(\pi_{k^\star})} - \frac{2(D_{k'} + 1)\varepsilon_n}{\mathrm{cost}(\pi_{k'}) - 2(D_{k'} - 1)\varepsilon_n} \right) \\
&\ge \mathrm{cost}(\pi_{k'}) \left(\frac{\mathrm{reward}(\pi_{k^\star})}{\mathrm{cost}(\pi_{k^\star})} - 4(D_K + 1)\varepsilon_n \right)
\end{aligned}$$

where in the last inequality we lower bounded the denominator by $1/2$ using $\mathrm{cost}(\pi_{k'}) \ge 1$ and $\varepsilon_n \le \varepsilon_{N_{\mathrm{ex}}} \le 1/2$ which follows by $n \ge N_{\mathrm{ex}} \ge 8D_K^2 \ln(4KN_{\mathrm{ex}}/\delta)$ and the monotonicity of $k \mapsto D_k$. Therefore, for all optimal policies $\pi_{k^\star}$, the total expected reward of Algorithm 1 divided by its total expected cost (i.e., the negative addend in the regret (2)) is at least

$$\begin{aligned}
&\frac{\mathbb{E}\big[-N_{\mathrm{ex}} + (N - N_{\mathrm{ex}})\,\mathrm{reward}(\pi_{k'})\big]}{\mathbb{E}\big[2\sum_{n=1}^{N_{\mathrm{ex}}} D_{\max(C_n)} + (N - N_{\mathrm{ex}})\,\mathrm{cost}(\pi_{k'})\big]} \\
&\qquad \ge \frac{-N_{\mathrm{ex}}}{2\sum_{n=1}^{N_{\mathrm{ex}}} \mathbb{E}\big[D_{\max(C_n)}\big] + (N - N_{\mathrm{ex}})\,\mathbb{E}\big[\mathrm{cost}(\pi_{k'})\big]} \\
&\qquad\quad + \frac{(N - N_{\mathrm{ex}})\,\mathbb{E}\big[\mathrm{cost}(\pi_{k'})\big]}{2\sum_{n=1}^{N_{\mathrm{ex}}} \mathbb{E}\big[D_{\max(C_n)}\big] + (N - N_{\mathrm{ex}})\,\mathbb{E}\big[\mathrm{cost}(\pi_{k'})\big]} \left(\frac{\mathrm{reward}(\pi_{k^\star})}{\mathrm{cost}(\pi_{k^\star})} - 4(D_K + 1)\varepsilon_n \right) \\
&\qquad \ge \frac{\mathrm{reward}(\pi_{k^\star})}{\mathrm{cost}(\pi_{k^\star})} - 4(D_K + 1)\varepsilon_n - \frac{N_{\mathrm{ex}} + 2\sum_{n=1}^{N_{\mathrm{ex}}} \mathbb{E}\big[D_{\max(C_n)}\big]}{2\sum_{n=1}^{N_{\mathrm{ex}}} \mathbb{E}\big[D_{\max(C_n)}\big] + (N - N_{\mathrm{ex}})\,\mathbb{E}\big[\mathrm{cost}(\pi_{k'})\big]} \\
&\qquad \ge \frac{\mathrm{reward}(\pi_{k^\star})}{\mathrm{cost}(\pi_{k^\star})} - 4(D_K + 1)\varepsilon_n - \frac{(2D_K + 1)N_{\mathrm{ex}}}{N}
\end{aligned}$$

where we used $\frac{a}{b+a}(x - y) \ge x - y - \frac{b}{b+a}$ for all $a, b, y > 0$ and all $x \le 1$ to lower bound the third line, then the monotonicity of $k \mapsto D_k$ and $2\mathbb{E}\big[D_{\max(C_n)}\big] \ge \mathbb{E}\big[\mathrm{cost}(\pi_{k'})\big] \ge 1$ for the last inequality. Rearranging the terms of the first and last hand side in the previous display, using the monotonicity of $k \mapsto D_k$, and plugging in the value of ε_n, gives

$$R_T \le 4(D_K + 1)\varepsilon_n + \frac{(2D_K + 1)N_{\mathrm{ex}}}{N} = (D_K + 1)\sqrt{\frac{8\ln(4KN_{\mathrm{ex}}/\delta)}{N_{\mathrm{ex}}}} + \frac{(2D_K + 1)N_{\mathrm{ex}}}{N}$$

□

5 Extension to Countable Sets of Policies (ESC-CAPE)

In this section, we show how a countable set of policies can be reduced to a finite one containing all optimal policies with high probability (Algorithm 2, ESC). After this is done, one can run Algorithm 1 (CAPE) on the smaller policy set, obtaining theoretical guarantees for the resulting algorithm.

