



HAL
open science

Fast and Accurate Repeated Decision Making

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

► **To cite this version:**

Nicolò Cesa-Bianchi, Tommaso R Cesari, Yishay Mansour, Vanney Perchet. Fast and Accurate Repeated Decision Making. 2020. hal-02976864v1

HAL Id: hal-02976864

<https://hal.science/hal-02976864v1>

Preprint submitted on 23 Oct 2020 (v1), last revised 22 Dec 2021 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fast and Accurate Repeated Decision Making

Nicolò Cesa-Bianchi

Università degli Studi di Milano & DSRC

Tommaso R. Cesari

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

Yishay Mansour

Tel Aviv University & Google research

Vannee Perchet

CREST, ENSAE & Criteo AI Lab, Paris

October 23, 2020

Abstract

We study a setting in which a learner faces a sequence of decision tasks and is required to make good decisions as quickly as possible. Each task n is associated with a pair (X_n, μ_n) , where X_n is a random variable and μ_n is its (unknown and potentially negative) expectation. The learner can draw arbitrarily many i.i.d. samples of X_n but its expectation μ_n is never revealed. After some sampling is done, the learner can decide to stop and either accept the task, gaining μ_n as a reward, or reject it, getting zero reward instead. A distinguishing feature of our model is that the learner's performance is measured as the expected cumulative reward divided by the expected cumulative number of drawn samples. The learner's goal is to converge to the per-sample reward of the optimal policy within a fixed class. 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

Repeated decision problems are pervasive in many applications domains due to their great modeling potential. In this paper, we introduce a novel setting in which the learner faces a sequence of binary decision tasks and is required to accumulate as much value as possible in the shortest amount of time.

Our setting. We view each task as the assessment of some innovation proposed to the learner. Each task n is associated with a pair (X_n, μ_n) , where X_n is a random variable with expectation μ_n and μ_n represents the true *value* of the n -th innovation. As such, μ_n 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 learner can draw i.i.d. samples from X_n in order to estimate μ_n , and use this estimate to decide whether the innovation is worth accepting. In other words, samples drawn from X_n represent noisy observations of the true value of the n -th innovation. Drawing more samples thus improves the estimate of μ_n , but also makes the assessment process run longer. This becomes an issue when many decisions have to be made sequentially, and drawing samples is costly. For this reason, the learner's 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 Appendix A.

A concrete application. Consider an online advertising company that keeps on testing out innovations in order to increase its revenue. Before deploying them, the company wants to figure out whether the innovations are actually more profitable than the technologies that are currently in place. As long as a reasonable metric is available (e.g., time spent on a page, click-through rates, conversion rates, etc.), the company can perform randomized tests and make statistically sound decisions. In real-life applications companies are often interested in spending as little time as possible to make these decisions because of budget constraints, and want to reject innovations that do not prove their worth in a reasonable amount of time. Indeed, running experiments is expensive and slows down the regular work flow. Therefore, discarding innovations that have a small positive margin could be significantly better in the long run than investing a large amount of resources into testing and implementing them.

I.I.D. assumption. We assume that the pair (X_n, μ_n) associated with the value of the n -th innovation is drawn i.i.d. from an unknown but fixed distribution. This assumption is meaningful if past decisions do not influence future innovations whose quality remains stable over time. Also, it applies whenever innovation can progress along many orthogonal directions, each yielding a similar added value. If this is the case, it would only make practical sense to invest resources in different directions in order to maximize one’s progressive improvements, which is precisely what is captured by our setting. It is also possible that both the state of the learner’s system and that of the environment evolve over time, but the ratio of good versus bad innovations remains essentially the same. 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 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 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 $\pi = (\tau, \text{accept})$ is a *policy*. Policies defined by capped early stopping rules (see (4) for a formal definition) are of great practical importance [10, 11]. 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 $(X_1, \mu_1), (X_2, \mu_2), \dots$ are drawn i.i.d. from some unknown but fixed distribution, the goal is to learn efficiently, at the smallest cost, the best policy π_* 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 from our setting 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 Appendix D). The presence of this bias is a significant departure from settings like multiarmed and firing bandits [1, 8], 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 performance measure that we use 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

While to the best of our knowledge our setting is novel, it share some similarities with 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.

Differences with bandits. If the set of all policies (defined rigorously in Section 3) used by the decision-maker to determine weather or not to accept an innovation are thought of as arms, our setting becomes somewhat reminiscent of multiarmed bandits [17, 3, 15]. However, the two problems are significantly different. At the end of each round of a stochastic bandit problem, the learner gets to see an unbiased estimate of the expected reward of the arm they played. As this does not happen in our setting (see impossibility result in Appendix D), we cannot just run a bandit algorithm to solve our problem. In addition to that, bandit algorithms are typically analyzed under an additive notion of regret, whereas the regret which makes most sense for us—see definition (2)—is not additive. Thus, it is unclear how formal guarantees for bandit algorithms would translate to our setting.

Firing bandits [8] can also be seen as a variant of our framework, where μ_n belongs to $[0, 1]$, samples X_n are Bernoulli random variables 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 one is allowed to go back and forth sampling from any of the past X_n or draw any number of new (X_m, μ_m) . This is a reasonable assumption for them, as each one of their μ_n is thought of as the value of a project in a crowdfunding platform, and drawing samples from X_n corresponds to displaying projects on web pages. However, in our setting each μ_n represents the theoretical increment (or decrement) of a company’s profit by means of an innovation. With this in mind, 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. Hence, when the sampling of a X_n stops, an irrevocable decision has to be taken. After that, no more samples of X_n can be drawn in the future and the learner moves on from the current pair (X_n, μ_n) . Finally, as in multiarmed bandits, the different form of the regret makes it so that even if a firing bandit algorithm could be adapted to our setting, its regret guarantees would not probably hold.

Differences with repeated A/B testing. Our setting 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 [18]. 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 [6, 5, 7, 9, 2, 12, 16]. The use of such data-driven sequential decisions processes has been successfully used by companies like Amazon, Bing, Criteo, Facebook, Google, and Uber [2].

A popular metric to optimize sequential A/B-tests is the so-called *false discovery rate* (FDR) —see [13, 20] 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 . This unfortunately disregards the relative values of tests μ_n . Our approach departs from online FDR [4, 14] by taking into account that the effect of many slightly negative accepted tests could be overcome by a few largely positive ones. For instance, assume that the samples X_n belong to $\{-1, 1\}$, and their expected value μ_n is uniformly distributed on $\{-\varepsilon, \varepsilon\}$. To control the FDR, one would have to run each A/B test 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. A direct computation shows that this policy, which fits our setting, achieves a significantly better performance of order ε .

