On Regret with Multiple Best Arms

Yinglun Zhu
Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706
yinglun@cs.wisc.edu

Robert Nowak
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706
rdnowak@wisc.edu

Abstract

We study a regret minimization problem with the existence of multiple best/near-optimal arms in the multi-armed bandit setting. We consider the case when the number of arms/actions is comparable or much larger than the time horizon, and make *no* assumptions about the structure of the bandit instance. Our goal is to design algorithms that can automatically adapt to the *unknown* hardness of the problem, i.e., the number of best arms. Our setting captures many modern applications of bandit algorithms where the action space is enormous and the information about the underlying instance/structure is unavailable. We first propose an adaptive algorithm that is agnostic to the hardness level and theoretically derive its regret bound. We then prove a lower bound for our problem setting, which indicates: (1) no algorithm can be minimax optimal simultaneously over all hardness levels; and (2) our algorithm achieves a rate function that is Pareto optimal. With additional knowledge of the expected reward of the best arm, we propose another adaptive algorithm that is minimax optimal, up to polylog factors, over *all* hardness levels. Experimental results confirm our theoretical guarantees and show advantages of our algorithms over the previous state-of-the-art.

1 Introduction

Multi-armed bandit problems describe exploration-exploitation trade-offs in sequential decision making. Most existing bandit algorithms tend to provide regret guarantees when the number of available arms/actions is smaller than the time horizon. In modern applications of bandit algorithm, however, the action space is usually comparable or even much larger than the allowed time horizon so that many existing bandit algorithms cannot even complete their initial exploration phases. Consider a problem of personalized recommendations, for example. For most users, the total number of movies, or even the amount of sub-categories, far exceeds the number of times they visit a recommendation site. Similarly, the enormous amount of user-generated content on YouTube and Twitter makes it increasingly challenging to make optimal recommendations. The tension between a very large action space and a limited time horizon poses a realistic problem in which deploying algorithms that converge to an optimal solution over an asymptotically long time horizon *do not* give satisfying results. There is a need to design algorithms that can exploit the highest possible reward within a *limited* time horizon. Past work has partially addressed this challenge. The quantile regret proposed in [12] to calculate regret with respect to an satisfactory action rather than the best one. The discounted regret analyzed in [25, 24] is used to emphasize short time horizon performance. Other existing works consider the extreme case when the number of actions is indeed infinite, and tackle such problems with one of two main assumptions: (1) the discovery of a near-optimal/best arm follows some probability measure with *known* parameters [6, 30, 4, 15]; (2) the existence of a *smooth* function represents the mean-payoff over a continuous subset [1, 20, 19, 8, 23, 17]. However, in many situations, neither assumption may be realistic. We make minimal assumptions in this paper. We study the regret minimization problem over a time horizon T, which might be unknown, with respect

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

to a bandit instance with n total arms, out of which m are best/near-optimal arms. We emphasize that the allowed time horizon and the given bandit instance should be viewed as features of *one* problem and together they indicate an intrinsic hardness level. We consider the case when the number of arms n is comparable or larger than the time horizon T so that no standard algorithm provides satisfying result. Our goal is to design algorithms that could adapt to the *unknown* m and achieve optimal regret.

1.1 Contributions and paper organization

We make the following contributions. In Section 2, we formally define the regret minimization problem that represents the tension between a very large action space and a limited time horizon; and capture the hardness level in terms of the number of best arms. We provide an adaptive algorithm that is agnostic to the *unknown* number of best arms in Section 3, and theoretically derive its regret bound. In Section 4, we prove a lower bound for our problem setting that indicates that there is no algorithm that can be optimal simultaneously over all hardness levels. Our lower bound also shows that our algorithm provided in Section 3 is Pareto optimal. With additional knowledge of the expected reward of the best arm, in Section 5, we provide an algorithm that achieves the non-adaptive minimax optimal regret, up to polylog factors, without the knowledge of the number of best arms. Experiments conducted in Section 6 confirm our theoretical guarantees and show advantages of our algorithms over previous state-of-the-art. We conclude our paper in Section 7. Most of the proofs are deferred to the Appendix due to lack of space.

1.2 Related work

Time sensitivity and large action space. As bandit models are getting much more complex, usually with large or infinite action spaces, researchers have begun to pay attention to tradeoffs between regret and time horizons when deploying such models. [13] study a linear bandit problem with ultra-high dimension, and provide algorithms that, under various assumptions, can achieve good reward within short time horizon. [24] also take time horizon into account and model time preference by analyzing a discounted regret. [12] consider a quantile regret minimization problem where they define their regret with respect to expected reward ranked at $(1-\rho)$-th quantile. One could easily transfer their problem to our setting; however, their regret guarantee is sub-optimal. [18, 4] also consider the problem with m best/near-optimal arms with no other assumptions, but they focus on the pure exploration setting; [4] additionally requires the knowledge of m. Another line of research considers the extreme case when the number arms is infinite, but with some *known* regularities. [6] proposes an algorithm with a minimax optimality guarantee under the situation where the reward of each arm follows *strictly* Bernoulli distribution; [27] provides an anytime algorithm that works under the same assumption. [30] relaxes the assumption on Bernoulli reward distribution, however, some other parameters are assumed to be known in their setting.

Continuum-armed bandit. Many papers also study bandit problems with continuous action spaces, where they embed each arm x into a bounded subset $\mathcal{X} \subseteq \mathbb{R}^d$ and assume there exists a smooth function f governing the mean-payoff for each arm. This setting is firstly introduced by [1]. When the smoothness parameters are known to the learner or under various assumptions, there exists algorithms [20, 19, 8] with near-optimal regret guarantees. When the smoothness parameters are unknown, however, [23] proves a lower bound indicating no strategy can be optimal simultaneously over all smoothness classes; under extra information, they provide adaptive algorithms with near-optimal regret guarantees. Although achieving optimal regret for all settings is impossible, [17] design adaptive algorithms and prove that they are Pareto optimal. Our algorithms are mainly inspired by the ones in [17, 23]. A closely related line of work [28, 16, 5, 26] aims at minimizing simple regret in the continuum-armed bandit setting.

Adaptivity to unknown parameters. [9] argues the awareness of regularity is flawed and one should design algorithms that can *adapt* to the unknown environment. In situations where the goal is pure exploration or simple regret minimization, [18, 28, 16, 5, 26] achieve near-optimal guarantees with unknown regularity because their objectives trade-off exploitation in favor of exploration. In the case of cumulative regret minimization, however, [23] shows no strategy can be optimal simultaneously over all smoothness classes. In special situations or under extra information, [9, 10, 23] provide algorithms that adapt in different ways. [17] borrows the concept of Pareto optimality from economics and provide algorithms with rate functions that are Pareto optimal. Adaptivity is studied in statistics as well: in some cases, only additional logarithmic factors are required [22, 7]; in others, however, there exists an additional polynomial cost of adaptation [11].