[9] Otherwise k' would belong to the set $\arg\max_{k \in C_{N_{\mathrm{ex}}+1}} \big(\widehat{r}^{+}_{N_{\mathrm{ex}}}(k)/\widehat{c}^{-}_{N_{\mathrm{ex}}}(k)\big)$ which in turn would be included in the set $\big\{k \in C_{N_{\mathrm{ex}}+1} : \widehat{r}^{+}_{N_{\mathrm{ex}}}(k) \ge 0\big\}$ and this would contradict the fact that $\widehat{r}^{+}_{N_{\mathrm{ex}}}(k') < 0$.

More precisely, we will focus on sets of policies $\Pi = \{\pi_k\}_{k\in\mathbb{N}} = \bigl\{(\tau_k, \text{accept})\bigr\}_{k\in\mathbb{N}}$ where accept is an arbitrary decision and $\tau_1, \tau_2, \dots$ is any sequence of durations which again, are assumed to be sorted by index and bounded by $D_1 \le D_2 \le \dots$ (note that now durations are no longer uniformly bounded).

Let us first introduce three handy notations. Firstly, in the case where $2D_k$ samples are drawn during each of n_2 consecutive tasks $n_1+1, n_1+2, \dots, n_1+n_2$, we define, for all $\varepsilon > 0$, the following lower confidence bound on reward(π_k) (similarly to (5))

$$\widehat{r}_k^{\,-}(n_1, n_2, \varepsilon) = \frac{1}{n_2} \sum_{n=n_1+1}^{n_1+n_2} \sum_{i=1}^{D_k} \frac{X_{n,D_k+i}}{D_k} \operatorname{accept}\bigl(\tau_k(\boldsymbol{X}_n), \boldsymbol{X}_n\bigr) - 2\varepsilon \tag{17}$$

Secondly, whenever the policy $(\tau_k, 0)$ is run for m_0 consecutive tasks $n_0+1, n_0+2, \dots, n_0+m_0$, we denote the empirical average of its duration (similarly to (12)) by

$$\overline{c}_k(n_0, m_0) = \bigl(\tau_k(\boldsymbol{X}_{n_0+1}) + \dots + \tau_k(\boldsymbol{X}_{n_0+m_0})\bigr)/m_0 \tag{18}$$

Lastly, let $M_0 = 0$ and for all $\varepsilon, \delta > 0$ and all $j \in \mathbb{N}$, we let

$$m_j(\varepsilon, \delta) = \bigl\lceil \ln\bigl(j(j+1)/\delta\bigr)/2\varepsilon^2 \bigr\rceil \qquad \text{and} \qquad M_j = m_1 + \dots + m_j \tag{19}$$

Algorithm 2: Extension to Countable (ESC)

```
  Input: countable policy set $\Pi$, number of tasks $N$, confidence $\delta$, accuracy levels $\varepsilon_1, \varepsilon_2, \dots > 0$
  Initialization: for all $j \in \mathbb{N}$, let $m_j \leftarrow m_j(\varepsilon_j, \delta)$ (see (19))
1 for $j = 1, 2, \dots$ do
2     run policy $\bigl(2D_{2^j}, 0\bigr)$ for $m_j$ tasks and compute $\widehat{r}_{2^j}^{\,-} \leftarrow \widehat{r}_{2^j}^{\,-}(M_{j-1}, m_j, \varepsilon_j)$ (see (17))
3     if $\widehat{r}_{2^j}^{\,-} > 0$ then
4         let $j_0 \leftarrow j$ and $k_0 \leftarrow 2^{j_0}$
5         for $l = j_0+1, j_0+2, \dots$ do
6             run policy $\bigl(\tau_{2^l}, 0\bigr)$ for $m_l$ tasks and compute $\overline{c}_{2^l} \leftarrow \overline{c}_{2^l}(M_{l-1}, m_l)$ (see (18))
7             if $\overline{c}_{2^l} > D_{2^l}\,\varepsilon_l + D_{k_0}/\widehat{r}_{k_0}^{\,-}$ then let $j_1 \leftarrow l$ and return $K \leftarrow 2^{j_1}$
```

The key idea behind Algorithm 2 (ESC) is simple. Since all optimal policies $\pi_{k^\star}$ have to satisfy the relationships $\text{reward}(\pi_k)/\text{cost}(\pi_k) \le \text{reward}(\pi_{k^\star})/\text{cost}(\pi_{k^\star}) \le 1/\text{cost}(\pi_{k^\star})$, then, for all policies π_k with $\text{reward}(\pi_k) > 0$, the cost of any optimal policy $\pi_{k^\star}$ must satisfy the relationship $\text{cost}(\pi_{k^\star}) \le \text{cost}(\pi_k)/\text{reward}(\pi_k)$. In other words, optimal policies cannot draw too many samples and their cost can be controlled by estimating the reward and cost of *any* policy with positive reward.

