On the complexity of All ε-Best Arms Identification

Aymen Al Marjani[1]✉, Tomas Kocak[2], and Aurélien Garivier[1]

[1] UMPA, ENS Lyon, Lyon, France
{aymen.al_marjani,aurelien.garivier}@ens-lyon.fr
[2] University of Potsdam tomas.kocak@gmail.com

Abstract. We consider the question introduced by [16] of identifying all the ε-optimal arms in a finite stochastic multi-armed bandit with Gaussian rewards. We give two lower bounds on the sample complexity of any algorithm solving the problem with a confidence at least $1-\delta$. The first, unimprovable in the asymptotic regime, motivates the design of a Track-and-Stop strategy whose average sample complexity is asymptotically optimal when the risk δ goes to zero. Notably, we provide an efficient numerical method to solve the convex max-min program that appears in the lower bound. Our method is based on a complete characterization of the alternative bandit instances that the optimal sampling strategy needs to rule out, thus making our bound tighter than the one provided by [16]. The second lower bound deals with the regime of high and moderate values of the risk δ, and characterizes the behavior of any algorithm in the initial phase. It emphasizes the linear dependency of the sample complexity in the number of arms. Finally, we report on numerical simulations demonstrating our algorithm's advantage over state-of-the-art methods, even for moderate risks.

Keywords: Multi-armed bandits · Best-arm identification · Pure exploration.

1 Introduction

The problem of finding all the ε-good arms was recently introduced by [16]. For a finite family of distributions $(\boldsymbol{\nu}_a)_{a\in[K]}$ with vector of mean rewards $\boldsymbol{\mu} = (\mu_a)_{a\in[K]}$, the goal is to return $G_\varepsilon(\boldsymbol{\mu}) \triangleq \{a \in [K] : \mu_a \geq \max_i \mu_i - \varepsilon\}$ in the additive case and $G_\varepsilon(\boldsymbol{\mu}) \triangleq \{a \in [K] : \mu_a \geq (1-\varepsilon)\max_i \mu_i\}$ in the multiplicative case. This problem is closely related to two other pure-exploration problems in the multi-armed bandit literature, namely the TOP$-k$ arms selection and the THRESHOLD bandits. The former aims to find the k arms with the highest means, while the latter seeks to identify all arms with means larger than a given threshold s. As argued by [16], finding all the ε-good arms is a more robust objective than the TOP-K and THRESHOLD problems, which require some prior knowledge of the distributions in order to return a relevant set of solutions. Take for example drug discovery applications, where the goal is to perform an initial selection of potential drugs through *in vitro* essays before conducting more expensive clinical trials: setting the number of arms k too high or the threshold s too low may result into poorly performing solutions. Conversely, if we set k to a small number or the threshold s too high we might miss promising drugs that will prove to be more efficient under careful examination. The All-ε objective circumvents this issues by requiring to return all drugs whose efficiency lies within a certain range from the best. In this paper, we want to identify $G_\varepsilon(\boldsymbol{\mu})$ in a PAC learning framework with fixed confidence: for a risk level δ, the algorithm samples arms $a \in [K]$ in a sequential manner to gather information about the distribution means $(\mu_a)_{a\in[K]}$ and returns an estimate $\widehat{G}_\varepsilon$ such that $\mathbb{P}_{\boldsymbol{\mu}}(\widehat{G}_\varepsilon \neq G_\varepsilon(\boldsymbol{\mu})) \leq \delta$. Such an algorithm is called δ-PAC and its performance is measured by the expected number of samples $\mathbb{E}[\tau_\delta]$, also called the *sample complexity*, needed to return a good answer with high probability. [16] provided two lower bounds on the sample complexity: fhe first bound is based on a classical change-of-measure argument and exhibits the behavior of sample complexity in the low confidence regime ($\delta \to 0$). The second bound resorts to the Simulator technique [18] combined with an algorithmic reduction to Best Arm Identification and shows the dependency of the sample complexity on the number of arms K for moderate values of δ. They also proposed FAREAST, an algorithm matching the first lower bound, up to some numerical constants and log factors, in the asymptotic regime $\delta \to 0$. Our contributions can be summarized as follows:

- Usual lower bounds on the sample complexity write as $f(\nu)\log(1/\delta) + g(\nu)$ for an instance ν. We derive a tight bound in terms of the first-order term which writes as $T^*_\varepsilon(\boldsymbol{\mu})\log(1/\delta)$, where the characteristic time $T^*_\varepsilon(\boldsymbol{\mu})$ is the value of a concave max-min optimization program. Our bound is tight in the sense that any lower bound of the form $f(\nu)\log(1/\delta)$ that holds for all $\delta \in (0,1)$ is such that $f(\nu) \leq T^*_\varepsilon(\boldsymbol{\mu})$. To do so, we investigate all the possible alternative instances $\boldsymbol{\lambda}$ that one can obtain from the original problem $\boldsymbol{\mu}$ by a change-of-measure, including (but not only) the ones that were considered by [16].
- We derive a second lower bound that writes as $g(\nu)$ in Theorem 2. $g(\nu)$ shows an additional linear dependency on the number of arms which is negligible when $\delta \to 0$ but can be dominant for moderate values of the risk. This result generalizes Theorem 4.1 in [16], since it also includes cases where there can be several arms with means close to the top arm. The proof of this result relies on a personal rewriting of the Simulator method of [18] which was proposed for the Best Arm Identification and TOP-k problems. As we explain in Section 3.3, our proof can be adapted to derive lower bounds for other pure exploration problems, *without resorting to algorithmic reduction of these problems to Best Arm Identification.* Therefore, we believe that the proof itself constitutes a significant contribution.
- We present two efficient methods to solve the minimization sub-problem (resp. the entire max-min program) that defines the characteristic time. These methods are used respectively in the stopping and sampling rule of

our Track-and-Stop algorithm, whose sample complexity matches the lower bound when δ tends to 0.

– Finally, to corroborate our asymptotic results, we conduct numerical experiments for a wide range of the confidence parameters and number of arms. Empirical evaluation shows that Track-and-Stop is optimal either for a small number of arms K or when δ goes to 0, and excellent in practice for much larger values of K and δ. We believe these are significant improvements in performance to be of interest for ML practitioners seeking solutions for this kind of problem.

In Section 2 we introduce the setting and the notation. Section 3 is devoted to our lower bounds on the sample complexity of identifying the set of ε-good arms and the pseudo-code of our algorithm, along with the theoretical guarantees on its sample complexity. In Sections 4 and 5, we present our method for solving the optimization program that defines the characteristic time, which is at the heart of the sampling and stopping rules of our algorithm.