2 Problem statement and notation

We consider the multi-armed bandit instance $\underline{\nu} = (\nu_1, \dots, \nu_n)$ with n probability distributions with means $\mu_i = \mathbb{E}_{X \sim \nu_i}[X] \in [0, 1]$. Let $\mu_\star = \max_{i \in [n]}\{\mu_i\}$ be the highest mean and $S_\star = \{i \in [n] : \mu_i = \mu_\star\}$ denote the subset of best arms.[1] The cardinality $|S_\star| = m$ is *unknown* to the learner. We could also generalize our setting to $S'_\star = \{i \in [n] : \mu_i \geq \mu_\star - \epsilon(T)\}$ with unknown $|S'_\star|$ (i.e., situations where there is an unknown number of near-optimal arms). Setting ϵ to be dependent on T is to avoid an additive term linear in T, e.g., $\epsilon \leq 1/\sqrt{T} \Rightarrow \epsilon T \leq \sqrt{T}$. All theoretical results and algorithms presented in this paper are applicable to this generalized setting with minor modifications. For ease of exposition, we focus on the case with multiple best arms throughout the paper. At each time step $t \in [T]$, the algorithm/learner selects an action $A_t \in [n]$ and receives an independent reward $X_t \sim \nu_{A_t}$. We assume that $X_t - \mu_{A_t}$ is $(1/2)$-sub-Gaussian conditioned on A_t.[2] We measure the success of an algorithm through the expected cumulative (pseudo) regret:

$$R_T = T \cdot \mu_\star - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{A_t}\right].$$

We use $\mathcal{R}(T, n, m)$ to denote the set of regret minimization problems with allowed time horizon T and any bandit instance $\underline{\nu}$ with n total arms and m best arms.[3] We emphasize that T is part of the problem instance. We are particularly interested in the case when n is comparable or even larger than T, which captures many modern applications where the available action space far exceeds the allowed time horizon. Although learning algorithms may not be able to pull each arm once, one should notice that the true/intrinsic hardness level of the problem could be viewed as n/m: selecting a subset uniformly at random with cardinality $\Theta(n/m)$ guarantees, with constant probability, the access to at least one best arm; but of course it is impossible to do this without knowing m. We quantify the *intrinsic* hardness level over a set of regret minimization problems $\mathcal{R}(T, n, m)$ as

$$\psi(\mathcal{R}(T, n, m)) = \inf\{\alpha \geq 0 : n/m \leq 2T^\alpha\},$$

where the constant 2 in front of T^α is added to avoid otherwise the trivial case with all best arms when the infimum is 0. $\psi(\mathcal{R}(T, n, m))$ is used here as it captures the *minimax* optimal regret over the set of regret minimization problem $\mathcal{R}(T, n, m)$, as explained later in our review of the `MOSS` algorithm and the lower bound. As smaller $\psi(\mathcal{R}(T, n, m))$ indicates easier problems, we then define the family of regret minimization problems with hardness level at most α as

$$\mathcal{H}_T(\alpha) = \{\cup \mathcal{R}(T, n, m) : \psi(\mathcal{R}(T, n, m)) \leq \alpha\},$$

with $\alpha \in [0, 1]$. Although T is necessary to define a regret minimization problem, we actually encode the hardness level into a single parameter α, which captures the *tension* between the complexity of bandit instance at hand and the allowed time horizon T: problems with different time horizons but the same α are equally difficult in terms of the achievable minimax regret (the exponent of T). We thus mainly study problems with T large enough so that we could mainly focus on the polynomial terms of T. We are interested in designing algorithms with *minimax* guarantees over $\mathcal{H}_T(\alpha)$, but *without* the knowledge of α.

`MOSS` and upper bound. In the classical setting, `MOSS` , designed by [2] and further generalized to the sub-Gaussian case [21] and improved in terms of constant factors [14], achieves the minimax optimal regret. In this paper, we will use `MOSS` as a subroutine with regret upper bound $O(\sqrt{nT})$ when $T \geq n$. For any problem in $\mathcal{H}_T(\alpha)$ with *known* α, one could run `MOSS` on a subset selected uniformly at random with cardinality $\widetilde{O}(T^\alpha)$ and achieve regret $\widetilde{O}(T^{(1+\alpha)/2})$.

[1] Throughout the paper, we denote by $[K]$ the set $\{1, \dots, K\}$ for any positive integer K.

[2] We say a random variable X is σ-sub-Gaussian if $\mathbb{E}[\exp(\lambda X)] \leq \exp(\sigma^2\lambda^2/2)$ for all $\lambda \in \mathbb{R}$.

[3] Our setting could be generalized to the case with infinite arms: one can consider embedding arms into an arm space $\mathcal{X}$ and let p be the probability that an arm sampled uniformly at random is (near-) optimal. $1/p$ will then serve a similar role as n/m does in the original definition.

Lower bound. The lower bound $\Omega(\sqrt{nT})$ in the classical setting does not work for our setting as its proof heavily relies on the existence of single best arm [21]. However, for problems in $\mathcal{H}_T(\alpha)$, we do have a matching lower bound $\Omega(T^{(1+\alpha)/2})$ as one could always apply the standard lower bound on an bandit instance with $n = \lfloor T^\alpha \rfloor$ and $m = 1$. For general value of m, a lower bound of the order $\Omega(\sqrt{T(n-m)/m}) = \Omega(T^{(1+\alpha)/2})$ for the m-best arms case could be obtained following similar analysis in Chapter 15 of [21].

Although $\log T$ may appear in our bounds, throughout the paper, we focus on problems with $T \geq 2$ as otherwise the bound is trivial.

3 An adaptive algorithm

Algorithm 1 takes time horizon T and a user-specified $\beta \in [1/2, 1]$ as input, and it is mainly inspired by [17]. Algorithm 1 operates in iterations with geometrically-increasing length $\Delta T_i = 2^{p+i}$ with $p = \lceil \log_2 T^\beta \rceil$. At each iteration i, it restarts `MOSS` on a set S_i consisting of $K_i = 2^{p+2-i}$ real arms selected uniformly at random *plus* a set of "virtual" *mixture-arms* (one from each of the $1 \leq j < i$ previous iterations, none if $i = 1$). The mixture-arms are constructed as follows. After each iteration i, let $\widehat{p}_i$ denote the vector of empirical sampling frequencies of the arms in that iteration (i.e., the k-th element of $\widehat{p}_i$ is the number of times arm k, including all previously constructed mixture-arms, was sampled in iteration i divided by the total number of samples ΔT_i). The mixture-arm for iteration i is the $\widehat{p}_i$-mixture of the arms, denoted by $\widetilde{\nu}_i$. When `MOSS` samples from $\widetilde{\nu}_i$ it first draws $i_t \sim \widehat{p}_i$, then draws a sample from the corresponding arm ν_{i_t} (or $\widetilde{\nu}_{i_t}$). The mixture-arms provide a convenient summary of the information gained in the previous iterations, which is key to our theoretical analysis. Although our algorithm is working on fewer regular arms in later iterations, information summarized in mixture-arms is good enough to provide guarantees. We name our algorithm `MOSS++` as it restarts `MOSS` at each iteration with past information summarized in mixture-arms. We provide an anytime version of Algorithm 1 in Appendix A.2 via the standard doubling trick.

Algorithm 1: `MOSS++`

Input: Time horizon T and user-specified parameter $\beta \in [1/2, 1]$.
1: **Set:** $p = \lceil \log_2 T^\beta \rceil$, $K_i = 2^{p+2-i}$ and $\Delta T_i = \min\{2^{p+i}, T\}$.
2: **for** $i = 1, \ldots, p$ **do**
3: Run `MOSS` on a subset of arms S_i for ΔT_i rounds. S_i contains K_i real arms selected uniformly at random *and* the set of virtual mixture-arms from previous iterations, i.e., $\{\widetilde{\nu}_j\}_{j<i}$.
4: Construct a virtual mixture-arm $\widetilde{\nu}_i$ based on empirical sampling frequencies of `MOSS` above.
5: **end for**

3.1 Analysis and discussion

We use $\mu_S = \max_{\nu \in S}\{\mathbb{E}_{X\sim\nu}[X]\}$ to denote the highest expected reward over a set of distributions/arms S. For any algorithm that only works on S, we can decompose the regret into approximation error and learning error:

$$R_T = \underbrace{T \cdot (\mu_\star - \mu_S)}_{\text{approximation error due to the selection of } S} + \underbrace{T \cdot \mu_S - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{A_t}\right]}_{\text{learning error due to the sampling rule } \{A_t\}_{t=1}^T}. \tag{1}$$

This type of regret decomposition was previously used in [20, 3, 17] to deal with the continuum-armed bandit problem. We consider here a probabilistic version, with randomness in the selection of S, for the classical setting.