Thus, Algorithm 2 (ESC) first finds a policy π_{k_0} with $\text{reward}(\pi_{k_0}) > 0$ (lines 1–4), memorizing an upper estimate $D_{k_0}/\widehat{r}_{k_0}^{\,-}$ of the ratio $\text{cost}(\pi_{k_0})/\text{reward}(\pi_{k_0})$. By the argument above, this estimate upper bounds the expected number of samples $\text{cost}(\pi_{k^\star})$ drawn by all optimal policies $\pi_{k^\star}$. Then ESC simply proceeds to finding the smallest (up to a factor of 2) K such that $\text{cost}(\pi_K) \ge D_{k_0}/\widehat{r}_{k_0}^{\,-}$ (lines 5–7). Being $D_{k_0}/\widehat{r}_{k_0}^{\,-} \ge \text{cost}(\pi_{k_0})/\text{reward}(\pi_{k_0}) \ge \text{cost}(\pi_{k^\star})$ by construction, the index K determined this way upper bounds $k^\star$ for all optimal policies $\pi_{k^\star}$. (All the previous statements are intended to hold with high probability.) This is formalized in the following key lemma.

Lemma 5. *Let Π be countable. If ESC is run with $\delta \in (0,1)$, $\varepsilon_1, \varepsilon_2, \dots > 0$, and halts returning K, then $k^\star \le K$ for all optimal policies $\pi_{k^\star}$ with probability at least $1-\delta$.*

Proof. Note fist that $\widehat{r}_{2^j}^{\,-} + 2\varepsilon_j$ (line 2) is an empirical average of m_j i.i.d. unbiased estimators of $\text{reward}(\pi_{2^j})$. Indeed, being $\text{accept}(k, \boldsymbol{x})$ independent of the variables $(x_{k+1}, x_{k+2}, \dots)$ by definition of duration and the conditional independence of the samples (recall the properties of samples in step 3 of our online protocol, Section 3), for all tasks n performed at line 2 during iteration j and all $i > D_{2^j}$,

$$\begin{aligned}\mathbb{E}\left[X_{n,i}\,\mathrm{accept}\big(\tau_{2^j}(\boldsymbol{X}_n),\boldsymbol{X}_n\big)\;\middle|\;\mu_n\right] &= \mathbb{E}\left[X_{n,i}\mid\mu_n\right]\mathbb{E}\Big[\mathrm{accept}\big(\tau_{2^j}(\boldsymbol{X}_n),\boldsymbol{X}_n\big)\;\Big|\;\mu_n\Big]\\ &= \mu_n\,\mathbb{E}\Big[\mathrm{accept}\big(\tau_{2^j}(\boldsymbol{X}_n),\boldsymbol{X}_n\big)\;\Big|\;\mu_n\Big] = \mathbb{E}\Big[\mu_n\,\mathrm{accept}\big(\tau_{2^j}(\boldsymbol{X}_n),\boldsymbol{X}_n\big)\;\Big|\;\mu_n\Big]\end{aligned}$$

Taking expectations to both sides proves the claim. Thus, Hoeffding's inequality implies

$$\mathbb{P}\Big(\widehat{r}_{2^j}^{\,-} > \mathrm{reward}(\pi_{2^j})\Big) = \mathbb{P}\Big(\big(\widehat{r}_{2^j}^{\,-} + 2\varepsilon_j\big) - \mathrm{reward}(\pi_{2^j}) > 2\varepsilon_j\Big) \le \frac{\delta}{j(j+1)}$$

for all $j \le j_0$. Similarly, for all $l > j_0$, $\mathbb{P}\big(\overline{c}_{2^l} - \mathrm{cost}(\pi_{2^l}) > D_{2^l}\,\varepsilon_l\big) \le \frac{\delta}{l(l+1)}$. Hence, the event

$$\big\{\widehat{r}_{2^j}^{\,-} \le \mathrm{reward}(\pi_{2^j})\big\} \;\wedge\; \big\{\overline{c}_{2^l} \le \mathrm{cost}(\pi_{2^l})) + D_{2^l}\,\varepsilon_l\big\} \qquad \forall j \le j_0, \forall l > j_0 \tag{20}$$

occurs with probability at least

$$1 - \sum_{j=1}^{j_0}\frac{\delta}{j(j+1)} - \sum_{l=j_0+1}^{j_1}\frac{\delta}{l(l+1)} \ge 1 - \delta\sum_{j\in\mathbb{N}}\frac{1}{j(j+1)} = 1-\delta$$

Note now that for each policy π_k with $\mathrm{reward}(\pi_k) \ge 0$ and each optimal policy $\pi_{k^\star}$,

$$\frac{\mathrm{reward}(\pi_k)}{D_k} \le \frac{\mathrm{reward}(\pi_k)}{\mathrm{cost}(\pi_k)} \le \frac{\mathrm{reward}(\pi_{k^\star})}{\mathrm{cost}(\pi_{k^\star})} \le \frac{1}{\mathrm{cost}(\pi_{k^\star})} \tag{21}$$