Some other A/B testing settings are more closely related to ours. However, in the existing literature, more assumptions or preliminary knowledge is needed in order to obtain theoretical guarantees. For example, in [2], smoothness assumptions are made on the distributions of both X_n and μ_n . In [16], the authors assume that the distribution of μ_n is known, and the distribution of its samples belongs to a single parameter exponential family, also known beforehand.

3 Preliminaries and definitions

In this section, we formally introduce the repeated decision-making protocol for a learner 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 learner sequentially observes samples¹ $x_i \in [-1, 1]$ representing realizations of stochastic observations of the current innovation value. A map $\tau: [-1, 1]^{\mathbb{N}} \rightarrow \mathbb{N}$ is a *duration* (of a decision task) if for all $\mathbf{x} \in [-1, 1]^{\mathbb{N}}$, its value $d = \tau(\mathbf{x}) \in \mathbb{N}$ at \mathbf{x} depends only on the first d components x_1, x_2, \dots, x_d of $\mathbf{x} = (x_1, x_2, \dots)$. This definition reflects the fact that the components x_1, x_2, \dots of the sequence $\mathbf{x} = (x_1, x_2, \dots)$ 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 learner has to make a decision: either accepting or rejecting the current innovation. Formally, we say that a function $\text{accept}: \mathbb{N} \times [-1, 1]^{\mathbb{N}} \rightarrow \{0, 1\}$ is a *decision* (to accept) if for all $k \in \mathbb{N}$ and $\mathbf{x} \in [-1, 1]^{\mathbb{N}}$, its value $\text{accept}(k, \mathbf{x}) \in \{0, 1\}$ at (k, \mathbf{x}) depends only on the first k components x_1, \dots, x_k of $\mathbf{x} = (x_1, x_2, \dots)$. Again, this definition reflects the fact that the decision $\text{accept}(k, \mathbf{x})$ to either accept ($\text{accept}(k, \mathbf{x}) = 1$) or reject ($\text{accept}(k, \mathbf{x}) = 0$) the current innovation after observing the first k values x_1, \dots, x_k of $\mathbf{x} = (x_1, x_2, \dots)$ is oblivious to all future observations x_{k+1}, x_{k+2}, \dots . Following up on the concrete example above, the decision function is accepting the current innovation if and only if the empirical average of the observed values x_i surpasses a certain threshold.²

Thus, the only two choices that a learner can make 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 learner during each task is fully characterized by the choice of a pair $\pi = (\tau, \text{accept})$ 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 policies $\Pi = \{\pi_k\}_{k \in \mathcal{K}} = \{(\tau_k, \text{accept})\}_{k \in \mathcal{K}}$ (with \mathcal{K} either finite or countable) and a distribution³ μ on $[-1, 1]$. Naturally, the former is known beforehand but the latter is unknown and must be learned.

¹We 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.

²Note 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 $\mathbb{1}\{\text{mean} \geq \varepsilon_k\}$, as it also includes all decisions of the form $\mathbb{1}\{\text{mean} \in A_k\}$, for all measurable $A_k \subset \mathbb{R}$.

³Once 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

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

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

1. A sample μ_n (unknown to the learner), that we call *value*,⁴ is drawn i.i.d. according to μ
2. Let X_n be a $[-1, 1]$ -valued r.v. with $\mathbb{E}[X_n | \mu_n] = \mu_n$, independent of past tasks
3. The learner picks a $k_n \in \mathcal{K}$ or, equivalently, a policy $\pi_{k_n} = (\tau_{k_n}, \text{accept}) \in \Pi$
4. The learner draws the first $d_n = \tau_{k_n}(\mathbf{X}_n)$ *samples*⁵ of the i.i.d. (given μ_n) sequence of random variables $\mathbf{X}_n = (X_{n,1}, X_{n,2}, \dots)$, where $X_{n,i}$ has the same distribution as X_n
5. The learner makes the decision $\text{accept}(d_n, \mathbf{X}_n)$

For short, we say that the learner *runs a policy* $\pi_{k_n} = (\tau_{k_n}, \text{accept})$ (on a value μ_n) when steps 3–5 occur. We also say that they *accept* (resp., *rejects*) μ_n if their decision at step 5 is equal to 1 (resp., 0). Moreover, we say that the *reward* obtained and the *cost* paid by running a policy $\pi_k = (\tau_k, \text{accept})$ on a value μ_n are, respectively,

$$\text{reward}(\pi_k, \mu_n) = \mu_n \text{accept}(\tau_k(\mathbf{X}_n), \mathbf{X}_n) \quad \text{and} \quad \text{cost}(\pi_k, \mu_n) = \tau_k(\mathbf{X}_n) \quad (1)$$

The objective of the learner is to minimize the regret R_N after N consecutive tasks, defined as

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

where μ_0 is drawn i.i.d. according to the distribution μ , the random quantities \mathbf{X}_0 , $\text{reward}(\pi_{k_0}, \mu_0)$, and $\text{cost}(\pi_{k_0}, \mu_n)$ are defined as in bullet points 2, 4, and equation (1) respectively (with $n = 0$ and $k = k_0$), and the expectations are taken with respect to μ_n , \mathbf{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

$$\text{reward}(\pi) = \mathbb{E}[\text{reward}(\pi, \mu_0)] \quad \text{and} \quad \text{cost}(\pi) = \mathbb{E}[\text{cost}(\pi, \mu_0)] \quad (3)$$

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

For each policy $(\tau, \text{accept}) \in \Pi$ and all tasks n , we let the learner reject any value regardless of the outcome of the sampling. Formally, the learner 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 learner draw arbitrarily many extra samples in addition to the number $\tau(\mathbf{X}_n)$ that they would otherwise draw when running a policy $(\tau, \text{accept}) \in \Pi$ 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 learner can always draw $\tau(\mathbf{X}_n) + k$ many samples (for any $k \in \mathbb{N}$) before making the decision $\text{accept}(\tau(\mathbf{X}_n), \mathbf{X}_n)$, where we stress that the first argument of the decision function accept is $\tau(\mathbf{X}_n)$ and not $\tau(\mathbf{X}_n) + k$.

Note that invoking the power to reject a value μ_n after observing $\tau(\mathbf{X}_n)$ samples increases the cost of sampling in the denominator of (2) by $\mathbb{E}[\tau(\mathbf{X}_n)]$ 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 tool that makes the analysis cleaner. 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 Appendix D).

be extended in a straightforward manner if μ is any (shifted) subgaussian distribution.