2 Setting and notation

The stochastic multi-armed bandit is a sequential learning framework where a learner faces a set of unknown probability distributions $(\nu_a)_{a\in[K]}$ with means $(\mu_a)_{a\in[K]}$, traditionally called *arms*. The learner collects information on the distributions by, at each time step t, choosing an arm based on past observations, and receiving an independent sample of this arm. The goal of *fixed-confidence pure exploration* is to answer some question about this set of distributions while using a minimum number of samples. In our case, we define the set of ε-good arms as $G_\varepsilon(\boldsymbol{\mu}) \triangleq \{a \in [K] : \mu_a \geq \max_i \mu_i - \varepsilon\}$; we wish to devise an algorithm that will collect samples and stop as soon as it can produce an estimate of $G_\varepsilon(\boldsymbol{\mu})$ that is certified to be correct with a prescribed probability $1-\delta$. This algorithm has three components. The *sampling rule* is $\{\pi_t\}_{t\geq 1}$, where $\pi_t(a|\ a_1, r_1, \ldots, a_{t-1}, r_{t-1})$ denotes the probability of choosing arm a at step t after a sequence of choices $(a_1, \ldots, a_{t-1})$ and the corresponding observations $(r_1, \ldots, r_t)$. The *stopping rule* τ_δ is a stopping time w.r.t the filtration of sigma-algebras $\mathcal{F}_t = \sigma(a_1, r_1, \ldots, a_{t-1}, r_{t-1})$ generated by the observations up to time t. Finally, the *recommendation rule* $\widehat{G}_\varepsilon$ is measurable w.r.t $\mathcal{F}_{\tau_\delta}$ and should satisfy $\mathbb{P}_{\nu,\mathcal{A}}(\widehat{G}_\varepsilon = G_\varepsilon(\boldsymbol{\mu})) \geq 1-\delta$. Algorithms obeying this inequality are called *δ-correct*, and among all of them we aim to find one with a minimal expected stopping time $\mathbb{E}_{\nu,\mathcal{A}}[\tau_\delta]$. In this work, like in [16], we restrict our attention to Gaussian arms with variance one. Even though this assumption is not mandatory, it considerably simplifies the presentation of the results[3].

[3] For σ^2-subgaussian distributions, we only need to multiply our bounds by σ^2. For bandits coming from another single-parameter exponential family, we lose the closed-form expression of the best response oracle that we have in the Gaussian case, but one can use binary search to solve the best response problem.

3 Lower bounds and asymptotically matching algorithm

We start by proving a lower bound on the sample complexity of any δ-correct algorithm. This lower bound will later motivate the design of our algorithm.

3.1 First lower bound

Let Δ_K denote the K-dimensional simplex and $\mathrm{kl}(p,q)$ be the KL-divergence between two Bernoulli distributions with parameters p and q. Finally, define the set of *alternative* bandit problems $\mathrm{Alt}(\boldsymbol{\mu}) = \{\boldsymbol{\lambda} \in \mathbb{R}^K : G_\varepsilon(\boldsymbol{\mu}) \neq G_\varepsilon(\boldsymbol{\lambda})\}$. Using change-of-measure arguments introduced by [13] , we derive the following lower bound on the sample complexity in our special setting.

Proposition 1. *For any δ-correct strategy $\mathcal{A}$ and any bandit instance $\boldsymbol{\mu}$, the expected stopping time τ_δ can be lower-bounded as*

$$\mathbb{E}_{\nu,\mathcal{A}}[\tau_\delta] \geq T^*_\varepsilon(\boldsymbol{\mu}) \log(1/2.4\delta)$$

where

$$T^*_\varepsilon(\boldsymbol{\mu})^{-1} \triangleq \sup_{\boldsymbol{\omega} \in \Delta_K} T_\varepsilon(\boldsymbol{\mu},\boldsymbol{\omega})^{-1} \qquad \textit{and} \tag{1}$$

$$T_\varepsilon(\boldsymbol{\mu},\boldsymbol{\omega})^{-1} \triangleq \inf_{\boldsymbol{\lambda} \in \mathrm{Alt}(\boldsymbol{\mu})} \sum_{a \in [K]} \omega_a \frac{(\mu_a - \lambda_a)^2}{2} \ . \tag{2}$$

The characteristic time $T^*_\varepsilon(\boldsymbol{\mu})$ above is an instance-specific quantity that determines the difficulty of our problem. The optimization problem in the definition of $T^*_\varepsilon(\boldsymbol{\mu})$ can be seen as a two-player game between an algorithm which samples each arm a proportionally to ω_a and an adversary who chooses an alternative instance $\boldsymbol{\lambda}$ that is difficult to distinguish from $\boldsymbol{\mu}$ under the algorithm's sampling scheme. This suggests that an optimal strategy should play the optimal allocation $\boldsymbol{\omega}^*$ that maximizes the optimization problem (1) and, as a consequence, rules out all alternative instances as fast as possible. This motivates our algorithm, presented in Section 3.2.

3.2 Algorithm

We propose a simple Track-and-Stop strategy similar to the one proposed by [8] for the problem of Best-Arm Identification. It starts by sampling once from every arm $a \in [K]$ and constructs an initial estimate $\widehat{\boldsymbol{\mu}}_K$ of the vector of mean rewards $\boldsymbol{\mu}$. After this burn-in phase, the algorithm enters a loop where at every iteration it plays arms according to the estimated optimal sampling rule (3) and updates its estimate $\widehat{\boldsymbol{\mu}}_t$ of the arms' expectations. Finally, the algorithm checks if the stopping rule (4) is satisfied, in which case it stops and returns the set of empirically ε-good arms.

Sampling rule: our sampling rule performs so-called C-tracking: first, we compute $\widetilde{\boldsymbol{\omega}}(\widehat{\boldsymbol{\mu}}_t)$, an allocation vector which is $\frac{1}{\sqrt{t}}$-optimal in the lower-problem (1) for the instance $\widehat{\boldsymbol{\mu}}_t$. Then we project $\widetilde{\boldsymbol{\omega}}(\widehat{\boldsymbol{\mu}}_t)$ on the set $\Delta_K^{\eta_t} = \Delta_K \cap [\eta_t, 1]^K$. Given the projected vector $\widetilde{\boldsymbol{\omega}}^{\eta_t}(\widehat{\boldsymbol{\mu}}_t)$, the next arm to sample from is defined by:

$$a_{t+1} = \arg\min_a N_a(t) - \sum_{s=1}^{t} \widetilde{\boldsymbol{\omega}}_a^{\eta_t}(\widehat{\boldsymbol{\mu}}_s) \tag{3}$$

where $N_a(t)$ is the number of times arm a has been pulled up to time t. In other words, we sample the arm whose number of visits is farther behind its corresponding sum of empirical optimal allocations. In the long run, as our estimate $\widehat{\boldsymbol{\mu}}_t$ tends to the true value $\boldsymbol{\mu}$, the sampling frequency $N_a(t)/t$ of every arm a will converge to the oracle optimal allocation $\boldsymbol{\omega}_a^*(\boldsymbol{\mu})$. The projection on $\Delta_K^{\eta_t}$ ensures exploration at minimal rate $\eta_t = \frac{1}{2\sqrt{(K^2+t)}}$ so that no arm is left-behind because of bad initial estimates.