The main idea behind providing guarantees for `MOSS++` is to decompose its regret at each iteration, using Eq. (1), and then bound the expected approximation error and learning error separately. The expected learning error at each iteration could always be controlled as $\widetilde{O}(T^\beta)$ thanks to regret guarantees for `MOSS` and specifically chosen parameters p, K_i, ΔT_i. Let $i_\star$ be the largest integer such that $K_i \geq 2T^\alpha \log \sqrt{T}$ still holds. The expected approximation error in iteration $i \leq i_\star$ could be upper bounded by $\sqrt{T}$ following an analysis on hypergeometric distribution. As a result, the expected regret in iteration $i \leq i_\star$ is $\widetilde{O}(T^{\beta})$. Since the mixture-arm $\widetilde{\nu}_{i_\star}$ is included in all following iterations, we could further bound the expected approximation error in iteration $i > i_\star$ by $\widetilde{O}(T^{1+\alpha-\beta})$ after a careful analysis on $\Delta T_i / \Delta T_{i_\star}$. This intuition is formally stated and proved in Theorem 1.

Theorem 1. *Run* `MOSS++` *with time horizon T and an user-specified parameter $\beta \in [1/2, 1]$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq C \left(\log_2 T\right)^{5/2} \cdot T^{\min\{\max\{\beta, 1+\alpha-\beta\}, 1\}},$$

where C is a universal constant.

Remark 1. *We primarily focus on the polynomial terms in T when deriving the bound, but put no effort in optimizing the polylog term. The $5/2$ exponent of $\log_2 T$ might be tightened as well.*

The theoretical guarantee is closely related to the user-specified parameter β: when $\beta > \alpha$, we suffer a multiplicative cost of adaptation $\widetilde{O}(T^{|(2\beta-\alpha-1)/2|})$, with $\beta = (1+\alpha)/2$ hitting the sweet spot, comparing to non-adaptive minimax regret; when $\beta \leq \alpha$, there is essentially no guarantees. One may hope to improve this result. However, our analysis in Section 4 indicates: (1) achieving minimax optimal regret for all settings simultaneously is *impossible*; and (2) the rate function achieved by `MOSS++` is already *Pareto optimal*.

4 Lower bound and Pareto optimality

4.1 Lower bound

In this section, we show that designing algorithms with the non-adaptive minimax optimal guarantee over all values of α is impossible. We first state the result in the following general theorem.

Theorem 2. *For any $0 \leq \alpha' < \alpha \leq 1$, assume $T^{\alpha} \leq B$ and $\lfloor T^{\alpha} \rfloor - 1 \geq \max\{T^{\alpha}/4, 2\}$. If an algorithm is such that $\sup_{\omega \in \mathcal{H}_T(\alpha')} R_T \leq B$, then the regret of this algorithm is lower bounded on $\mathcal{H}_T(\alpha)$:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \geq 2^{-10} T^{1+\alpha} B^{-1}. \tag{2}$$

To give an interpretation of Theorem 2, we consider any algorithm/policy π together with regret minimization problems $\mathcal{H}_T(\alpha')$ and $\mathcal{H}_T(\alpha)$ satisfying corresponding requirements. On one hand, if algorithm π achieves a regret that is order-wise larger than $\widetilde{O}(T^{(1+\alpha')/2})$ over $\mathcal{H}_T(\alpha')$, it is already not minimax optimal for $\mathcal{H}_T(\alpha')$. Now suppose π achieves a near-optimal regret, i.e., $\widetilde{O}(T^{(1+\alpha')/2})$, over $\mathcal{H}_T(\alpha')$; then, according to Eq. (2), π must incur a regret of order at least $\widetilde{\Omega}(T^{1/2+\alpha-\alpha'/2})$ on one problem in $\mathcal{H}_T(\alpha')$. This, on the other hand, makes algorithm π strictly sub-optimal over $\mathcal{H}_T(\alpha)$.

4.2 Pareto optimality

We capture the performance of any algorithm by its dependence on polynomial terms of T in the asymptotic sense. Note that the hardness level of a problem is encoded in α.

Definition 1. *Let $\theta : [0,1] \to [0,1]$ denote a non-decreasing function. An algorithm achieves the rate function θ if*

$$\forall \epsilon > 0, \forall \alpha \in [0,1], \quad \limsup_{T \to \infty} \frac{\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T}{T^{\theta(\alpha)+\epsilon}} < +\infty.$$

Recall that a function θ' is strictly smaller than another function θ in pointwise order if $\theta'(\alpha) \leq \theta(\alpha)$ for all α and $\theta'(\alpha_0) < \theta(\alpha_0)$ for at least one value of α_0. As there may not always exist a pointwise ordering over rate functions, following [17], we consider the notion of Pareto optimality over rate functions achieved by some algorithms.

Definition 2. *A rate function θ is Pareto optimal if it is achieved by an algorithm, and there is no other algorithm achieving a strictly smaller rate function θ' in pointwise order. An algorithm is Pareto optimal if it achieves a Pareto optimal rate function.*

Combining the results in Theorem 1 and Theorem 2 with above definitions, we could further obtain the following result in Theorem 3.

Theorem 3. *The rate function achieved by* MOSS++ *with any $\beta \in [1/2, 1]$, i.e.,*

$$\theta_\beta : \alpha \mapsto \min\{\max\{\beta, 1 + \alpha - \beta\}, 1\}, \tag{3}$$

is Pareto optimal.

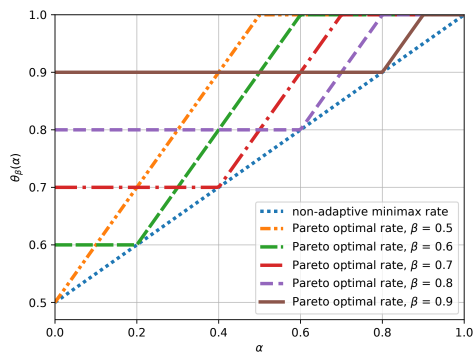


Figure 1: Pareto optimal rates

Fig. 1 provides an illustration of the rate functions achieved by MOSS++ with different β as input, as well as the non-adaptive minimax optimal rate.

Remark 2. *One should notice that the naive algorithm running* MOSS *on a subset selected uniformly at random with cardinality $\widetilde{O}(T^{\beta'})$ is not Pareto optimal, since running* MOSS++ *with $\beta = (1+\beta')/2$ leads to a strictly smaller rate function. The algorithm provided in [12], if transferred to our setting and allowing time horizon dependent quantile, is not Pareto optimal as well since it corresponds to the rate function $\theta(\alpha) = \max\{2.89\,\alpha, 0.674\}$.*

5 Learning with extra information

Although previous Section 4 gives negative results on designing algorithms that could optimally adapt to all settings, one could actually design such an algorithm *with* extra information. In this section, we provide an algorithm that takes the expected reward of the best arm $\mu_\star$ (or an estimated one with error up to $1/\sqrt{T}$) as extra information, and achieves near minimax optimal regret over all settings simultaneously. Our algorithm is mainly inspired by [23].

5.1 Algorithm

We name our Algorithm 3 `Parallel` as it maintains $\lceil \log T \rceil$ instances of subroutine, i.e., Algorithm 2, in parallel. Each subroutine $\texttt{SR}_i$ is initialized with time horizon T and hardness level $\alpha_i = i/\lceil \log T \rceil$. We use $T_{i,t}$ to denote the number of samples allocated to $\texttt{SR}_i$ up to time t, and represent its empirical regret at time t as $\widehat{R}_{i,t} = T_{i,t} \cdot \mu_\star - \sum_{t=1}^{T_{i,t}} X_{i,t}$ with $X_{i,t} \sim \nu_{A_{i,t}}$ being the t-th empirical reward obtained *by* $\texttt{SR}_i$ and $A_{i,t}$ being the index of the t-th arm pulled *by* $\texttt{SR}_i$.

Algorithm 2: MOSS Subroutine (SR)

Input: Time horizon T and hardness level α.

1: Select a subset of arms S_α uniformly at random with $|S_\alpha| = \lceil 2T^\alpha \log \sqrt{T} \rceil$ and run MOSS on S_α.