Hence, all optimal policies $\pi_{k^\star}$ satisfy $\mathrm{cost}(\pi_{k^\star}) \le D_k/\mathrm{reward}(\pi_k)$ for all policies π_k such that $\mathrm{reward}(\pi_k) > 0$. Being durations sorted by index, for all $k \le h$

$$\mathrm{cost}(\pi_k) = \mathbb{E}\big[\mathrm{cost}(\pi_k,\mu_0)\big] \le \mathbb{E}\big[\mathrm{cost}(\pi_h,\mu_0)\big] = \mathrm{cost}(\pi_h) \tag{22}$$

Thus, with probability at least $1-\delta$, for all $k > K$

$$\mathrm{cost}(\pi_k) \overset{(22)}{\ge} \mathrm{cost}(\pi_K) \overset{(20)}{\ge} \overline{c}_K - D_K\,\varepsilon_{\log_2 K} \overset{\text{line } 7}{>} \frac{D_{k_0}}{\widehat{r}_{k_0}^{\,-}} \ge \frac{D_{k_0}}{\mathrm{reward}(k_0)}$$

where $\mathrm{reward}(k_0) \ge \widehat{r}_{k_0}^{\,-} > 0$ by (20) and line (3); i.e., π_k do not satisfy (21). Therefore, with probability at least $1-\delta$, all optimal policies $\pi_{k^\star}$ satisfy $k^\star \le K$. □

Before stating the main result of this section, we need a final lemma upper bounding the expected cost of Algorithm 2.

Lemma 6. *Let Π be countable. If ESC is run with $\delta \in (0,1)$, $\varepsilon_1, \varepsilon_2, \ldots > 0$, and halts returning K, then the total number of samples it draws before stopping (i.e., its cost) is upper bounded by $\widetilde{\mathcal{O}}\left(({}^{D_K}/_{\varepsilon^2})\log({}^{1}/_{\delta})\right)$ where $\varepsilon = \min\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{\log_2 K}\}$.*

Proof. Recall the definition of $m_j(\varepsilon,\delta)$ (19) and m_j (initialization of Algorithm 2). Note that, by definition, $\varepsilon = \min\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{j_1}\} > 0$. Algorithm 2 (ESC) draw samples only when lines 2 or 6 are executed. Whenever line 2 is executed ($j = 1, \ldots, j_0$) the algorithm performs m_j tasks drawing $2D_{2^j}$ samples each time. Similarly, whenever line 6 is executed ($l = j_0+1, \ldots, j_1$) the algorithm draws at most D_{2^l} samples during each of the m_l tasks. Therefore, recalling that $j_1 = \log_2 K$, the total number of samples drawn by ESC before stopping is upper bounded by

$$\sum_{j=1}^{j_0} 2D_{2^j}m_j + \sum_{l=j_0+1}^{j_1} D_{2^l}m_l \le 2\sum_{j=1}^{j_1} D_{2^j}m_j \le 2j_1 D_{2^{j_1}}m_{j_1}(\varepsilon,\delta) \le 4\log_2(K)D_K\frac{\ln\big(\frac{\log_2 K}{\delta}\big)}{\varepsilon^2}$$

□

We can now join together our two algorithms obtaining a new one, that we call ESC-CAPE, which takes as input a countable policy set Π, the number of tasks N, a confidence parameter δ, some accuracy levels $\varepsilon_1, \varepsilon_2, \ldots$, and an exploration cap N_{ex}. The joint algorithm runs ESC first with parameters $\Pi, N, \delta, \varepsilon_1, \varepsilon_2, \ldots$. Then, if ESC halts returning K, it runs CAPE with parameters $\bigl\{(\tau_k, \mathrm{accept})\bigr\}_{k=1}^{K}, N, \delta, N_{\mathrm{ex}}$.

Since all mutations are rejected during the run of ESC, the sum of the rewards accumulated during the preprocessing step is zero. The only effect on the regret (2) is then an increment on the total cost in the denominator of the second term, which can be controlled by minimizing its upper bound in Lemma 6. This is not a simple matter of taking all ε_j as large as possible. Indeed, if all ε_j are large, the **if** clause at line 3 might never be verified. In other words, the returned index K depends on ε and grows unbounded in general as ε approaches $1/2$.

Thus, there is a trade-off between having a small K (for which small ε_j are required in general) and a small $1/\varepsilon^2$ (for which large ε_j are needed). A direct computation shows that combining Algorithm 2 and Algorithm 1 (ESC-CAPE) with a constant accuracy level $\varepsilon_j = N^{-1/3}$ achieves the best of both worlds and immediately gives Theorem 2, below.

Note that contrary to vanilla CAPE, we do not get the $1/N$ rate for ESC-CAPE when $\Delta \gg 0$ (recall bound (8)). Indeed, the optimal choice of $\varepsilon_j = N^{-1/3}$ still makes the regret rate degrade to order $N^{-1/3}$ by Lemma 6.