Stopping rule: To be sample-efficient, the algorithm should should stop as soon as the collected samples are sufficiently informative to declare that $G_\varepsilon(\widehat{\boldsymbol{\mu}}_t) = G_\varepsilon(\boldsymbol{\mu})$ with probability larger than $1-\delta$. For this purpose we use the Generalized Likelihood Ratio (GLR) test [3]. We define the Z-statistic:

$$Z(t) = t \times T_\varepsilon\left(\widehat{\boldsymbol{\mu}}_t, \frac{\boldsymbol{N}(t)}{t}\right)^{-1}$$

where $\boldsymbol{N}(t) = \big(N_a(t)\big)_{a\in[K]}$. As shown in [8,6], the Z-statistic is equal to the ratio of the likelihood of observations under the most likely model where $G_\varepsilon(\widehat{\boldsymbol{\mu}}_t)$ is the correct answer, i.e. $\widehat{\boldsymbol{\mu}}_t$, to the likelihood of observations under the most likely model where $G_\varepsilon(\widehat{\boldsymbol{\mu}}_t)$ is not the set of ε-good arms. The algorithm rejects the hypothesis $G_\varepsilon(\widehat{\boldsymbol{\mu}}_t) \neq G_\varepsilon(\boldsymbol{\mu})$ and stops as soon as this ratio of likelihoods becomes larger than a certain threshold $\beta(\delta, t)$, properly tuned to ensure that the algorithm is δ-PAC. The stopping rule is defined as:

$$\tau_\delta = \inf\big\{t \in \mathbb{N} \ : \ Z(t) > \beta(t,\delta)\big\} \tag{4}$$

One can find many suitable thresholds from the bandit literature [7], [15], [12], all of which are of the order $\beta(\delta,t) \approx \log(1/\delta) + \frac{K}{2}\log(\log(t/\delta))$ is enough to ensure that $\mathbb{P}\big(G_\varepsilon(\widehat{\boldsymbol{\mu}}_{\tau_\delta}) \neq G_\varepsilon(\boldsymbol{\mu})\big) \le \delta$, i.e. that the algorithm is δ-correct.

Now we state our sample complexity result which we adapted from Theorem 14 in [8]. Notably, while their Track-and-Stop strategy relies on tracking the exact optimal weights to prove that the expected stopping time matches the lower bound when δ tends to zero, our proof shows that it is enough to track some slightly sub-optimal weights with a decreasing gap in the order of $\frac{1}{\sqrt{t}}$ to enjoy the same sample complexity guarantees.

Theorem 1. *For all $\delta \in (0,1)$, Track-and-Stop terminates almost-surely and its stopping time τ_δ satisfies:*

$$\limsup_{\delta\to 0} \frac{\mathbb{E}[\tau_\delta]}{\log(1/\delta)} \le T_\varepsilon^*(\boldsymbol{\mu}).$$

Algorithm 1: Track and Stop

```
**Input:** Confidence level $\delta$, accuracy parameter $\varepsilon$.
Pull each arm once and observe rewards $(r_a)_{a\in[K]}$.
Set initial estimate $\widehat{\boldsymbol{\mu}}_K = (r_1, \dots, r_K)^T$.
Set $t \leftarrow K$ and $N_a(t) \leftarrow 1$ for all arms $a$.
while Stopping condition (4) is not satisfied do
    Compute $\tilde{\boldsymbol{\omega}}(\widehat{\boldsymbol{\mu}}_t)$, a $\frac{1}{\sqrt{t}}$-optimal vector for (1) using mirror-ascent.
    Pull next arm $a_{t+1}$ given by (3) and observe reward $r_t$.
    Update $\widehat{\boldsymbol{\mu}}_t$ according to $r_t$.
    Set $t \leftarrow t+1$ and update $\big(N_a(t)\big)_{a\in[K]}$.
end
**Output:** Empirical set of $\varepsilon$-good arms: $G_\varepsilon(\widehat{\boldsymbol{\mu}}_{\tau_\delta})$
```

Remark 1. Suppose that the arms are ordered decreasingly $\mu_1 \geq \mu_2 \geq \dots \geq \mu_K$. [16] define the upper margin $\alpha_\varepsilon = \min\limits_{k\in G_\varepsilon(\boldsymbol{\mu})} \mu_k - (\mu_1 - \varepsilon)$ and provide a lower bound of the form $f(\nu)\log(1/\delta)$ where:

$$f(\nu) \triangleq 2\sum_{a=1}^{K} \max\left(\frac{1}{(\mu_1 - \varepsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\varepsilon - \mu_a)^2}\right).$$

It can be seen directly (or deduced from Theorem 1) that $f(\nu) \leq T^*_\varepsilon(\boldsymbol{\mu})$. In a second step, they proposed FAREAST, an algorithm whose sample complexity in the asymptotic regime $\delta \to 0$ matches their bound up to some universal constant c that does not depend on the instance ν. From Proposition 1, we deduce that $T^*_\varepsilon(\boldsymbol{\mu}) \leq cf(\nu)$, which can be seen directly from the particular changes of measure considered in that paper. The sample complexity of our algorithm improves upon previous work by multiplicative constants that can possibly be large, as illustrated in Section 6.

3.3 Lower bound for moderate confidence regime

The lower bound in Proposition 1 and the upper bound in Theorem 1 show that in the asymptotic regime $\delta \to 0$ the optimal sample complexity scales as $T^*_\varepsilon(\boldsymbol{\mu})\log(1/\delta)$. However, one may wonder whether this bound catches all important aspects of the complexity, especially for large or moderate values of the risk δ. Towards answering this question, we present the following lower bound which shows that there is an additional cost, linear in the number of arms, that any δ-PAC algorithm must pay in order to learn the set of All-ε good arms. Before stating our result, let us introduce some notation. We denote by $\mathbf{S}_K$ the group of permutations over $[K]$. For a bandit instance $\nu = (\nu_1, \dots, \nu_K)$ we define the *permuted instance* $\pi(\nu) = (\nu_{\pi(1)}, \dots, \nu_{\pi(K)})$. $\mathbf{S}_K(\nu) = \{\pi(\nu),\ \pi \in \mathbf{S}_K\}$ refers to the set of all permuted instances of ν. Finally, we will write $\pi \sim \mathbf{S}_K$ to indicate that a permutation is drawn uniformly at random from $\mathbf{S}_K$. These results are much inspired from [16], but come with quite different proofs that we hope can be useful to the community.

Theorem 2. *Fix $\delta \leq 1/10$ and $\varepsilon > 0$. Consider an instance ν such that there exists at least one bad arm: $G_\varepsilon(\boldsymbol{\mu}) \neq [K]$. Without loss of generality, suppose the arms are ordered decreasingly $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$ and define the lower margin $\beta_\varepsilon = \min_{k \notin G_\varepsilon(\boldsymbol{\mu})} \mu_1 - \varepsilon - \mu_k$. Then any δ-PAC algorithm has an average sample complexity over all permuted instances satisfying*

$$\mathbb{E}_{\pi \sim \mathbf{S}_K} \mathbb{E}_{\pi(\nu)}[\tau_\delta] \geq \frac{1}{12|G_{\beta_\varepsilon}(\boldsymbol{\mu})|^3} \sum_{b=1}^{K} \frac{1}{(\mu_1 - \mu_b + \beta_\varepsilon)^2},$$

The proof of the lower bound can be found in Appendix C. In the special case where $|G_{2\beta_\varepsilon}| = 1$, then $|G_{\beta_\varepsilon}| = 1$ also (since $\{1\} \subset G_{\beta_\varepsilon} \subset G_{2\beta_\varepsilon}$) and we recover the bound in Theorem 4.1 of [16]. The lower bound above informs us that we must pay a linear cost in K, *even when there are several arms close to the top one*, provided that their cardinal does not scale with the total number of arms, i.e. $|G_{\beta_\varepsilon}| = \mathcal{O}(1)$.