⁴Since it represents of the value of the current innovation being tested.

⁵Note that given μ_n , the random variable d_n is a stopping time with respect to the natural filtration associated to the stochastic process \mathbf{X}_n (by definition of duration).

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

$$\tau_k(\mathbf{x}) = \min \left(k, \inf \left\{ n \in \mathbb{N} : |\bar{x}_n| \geq c \sqrt{\frac{\ln \frac{KN}{\delta}}{n}} \right\} \right), \quad \text{accept}(n, \mathbf{x}) = \mathbb{I} \left\{ \bar{x}_n \geq c \sqrt{\frac{\ln \frac{KN}{\delta}}{n}} \right\} \quad (4)$$

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

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 $\Pi = \{\pi_k\}_{k \in \{1, \dots, K\}} = \{(\tau_k, \text{accept})\}_{k \in \{1, \dots, K\}}$, where accept is an arbitrary decision and τ_1, \dots, τ_K is any sequence of durations which, for the sake of convenience, we assume to be sorted by index ($\tau_k \leq \tau_h$ if $k \leq h$) and bounded ($\tau_k \leq D_k$ for all k , with $D_k \leq D_h$ if $k \leq h$). 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 policies.

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 $\text{accept}(\tau_{\max(C_n)}(\mathbf{X}_n), \mathbf{X}_n)$ (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 $\hat{c}_n^-(k)$ 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 $\Delta = \min_{k \neq k^*} \frac{\text{reward}(\pi_{k^*})}{\text{cost}(\pi_{k^*})} - \frac{\text{reward}(\pi_k)}{\text{cost}(\pi_k)}$ where we recall that $k^* \in \arg \max_k (\text{reward}(\pi_k)/\text{cost}(\pi_k))$ 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 regret satisfying, with probability at least $1 - \delta$,*

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

as soon as $N \geq D^3$ (where the $\tilde{\mathcal{O}}$ 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 (for a complete proof, see Appendix B).

Proof sketch. The proof of this theorem relies on four technical lemmas (Lemmas 2-5) proven in Appendix B.1. 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 $D = D_K$.

Algorithm 1: Capped Policy Elimination (CAPE)

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

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

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

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

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

$$k' \in \begin{cases} \arg \max_{k \in C_{N_{\text{ex}}+1}} (\widehat{r}_{N_{\text{ex}}}^+(k) / \widehat{c}_{N_{\text{ex}}}^-(k)) & \text{if } \widehat{r}_{N_{\text{ex}}}^+(k) \geq 0 \text{ for some } k \in C_{N_{\text{ex}}+1} \\ \arg \max_{k \in C_{N_{\text{ex}}+1}} (\widehat{r}_{N_{\text{ex}}}^+(k) / \widehat{c}_{N_{\text{ex}}}^+(k)) & \text{if } \widehat{r}_{N_{\text{ex}}}^+(k) < 0 \text{ for all } k \in C_{N_{\text{ex}}+1} \end{cases} \quad (7)$$

With a concentration argument (Lemma 2), we 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

$$\widehat{r}_n^-(k) \leq \text{reward}(\pi_k) \leq \widehat{r}_n^+(k) \quad \text{and} \quad \widehat{c}_n^-(k) \leq \text{cost}(\pi_k) \leq \widehat{c}_n^+(k) \quad (9)$$

occurs simultaneously for all $n \leq N_{\text{ex}}$ and all $k \leq \max(C_n)$. 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 3) that suboptimal policies are eliminated after at most N'_{ex} tasks, where $N'_{\text{ex}} \leq 288 D_K^2 \ln(4KN_{\text{ex}}/\delta)/\Delta^2 + 1$. To prove it we upper bound the length of the confidence interval for $\text{reward}(\pi_k)/\text{cost}(\pi_k)$:

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

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

Afterwards, we analyze separately the case in which the test at line 6 is true for some task $N'_{\text{ex}} \leq N_{\text{ex}}$ 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 4) yields

$$R_N \leq \min \left(\frac{(2D_K + 1)N_{\text{ex}}}{N}, \frac{(2D_K + 1)(288 (D_K/\Delta)^2 \ln(4KN_{\text{ex}}/\delta) + 1)}{N} \right) \quad (10)$$

Finally, we consider the case in which the test at line 6 is false for all tasks $n \leq N_{\text{ex}}$, and line 7 is executed with $C_{N_{\text{ex}}+1}$ containing two or more policies. The key idea here is to use the definition of k' in Equation (7) to lower-bound $\text{reward}(\pi_{k'})$ in terms of $\text{reward}(\pi_{k^*})/\text{cost}(\pi_{k^*})$. This, together with some additional technical estimations (Lemma 5) leads to the result. \square

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.

More precisely, we will focus on sets of policies $\Pi = \{\pi_k\}_{k \in \mathbb{N}} = \{(\tau_k, \text{accept})\}_{k \in \mathbb{N}}$ where accept is an arbitrary decision and τ_1, τ_2, \dots is any sequence of durations which again, are assumed to be sorted by index and bounded by $D_1 \leq D_2 \leq \dots$ (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 $\text{reward}(\pi_k)$ (similarly to (5))

$$\widehat{r}_k^-(n_1, n_2, \varepsilon) = \frac{1}{n_2} \sum_{n=n_1+1}^{n_1+n_2} \sum_{i=1}^{D_k} \frac{X_{n, D_k+i}}{D_k} \text{accept}(\tau_k(\mathbf{X}_n), \mathbf{X}_n) - 2\varepsilon \quad (11)$$

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 (21)) by

$$\bar{\tau}_k(n_0, m_0) = (\tau_k(\mathbf{X}_{n_0+1}) + \dots + \tau_k(\mathbf{X}_{n_0+m_0}))/m_0 \quad (12)$$

Algorithm 2: Extension to Countable (ESC)

Input: countable policy set Π , number of tasks N , confidence δ , 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 (13))

- 1 **for** $j = 1, 2, \dots$ **do**
- 2 run policy $(2D_{2^j}, 0)$ for m_j tasks and compute $\widehat{r}_{2^j}^- \leftarrow \widehat{r}_{2^j}^-(M_{j-1}, m_j, \varepsilon_j)$ (see (11))
- 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 $(\tau_{2^l}, 0)$ for m_l tasks and compute $\bar{c}_{2^l} \leftarrow \bar{c}_{2^l}(M_{l-1}, m_l)$ (see (12))
- 7 **if** $\bar{c}_{2^l} > D_{2^l} \varepsilon_l + D_{k_0} / \widehat{r}_{k_0}^-$ **then** let $j_1 \leftarrow l$ and **return** $K \leftarrow 2^{j_1}$