`Parallel` operates in iterations of length $\lceil \sqrt{T} \rceil$. At the beginning of each iteration, i.e., at time $t = i \cdot \lceil \sqrt{T} \rceil$ for $i \in \{0\} \cup [\lceil \sqrt{T} \rceil - 1]$, `Parallel` first selects the subroutine with the lowest (breaking ties arbitrarily) empirical regret so far, i.e., $k = \arg\min_{i \in [\lceil \log T \rceil]} \widehat{R}_{i,t}$; it then *resumes* the learning process of $\texttt{SR}_k$, from where it halted, for another $\lceil \sqrt{T} \rceil$ more pulls. All the information is updated at the end of that iteration. An anytime version of Algorithm 3 is provided in Appendix C.3.

5.2 Analysis

As `Parallel` discretizes the hardness parameter over a grid with interval $1/\lceil \log T \rceil$, we first show that running the best subroutine alone leads to regret $\widetilde{O}(T^{(1+\alpha)/2})$.

Algorithm 3: `Parallel`

Input: Time horizon T and the optimal reward $\mu_\star$.

set: $p = \lceil \log T \rceil$, $\Delta = \lceil \sqrt{T} \rceil$ and $t = 0$.
for $i = 1, \ldots, p$ **do**
  Set $\alpha_i = i/p$, initialize $\texttt{SR}_i$ with α_i, T; set $T_{i,t} = 0$, and $\widehat{R}_{i,t} = 0$.
end for
for $i = 1, \ldots, \Delta - 1$ **do**
  Select $k = \arg\min_{i \in [p]} \widehat{R}_{i,t}$ and run $\texttt{SR}_k$ for Δ rounds.
  Update $T_{k,t} = T_{k,t} + \Delta$, $\widehat{R}_{k,t} = T_{k,t} \cdot \mu_\star - \sum_{t=1}^{T_{k,t}} X_{k,t}$, $t = t + \Delta$.
end for

Lemma 1. *Suppose α is the true hardness parameter and $\alpha_i - 1/\lceil \log T \rceil < \alpha \leq \alpha_i$, run Algorithm 2 with time horizon T and α_i leads to the following regret bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq C \, \log T \cdot T^{(1+\alpha)/2},$$

where C is a universal constant.

Since `Parallel` always allocates new samples to the subroutine with the lowest empirical regret so far, we know that the regret of every subroutine should be roughly of the same order at time T. In particular, all subroutines should achieve regret $\widetilde{O}(T^{(1+\alpha)/2})$, as the best subroutine does. `Parallel` then achieves the non-adaptive minimax optimal regret, up to polylog factors, *without* knowing the true hardness level α.

Theorem 4. *For any $\alpha \in [0, 1]$ unknown to the learner, run* `Parallel` *with time horizon T and optimal expected reward $\mu_\star$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq C \, (\log T)^2 \, T^{(1+\alpha)/2},$$

where C is a universal constant.

6 Experiments

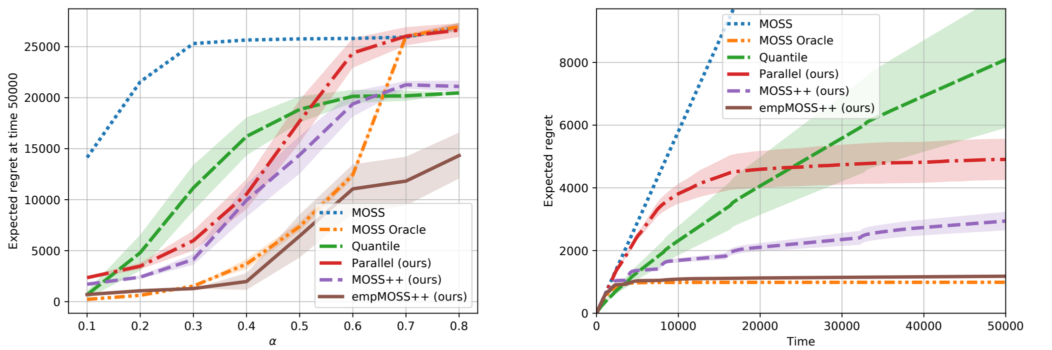
We conduct three experiments to compare our algorithms with baselines. In Section 6.1, we compare the performance of each algorithm on problems with varying hardness levels. We examine how the regret curve of each algorithm increases on synthetic and real-world datasets in Section 6.2 and Section 6.3, respectively.

We first introduce the nomenclature of the algorithms. We use `MOSS` to denote the standard `MOSS` algorithm; and `MOSS Oracle` to denote Algorithm 2 with *known* α. `Quantile` represents the algorithm (QRM2) proposed by [12] to minimize the regret with respect to the $(1-\rho)$-th quantile of means among arms, without the knowledge of ρ. One could easily transfer `Quantile` to our settings with top-ρ fraction of arms treated as best arms. As suggested in [12], we reuse the statistics obtained in previous iterations of `Quantile` to improve its sample efficiency. We use `MOSS++` to represent the vanilla version of Algorithm 1; and use `empMOSS++` to represent an empirical version such that: (1) `empMOSS++` reuse statistics obtained in previous round, as did in `Quantile`; and (2) instead of selecting K_i real arms uniformly at random at the i-th iteration, `empMOSS++` selects K_i arms with the highest empirical mean for $i > 1$. We choose $\beta = 0.5$ for `MOSS++` and `empMOSS++` in all experiments.[4] All results are averaged over 100 experiments. Shaded area represents 0.5 standard deviation for each algorithm.

6.1 Adaptivity to hardness level

We compare our algorithms with baselines on regret minimization problems with different hardness levels. For this experiment, we generate best arms with expected reward 0.9 and sub-optimal arms

[4] Increasing β generally leads to worse performance on problems with small α but better performance on problems with large α.



(a) (b)

Figure 2: (a) Comparison with varying hardness levels (b) Regret curve comparison with $\alpha = 0.25$.

with expected reward evenly distributed among $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. All arms follow Bernoulli distribution. We set the time horizon to $T = 50000$ and consider the total number of arms $n = 20000$. We vary α from 0.1 to 0.8 (with interval 0.1) to control the number of best arms $m = \lceil n/2T^{\alpha} \rceil$ and thus the hardness level. In Fig. 2(a), the regret of any algorithm gets larger as α increases, which is expected. `MOSS` does not provide satisfying performance due to the large action space and the relatively small time horizon. Although implemented in an anytime fashion, `Quantile` could be roughly viewed as an algorithm that runs `MOSS` on a subset selected uniformly at random with cardinality $T^{0.347}$. `Quantile` displays good performance when $\alpha = 0.1$, but suffers regret much worse than `MOSS++` and `empMOSS++` when α gets larger. Note that the regret curve of `Quantile` gets flattened at 20000 is expected: it simply learns the best sub-optimal arm and suffers a regret $50000 \times (0.9 - 0.5)$. Although `Parallel` enjoys near minimax optimal regret, the regret it suffers from is the summation of 11 subroutines, which hurts its empirical performance. `empMOSS++` achieves performance comparable to `MOSS Oracle` when α is small, and achieve the best empirical performance when $\alpha \geq 0.3$. When $\alpha \geq 0.7$, `MOSS Oracle` needs to explore most/all of the arms to statistically guarantee the finding of at least one best arm, which hurts its empirical performance.

6.2 Regret curve comparison

We compare how the regret curve of each algorithm increases in Fig. 2(b). We consider the same regret minimization configurations as described in Section 6.1 with $\alpha = 0.25$. `empMOSS++`, `MOSS++` and `Parallel` all outperform `Quantile` with `empMOSS++` achieving the performance closest to `MOSS Oracle`. `MOSS Oracle`, `Parallel` and `empMOSS++` have flattened their regret curve indicating they could confidently recommend the best arm. The regret curves of `MOSS++` and `Quantile` do not flat as the random-sampling component in each of their iterations encourage them to explore new arms. Comparing to `MOSS++`, `Quantile` keeps increasing its regret at a much faster rate and with a much larger variance, which empirically confirms the sub-optimality of their regret guarantees.