The bound of Thm 2 can be arbitrarily large compared to $T^*_\varepsilon(\boldsymbol{\mu}) \log(1/\delta)$. Fix $\delta = 0.1$ and let $\varepsilon, \beta > 0$ with $\beta \ll \varepsilon$ and consider the instance such that $\mu_1 = \beta, \mu_K = -\varepsilon$ and $\mu_a = -\beta$ for $a \in [|2, K-1|]$. Then we show in Appendix C that $T^*_\varepsilon(\boldsymbol{\mu}) \log(1/\delta) = \mathcal{O}(1/\beta^2 + K/\varepsilon^2)$. In contrast the lower bound above scales as $\Omega(K/\beta^2)$. Since $\beta \ll \varepsilon$, the second bound exhibits a better scaling w.r.t the number of arms.

The intuition behind this result comes from the following observations: first, note that arms in $G_{\beta_\varepsilon}(\boldsymbol{\mu})$ must be sampled at least $\Omega(1/\beta_\varepsilon^2)$ times, because otherwise we might underestimate their means and misclassify the arms in $\arg\min_{k \notin G_\varepsilon(\boldsymbol{\mu})} \mu_1 - \varepsilon - \mu_k$ as good arms. Second, in the initial phase the algorithm does not know which arms belong to $G_{\beta_\varepsilon}(\boldsymbol{\mu})$ and we need at least $\Omega(1/(\mu_1 - \mu_b)^2)$ samples to distinguish any arm b from arms in $G_{\beta_\varepsilon}(\boldsymbol{\mu})$. Together, these observations tell us that we must pay a cost of $\Omega(\min(1/\beta_\varepsilon^2, 1/(\mu_1 - \mu_b)^2))$ samples to either declare that b is not in $G_{\beta_\varepsilon}(\boldsymbol{\mu})$ or learn its mean up to $\mathcal{O}(\beta_\varepsilon)$ precision. More generally, consider a pure exploration problem with a unique answer, where some particular arm $i^\star$[4] needs to be estimated up to some precision $\eta > 0$ in order to return the correct answer. In this case, one can adapt our proof, *without using any algorithmic reduction to Best Arm Identification*, to show that every arm a must be played at least $\Omega(1/(|\mu_{i^\star} - \mu_a| + \eta)^2)$ times. For example, consider the problem of testing whether the minimum mean of a multi-armed bandit is above or below some threshold γ. Let ν be an instance such that $\{a \in [K] : \ \mu_a < \gamma\} = \{i^\star\}$ and define $\eta \triangleq \gamma - \mu_{i^\star} > 0$. Then our proof

[4] or a subset of arms, as in our case.

can be adapted in a straightforward fashion to prove that any δ-PAC algorithm for this task has a sample complexity of at least $\Omega\big(\sum_{a=1}^{K}\frac{1}{(\mu_a-\mu_{i^\star}+\eta)^2}\big)$.[5]

4 Solving the min problem: Best response oracle

Note that Algorithm 1 requires to solve the best response problem, i.e. the minimization problem in (2), in order to be able to compute the Z-statistic of the stopping rule, and also to solve the entire lower bound problem in (1) to compute the optimal weights for the sampling rule. The rest of the paper is dedicated to presenting the tools necessary to solve these two problems. For a given vector $\boldsymbol{\omega}$, we want to compute the best response

$$\boldsymbol{\lambda}^*_{\varepsilon,\boldsymbol{\mu}}(\boldsymbol{\omega}) \triangleq \mathop{\arg\min}_{\boldsymbol{\lambda}\in \mathrm{Alt}(\boldsymbol{\mu})} \sum_{a\in[K]} \omega_a \frac{(\mu_a-\lambda_a)^2}{2}. \tag{5}$$

For the simplicity of the presentation, we assume that the arms are ordered decreasingly $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$ and start by presenting the additive case (i.e. $G_\varepsilon(\boldsymbol{\mu}) \triangleq \{a \in [K] : \mu_a \geq \max_i \mu_i - \varepsilon\}$). The multiplicative case can be treated in the same fashion and is deferred to appendix A. Finally, we denote by $B_\varepsilon(\boldsymbol{\mu}) \triangleq [K] \setminus G_\varepsilon(\boldsymbol{\mu})$ the set of bad arms.

Since an alternative problem $\boldsymbol{\lambda} \in \mathrm{Alt}(\boldsymbol{\mu})$ must have a different set of ε-optimal arms than the original problem $\boldsymbol{\mu}$, we can obtain it from $\boldsymbol{\mu}$ by changing the expected reward of some arms. We have two options to create an alternative problem $\boldsymbol{\lambda}$:

- **Making one of the ε-optimal arms bad**. We can achieve it by decreasing the expectation of some ε-optimal arm k while increasing the expectation of some other arm ℓ to the point where k is no more ε-optimal. This is illustrated in Figure 1.
- **Making one of the ε-sub-optimal arms good.** We can achieve it by increasing the expectation of some sub-optimal arm k while decreasing the expectations of the arms with the largest means -as many as it takes- to the point where k becomes ε-optimal. This is illustrated in Figure 1.

In the following, we solve both cases separately.

Case 1: Making one of the ε-optimal arms bad. Let $k \in G_\varepsilon(\boldsymbol{\mu})$ be one of the ε-optimal arms. In order to make arm k sub-optimal, we need to set the

[5] The phenomenon discussed above is essentially already discussed in [16], a very rich study of the problem. However, we do not fully understand the proof of Theorem 4.1. Define a sub-instance to be a bandit $\widetilde{\nu}$ with fewer arms $m \leq K$ such that $\{\widetilde{\nu}_1,\dots,\widetilde{\nu}_m\} \subset \{\nu_1,\dots,\nu_K\}$. Lemma D.5 in [16] actually shows that there exists some sub-instance of ν on which the algorithm must pay $\Omega(\sum_{b=2}^{m} 1/(\mu_1-\mu_b)^2)$ samples. But this does not imply that such cost must be paid for the instance of interest ν instead of some sub-instance with very few arms.