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

$$m_j(\varepsilon, \delta) = \lceil \ln(j(j+1)/\delta) / 2\varepsilon^2 \rceil \quad \text{and} \quad M_j = m_1 + \dots + m_j \quad (13)$$

The key idea behind Algorithm 2 (ESC) is simple. Since all optimal policies π_{k^*} have to satisfy the relationships $\text{reward}(\pi_k) / \text{cost}(\pi_k) \leq \text{reward}(\pi_{k^*}) / \text{cost}(\pi_{k^*}) \leq 1 / \text{cost}(\pi_{k^*})$, then, for all policies π_k with $\text{reward}(\pi_k) > 0$, the cost of any optimal policy π_{k^*} must satisfy the relationship $\text{cost}(\pi_{k^*}) \leq \text{cost}(\pi_k) / \text{reward}(\pi_k)$. 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 $\text{cost}(\pi_{k^*})$ drawn by all optimal policies π_{k^*} . Then ESC simply proceeds to finding the smallest (up to a factor of 2) K such that $\text{cost}(\pi_K) \geq D_{k_0} / \widehat{r}_{k_0}^-$ (lines 5–7). Being $D_{k_0} / \widehat{r}_{k_0}^- \geq \text{cost}(\pi_{k_0}) / \text{reward}(\pi_{k_0}) \geq \text{cost}(\pi_{k^*})$ by construction, the index K determined this way upper bounds k^* for all optimal policies π_{k^*} . (All the previous statements are intended to hold with high probability.) This is formalized in the following key lemma.

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

We sketch a proof for this result. For all missing details, see Appendix C.

Proof sketch. Note first that $\widehat{r}_{2^j}^- + 2\varepsilon_j$ (line 2) is an empirical average of m_j i.i.d. unbiased estimators of $\text{reward}(\pi_{2^j})$. Thus, Hoeffding's inequality implies that $\mathbb{P}(\widehat{r}_{2^j}^- > \text{reward}(\pi_{2^j})) \leq \frac{\delta}{j(j+1)}$ for all $j \leq j_0$. Similarly, for all $l > j_0$, $\mathbb{P}(\bar{c}_{2^l} - \text{cost}(\pi_{2^l}) > D_{2^l} \varepsilon_l) \leq \frac{\delta}{l(l+1)}$. Hence, the event

$$\{\widehat{r}_{2^j}^- \leq \text{reward}(\pi_{2^j})\} \wedge \{\bar{c}_{2^l} \leq \text{cost}(\pi_{2^l}) + D_{2^l} \varepsilon_l\} \quad \forall j \leq j_0, \forall l > j_0 \quad (14)$$

occurs with probability at least $1 - \sum_{j=1}^{j_0} \frac{\delta}{j(j+1)} - \sum_{l=j_0+1}^{j_1} \frac{\delta}{l(l+1)} \geq 1 - \delta$. Now, as we pointed out above, all optimal policies π_{k^*} have to satisfy $\text{cost}(\pi_{k^*}) \leq D_k / \text{reward}(\pi_k)$ for all policies π_k with $\text{reward}(\pi_k) > 0$. However, no policy π_k with $k > K$ satisfies this condition (with high probability), since being durations sorted by index, we have, for all $k > K$

$$\text{cost}(\pi_k) \geq \text{cost}(\pi_K) \stackrel{(14)}{\geq} \bar{c}_K - D_K \varepsilon_{\log_2 K} \stackrel{\text{line 7}}{>} \frac{D_{k_0}}{\widehat{r}_{k_0}^-} \stackrel{(14)}{\geq} \frac{D_{k_0}}{\text{reward}(k_0)}$$

with probability at least $1 - \delta$, where $\text{reward}(k_0) \geq \widehat{r}_{k_0}^- > 0$ by (26) and line (3). Therefore, with probability at least $1 - \delta$, all optimal policies π_{k^*} satisfy $k^* \leq K$. \square

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 $\varepsilon_1, \varepsilon_2, \dots$, and an exploration cap N_{ex} . The joint algorithm runs ESC first with parameters $\Pi, N, \delta, \varepsilon_1, \varepsilon_2, \dots$. Then, if ESC halts returning K , it runs CAPE with parameters $\{(\tau_k, \text{accept})\}_{k=1}^K, N, \delta, N_{\text{ex}}$.

We conclude this section by stating the theoretical guarantees of our final algorithm ESC-CAPE against infinite policy classes. For additional details, see Appendix C.

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

Acknowledgements

An earlier version of this work was done during Tommaso R. 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 (ALGADI-MAR). 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] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.
- [2] Eduardo M Azevedo, Alex Deng, Jose Luis Montiel Olea, Justin Rao, and E Glen Weyl. The A/B testing problem. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 461–462. ACM, 2018.
- [3] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.
- [4] Shiyun Chen and Shiva Kasiviswanathan. Contextual online false discovery rate control. *arXiv preprint arXiv:1902.02885*, 2019.
- [5] Dean P Foster and Robert A Stine. α -investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):429–444, 2008.
- [6] Christopher R Genovese, Kathryn Roeder, and Larry Wasserman. False discovery control with p -value weighting. *Biometrika*, 93(3):509–524, 2006.
- [7] Philipp Heesen and Arnold Janssen. Dynamic adaptive multiple tests with finite sample FDR control. *Journal of Statistical Planning and Inference*, 168:38–51, 2016.
- [8] Lalit Jain and Kevin Jamieson. Firing bandits: Optimizing crowdfunding. In *International Conference on Machine Learning*, pages 2211–2219, 2018.
- [9] Adel Javanmard, Andrea Montanari, et al. Online rules for control of false discovery rate and false discovery exceedance. *The Annals of statistics*, 46(2):526–554, 2018.