We conclude this section (and this paper) by stating the theoretical guarantees of our final algorithm ESC-CAPE against infinite policy classes.

Theorem 2. *Assuming Π is countable, then ESC-CAPE run with constant accuracy levels $\varepsilon_j = N^{-1/3}$, $\delta \in (0,1)$, and $N_{\mathrm{ex}} = \lceil N^{2/3} \rceil$ has a regret satisfying $R_N = \widetilde{\mathcal{O}}\bigl(D_K / N^{1/3}\bigr)$ with probability at least $1 - 2\delta$, where the $\widetilde{\mathcal{O}}$ notation hides only logarithmic terms, including a $\log(1/\delta)$ term.*

6 Choice of Performance Measure

In this section we discuss the choice of performance measurements of policies π, namely the ratio of expectations $\mathrm{reward}(\pi)/\mathrm{cost}(\pi)$. We compare several different benchmarks and investigate the differences if the agent had a budget of samples and a variable number tasks, rather than the other way around. We will show that all "natural" choices essentially go in the same direction, except for one (perhaps the most natural) which is surprisingly poorly suited.

At a high level, an agent constrained by a budget would like to maximize its reward per "time step" (interpreting the draw of each sample as a time step gone by). This can be done in several different ways. If the constraint is on the number N of tasks, then the agent aims at maximizing (over $\pi = (\tau, \mathrm{accept}) \in \Pi$) the objective $g_1(\pi, N)$ defined by

$$g_1(\pi, N) = \mathbb{E}\left[\frac{\sum_{n=1}^{N} \mathrm{reward}(\pi, \mu_n)}{\sum_{m=1}^{N} \mathrm{cost}(\pi, \mu_m)} \right]$$

This is equivalent to the maximization of the ratio

$$\frac{\mathrm{reward}(\pi)}{\mathrm{cost}(\pi)} = \frac{\mathbb{E}\bigl[\mathrm{reward}(\pi, \mu_0)\bigr]}{\mathbb{E}\bigl[\mathrm{cost}(\pi, \mu_0)\bigr]}$$

in the sense that, multiplying both the numerator and the denominator in $g_1(\pi, N)$ by $1/N$ and applying Hoeffding's inequality, we get $g_1(\pi, N) = \Theta\bigl(\mathrm{reward}(\pi)/\mathrm{cost}(\pi)\bigr)$. Furthermore, by the law of large numbers and Lebesgue's dominated convergence theorem, $g_1(\pi, N) \to \mathrm{reward}(\pi)/\mathrm{cost}(\pi)$ when $N \to \infty$ for any $\pi \in \Pi$.

Assume now that the constraint is on the total number of samples instead. We say that the agent has a *budget of samples* T if as soon as the total number of samples reaches T during task N (which is now a random variable), the agent has to interrupt the run of the current policy, reject the current value μ_N, and end the process. Formally, the random variable N that counts the total number of tasks performed by repeatedly running a policy $\pi = (\tau, \mathrm{accept})$ is defined by

$$N = \min \left\{ m \in \mathbb{N} \,\middle|\, \sum_{n=1}^{m} \tau(\boldsymbol{X}_n) \geq T \right\}$$

In this case, the agent aims at maximizing the objective

$$g_2(\pi, T) = \mathbb{E}\left[\frac{\sum_{n=1}^{N-1} \mathrm{reward}(\pi, \mu_n)}{T} \right]$$

where the sum is 0 if $N = 1$ and it stops at $N - 1$ because the the last task is interrupted and no reward is gained. As before, assume that $\tau \leq D$, for some $D \in \mathbb{N}$. Note first that by the independence of μ_n and $\boldsymbol{X}_n$ from past tasks, for all deterministic functions f and all $n \in \mathbb{N}$, the two random variables $f(\mu_n, \boldsymbol{X}_n)$ and $\mathbb{I}\{N \geq n\}$ are independent, because $\mathbb{I}\{N \geq n\} = \mathbb{I}\big\{\sum_{i=1}^{n-1} \tau(\boldsymbol{X}_i) < T\big\}$ depends only on the random variables $\tau(\boldsymbol{X}_1), \ldots, \tau(\boldsymbol{X}_{n-1})$. Hence

$$\begin{aligned} \mathbb{E}\Big[\mathrm{reward}(\pi, \mu_n)\, \mathbb{I}\{N \geq n\}\Big] &= \mathrm{reward}(\pi)\, \mathbb{P}(N \geq n) \\ \mathbb{E}\big[\mathrm{cost}(\pi, \mu_n)\, \mathbb{I}\{N \geq n\}\big] &= \mathrm{cost}(\pi)\, \mathbb{P}(N \geq n) \end{aligned}$$