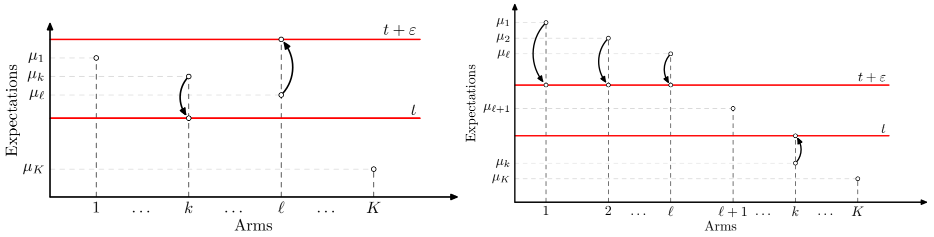

Fig. 1. Left: Making One of the ε-Optimal Arms Bad. Right: Making One of the ε-Sub-Optimal Arms Good.

expectation of arm k to some value $\lambda_k = t$ and the maximum expectation over all arms to $\max_a \lambda_a = t + \varepsilon$. Note that the index of the arm ℓ with maximum expectation can be chosen in $G_\varepsilon(\boldsymbol{\mu})$. Indeed, if we choose some arm from $B_\varepsilon(\mu)$ to become the arm with maximum expectation in λ then we would make an ε-suboptimal arm good which is covered in the other case below. The expectations of all the other arms should stay the same as in the instance $\boldsymbol{\mu}$, since changing their values would only increase the value of the objective. Now given indices k and ℓ, computing the optimal value of t is rather straightforward since the objective function simplifies to

$$\omega_k \frac{(\mu_k - t)^2}{2} + \omega_\ell \frac{(\mu_\ell - t - \varepsilon)^2}{2}$$

for which the optimal value of t is:

$$t = \overline{\boldsymbol{\mu}}_\varepsilon^{k,\ell}(\boldsymbol{\omega}) \triangleq \frac{\omega_k \mu_k + \omega_\ell(\mu_\ell - \varepsilon)}{\omega_k + \omega_\ell}.$$

and the corresponding alternative bandit is:

$$\boldsymbol{\lambda}_\varepsilon^{k,\ell}(\boldsymbol{\omega}) \triangleq (\mu_1, \ldots, \underbrace{\overline{\boldsymbol{\mu}}_\varepsilon^{k,\ell}(\boldsymbol{\omega})}_{\text{index } k}, \ldots, \underbrace{\overline{\boldsymbol{\mu}}_\varepsilon^{k,\ell}(\boldsymbol{\omega}) + \varepsilon}_{\text{index } \ell}, \ldots, \mu_K)^\mathsf{T}.$$

The last step is taking the pair of indices $(k, \ell) \in G_\varepsilon(\boldsymbol{\mu}) \times (G_\varepsilon(\boldsymbol{\mu}) \setminus \{k\})$ with the minimal value in the objective (2).

Case 2: Making one of the sub-optimal arms good. Let $k \in B_\varepsilon(\boldsymbol{\mu})$ be a sub-optimal arm, if such arm exists, and denote by t the value of its expectation in $\boldsymbol{\lambda}$. In order to make this arm ε-optimal, we need to decrease the expectations of all the arms that are above the threshold $t+\varepsilon$. We pay a cost of $\frac{1}{2}\omega_k(t-\mu_k)^2$ for moving arm k and of $\frac{1}{2}\omega_i(t+\varepsilon-\mu_i)^2$ for every arm i such that $\mu_i > t+\varepsilon$. Consider the functions:

$$f_k(t) = \frac{1}{2}\omega_k(t-\mu_k)^2$$

and for $i \in [K] \setminus \{k\}$

$$f_i(t) = \begin{cases} \frac{1}{2}\omega_i(t+\varepsilon-\mu_i)^2 & \text{for } t < \mu_i - \varepsilon, \\ 0 & \text{for } t \geq \mu_i - \varepsilon. \end{cases}$$

Each of these functions is convex. Therefore the function $f(t) = \sum_{i=1}^{K} f_i(t)$ is convex and has a unique minimizer t^*. One can easily check that $f'(\mu_k) \leq 0$ and $f'(\mu_1 - \varepsilon) \geq 0$, implying that $\mu_k - \varepsilon < \mu_k \leq t^* \leq \mu_1 - \varepsilon$. Therefore:

$$\ell = \min\{i \geq 1 \ : \ t^* > \mu_i - \varepsilon\} - 1$$

is well defined and satisfies $\ell \in [|1, k-1|]$. Note that by definition $\mu_{\ell+1} - \varepsilon < t^*$ and $t^* \leq \mu_a - \varepsilon$ for all $a \leq \ell$, hence:

$$0 = f'(t^*) = \omega_k(t^* - \mu_k) + \sum_{a=1}^{\ell} \omega_a(t^* + \varepsilon - \mu_a).$$

Implying that[6]:

$$t^* = \overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) \triangleq \frac{\omega_k \mu_k + \sum_{a=1}^{\ell} \omega_a(\mu_a - \varepsilon)}{\omega_k + \sum_{a=1}^{\ell} \omega_a}$$

and the alternative bandit in this case writes as:

$$\boldsymbol{\lambda}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) \triangleq (\underbrace{\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) + \varepsilon}_{\text{indices 1to } \ell}, \mu_{\ell+1}, \ldots, \underbrace{\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega})}_{\text{index } k}, \ldots, \mu_K)^{\intercal}.$$

Observe that since ℓ depends on t^*, we can't directly compute t^* from the expression above. Instead, we use the fact that ℓ is unique by definition. Therefore, to determine t^* one can compute $\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega})$ for all values of $\ell \in [|1, k-1|]$ and search for the index ℓ satisfying $\mu_{\ell+1} - \varepsilon < \overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) \leq \mu_\ell - \varepsilon$ and with minimum value in the objective (2).

As a summary, we have reduced the minimization problem over the infinite set $\mathrm{Alt}(\boldsymbol{\mu})$ to a combinatorial search over a finite number of alternative bandit instances whose analytical expression is given in the next definition.

Definition 1. *Let $\boldsymbol{\lambda}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega})$ be a vector created form $\boldsymbol{\mu}$ by replacing elements on positions k and ℓ (resp. 1 to ℓ), defined as:*

$$\boldsymbol{\lambda}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) \triangleq (\mu_1, \ldots, \underbrace{\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega})}_{\text{index } k}, \ldots, \underbrace{\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) + \varepsilon}_{\text{index } \ell}, \ldots, \mu_K)^{\intercal}$$

for $k \in G_\varepsilon(\boldsymbol{\mu})$ and

$$\boldsymbol{\lambda}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) \triangleq (\underbrace{\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) + \varepsilon}_{\text{indices 1to } \ell}, \mu_{\ell+1}, \ldots, \underbrace{\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega})}_{\text{index } k}, \ldots, \mu_K)^{\intercal}$$

for $k \in B_\varepsilon(\boldsymbol{\mu})$ where $\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega})$ is a weighted average of elements on positions k and ℓ (resp. 1 to ℓ) defined as:

$$\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega}) \triangleq \frac{\omega_k \mu_k + \omega_\ell(\mu_\ell - \varepsilon)}{\omega_k + \omega_\ell}$$

[6] $\overline{\boldsymbol{\mu}}_{\varepsilon}^{k,\ell}(\boldsymbol{\omega})$ has a different definition depending on k being a good or a bad arm.

for $k \in G_\varepsilon(\boldsymbol{\mu})$ and

$$\overline{\boldsymbol{\mu}}_\varepsilon^{k,\ell}(\boldsymbol{\omega}) \triangleq \frac{\omega_k \mu_k + \sum_{a=1}^{\ell} \omega_a(\mu_a - \varepsilon)}{\omega_k + \sum_{a=1}^{\ell} \omega_a}$$

for $k \in B_\varepsilon(\boldsymbol{\mu})$.