- [10] Ramesh Johari, Pete Koomen, Leonid Pekelis, and David Walsh. 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*, pages 1517–1525, 2017.
- [11] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1168–1176, 2013.
- [12] Ang Li and Rina Foygel Barber. Multiple testing with the structure-adaptive Benjamini–Hochberg algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(1):45–74, 2019.
- [13] Aaditya Ramdas, Fanny Yang, Martin J Wainwright, and Michael I Jordan. Online control of the false discovery rate with decaying memory. In *Advances In Neural Information Processing Systems*, pages 5650–5659, 2017.
- [14] David S Robertson and James Wason. Online control of the false discovery rate in biomedical research. *arXiv preprint arXiv:1809.07292*, 2018.
- [15] Dinah Rosenberg, Eilon Solan, and Nicolas Vieille. Social learning in one-arm bandit problems. *Econometrica*, 75(6):1591–1611, 2007.
- [16] Sven Schmit, Virag Shah, and Ramesh Johari. Optimal testing in the experiment-rich regime. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 626–633, 2019.
- [17] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.
- [18] John Wilder Tukey. The problem of multiple comparisons. *Unpublished*, 1953.
- [19] Abraham Wald. On cumulative sums of random variables. *The Annals of Mathematical Statistics*, 15(3):283–296, 1944.
- [20] Fanny Yang, Aaditya Ramdas, Kevin G Jamieson, and Martin J Wainwright. A framework for multi-a (rmed)/b (andit) testing with online FDR control. In *Advances in Neural Information Processing Systems*, pages 5957–5966, 2017.

A Choice of Performance Measure

In this section we discuss our choice of measuring the performance of our policies π with the ratio of expectations $\text{reward}(\pi)/\text{cost}(\pi)$. We compare several different benchmarks and investigate how things differ if the learner has a budget of samples and a variable number tasks, rather than the other way around. We will show that all “natural” choices go essentially in the same direction, except for one (perhaps the most natural) which is surprisingly poorly suited to model our problem.

At a high level, a learner 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 learner might want to maximize (over $\pi = (\tau, \text{accept}) \in \Pi$) the objective $g_1(\pi, N)$ defined by

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

This is equivalent to our choice of maximizing

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

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 $g_1(\pi, N) = \Theta(\text{reward}(\pi)/\text{cost}(\pi))$. Furthermore, by the law of large numbers and Lebesgue's dominated convergence theorem, $g_1(\pi, N) \rightarrow \text{reward}(\pi)/\text{cost}(\pi)$ when $N \rightarrow \infty$ for any $\pi \in \Pi$.

Assume now that the constraint is on the total number of samples instead. We say that the learner 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 learner 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 $\pi = (\tau, \text{accept})$ is defined by

$$N = \min \left\{ m \in \mathbb{N} \mid \sum_{n=1}^m \tau(\mathbf{X}_n) \geq T \right\}$$

In this case, the learner might want to maximize the objective

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

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 \mathbf{X}_n from past tasks, for all deterministic functions f and all $n \in \mathbb{N}$, the two random variables $f(\mu_n, \mathbf{X}_n)$ and $\mathbb{I}\{N \geq n\}$ are independent, because $\mathbb{I}\{N \geq n\} = \mathbb{I}\{\sum_{i=1}^{n-1} \tau(\mathbf{X}_i) < T\}$ depends only on the random variables $\tau(\mathbf{X}_1), \dots, \tau(\mathbf{X}_{n-1})$. Hence

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

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

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

We can therefore apply Wald's identity [19] to deduce

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

which, together with

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

and

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

yields

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

if the denominator on the right hand side is positive, which happens as soon as $T > D^2$ by $ND \geq \sum_{n=1}^N \tau(\mathbf{X}_n) \geq T$ and $\text{cost}(\pi) \geq 1$. I.e., $g_2(\pi, T) = \Theta(\text{reward}(\pi)/\text{cost}(\pi))$ and noting that $\mathbb{E}[N] \geq T/D \rightarrow \infty$ if $T \rightarrow \infty$, we have once more that $g_2(\pi, T) \rightarrow \text{reward}(\pi)/\text{cost}(\pi)$ when $T \rightarrow \infty$ 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 our 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 model our problem. Consider as objective the average payoff of accepted values per amount of time used to make the decision, i.e.,

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

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 learner, consider the natural policy class $\{\pi_k\}_{k \in \{1, \dots, K\}} = \{(\tau_k, \text{accept})\}_{k \in \{1, \dots, K\}}$ given by

$$\tau_k(\mathbf{x}) = \min \left(k, \inf \left\{ n \in \mathbb{N} : |\bar{x}_n| \geq c \sqrt{\frac{\ln \frac{KN}{\delta}}{n}} \right\} \right), \quad \text{accept}(n, \mathbf{x}) = \mathbb{I} \left\{ \bar{x}_n \geq c \sqrt{\frac{\ln \frac{KN}{\delta}}{n}} \right\}$$

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

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

$$\left(\tau_K(\mathbf{X}_0), \text{accept}(\tau_K(\mathbf{X}_0), \mathbf{X}_0) \right) = \begin{cases} (k, 1) & \text{if } \mu_1 = 1 \\ (k, 0) & \text{if } \mu_1 = -1 \\ (K, 0) & \text{if } \mu_1 = 0 \end{cases}$$

i.e., the learner 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 $\text{reward}(\pi_K)/\text{cost}(\pi_K)$. Indeed, in this instance

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

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.

B Proof of Theorem 1

In this section, we give a detailed proof of Theorem 1, that we restate for ease of reading.

Theorem (Theorem 1, restated). *If Π is finite and Algorithm 1 is run with confidence $\delta \in (0, 1)$, exploration cap $N_{\text{ex}} \in \{1, \dots, N - 1\}$, and $N_{\text{ex}} \geq 8(D_K - 1)^2 \ln(4KN/\delta)$, then, with probability at least $1 - \delta$, its regret*

satisfies

$$R_N = \mathcal{O} \left(D_K \frac{N_{\text{ex}}}{N} + D_K \sqrt{\frac{\log(K N_{\text{ex}}/\delta)}{N_{\text{ex}}}} \right) \quad (15)$$

if the test at line 6 is false for all tasks $n \leq N_{\text{ex}}$ and

$$R_N = \mathcal{O} \left(\min \left(D_K \frac{N_{\text{ex}}}{N}, \frac{D_K^3 \log(K N_{\text{ex}}/\delta)}{\Delta^2 N} \right) \right) \quad (16)$$

otherwise, where we set $1/\Delta = \infty$ when $\Delta = 0$ (i.e., when there are at least two optimal policies).

In particular, if under the same assumptions we pick $N_{\text{ex}} = \lceil N^{2/3} \rceil$, then

$$R_N = \tilde{\mathcal{O}} \left(\min \left(\frac{D_K^3}{\Delta^2 N}, \frac{D_K}{N^{1/3}} \right) \right)$$

with probability at least $1 - \delta$, where the $\tilde{\mathcal{O}}$ notation hides only logarithmic terms.

Proof. The proof of this theorem relies on four technical lemmas, whose proofs we defer to the next section.