6.3 Real-world dataset

We also compare all algorithms in a realistic setting of recommending funny captions to website visitors. We use a real-world dataset from the *New Yorker Magazine* Cartoon Caption Contest[5]. The dataset of 1-3 star caption ratings/rewards for Contest 652 consists of $n = 10025$ captions[6]. We use the ratings to compute Bernoulli reward distributions for each caption as follows. The mean of each caption/arm i is calculated as the percentage p_i of its ratings that were funny or somewhat funny (i.e., 2 or 3 stars). We normalize each p_i with the best one and then threshold each: if $p_i \geq 0.8$, then put $p_i = 1$; otherwise leave p_i unaltered. This produces a set of $m = 54$ best arms with rewards 1 and all

[5] `https://www.newyorker.com/cartoons/contest`.

[6] Available online at `https://nextml.github.io/caption-contest-data`.

other 9971 arms with rewards among $[0, 0.8]$. We set $T = 10^5$ and this results in a hardness level around $\alpha \approx 0.43$.

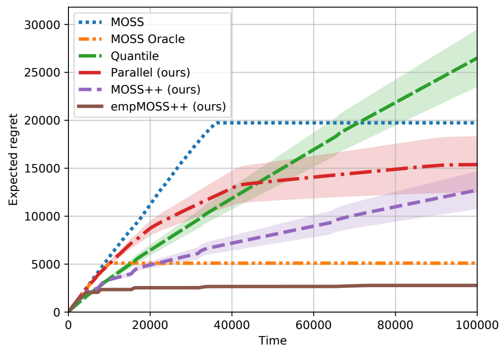


Figure 3: Regret comparison with real-world dataset

Using these Bernoulli reward models, we compare the performance of each algorithm, as shown in Fig. 3. `MOSS`, `MOSS Oracle`, `Parallel` and `empMOSS++` have flattened their regret curve indicating they could confidently recommend the funny captions (i.e., best arms). Although `MOSS` could eventually identify a best arm in this problem, its cumulative regret is more than 7x of the regret achieved by `empMOSS++` due to its initial exploration phase. The performance of `Quantile` is even worse, and its cumulative regret is more than 9x of the regret achieved by `empMOSS++`. One surprising phenomenon is that `empMOSS++` outperforms `MOSS Oracle` in this realistic setting. Our hypothesis is that `MOSS Oracle` is a little bit conservative and selects an initial set with cardinality too large. This experiment demonstrates the effectiveness of `empMOSS++` and `MOSS++` in modern applications of bandit algorithm with large action space and limited time horizon.

7 Conclusion

We study a regret minimization problem with large action space but limited time horizon, which captures many modern applications of bandit algorithms. Depending on the number of best/near-optimal arms, we encode the hardness level, in terms of minimax regret achievable, of the given regret minimization problem into a single parameter α, and we design algorithms that could adapt to this *unknown* hardness level. Our first algorithm `MOSS++` takes a user-specified parameter β as input and provides guarantees as long as $\alpha < \beta$; our lower bound further indicates the rate function achieved by `MOSS++` is Pareto optimal. Although no algorithm can achieve near minimax optimal regret over all α simultaneously, as demonstrated by our lower bound, we overcome this limitation with an (often) easily-obtained extra information and propose `Parallel` that is near-optimal for all settings. Inspired by `MOSS++`, We also propose `empMOSS++` with excellent empirical performance. Experiments on both synthetic and real-world datasets demonstrate the efficiency of our algorithms over the previous state-of-the-art.

Broader Impact

This paper provides efficient algorithms that work well in modern applications of bandit algorithms with large action space but limited time horizon. We make *minimal* assumption about the setting, and our algorithms can automatically adapt to *unknown* hardness levels. Worst-case regret guarantees are provided for our algorithms; we also show `MOSS++` is Pareto optimal and `Parallel` is minimax optimal, up to polylog factors. `empMOSS++` is provided as a practical version of `MOSS++` with excellent empirical performance. Our algorithms are particularly useful in areas such as e-commence and movie/content recommendation, where the action space is enormous but possibly contains multiple best/satisfactory actions. If deployed, our algorithms could automatically adapt to the hardness level of the recommendation task and benefit both service-providers and customers through efficiently delivering satisfactory content. One possible negative outcome is that items recommended to a *specific* user/customer might only come from a subset of the action space. However, this is *unavoidable* when the number of items/actions exceeds the allowed time horizon. In fact, one should notice that all items/actions will be selected with essentially the same probability, thanks to the incorporation of random selection processes in our algorithms. Our algorithms will *not* leverage/create biases due to the same reason. Overall, we believe this paper's contribution will have a net positive impact.

Acknowledgments and Disclosure of Funding

The authors would like to thank anonymous reviewers for their comments and suggestions. This work was partially supported by NSF grant no. 1934612.

References

[1] Rajeev Agrawal. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951, 1995.

[2] Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, pages 217–226, 2009.

[3] Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *International Conference on Computational Learning Theory*, pages 454–468. Springer, 2007.

[4] Maryam Aziz, Jesse Anderton, Emilie Kaufmann, and Javed Aslam. Pure exploration in infinitely-armed bandit models with fixed-confidence. *arXiv preprint arXiv:1803.04665*, 2018.

[5] Peter L Bartlett, Victor Gabillon, and Michal Valko. A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. *arXiv preprint arXiv:1810.00997*, 2018.

[6] Donald A Berry, Robert W Chen, Alan Zame, David C Heath, and Larry A Shepp. Bandit problems with infinitely many arms. *The Annals of Statistics*, pages 2103–2116, 1997.

[7] Lucien Birgé and Pascal Massart. From model selection to adaptive estimation. In *Festschrift for lucien le cam*, pages 55–87. Springer, 1997.

[8] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12(May):1655–1695, 2011.

[9] Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu. Lipschitz bandits without the lipschitz constant. In *International Conference on Algorithmic Learning Theory*, pages 144–158. Springer, 2011.

[10] Adam D Bull et al. Adaptive-treed bandits. *Bernoulli*, 21(4):2289–2307, 2015.

[11] T Tony Cai, Mark G Low, et al. On adaptive estimation of linear functionals. *The Annals of Statistics*, 33(5):2311–2343, 2005.

[12] Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Quantile-regret minimisation in infinitely many-armed bandits. In *UAI*, pages 425–434, 2018.

[13] Yash Deshpande and Andrea Montanari. Linear bandits in high dimension and recommendation systems. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1750–1754. IEEE, 2012.

[14] Aurélien Garivier, Hédi Hadiji, Pierre Menard, and Gilles Stoltz. Kl-ucb-switch: optimal regret bounds for stochastic bandits from both a distribution-dependent and a distribution-free viewpoints. *arXiv preprint arXiv:1805.05071*, 2018.

[15] Ganesh Ghalme, Swapnil Dhamal, Shweta Jain, Sujit Gujar, and Y Narahari. Ballooning multi-armed bandits. *arXiv preprint arXiv:2001.10055*, 2020.

[16] Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. In *Advances in Neural Information Processing Systems*, pages 667–675, 2015.

[17] Hédi Hadiji. Polynomial cost of adaptation for x-armed bandits. In *Advances in Neural Information Processing Systems*, pages 1027–1036, 2019.

[18] Julian Katz-Samuels and Kevin Jamieson. The true sample complexity of identifying good arms. *arXiv preprint arXiv:1906.06594*, 2019.

[19] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690, 2008.

[20] Robert D Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pages 697–704, 2005.

[21] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[22] OV Lepskii. On a problem of adaptive estimation in gaussian white noise. *Theory of Probability & Its Applications*, 35(3):454–466, 1991.

[23] Andrea Locatelli and Alexandra Carpentier. Adaptivity to smoothness in x-armed bandits. In *Conference on Learning Theory*, pages 1463–1492, 2018.

[24] Daniel Russo and Benjamin Van Roy. Satisficing in time-sensitive bandit learning. *arXiv preprint arXiv:1803.02855*, 2018.

[25] Ilya O Ryzhov, Warren B Powell, and Peter I Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195, 2012.