The next lemma then states that the best response oracle belongs to the finite set of $(\boldsymbol{\lambda}_\varepsilon^{k,\ell}(\boldsymbol{\omega}))_{k,\ell}$.

Lemma 1. *Using the previous definition, $\boldsymbol{\lambda}^*_{\varepsilon,\boldsymbol{\mu}}(\boldsymbol{\omega})$ can be computed as*

$$\boldsymbol{\lambda}^*_{\varepsilon,\boldsymbol{\mu}}(\boldsymbol{\omega}) = \operatorname*{arg\,min}_{\boldsymbol{\lambda} \in \Lambda_G \cup \Lambda_B} \sum_{a \in [K]} \omega_a \frac{(\mu_a - \lambda_a)^2}{2}$$

where

$$\Lambda_G = \{\boldsymbol{\lambda}_\varepsilon^{k,\ell}(\boldsymbol{\omega}) : k \in G_\varepsilon(\boldsymbol{\mu}), \ell \in G_\varepsilon(\boldsymbol{\mu})/\{k\}\}$$

and

$$\Lambda_B = \{\boldsymbol{\lambda}_\varepsilon^{k,\ell}(\boldsymbol{\omega}) : k \in B_\varepsilon(\boldsymbol{\mu}), \ell \in [|1, k-1|] \\ \text{s.t. } \mu_\ell \geq \overline{\boldsymbol{\mu}}_\varepsilon^{k,\ell}(\boldsymbol{\omega}) + \varepsilon > \mu_{\ell+1}\}.$$

5 Solving the max-min problem: Optimal weights

First observe that we can rewrite $T_\varepsilon(\boldsymbol{\mu}, .)^{-1}$ as a minimum of linear functions:

$$T_\varepsilon(\boldsymbol{\mu}, \boldsymbol{\omega})^{-1} = \inf_{\boldsymbol{d} \in \mathcal{D}_{\varepsilon,\boldsymbol{\mu}}} \boldsymbol{\omega}^\intercal \boldsymbol{d} \tag{6}$$

where

$$\mathcal{D}_{\varepsilon,\boldsymbol{\mu}} \triangleq \left\{ \left(\frac{(\lambda_a - \mu_a)^2}{2} \right)^\intercal_{a \in [K]} \mid \boldsymbol{\lambda} \in \mathrm{Alt}(\boldsymbol{\mu}) \right\}.$$

Note that by using $\mathcal{D}_{\varepsilon,\boldsymbol{\mu}}$ instead of $\mathrm{Alt}(\boldsymbol{\mu})$, the optimization function becomes simpler for the price of more complex domain (see Figure 2 for an example). As a result, $T_\varepsilon(\boldsymbol{\mu}, .)^{-1}$ is concave and we can compute its supergradients thanks to Danskin's Theorem [4] which we recall in the lemma below.

Lemma 2. *(Danskin's Theorem) Let $\boldsymbol{\lambda}^*(\boldsymbol{\omega})$ be a best response to $\boldsymbol{\omega}$ and define $\boldsymbol{d}^*(\boldsymbol{\omega}) \triangleq \big(\frac{(\boldsymbol{\lambda}^*(\boldsymbol{\omega})_a - \mu_a)^2}{2}\big)^\intercal_{a \in [K]}$. Then $\boldsymbol{d}^*(\boldsymbol{\omega})$ is a supergradient of $T_\varepsilon(\boldsymbol{\mu}, .)^{-1}$ at $\boldsymbol{\omega}$.*

Next we prove that $T_\varepsilon(\boldsymbol{\mu}, .)^{-1}$ is Liptschiz.

Lemma 3. *The function $\boldsymbol{\omega} \mapsto T_\varepsilon(\boldsymbol{\mu}, \boldsymbol{\omega})^{-1}$ is L-Lipschitz with respect to $\|\cdot\|_1$ for any*

$$L \geq \max_{a,b \in [K]} \frac{(\mu_a - \mu_b + \varepsilon)^2}{2} .$$

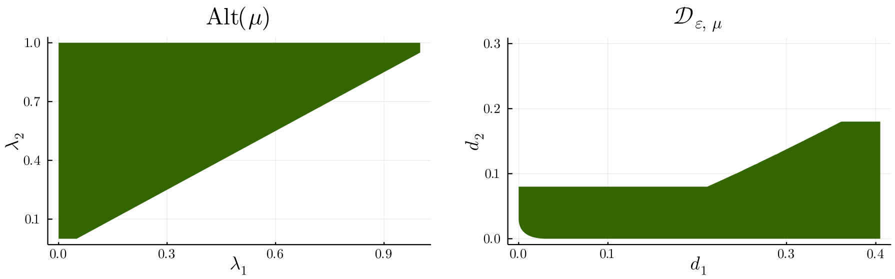

Fig. 2. Comparison of $\mathrm{Alt}(\boldsymbol{\mu})$ with Simple Linear Boundaries (First Figure) and $\mathcal{D}_{\varepsilon,\boldsymbol{\mu}}$ with Non-Linear Boundaries (Second Figure) for $\boldsymbol{\mu} = [0.9,\, 0.6]$ and $\varepsilon = 0.05$.

Proof. As we showed in Lemma 1, the best response $\boldsymbol{\lambda}^*_{\varepsilon,\boldsymbol{\mu}}(\boldsymbol{\omega})$ to $\boldsymbol{\omega}$ is created from $\boldsymbol{\mu}$ by replacing some of the elements by $\overline{\boldsymbol{\mu}}^{k,\ell}_{\varepsilon}(\boldsymbol{\omega})$ or $\overline{\boldsymbol{\mu}}^{k,\ell}_{\varepsilon}(\boldsymbol{\omega}) + \varepsilon$. We also know that $\overline{\boldsymbol{\mu}}^{k,\ell}_{\varepsilon}(\boldsymbol{\omega})$ is a weighted average of an element of $\boldsymbol{\mu}$ with one or more elements of $\boldsymbol{\mu}$ decreased by ε. This means that:

$$\max_{a\in[K]} \mu_a \geq \overline{\boldsymbol{\mu}}^{k,\ell}_{\varepsilon}(\boldsymbol{\omega}) \geq \min_{a\in[K]} \mu_a - \varepsilon$$

and, as a consequence, we have:

$$|\mu_i - \boldsymbol{\lambda}^*_{\varepsilon,\boldsymbol{\mu}}(\boldsymbol{\omega})_i| \leq \max_{a,b\in[K]} (\mu_a - \mu_b + \varepsilon)$$

for any $i \in [K]$. Let $f(\boldsymbol{\omega}) \triangleq T_\varepsilon(\boldsymbol{\mu}, \boldsymbol{\omega})^{-1}$. Using the last inequality and the definition of $\boldsymbol{d}^*(\boldsymbol{\omega})$, we can obtain:

$$\begin{aligned} f(\boldsymbol{\omega}) - f(\boldsymbol{\omega}') &\leq (\boldsymbol{\omega} - \boldsymbol{\omega}')^\intercal \boldsymbol{d}^*(\boldsymbol{\omega}') \\ &\leq \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_1 \|\boldsymbol{d}^*(\boldsymbol{\omega}')\|_\infty \\ &\leq \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_1 \max_{a,b\in[K]} \frac{(\mu_a - \mu_b + \varepsilon)^2}{2} \end{aligned}$$

for any $\boldsymbol{\omega}$, $\boldsymbol{\omega}' \in \Delta_K$.