By Lemma 2, the event

$$\widehat{r}_n^-(k) \leq \text{reward}(\pi_k) \leq \widehat{r}_n^+(k) \quad \text{and} \quad \widehat{c}_n^-(k) \leq \text{cost}(\pi_k) \leq \widehat{c}_n^+(k) \quad (17)$$

occurs simultaneously for all $n = 1, \dots, N_{\text{ex}}$ and all $k = 1, \dots, \max(C_n)$ with probability at least $1 - \delta$. In order to avoid repetitions, from here on out we assume that (17) holds, i.e., all subsequent statements are assumed to hold over the common high-probability event (17).

By Lemma 3, if $\Delta > 0$ (i.e., if there is a unique optimal policy), then all suboptimal policies are eliminated after at most N'_{ex} tasks, where

$$N'_{\text{ex}} \leq \frac{288 D_K^2 \ln(4K N_{\text{ex}}/\delta)}{\Delta^2} + 1 \quad (18)$$

We now analyze separately the case in which the test at line 6 is true for some task $N'_{\text{ex}} \leq N_{\text{ex}}$, and the case in which the same test is always false.

Assume first that the test at line 6 is true for some task $N'_{\text{ex}} \leq N_{\text{ex}}$. Then, by Lemma 4,

$$R_N \leq \min \left(\frac{(2D_K + 1)N_{\text{ex}}}{N}, \frac{(2D_K + 1)(288 (D_K/\Delta)^2 \ln(4K N_{\text{ex}}/\delta) + 1)}{N} \right) \quad (19)$$

This proves (16).

Now, we upper bound the regret in the case in which the test at line 6 is false for all tasks $n \leq N_{\text{ex}}$, and line 7 is executed with $C_{N_{\text{ex}}+1}$ containing two or more policies. In this case, Lemma 5 yields

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

This, together with (16), gives (15). Finally, we let $N_{\text{ex}} = \lceil N^{2/3} \rceil$ and prove that up to logarithmic terms

$$R_N \leq \min \left(\frac{D_K^3}{\Delta^2 N}, \frac{D_K}{N^{1/3}} \right)$$

Assume first that $\Delta \leq 24(D_K/N^{1/3})\sqrt{\ln(4KN_{\text{ex}}/\delta)}$. Then the general regret bound (15) gives, up to logarithmic terms,

$$R_N \leq \frac{D_K}{N^{1/3}} = \min\left(\frac{D_K^3}{\Delta^2 N}, \frac{D_K}{N^{1/3}}\right)$$

Assume now that $\Delta > 24(D_K/N^{1/3})\sqrt{\ln(4KN_{\text{ex}}/\delta)}$. Then all suboptimal policies are eliminated after at most $N'_{\text{ex}} \leq N_{\text{ex}}$ tasks by (18). We can therefore apply our distribution-dependent bound (19) obtaining, up to logarithmic terms,

$$R_N \leq \frac{D_K^3}{\Delta^2 N} = \min\left(\frac{D_K^3}{\Delta^2 N}, \frac{D_K}{N^{1/3}}\right)$$

This gives (8) and concludes the proof of the theorem. \square

B.1 Technical lemmas

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

Lemma 2. *Under the assumptions of Theorem 1, the event*

$$\widehat{r}_n^-(k) \leq \text{reward}(\pi_k) \leq \widehat{r}_n^+(k) \quad \text{and} \quad \widehat{c}_n^-(k) \leq \text{cost}(\pi_k) \leq \widehat{c}_n^+(k) \quad (20)$$

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

Proof. Let, for all n, k ,

$$\varepsilon_n = \sqrt{\frac{\ln(4KN_{\text{ex}}/\delta)}{2n}}, \quad \bar{r}_n(k) = \widehat{r}_n^+(k) - 2\varepsilon_n, \quad \bar{c}_n(k) = \widehat{c}_n^+(k) - (D_k - 1)\varepsilon_n \quad (21)$$

Note that $\bar{c}_n(k)$ is the empirical average of n i.i.d. samples of $\text{cost}(\pi_k)$ for all n, k by definitions (21), (6), (1), (3), and point 4 in the formal definition of our protocol (Section 3). We show now that $\bar{r}_n(k)$ is the empirical average of n i.i.d. samples of $\text{reward}(\pi_k)$ for all n, k ; then claim (17) follows by Hoeffding's inequality. Indeed, by the conditional independence of the samples and being $\text{accept}(k, \mathbf{x})$ 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 $i > D_{\max(C_n)} (\geq D_k)$ by monotonicity of $k \mapsto D_k$,

$$\begin{aligned} \mathbb{E}\left[X_{n,i} \text{accept}(\tau_k(\mathbf{X}_n), \mathbf{X}_n) \mid \mu_n\right] &= \mathbb{E}\left[X_{n,i} \mid \mu_n\right] \mathbb{E}\left[\text{accept}(\tau_k(\mathbf{X}_n), \mathbf{X}_n) \mid \mu_n\right] \\ &= \mu_n \mathbb{E}\left[\text{accept}(\tau_k(\mathbf{X}_n), \mathbf{X}_n) \mid \mu_n\right] \\ &= \mathbb{E}\left[\mu_n \text{accept}(\tau_k(\mathbf{X}_n), \mathbf{X}_n) \mid \mu_n\right] \end{aligned}$$

Taking expectations with respect to μ_n on both sides of the above, and recalling definitions (21), (5), (1), (3), (4) proves the claim. Thus, Hoeffding's inequality implies, for all fixed n, k ,

$$\begin{aligned} \mathbb{P}\left(\widehat{r}_n^-(k) \leq \text{reward}(\pi_k) \leq \widehat{r}_n^+(k)\right) &= \mathbb{P}\left(|\bar{r}_n(k) - \text{reward}(\pi_k)| \leq 2\varepsilon_n\right) \geq 1 - \frac{\delta}{2KN_{\text{ex}}} \\ \mathbb{P}\left(\widehat{c}_n^-(k) \leq \text{cost}(\pi_k) \leq \widehat{c}_n^+(k)\right) &= \mathbb{P}\left(|\bar{c}_n(k) - \text{cost}(\pi_k)| \leq (D_k - 1)\varepsilon_n\right) \geq 1 - \frac{\delta}{2KN_{\text{ex}}} \end{aligned}$$

Applying a union bound shows that event (17) occurs simultaneously for all $n \in \{1, \dots, N_{\text{ex}}\}$ and $k \in \{1, \dots, \max(C_n)\}$ with probability at least $1 - \delta$. \square