[26] Xuedong Shang, Emilie Kaufmann, and Michal Valko. General parallel optimization a without metric. In *Algorithmic Learning Theory*, pages 762–788, 2019.

[27] Olivier Teytaud, Sylvain Gelly, and Michèle Sebag. Anytime many-armed bandits. In *CAP07*, Grenoble, France, 2007.

[28] Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *International Conference on Machine Learning*, pages 19–27, 2013.

[29] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[30] Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *Advances in Neural Information Processing Systems*, pages 1729–1736, 2009.

A Omitted proofs for Section 3

We introduce the notation $R_{T|\mathcal{F}} = T \cdot \mu_\star - \mathbb{E}[\sum_{t=1}^{T} X_t | \mathcal{F}]$ for any σ-algebra. One should also notice that $\mathbb{E}[R_{T|\mathcal{F}}] = R_T$.

A.1 Proof of Theorem 1

Lemma 2. *For an instance with n total arms and m best arms, and for a subset S selected uniformly at random with cardinality k, the probability that none of the best arms are selected in S is upper bounded by* $\exp(-mk/n)$.

Proof. Consider selecting k items out of n items without replacement; and suppose there are m target items. Let $\mathcal{E}$ denote the event where none of the target items are selected, we then have

$$\begin{aligned}
\mathbb{P}(\mathcal{E}) &= \frac{\binom{n-m}{k}}{\binom{n}{k}} = \frac{\frac{(n-m)!}{(n-m-k)!k!}}{\frac{n!}{(n-k)!k!}} \\
&= \frac{(n-m)!}{(n-m-k)!} \cdot \frac{(n-k)!}{n!} \\
&= \prod_{i=0}^{k-1} \frac{n-m-i}{n-i} \\
&\leq \left(\frac{n-m}{n}\right)^k && (4)\\
&\leq \exp\left(-\frac{m}{n} \cdot k\right), && (5)
\end{aligned}$$

where Eq. (4) comes from the fact that $\frac{n-m-i}{n-i}$ is decreasing in i; and Eq. (5) comes from the fact that $1 - x \leq \exp(-x)$ for all $x \in \mathbb{R}$.

Selecting arms with replacement gives the same guarantee (which directly goes to Eq. (4)), and can be used in corner cases when $k > n$. □

Theorem 1. *Run* `MOSS++` *with time horizon T and an user-specified parameter $\beta \in [1/2, 1]$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq C (\log_2 T)^{5/2} \cdot T^{\min\{\max\{\beta, 1+\alpha-\beta\}, 1\}},$$

where C is a universal constant.

Proof. Let $T_i = \sum_{j=1}^{i} \Delta T_j$. We first notice that Algorithm 1 is a valid algorithm in the sense that it selects an arm A_t for any $t \in [T]$, i.e., it does not terminate before time T: the argument is clearly true if there exists $i \in [p]$ such that $\Delta T_i = T$; otherwise, we can show that

$$T_p = \sum_{i=1}^{p} \Delta T_i = 2(2^{2p} - 1) \geq 2^{2p} \geq T,$$

for all $\beta \in [1/2, 1]$.

We will only consider the case when $\alpha < \beta$ in the following since otherwise Theorem 1 trivially holds due to $T^{1+\alpha-\beta} \geq T$.

Let $\mathcal{F}_{i-1}$ represents the information available at the beginning of iteration i, including the random selection process of generating S_i. We denote $R_{\Delta T_i} = \Delta T_i \cdot \mu_\star - \mathbb{E}[\sum_{t=T_{i-1}+1}^{T_i} X_t]$ the expected cumulative regret at iteration i. Recall that we use $R_{\Delta T_i | \mathcal{F}_{i-1}}$ to represent the expected regret conditioned on $\mathcal{F}_{i-1}$ and have $\mathbb{E}[R_{\Delta T_i | \mathcal{F}_{i-1}}] = R_{\Delta T_i}$; we also use $\mu_{S_i} = \max_{\nu \in S_i}\{\mathbb{E}_{X\sim\nu}[X]\}$. Applying Eq. (1) on $R_{\Delta T_i | \mathcal{F}_{i-1}}$ leads to

$$R_{\Delta T_i | \mathcal{F}_{i-1}} = \Delta T_i \cdot (\mu_\star - \mu_{S_i}) + \left(\Delta T_i \cdot \mu_{S_i} - \mathbb{E}\left[\sum_{t=T_{i-1}+1}^{T_i} \mu_{A_t} \,\middle|\, \mathcal{F}_{i-1}\right]\right), \tag{6}$$

where, by a slightly abuse of notations, we use μ_{A_t} to refer to the mean of arm $A_t \in S_i$, which could also be the mean of a virtual arm constructed in one of the previous iterations.

Note that we will be working on a bandit instance with $(\sqrt{2}/2)$-sub-Gaussian noise when the extra virtual arms are included: let X be a sample from a virtual mixture-arm $\widetilde{\nu}_i$, which is realized by first sampling an index i_t (of a real arm) from the empirical measure, and then draw X from the real arm ν_{i_t}. We have $X - \mathbb{E}[X] = (X - \mu_{i_t}) + (\mu_{i_t} - \mathbb{E}[X])$ and thus for any $\lambda \in \mathbb{R}$,

$$
\begin{aligned}
\mathbb{E}\left[\exp\left(\lambda\left(X - \mathbb{E}[X]\right)\right)\right] &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(\lambda\left(X - \mathbb{E}[X]\right)\right) \middle| i_t\right]\right] \\
&= \mathbb{E}\left[\exp\left(\lambda(\mu_{i_t} - \mathbb{E}[X])\right) \mathbb{E}\left[\exp\left(\lambda\left(X - \mu_{i_t}\right)\right) \middle| i_t\right]\right] \\
&\leq \exp\left(\frac{\lambda^2/4}{2}\right) \mathbb{E}\left[\exp\left(\lambda(\mu_{i_t} - \mathbb{E}[X])\right)\right] \\
&\leq \exp\left(\frac{\lambda^2/4}{2} + \frac{\lambda^2/4}{2}\right) \qquad (7) \\
&= \exp\left(\frac{\lambda^2/2}{2}\right)
\end{aligned}
$$

where Eq. (7) comes from the fact that $\mu_{i_t} \in [0,1]$ and $\mathbb{E}[\mu_{i_t}] = \mathbb{E}[X]$. In the following, we'll directly plug in the regret bound of MOSS for the 1-sub-Gaussian case.

We first consider the learning error for any iteration $i \in [p]$. Although μ_{S_i} is random, it is fixed at time $T_{i-1} + 1$ [17]. Since MOSS restarts at each iteration, conditioning on the information available at the beginning of the i-th iteration, i.e., $\mathcal{F}_{i-1}$, and apply the regret bound for MOSS , we have:

$$
\begin{aligned}
\Delta T_i \cdot \mu_{S_i} - \mathbb{E}\left[\sum_{t=T_{i-1}+1}^{T_i} \mu_{A_t} \middle| \mathcal{F}_{i-1}\right] &\leq 39\sqrt{|S_i|\Delta T_i} + |S_i| \qquad (8) \\
&= 39\sqrt{(K_i + i - 1)\Delta T_i} + (K_i + i - 1) \\
&\leq 39\sqrt{K_i \Delta T_i + (p-1)\Delta T_i} + (K_i + p - 1) \qquad (9) \\
&\leq 39\sqrt{2^{2p+2} + (p-1)T} + 2^{p+1} + (p-1) \qquad (10) \\
&\leq 39\sqrt{16T^{2\beta} + \log_2(T^\beta)\, T} + 4T^\beta + \log_2 T^\beta \qquad (11) \\
&\leq 166\,(\log_2 T)^{1/2} \cdot T^\beta, \qquad (12)
\end{aligned}
$$

where Eq. (8) comes from the guarantee of MOSS [21]; Eq. (9) comes from $i \leq p$; Eq. (10) comes from the definition of K_i and ΔT_i; Eq. (11) comes from the fact that $p = \lceil \log_2 T^\beta \rceil \leq \log_2 T^\beta + 1$; Eq. (12) comes from some trivial boundings on the constant.[7]

Taking expectation over all randomness on Eq. (6), we obtain

$$
R_{\Delta T_i} \leq \Delta T_i \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_i})\right] + 166\,(\log_2 T)^{1/2} \cdot T^\beta. \qquad (13)
$$

Now, we only need to consider the first term, i.e., the expected approximation error over the i-th iteration. Let $\mathcal{E}_i$ denote the event that none of the best arms, among regular arms, is selected in S_i, according to Lemma 2, we further have

$$
\begin{aligned}
\Delta T_i \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_i})\right] &\leq \Delta T_i \cdot (0 \cdot \mathbb{P}(\neg\mathcal{E}_i) + 1 \cdot \mathbb{P}(\mathcal{E}_i)) \qquad (14) \\
&\leq \Delta T_i \cdot \exp(-K_i/(2T^\alpha)), \qquad (15)
\end{aligned}
$$

where we use the fact the $\mu_i \in [0,1]$ in Eq. (14); and directly plug $n/m \leq 2T^\alpha$ into Eq. (5) to get Eq. (15).