As a summary $T_\varepsilon(\boldsymbol{\mu}, .)^{-1}$ is concave, Lipschitz and we have a simple expression to compute its supergradients through the best response oracle. Therefore we have all the necessary ingredients to apply a gradient-based algorithm in order to find the optimal weights and therefore, the value of $T^*_\varepsilon(\boldsymbol{\mu})$. The algorithm of our choice is the mirror ascent algorithm which provides the following guarantees:

Proposition 2. *[2] Let $\boldsymbol{\omega}_1 = (\frac{1}{K}, \ldots, \frac{1}{K})^\intercal$ and learning rate $\alpha_n = \frac{1}{L}\sqrt{\frac{2\log K}{n}}$. Then using mirror ascent algorithm to maximize a L-Lipschitz function f, with respect to $\|\cdot\|_1$, defined on Δ_K with generalized negative entropy $\Phi(\boldsymbol{\omega}) = \sum_{a\in[K]} \omega_a \log(\omega_a)$ as the mirror map enjoys the following guarantees:*

$$f(\boldsymbol{\omega}^*) - f\left(\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\omega}_n\right) \leq L\sqrt{\frac{2\log K}{N}}\ .$$

Computational complexity of our algorithm. To simplify the presentation and analysis, we chose to focus on the vanilla version of Track and Stop. However, in practice this requires solving the optimization program that appears in the lower bound at every time step, which can result in large run times. Nonetheless, we note that there are many possible adaptations of Track and Stop that reduce the computational complexity, while retaining the guarantees of asymptotic optimality in terms of the sample complexity (and with a demonstrated small performance loss experimentally). A first solution is to use Franke-Wolfe style algorithms [17,19], which only perform a gradient step of the optimization program at every step. Once can also apply the Gaming approach initiated by [5] which only needs to solve the best response problem, and runs a no-regret learner such as AdaHedge to determine the weights to be tracked at each step. This approach was used for example by [10] in a similar setting of Pure Exploration with semi-bandit feedback. Another adaptation is the Lazy Track-and-Stop [9], which updates the weights that are tracked by the algorithms every once in a while. We chose the latter solution in our implementation, where we updated the weights every $100K$ steps.

6 Experiments

We conducted three experiments to compare Track-and-Stop with state-of-the-art algorithms, mainly $(\mathrm{ST})^2$ and FAREAST from [16]. In the first experiment, we simulate a multi-armed bandit with Gaussian rewards of means $\boldsymbol{\mu} = [1, 1, 1, 1, 0.05]$, variance one and a parameter $\varepsilon = 0.9$. We chose this particular instance $\boldsymbol{\mu}$ because its difficulty is two-fold: First, the last arm μ_5 is very close to the threshold $\max_a \mu_a - \varepsilon$. Second, the argmax is realized by more than one arm, which implies that any algorithm must estimate all the means to high precision to produce a confident guess of $G_\varepsilon(\boldsymbol{\mu})$. Indeed, a small underestimation error of $\max_a \mu_a$ would mean wrongly classifying μ_5 as a good arm. We run the three algorithms for several values of δ ranging from $\delta = 0.1$ to $\delta = 10^{-10}$, with $N = 100$ Monte-Carlo simulations for each risk level. Figure 3 shows the expected stopping time along with the 10% and 90% quantiles (shaded area) for each algorithm. Track-and-Stop consistently outperforms $(\mathrm{ST})^2$ and FAREAST, even for moderate values of δ. Also note that, as we pointed out in Remark 1, the sample complexity of Track-and-Stop is within some multiplicative constant of $(\mathrm{ST})^2$.

Next, we examine the performance of the algorithms w.r.t the number of arms. For any given K, we consider a bandit problem μ similar to the previous instance: $\forall a \in [|1, K-1|]$, $\mu_a = 1$ and $\mu_K = 0.05$. We fix $\varepsilon = 0.9$ and $\delta = 0.1$ and run $N = 30$ Monte-Carlo simulations for each K. Figure 4 shows, in log-scale, the ratio of the sample complexities of $(\mathrm{ST})^2$ and FAREAST w.r.t to the sample complexity of Track-and-Stop. We see that Track-and-Stop performs better than $(\mathrm{ST})^2$ (resp. FAREAST) for small values of K. However when the number of arms grows larger than $K = 40$ (resp. $K = 60$), $(\mathrm{ST})^2$ (resp. FAREAST) have a smaller sample complexity.

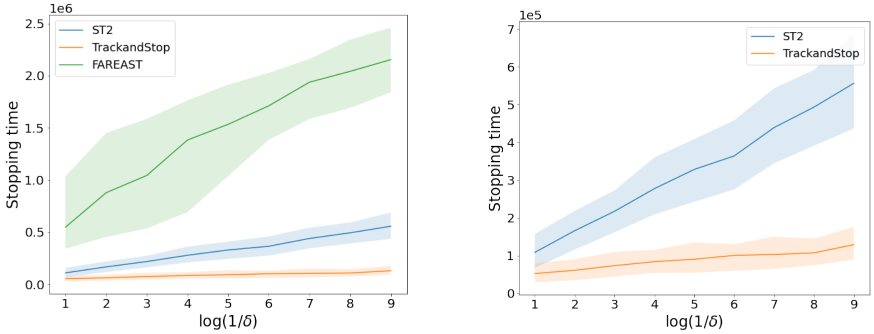

Fig. 3. Expected Stopping Time on $\boldsymbol{\mu} = [1, 1, 1, 1, 0.05]$. Left: All three Algorithms. Right: Track-and-Stop vs FAREAST.

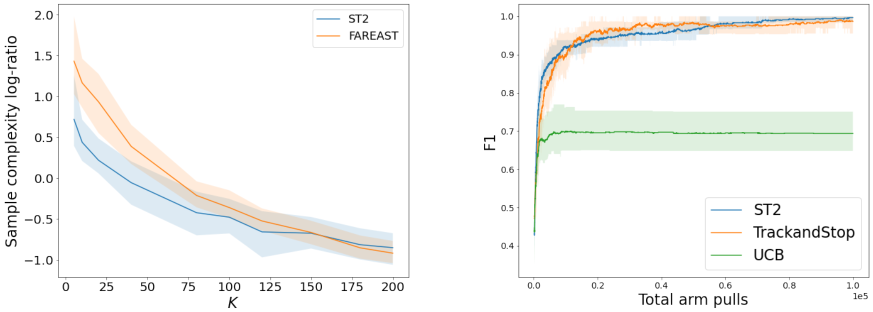

Fig. 4. Left: $\log_{10}\left(\mathbb{E}_{\mathrm{Alg}}[\tau_\delta]/\mathbb{E}_{\mathrm{TaS}}[\tau_\delta]\right)$ for Alg $\in \{(\mathrm{ST})^2, \mathrm{FAREAST}\}$ and TaS = Track-and-Stop, $K_{\min} = 5$ arms. Right: F1 scores for Cancer Drug discovery.