Lemma 3. *Under the assumptions of Theorem 1, if the event (20) occurs simultaneously for all $n = 1, \dots, N_{\text{ex}}$ 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*

$$N'_{\text{ex}} \leq \frac{288 D_K^2 \ln(4KN_{\text{ex}}/\delta)}{\Delta^2} + 1 \quad (22)$$

Proof. Note first that (20) implies, for all $n \geq 2D_K^2 \ln(4KN_{\text{ex}}/\delta)$ (guaranteed by line 5) and all $k \in C_n$

$$\begin{aligned} \frac{\widehat{r}_n^-(k)}{\widehat{c}_n^+(k)} &\leq \frac{\text{reward}(\pi_k)}{\text{cost}(\pi_k)} \leq \frac{\widehat{r}_n^+(k)}{\widehat{c}_n^-(k)} && \text{if } \widehat{r}_n^+(k) \geq 0 \\ \frac{\widehat{r}_n^-(k)}{\widehat{c}_n^-(k)} &\leq \frac{\text{reward}(\pi_k)}{\text{cost}(\pi_k)} \leq \frac{\widehat{r}_n^+(k)}{\widehat{c}_n^+(k)} && \text{if } \widehat{r}_n^+(k) < 0 \end{aligned}$$

In other words, the interval

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

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

$$\varepsilon_n = \sqrt{\frac{\ln(4KN_{\text{ex}}/\delta)}{2n}}, \quad \bar{r}_n(k) = \widehat{r}_n^+(k) - 2\varepsilon_n, \quad \bar{c}_n(k) = \widehat{c}_n^+(k) - (D_k - 1)\varepsilon_n \quad (23)$$

If $\widehat{r}_n^+(k) \geq 0$, by the definitions in (23), the length of this confidence interval is

$$\frac{\bar{r}_n(k) + 2\varepsilon_n}{\bar{c}_n(k) - (D_k - 1)\varepsilon_n} - \frac{\bar{r}_n(k) - 2\varepsilon_n}{\bar{c}_n(k) + (D_k - 1)\varepsilon_n} = \frac{2\varepsilon_n(2\bar{c}_n(k) + (D_k - 1)\bar{r}_n(k))}{\bar{c}_n(k)^2 - (D_k - 1)^2\varepsilon_n^2} \leq 12D_K\varepsilon_n$$

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

$$\frac{\bar{r}_n(k) + 2\varepsilon_n}{\bar{c}_n(k) + (D_k - 1)\varepsilon_n} - \frac{\bar{r}_n(k) - 2\varepsilon_n}{\bar{c}_n(k) - (D_k - 1)\varepsilon_n} = \frac{2\varepsilon_n(2\bar{c}_n(k) - (D_k - 1)\bar{r}_n(k))}{\bar{c}_n(k)^2 - (D_k - 1)^2\varepsilon_n^2} \leq 12D_K\varepsilon_n$$

where, in addition to the considerations above, we used $0 < -\widehat{r}_n^+(k) < -\bar{r}_n(k) \leq 1$. Hence, as soon as the upper bound $12D_K\varepsilon_n$ 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 $n \geq 2D_K^2 \ln(4KN_{\text{ex}}/\delta)$ satisfies

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

This proves the result. \square

Lemma 4. *Under the assumptions of Theorem 1, if the event (20) 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 $N'_{\text{ex}} \leq N_{\text{ex}}$, then*

$$R_N \leq \min \left(\frac{(2D_K + 1)N_{\text{ex}}}{N}, \frac{(2D_K + 1)(288(D_K/\Delta)^2 \ln(4KN_{\text{ex}}/\delta) + 1)}{N} \right) \quad (24)$$

Proof. Note that if the test at line 6 is true, than by (20) there exists a unique optimal policy, i.e., we have $\Delta > 0$. We can therefore apply Lemma 3, obtaining a deterministic upper bound N''_{ex} on the number N'_{ex} of tasks needed to identify the optimal policy, where

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

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

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

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

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

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

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

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

Lemma 5. *Under the assumptions of Theorem 1, if the event (20) 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 $n \leq N_{\text{ex}}$ (i.e., if line 7 is executed with $C_{N_{\text{ex}}+1}$ containing two or more policies), then*

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

Proof. Note first that by (20) and the definition of C_n (line 5), all optimal policies belong to $C_{N_{\text{ex}}+1}$. Let, for all n, k ,

$$\varepsilon_n = \sqrt{\frac{\ln(4K N_{\text{ex}}/\delta)}{2n}}, \quad \bar{r}_n(k) = \hat{r}_n^+(k) - 2\varepsilon_n, \quad \bar{c}_n(k) = \hat{c}_n^+(k) - (D_k - 1)\varepsilon_n \quad (25)$$

By (20) and the definitions of k' , $\hat{r}_n^\pm(k)$, and ε_n (line 7, (5), (5), and (25) respectively), for all optimal policies π_{k^*} , if $\hat{r}_{N_{\text{ex}}}^+(k^*) \geq 0$, then also $\hat{r}_{N_{\text{ex}}}^+(k') \geq 0^6$ and

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

where all the denominators are positive because $N_{\text{ex}} \geq 8(D_K - 1)^2 \ln(4K N_{\text{ex}}/\delta)$ and the last inequality follows by $(a + b)/(c - d) \leq a/c + (d + b)/(c - d)$ for all $a \leq 1, b \in \mathbb{R}, c \geq 1$, and $d < c$; next, if $\hat{r}_{N_{\text{ex}}}^+(k^*) < 0$ but $\hat{r}_{N_{\text{ex}}}^+(k') \geq 0$ the exact same chain of inequalities hold; finally, if both $\hat{r}_{N_{\text{ex}}}^+(k^*) < 0$ and $\hat{r}_{N_{\text{ex}}}^+(k') < 0$, then $\hat{r}_{N_{\text{ex}}}^+(k) < 0$ for all $k \in C_{N_{\text{ex}}+1}$ ⁷, hence, by definition of k' and the same arguments used above

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

⁶Indeed, $k' \in \arg \max_{k \in C_{N_{\text{ex}}+1}} (\hat{r}_{N_{\text{ex}}}^+(k) \hat{c}_{N_{\text{ex}}}^-(k))$ in this case, and $\hat{r}_{N_{\text{ex}}}^+(k') \geq 0$ follows by the two inequalities $\hat{r}_{N_{\text{ex}}}^+(k') \hat{c}_{N_{\text{ex}}}^-(k') \geq \hat{r}_{N_{\text{ex}}}^+(k^*) \hat{c}_{N_{\text{ex}}}^-(k^*) \geq 0$.