Moreover, note that during each task at least one sample is drawn, hence $N \leq T$ and

$$\begin{aligned} \sum_{n=1}^{\infty} \mathbb{E}\Big[\big|\mathrm{reward}(\pi, \mu_n)\big|\, \mathbb{I}\{N \geq n\}\Big] &\leq \sum_{n=1}^{T} \mathbb{E}\Big[\big|\mathrm{reward}(\pi, \mu_n)\big|\Big] \leq T < \infty \\ \sum_{n=1}^{\infty} \mathbb{E}\big[\mathrm{cost}(\pi, \mu_n)\, \mathbb{I}\{N \geq n\}\big] &\leq \sum_{n=1}^{T} \mathbb{E}\big[\mathrm{cost}(\pi, \mu_n)\big] = T\, \mathrm{cost}(\pi) \leq TD < \infty \end{aligned}$$

We can therefore apply Wald's identity [24] to deduce

$$\mathbb{E}\left[\sum_{n=1}^{N} \mathrm{reward}(\pi, \mu_n)\right] = \mathbb{E}[N]\, \mathrm{reward}(\pi) \qquad \text{and} \qquad \mathbb{E}\left[\sum_{n=1}^{N} \mathrm{cost}(\pi, \mu_n)\right] = \mathbb{E}[N]\, \mathrm{cost}(\pi)$$

which, together with

$$\mathbb{E}\left[\sum_{n=1}^{N} \mathrm{cost}(\pi, \mu_n)\right] \geq T \geq \mathbb{E}\left[\sum_{n=1}^{N} \mathrm{cost}(\pi, \mu_n)\right] - D$$

and

$$\mathbb{E}\left[\sum_{n=1}^{N} \mathrm{reward}(\pi, \mu_n)\right] - 1 \leq \mathbb{E}\left[\sum_{n=1}^{N-1} \mathrm{reward}(\pi, \mu_n)\right] \leq \mathbb{E}\left[\sum_{n=1}^{N} \mathrm{reward}(\pi, \mu_n)\right] + 1$$

yields

$$\frac{\mathbb{E}[N]\, \mathrm{reward}(\pi) - 1}{\mathbb{E}[N]\, \mathrm{cost}(\pi)} \leq g_2(\pi, T) \leq \frac{\mathbb{E}[N]\, \mathrm{reward}(\pi) + 1}{\mathbb{E}[N]\, \mathrm{cost}(\pi) - D}$$

if the denominator on the right hand side is positive, which happens as soon as $T > D^2$ by $ND \geq \sum_{n=1}^{N} \tau(\boldsymbol{X}_n) \geq T$ and $\mathrm{cost}(\pi) \geq 1$. I.e., $g_2(\pi, T) = \Theta\big(\mathrm{reward}(\pi)/\mathrm{cost}(\pi)\big)$ and noting that $\mathbb{E}[N] \geq T/D \to \infty$ if $T \to \infty$, we have once more that $g_2(\pi, T) \to \mathrm{reward}(\pi)/\mathrm{cost}(\pi)$ when $T \to \infty$ for any $\pi \in \Pi$.

This proves that having a budget of tasks, samples, or using any of the three natural objectives introduced so far is essentially the same.

Before concluding the section, we go back to the original setting and discuss a very natural definition of objective which should be avoided because, albeit easier to maximize, it is not well-suited to this problem. Consider as objective the average payoff of accepted values per amount of time used to make the decision, i.e.,

$$g_3(\pi) = \mathbb{E}\left[\frac{\mathrm{reward}(\pi, \mu_0)}{\mathrm{cost}(\pi, \mu_0)}\right]$$

We give some intuition on the differences between the ratio of expectations and the expectation of the ratio g_3 using the concrete example (4) and we make a case for the former being better than the latter.

More precisely, if N decision tasks have to be performed by the agent, consider the natural policy class $\{\pi_k\}_{k\in\{1,\dots,K\}} = \big\{(\tau_k, \mathrm{accept})\big\}_{k\in\{1,\dots,K\}}$ given by

$$\tau_k(\boldsymbol{x}) = \min\left(k,\ \inf\left\{n \in \mathbb{N} : |\overline{x}_n| \ge c\sqrt{\frac{\ln\frac{KN}{\delta}}{n}}\right\}\right), \qquad \mathrm{accept}(n, \boldsymbol{x}) = \mathbb{I}\left\{\overline{x}_n \ge c\sqrt{\frac{\ln\frac{KN}{\delta}}{n}}\right\}$$

for some $c > 0$ and $\delta \in (0, 1)$, where $\overline{x}_n = ({}^1\!/\!{}_n)\sum_{i=1}^{n} x_i$ is the average of the first n elements of the sequence $\boldsymbol{x} = (x_1, x_2, \dots)$.