Let $i_\star \in [p]$ be the largest integer, if exists, such that $K_{i_\star} \geq 2T^\alpha \log \sqrt{T}$, we then have that, for any $i \leq i_\star$,

$$
\Delta T_i \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_i})\right] \leq \Delta T_i/\sqrt{T} \leq T/\sqrt{T} \leq \sqrt{T}. \qquad (16)
$$

[7] One can remove the $(\log_2 T)^{1/2}$ term in many cases, e.g., when $\beta > 1/2$ and T is large enough (with respect to β). However, we mainly focus on the polynomial terms here.

Note that this choice of $i_\star$ indicates $T^\alpha \log T \leq K_{i_\star} < 2T^\alpha \log T$.

If we have $K_1 < 2T^\alpha \log \sqrt{T}$, we then set $i_\star = 1$. Notice that $K_1 = 2^{p+1} = 2^{\lceil \log_2 T^\beta \rceil + 1} \geq 2T^\beta > 2T^\alpha$, we then have

$$\Delta T_1 \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_1})\right] \leq \Delta T_1 \exp(-1) \leq 2^{p+1} \exp(-1) < 2T^\beta. \tag{17}$$

Combining Eq. (13) with Eq. (16) or Eq. (17), we have for any $i \leq i_\star$, and in particular for $i = i_\star$,

$$\begin{aligned} R_{\Delta T_i} &\leq \max\{\sqrt{T}, 2T^\beta\} + 166 \,(\log_2 T)^{1/2} \cdot T^\beta \\ &\leq 168 \,(\log_2 T)^{1/2} \cdot T^\beta. \end{aligned} \tag{18}$$

In the case when $i_\star = p$ or when $\Delta T_{i_\star} = \min\{2^{p+i}, T\} = T$, we know that MOSS++ will in fact stop at a time step no larger than $T_{i_\star}$ (since the allowed time horizon is T), and incur no regret in iterations $i > i_\star$. In the following, we only consider the case when $i_\star < p$ and $\Delta T_{i_\star} = 2^{p+i_\star}$. As a result, we have $K_{i_\star} \Delta T_{i_\star} = 2^{2p+2}$ and thus

$$\Delta T_{i_\star} = \frac{2^{2p+2}}{K_{i_\star}} > \frac{2^{2p+1}}{T^\alpha \log T}, \tag{19}$$

where Eq. (19) comes from the fact that $K_{i_\star} < \max\{2T^\alpha \log T, 2T^\alpha \log \sqrt{T}\} = 2T^\alpha \log T$ by definition of $i_\star$.

We now analysis the expected approximation error for iteration $i > i_\star$. Since the sampling information during the $i_\star$-th iteration is summarized in the virtual mixture-arm $\widetilde{\nu}_{i_\star}$, and being added to all S_i for all $i > i_\star$. Let $\widetilde{\mu}_{i_\star} = \mathbb{E}_{X \sim \widetilde{\nu}_{i_\star}}[X]$ denote the expected reward of sampling according to the virtual mixture-arm $\widetilde{\nu}_{i_\star}$. For any $i > i_\star$, we then have

$$\begin{aligned} \Delta T_i \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_i})\right] &\leq \Delta T_i \cdot (\mu_\star - \mathbb{E}[\widetilde{\mu}_{i_\star}]) \\ &= \frac{\Delta T_i}{\Delta T_{i_\star}} \cdot \left(\Delta T_{i_\star} \cdot (\mu_\star - \mathbb{E}[\widetilde{\mu}_{i_\star}])\right) \\ &= \frac{\Delta T_i}{\Delta T_{i_\star}} \cdot \left(\Delta T_{i_\star} \cdot \mu_\star - \sum_{t = T_{i_\star - 1}+1}^{T_{i_\star}} \mathbb{E}[\mu_{A_t}]\right) \\ &= \frac{\Delta T_i}{\Delta T_{i_\star}} \cdot R_{\Delta T_{i_\star}} \\ &< \frac{\Delta T_i}{\frac{2^{2p+1}}{T^\alpha \log T}} \cdot 168 \,(\log_2 T)^{1/2} \cdot T^\beta \\ &\leq \frac{T^{1+\alpha+\beta}}{2^{2p}} \cdot 84 \,(\log_2 T)^{3/2} \qquad (20) \\ &\leq 84 \,(\log_2 T)^{3/2} \cdot T^{1+\alpha-\beta}, \qquad (21) \end{aligned}$$

where Eq. (20) comes from the fact that $\Delta T_i \leq T$ and some rewriting; Eq. (21) comes from the fact that $p = \lceil \log_2 T^\beta \rceil \geq \log_2 T^\beta$.

Combining Eq. (21) and Eq. (13) gives the following regret bound for iterations $i > i_\star$:

$$R_{\Delta T_i} \leq 250 \,(\log_2 T)^{3/2} \cdot T^{\max\{\beta, 1+\alpha-\beta\}},$$

where the constant 250 simply comes from $84 + 166$.

Since the cumulative regret is non-decreasing in t, we have

$$\begin{aligned} R_T &\leq \sum_{i=1}^{p} R_{\Delta T_i} \\ &\leq 250 \, p \,(\log_2 T)^{3/2} \cdot T^{\max\{\beta, 1+\alpha-\beta\}} \\ &\leq 250 \,(\log_2 T + 1) \cdot (\log_2 T)^{3/2} \cdot T^{\max\{\beta, 1+\alpha-\beta\}} \qquad (22) \\ &\leq 251 \,(\log_2 T)^{5/2} \cdot T^{\max\{\beta, 1+\alpha-\beta\}}, \end{aligned}$$

where Eq. (22) comes from the fact that $p = \lceil \log_2(T^\beta) \rceil \leq \log_2(T^\beta) + 1 \leq \log_2 T + 1$. Our results follows after noticing $R_T \leq T$ is a trivial upper bound. $\square$

A.2 Anytime version

Algorithm 4: Anytime version of `MOSS++`

Input: User specified parameter $\beta \in [1/2, 1]$.

for $i = 0, 1, \dots$ **do**

  Run Algorithm 1 with parameter β for 2^i rounds (note that we will set $p = \lceil \log_2 2^{i\beta} \rceil = \lceil i\beta \rceil$).

end for

Corollary 1. *For any unknown time horizon T, run Algorithm 4 with an user-specified parameter $\beta \in [1/2, 1]$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq C (\log_2 T)^{5/2} \cdot T^{\min\{\max\{\beta, 1+\alpha-\beta\}, 1\}},$$

where C is a universal constant.

Proof. Let $t_\star$ be the smallest integer such that

$$\sum_{i=0}^{t_\star} 2^i = 2^{t_\star+1} - 1 \geq T.$$

We then only need to run Algorithm 1 for at most $t_\star$ times. By the definition of $t_\star$, we also know that $2^{t_\star} \leq T$, which leads to $t_\star \leq \log_2 T$.