⁷Otherwise k' would belong to the set $\arg \max_{k \in C_{N_{\text{ex}}+1}} (\hat{r}_{N_{\text{ex}}}^+(k) \hat{c}_{N_{\text{ex}}}^-(k))$ which in turn would be included in the set $\{k \in C_{N_{\text{ex}}+1} : \hat{r}_{N_{\text{ex}}}^+(k) \geq 0\}$ and this would contradict the fact that $\hat{r}_{N_{\text{ex}}}^+(k') < 0$.

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

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

where in the last inequality we lower bounded the denominator by $1/2$ using $\text{cost}(\pi_{k'}) \geq 1$ and $\varepsilon_n \leq \varepsilon_{N_{\text{ex}}} \leq 1/2$ which follows by $n \geq N_{\text{ex}} \geq 8D_K^2 \ln(4KN_{\text{ex}}/\delta)$ and the monotonicity of $k \mapsto D_k$. Therefore, for all optimal policies π_{k^*} , 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

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

where we used $\frac{a}{b+a}(x-y) \geq x-y - \frac{b}{b+a}$ for all $a, b, y > 0$ and all $x \leq 1$ to lower bound the third line, then the monotonicity of $k \mapsto D_k$ and $2\mathbb{E}[D_{\max(C_n)}] \geq \mathbb{E}[\text{cost}(\pi_{k'})] \geq 1$ 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

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

□

C Countable sets of policies

In this section, we present all missing results from Section 5. We begin by giving a full proof of Lemma 1, whose statement we recall here.

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

Proof. Note first that $\widehat{r}_{2j}^- + 2\varepsilon_j$ (line 2) is an empirical average of m_j i.i.d. unbiased estimators of $\text{reward}(\pi_{2j})$. Indeed, being $\text{accept}(k, \mathbf{x})$ 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 4 of our online protocol, Section 3), for all tasks n performed at line 2 during iteration j and all $i > D_{2j}$,

$$\begin{aligned} \mathbb{E} \left[X_{n,i} \text{accept}(\tau_{2j}(\mathbf{X}_n), \mathbf{X}_n) \mid \mu_n \right] &= \mathbb{E} [X_{n,i} \mid \mu_n] \mathbb{E} \left[\text{accept}(\tau_{2j}(\mathbf{X}_n), \mathbf{X}_n) \mid \mu_n \right] \\ &= \mu_n \mathbb{E} \left[\text{accept}(\tau_{2j}(\mathbf{X}_n), \mathbf{X}_n) \mid \mu_n \right] = \mathbb{E} \left[\mu_n \text{accept}(\tau_{2j}(\mathbf{X}_n), \mathbf{X}_n) \mid \mu_n \right] \end{aligned}$$

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

$$\mathbb{P}\left(\widehat{r}_{2^j}^- > \text{reward}(\pi_{2^j})\right) = \mathbb{P}\left((\widehat{r}_{2^j}^- + 2\varepsilon_j) - \text{reward}(\pi_{2^j}) > 2\varepsilon_j\right) \leq \frac{\delta}{j(j+1)}$$

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

$$\{\widehat{r}_{2^j}^- \leq \text{reward}(\pi_{2^j})\} \wedge \{\bar{c}_{2^l} \leq \text{cost}(\pi_{2^l}) + D_{2^l} \varepsilon_l\} \quad \forall j \leq j_0, \forall l > j_0 \quad (26)$$

occurs with probability at least

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

Note now that for each policy π_k with $\text{reward}(\pi_k) \geq 0$ and each optimal policy π_{k^*} ,

$$\frac{\text{reward}(\pi_k)}{D_k} \leq \frac{\text{reward}(\pi_k)}{\text{cost}(\pi_k)} \leq \frac{\text{reward}(\pi_{k^*})}{\text{cost}(\pi_{k^*})} \leq \frac{1}{\text{cost}(\pi_{k^*})} \quad (27)$$

Hence, all optimal policies π_{k^*} satisfy $\text{cost}(\pi_{k^*}) \leq D_k / \text{reward}(\pi_k)$ for all policies π_k with $\text{reward}(\pi_k) > 0$. Being durations sorted by index, for all $k \leq h$

$$\text{cost}(\pi_k) = \mathbb{E}[\text{cost}(\pi_k, \mu_0)] \leq \mathbb{E}[\text{cost}(\pi_h, \mu_0)] = \text{cost}(\pi_h) \quad (28)$$

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

$$\text{cost}(\pi_k) \stackrel{(28)}{\geq} \text{cost}(\pi_K) \stackrel{(26)}{\geq} \bar{c}_K - D_K \varepsilon_{\log_2 K} \stackrel{\text{line 7}}{>} \frac{D_{k_0}}{\widehat{r}_{k_0}^-} \geq \frac{D_{k_0}}{\text{reward}(k_0)}$$

where $\text{reward}(k_0) \geq \widehat{r}_{k_0}^- > 0$ by (26) and line (3); i.e., π_k do not satisfy (27). Therefore, with probability at least $1 - \delta$, all optimal policies π_{k^*} satisfy $k^* \leq K$. \square

We now state a lemma upper bounding the expected cost of Algorithm 2.

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

Proof. Recall the definition of $m_j(\varepsilon, \delta)$ (13) and m_j (initialization of Algorithm 2). Note that, by definition, $\varepsilon = \min\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{j_1}\} > 0$. Algorithm 2 (ESC) draw samples only when lines 2 or 6 are executed. Whenever line 2 is executed ($j = 1, \dots, j_0$) the algorithm performs m_j tasks drawing $2D_{2^j}$ samples each time. Similarly, whenever line 6 is executed ($l = j_0 + 1, \dots, j_1$) 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

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

\square

Consider any algorithm \mathcal{A} requiring the knowledge of an upper bound on k^* , for some optimal policy π_{k^*} , such as Algorithm 1 (CAPE). We can spend a portion of the N tasks to run Algorithm 2 (ESC) as a pre-processing step, then run \mathcal{A} with the upper bound K determined by ESC. 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.

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.

D An impossibility result

In this section we show 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 $\pi_1 = (1, \text{accept})$ 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 $\text{accept}(1, \mathbf{x}) = x_1$ for all $\mathbf{x} = (x_1, x_2, \dots)$. 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 Appendix A 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

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

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 $f: \{0, 1\} \rightarrow \mathbb{R}$ satisfies $\mathbb{E}[f(X)] = \mathbb{E}[X]^2$, then f also satisfies*

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

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

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

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