If $K \gg 1$, there are numerous policies in the class with a large cap. For concreteness, consider the last one $(\tau_K, \mathrm{accept})$ and let $k = \lceil c^2 \ln(KN/\delta)\rceil$. If μ_0 is uniformly distributed on $\{-1, 0, 1\}$, then

$$\Big(\tau_K(\boldsymbol{X}_0), \mathrm{accept}\big(\tau_K(\boldsymbol{X}_0), \boldsymbol{X}_0\big)\Big) = \begin{cases} (k, 1) & \text{if } \mu_1 = 1 \\ (k, 0) & \text{if } \mu_1 = -1 \\ (K, 0) & \text{if } \mu_1 = 0 \end{cases}$$

i.e., the agent understands quickly (drawing only k samples) that $\mu_0 = \pm 1$, accepting it or rejecting it accordingly, but takes exponentially longer ($K \gg k$ samples) to figure out that the mutation is nonpositive when $\mu_0 = 0$. The fact that for a constant fraction of tasks (1/3 of the total) π invests a long time (K samples) to earn no reward makes it a very poor choice of policy. This is not reflected in the definition of $g_3(\pi_K)$ but it is so in the definition of $\mathrm{reward}(\pi_K)/\mathrm{cost}(\pi_K)$. Indeed, in this instance

$$\mathbb{E}\left[\frac{\mathrm{reward}(\pi_K, \mu_0)}{\mathrm{cost}(\pi_K, \mu_0)}\right] = \Theta\left(\frac{1}{k}\right) \qquad \gg \qquad \Theta\left(\frac{1}{K}\right) = \frac{\mathrm{reward}(\pi_K)}{\mathrm{cost}(\pi_K)}$$

This is due to the fact that the expectation of the ratio "ignores" outcomes with null (or very small) rewards, even if a large number of samples is needed to learn them. On the other hand, the ratio of expectations weighs the total number of requested samples and it is highly influenced by it, a property we are interested to capture within our model.

7 An Impossibility Result

We conclude the paper by showing that given μ_n it is impossible to define an unbiased estimator of the reward of all policies using only the samples drawn by the policies themselves, unless μ_n is known beforehand.

Take a policy $\pi_1 = (1, \mathrm{accept})$ that draws exactly one sample. Note that such a policy is included in all sets of policies defined as capped versions of a base policy (4). More generally, π_1 is included in all sets of policies with durations τ_k bounded by D_k, if $D_k = 1$ for some k, so this is by no means a pathological example. For the sake of simplicity, assume that samples take values in $\{0, 1\}$ and consider any decision function accept such that $\mathrm{accept}(1, \boldsymbol{x}) = x_1$ for all $\boldsymbol{x} = (x_1, x_2, \dots)$. In words, policy π_1 looks at one single sample $x_1 \in \{0, 1\}$ and accepts if and only if $x_1 = 1$. As discussed in Section 2 (second paragraph of the A/B testing part), there are settings in which this policy performs near-optimally. Moreover, in Section 6 we show that π_1 is optimal if μ is concentrated around $[-1, 0] \cup \{1\}$.

The following lemma shows that in the simple, yet meaningful case of the policy π_1 described above, it is impossible to define an unbiased estimator of its reward

$$\mu_n \, \mathbb{E}[\text{accept}(1, \boldsymbol{X}_n) \mid \mu_n] = \mathbb{E}[X_{n,1} \mid \mu_n] \, \mathbb{E}[X_{n,1} \mid \mu_n] = \mathbb{E}[X_{n,1} \mid \mu_n]^2$$

given μ_n, using only $X_{n,1}$, unless μ_n is known beforehand.

Lemma 7. *Let X be a Bernoulli random variable with parameter μ, for some real number $\mu \in [0,1]$. If $f \colon \{0,1\} \to \mathbb{R}$ satisfies $\mathbb{E}\big[f(X)\big] = \mathbb{E}[X]^2$, then f also satisfies*

$$\begin{cases} f(0) = \mu & \textit{if } \mu = 0 \\ f(1) = \mu - f(0)\dfrac{1-\mu}{\mu} & \textit{if } \mu \neq 0 \end{cases}$$

Proof. Let $f \colon \{0,1\} \to \mathbb{R}$ be any function satisfying $\mathbb{E}\big[f(X)\big] = \mathbb{E}[X]^2$. The law of the unconscious statistician and the definition of expectation imply

$$f(1)\mu + f(0)(1-\mu) = \mathbb{E}\big[f(X)\big] = \mathbb{E}[X]^2 = \mu^2$$

Thus, if $\mu = 0$, we have $f(0) = 0 = \mu$. If $\mu \neq 0$, solving by $f(1)$ gives the result. □

Acknowledgements

An earlier version of this work was done during Tommaso Cesari's Ph.D. at the University of Milan. Nicolò Cesa-Bianchi and Tommaso R. Cesari gratefully acknowledge partial support by Criteo AI Lab through a Faculty Research Award and by the MIUR PRIN grant Algorithms, Games, and Digital Markets (ALGADIMAR). This work has also benefited from the AI Interdisciplinary Institute ANITI. ANITI is funded by the French "Investing for the Future – PIA3" program under the Grant agreement n. ANR-19-PI3A-0004. Yishay Mansour was supported in part by a grant from the Israel Science Foundation (ISF). Vianney Perchet was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH, in a joint call with Gaspard Monge Program for optimization, operations research and their interactions with data sciences. Vianney Perchet also acknowledges the support of the ANR under the grant ANR-19-CE23-0026.