Finally, we rerun the Cancer Drug Discovery experiment from [16]. Note that this experiment is more adapted to a *fixed budget setting* where we fix a sampling budget and the algorithm stops once it has reached this limit, which is different from the *fixed confidence* setting that our algorithm was designed for. The goal is to find, among a list of 189 chemical compounds, potential inhibitors to **ACRVL1**, a Kinaze that researchers [1] have linked to several forms of cancer. We use the same dataset as [16], where for each compound a percent control[7] is reported. We fix a budget of samples $N = 10^5$ and try to find all the ε-good compounds in the multiplicative case with $\varepsilon = 0.8$. For each algorithm, we compute the F1-score[8] of

[7] percent control is a metric expressing the efficiency of the compound as an inhibitor against the target Kinaze.

[8] F1 score is the harmonic mean of precision (the proportion of arms in $\widehat{G}$ that are actually good) and recall (the proportion of arms in $G_\varepsilon(\boldsymbol{\mu})$ that were correctly returned in $\widehat{G}$).

its current estimate $\widehat{G}_\varepsilon = \{i \ : \ \widehat{\mu}_i \geq (1-\varepsilon)\max_a \widehat{\mu}_a\}$ after every iteration. The F1-score in this fixed-budget setting reflects how good is the sampling scheme of an algorithm, independently of its stopping condition. In Figure 4 we plot the average F1-score along with the 10% and 90% quantiles (shaded area). We see that $(\mathrm{ST})^2$ and Track-and-Stop have comparable performance and that both outperform UCB's sampling scheme.

7 Conclusion

We shed a new light on the sample complexity of finding all the ε-good arms in a multi-armed bandit with Gaussian rewards. We derived two lower bounds, identifying the characteristic time that reflects the true hardness of the problem in the asymptotic regime. Moreover, we proved a second bound highlighting an additional cost that is linear in the number of arms and can be arbitrarily larger than the first bound for moderate values of the risk. Then, capitalizing on an algorithm solving the optimization program that defines the characteristic time, we proposed an efficient Track-and-Stop strategy whose sample complexity matches the lower bound for small values of the risk level. Finally, we showed through numerical simulations that our algorithm outperforms state-of-the-art methods for bandits with small to moderate number of arms. Several directions are worth investigating in the future. Notably, we observe that while Track-and-Stop performs better in the fixed-K-small-δ regime, the elimination based algorithms $(\mathrm{ST})^2$ and FAREAST become more efficient in the large-K-fixed-δ regime. It would be interesting to understand the underlying tradeoff between the number of arms and confidence parameter. This will help design pure exploration strategies having best of both worlds guarantees.

Acknowledgements The authors acknowledge the support of the Chaire SeqALO (ANR-20-CHIA-0020-01) and of Project IDEXLYON of the University of Lyon, in the framework of the Programme Investissements d'Avenir (ANR-16-IDEX-0005).

References

1. Bocci, M., Sjölund, J., Kurzejamska, E., Lindgren, D., Marzouka, N.a.d., Bartoschek, M., Höglund, M., Pietras, K.: Activin receptor-like kinase 1 is associated with immune cell infiltration and regulates clec14a transcription in cancer. Angiogenesis **22**, 117–131 (02 2019). `https://doi.org/10.1007/s10456-018-9642-5`
2. Bubeck, S.: Convex optimization: Algorithms and complexity. Foundations and Trends in Machine Learning (2015)
3. Chernoff, H.: Sequential design of Experiments. The Annals of Mathematical Statistics **30**(3), 755–770 (1959)
4. Danskin, J.M.: The theory of max-min, with applications. Siam Journal on Applied Mathematics **14**, 641–664 (1966)

5. Degenne, R., Koolen, W.M., Ménard, P.: Non-asymptotic pure exploration by solving games. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), `https://proceedings.neurips.cc/paper/2019/file/8d1de7457fa769ece8d93a13a59c8552-Paper.pdf`
6. Garivier, A., Kaufmann, E.: Non-Asymptotic Sequential Tests for Overlapping Hypotheses and application to near optimal arm identification in bandit models . Sequential Analysis **40**, 61–96 (2021)
7. Garivier, A.: Informational confidence bounds for self-normalized averages and applications. 2013 IEEE Information Theory Workshop (ITW) (Sep 2013). `https://doi.org/10.1109/itw.2013.6691311`, `http://dx.doi.org/10.1109/ITW.2013.6691311`
8. Garivier, A., Kaufmann, E.: Optimal best arm identification with fixed confidence. Proceedings of the 29th Conference On Learning Theory pp. 998–1027 (2016)
9. Jedra, Y., Proutiere, A.: Optimal best-arm identification in linear bandits. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 10007–10017. Curran Associates, Inc. (2020), `https://proceedings.neurips.cc/paper/2020/file/7212a6567c8a6c513f33b858d868ff80-Paper.pdf`
10. Jourdan, M., Mutn'y, M., Kirschner, J., Krause, A.: Efficient pure exploration for combinatorial bandits with semi-bandit feedback. In: ALT (2021)
11. Kaufmann, E., Cappé, O., Garivier, A.: On the Complexity of Best Arm Identification in Multi-Armed Bandit Models. Journal of Machine Learning Research (2015)
12. Kaufmann, E., Koolen, W.M.: Mixture martingales revisited with applications to sequential tests and confidence intervals. arXiv preprint **arXiv:1811.11419** (2018)
13. Lai, T., Robbins, H.: Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics **6**(1), 4–2 (1985)
14. Lattimore, T., Szepesvári, C.: Bandit Algorithms. Cambridge University Press (2019)
15. Magureanu, S., Combes, R., Proutiere, A.: Lipschitz bandits: Regret lower bounds and optimal algorithms. In: Conference on Learning Theory (2014)
16. Mason, B., Jain, L., Tripathy, A., Nowak, R.: Finding all \epsilon-good arms in stochastic bandits. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 20707–20718. Curran Associates, Inc. (2020), `https://proceedings.neurips.cc/paper/2020/file/edf0320adc8658b25ca26be5351b6c4a-Paper.pdf`
17. Ménard, P.: Gradient ascent for active exploration in bandit problems. arXiv e-prints p. arXiv:1905.08165 (May 2019)
18. Simchowitz, M., Jamieson, K., Recht, B.: The simulator: Understanding adaptive sampling in the moderate-confidence regime. In: Kale, S., Shamir, O. (eds.) Proceedings of the 2017 Conference on Learning Theory. Proceedings of Machine Learning Research, vol. 65, pp. 1794–1834. PMLR, Amsterdam, Netherlands (07–10 Jul 2017), `http://proceedings.mlr.press/v65/simchowitz17a.html`
19. Wang, P.A., Tzeng, R.C., Proutiere, A.: Fast pure exploration via frank-wolfe. Advances in Neural Information Processing Systems **34** (2021)