References

[1] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. 2012. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. Association for Computing Machinery, New York, NY, USA, 18–35.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[3] Eduardo M. Azevedo, Alex Deng, Jose Luis Montiel Olea, Justin Rao, and E. Glen Weyl. 2018. The A/B Testing Problem. In *Proceedings of the 2018 ACM Conference on Economics and Computation* (Ithaca, NY, USA) *(EC '18)*. Association for Computing Machinery, New York, NY, USA, 461–462. `https://doi.org/10.1145/3219166.3219204`

[4] Sébastien Bubeck and Nicolo Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5, 1 (2012), 1–122.

[5] Shiyun Chen and Shiva Kasiviswanathan. 2020. Contextual Online False Discovery Rate Control. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, Online, 952–961.

[6] Jose Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. 2019. Recent developments in prophet inequalities. *ACM SIGecom Exchanges* 17, 1 (2019), 61–70.

[7] Hossein Esfandiari, MohammadTaghi HajiAghayi, Brendan Lucier, and Michael Mitzenmacher. 2019. Online pandora's boxes and bandits. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (2019), 1885–1892.

[8] Dean P Foster and Robert A Stine. 2008. α-investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70, 2 (2008), 429–444.

[9] Christopher R Genovese, Kathryn Roeder, and Larry Wasserman. 2006. False discovery control with p-value weighting. *Biometrika* 93, 3 (2006), 509–524.

[10] Philipp Heesen and Arnold Janssen. 2016. Dynamic adaptive multiple tests with finite sample FDR control. *Journal of Statistical Planning and Inference* 168 (2016), 38–51.

[11] Lalit Jain and Kevin Jamieson. 2018. Firing Bandits: Optimizing Crowdfunding. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, Stockholmsmässan, Stockholm Sweden, 2206–2214.

[12] Adel Javanmard, Andrea Montanari, et al. 2018. Online rules for control of false discovery rate and false discovery exceedance. *The Annals of statistics* 46, 2 (2018), 526–554.

[13] Ramesh Johari, Pete Koomen, Leonid Pekelis, and David Walsh. 2017. Peeking at A/B Tests: Why It Matters, and What to Do about It. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 1517–1525.

[14] Robert Kleinberg, Bo Waggoner, and E. Glen Weyl. 2016. Descending Price Optimally Coordinates Search. In *Proceedings of the 2016 ACM Conference on Economics and Computation* (Maastricht, The Netherlands) *(EC '16)*. Association for Computing Machinery, New York, NY, USA, 23–24.

[15] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. 2013. Online Controlled Experiments at Large Scale. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Chicago, Illinois, USA) *(KDD '13)*. Association for Computing Machinery, New York, NY, USA, 1168–1176.

[16] Ang Li and Rina Foygel Barber. 2019. Multiple testing with the structure-adaptive Benjamini–Hochberg algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 81, 1 (2019), 45–74.

[17] Brendan Lucier. 2017. An economic view of prophet inequalities. *ACM SIGecom Exchanges* 16, 1 (2017), 24–47.

[18] Aaditya Ramdas, Fanny Yang, Martin J. Wainwright, and Michael I. Jordan. 2017. Online Control of the False Discovery Rate with Decaying Memory. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 5655–5664.

[19] David S. Robertson and James M. S. Wason. 2018. Online control of the false discovery rate in biomedical research. arXiv:1809.07292 [stat.ME]

[20] Dinah Rosenberg, Eilon Solan, and Nicolas Vieille. 2007. Social learning in one-arm bandit problems. *Econometrica* 75, 6 (2007), 1591–1611.

[21] Sven Schmit, Virag Shah, and Ramesh Johari. 2019. Optimal Testing in the Experiment-rich Regime. In *Proceedings of Machine Learning Research (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, Naha, Okinawa, Japan, 626–633.

[22] Aleksandrs Slivkins. 2019. Introduction to Multi-Armed Bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.

[23] John Wilder Tukey. 1953. The Problem of Multiple Comparisons.

[24] Abraham Wald. 1944. On cumulative sums of random variables. *The Annals of Mathematical Statistics* 15, 3 (1944), 283–296.

[25] Martin L. Weitzman. 1979. Optimal Search for the Best Alternative. *Econometrica* 47, 3 (1979), 641–654.

[26] Fanny Yang, Aaditya Ramdas, Kevin G Jamieson, and Martin J Wainwright. 2017. A framework for Multi-A(rmed)/B(andit) Testing with Online FDR Control. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., Long Beach, CA, USA, 5957–5966.