Let $\gamma = \min\{\max\{\beta, 1+\alpha-\beta\}, 1\}$. From Theorem 1 we know that the regret at $i \in [t_\star]$-th round, denoted as R_{2^i}, could be upper bounded by

$$R_{2^i} \leq 251 (\log_2 2^i)^{5/2} \cdot (2^i)^\gamma = 251\, i^{5/2} \cdot (2^\gamma)^i \leq 251\, t_\star^{5/2} \cdot (2^\gamma)^i \leq 251 (\log_2 T)^{5/2} \cdot (2^\gamma)^i.$$

For $i = 0$, we have $R_{2^0} \leq 1 \leq 251 (\log_2 T)^{5/2} \cdot (2^\gamma)^0$ as well as long as $T \geq 2$.

Now for the unknown time horizon T, we could upper bound the regret by

$$\begin{aligned}
R_T &\leq \sum_{i=0}^{t_\star} R_{2^i} \\
&\leq 251 (\log_2 T)^{5/2} \cdot \left(\sum_{i=0}^{t_\star} (2^\gamma)^i \right) \\
&\leq 251 (\log_2 T)^{5/2} \cdot \int_{x=0}^{t_\star+1} (2^\gamma)^x dx \qquad (23) \\
&= 251 (\log_2 T)^{5/2} \cdot \frac{1}{\log 2^\gamma} \cdot \left((2^\gamma)^{t_\star+1} - 1 \right) \\
&\leq \frac{2^\gamma}{\gamma \log 2} 251 (\log_2 T)^{5/2} \cdot T^\gamma \\
&\leq 1449 (\log_2 T)^{5/2} \cdot T^\gamma, \qquad (24)
\end{aligned}$$

where Eq. (23) comes from upper bounding summation by integral; and Eq. (24) comes from a trivial bound on the constant when $1/2 \leq \gamma \leq 1$. □

B Omitted proofs for Section 4

B.1 Proof of Theorem 2

Theorem 2. *For any $0 \leq \alpha' < \alpha \leq 1$, assume $T^\alpha \leq B$ and $\lfloor T^\alpha \rfloor - 1 \geq \max\{T^\alpha/4, 2\}$. If an algorithm is such that $\sup_{\omega \in \mathcal{H}_T(\alpha')} R_T \leq B$, then the regret of this algorithm is lower bounded on $\mathcal{H}_T(\alpha)$:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \geq 2^{-10} T^{1+\alpha} B^{-1}. \qquad (2)$$

The proof of Theorem 2 is mainly inspired by the proofs of lower bounds in [23, 17]. Before the start of the proof, we first state a generalized version of Pinsker's inequality developed in [17] (Lemma 3 therein).

Lemma 3. *Let $\mathbb{P}$ and $\mathbb{Q}$ be two probability measures. For any random variable $Z \in [0, 1]$, we have*

$$|\mathbb{E}_{\mathbb{P}}[Z] - \mathbb{E}_{\mathbb{Q}}[Z]| \leq \sqrt{\mathrm{KL}(\mathbb{P}, \mathbb{Q})/2}.$$

We consider $K + 1$ bandit instances $\{\underline{\nu}_i\}_{i=0}^K$ such that each bandit instance is a collection of n distributions $\underline{\nu}_i = (\nu_{i1}, \nu_{i2}, \ldots, \nu_{in})$ where each ν_{ij} represents a Gaussian distribution $\mathcal{N}(\mu_{ij}, 1/4)$ with $\mu_{ij} = \mathbb{E}[\nu_{ij}]$. For any given $0 \leq \alpha' < \alpha \leq 1$ and time horizon T large enough, we choose $n, m_0, m, K \in \mathbb{N}_+$ such that the following three conditions are satisfied:

1. $n = m_0 + Km$;
2. $n/m_0 \leq 2T^{\alpha'}$;
3. $n/m \leq 2T^{\alpha}$.

Proposition 1. *Integers satisfying the above three conditions exist. For instance, we could first fix $m \in \mathbb{N}_+$ and set $K = \lfloor T^\alpha \rfloor - 1 \geq 2$.[8] One could then set $m_0 = m\lceil T^{\alpha-\alpha'} \rceil$ and $n = m_0 + Km$.*

Proof. We notice that the first condition holds by construction. We now show that the second and the third conditions hold.

For the second condition, we have

$$\begin{aligned}
\frac{n}{m_0} &= \frac{m_0 + Km}{m_0} \\
&= 1 + \frac{m(\lfloor T^\alpha \rfloor - 1)}{m\left\lceil T^{\alpha-\alpha'} \right\rceil} \\
&\leq 1 + \frac{T^\alpha}{T^{\alpha-\alpha'}} \\
&\leq 2T^{\alpha'}.
\end{aligned}$$

For the third condition, we have

$$\begin{aligned}
\frac{n}{m} &= \frac{m_0 + Km}{m} \\
&= \frac{m\lceil T^{\alpha-\alpha'} \rceil + (\lfloor T^\alpha \rfloor - 1)m}{m} \\
&= \lceil T^{\alpha-\alpha'} \rceil + \lfloor T^\alpha \rfloor - 1 \\
&= \left(\lceil T^{\alpha-\alpha'} \rceil - 1\right) + \lfloor T^\alpha \rfloor \\
&\leq T^{\alpha-\alpha'} + T^\alpha \\
&\leq 2T^\alpha.
\end{aligned}$$

$\square$

Now we group n distribution into $K + 1$ different groups based on their indices: $S_0 = [m_0]$ and $S_i = [m_0 + i \cdot m] \backslash [m_0 + (i - 1) \cdot m]$. Let $\Delta \in (0, 1]$ be a parameter to be tuned later, we then define $K + 1$ bandit instances $\underline{\nu}_i$ for $i \in \{0\} \cup [K]$ by assigning different values to their means μ_{ij}:

$$\mu_{ij} = \begin{cases} \Delta/2 & \text{if } j \in S_0, \\ \Delta & \text{if } j \in S_i \text{ and } i \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

We could clearly see there are m_0 best arms in instance $\underline{\nu}_0$ and m best arms in instances $\underline{\nu}_i, \forall i \in [K]$. Based on our construction in Proposition 1, we could then conclude that, with time horizon T, the

[8] $K \geq 2$ holds for T large enough.

regret minimization problem with respect to $\underline{\nu}_0$ is in $\mathcal{H}_T(\alpha')$; and similarly the regret minimization problem with respect to $\underline{\nu}_i$ is in $\mathcal{H}_T(\alpha), \forall i \in [K]$.

For any $t \in [T]$, the tuple of random variables $H_t = (A_1, X_1, \dots, A_t, X_t)$ is the outcome of an algorithm interacting with an bandit instance up to time t. Let $\Omega_t = ([n] \times \mathbb{R})^t \subseteq \mathbb{R}^{2t}$ and $\mathcal{F}_t = \mathfrak{B}(\Omega_t)$; one could then define a measurable space $(\Omega_t, \mathcal{F}_t)$ for H_t. The random variables $A_1, X_1, \dots, A_t, X_t$ that make up the outcome are defined by their coordinate projections:

$$A_t(a_1, x_1, \dots, a_t, x_t) = a_t \quad \text{and} \quad X_t(a_1, x_1, \dots, a_t, x_t) = x_t.$$

For any fixed algorithm/policy π and bandit instance $\underline{\nu}_i, \forall i \in \{0\} \cup [K]$, we are now constructing a probability measure $\mathbb{P}_{i,t}$ over $(\Omega_t, \mathcal{F}_t)$. Note that a policy π is a sequence $(\pi_t)_{t=1}^T$, where π_t is a probability kernel from $(\Omega_{t-1}, \mathcal{F}_{t-1})$ to $([n], 2^{[n]})$. For each i, we define another probability kernel $p_{i,t}$ from $(\Omega_{t-1} \times [n], \mathcal{F}_{t-1} \otimes 2^{[n]})$ to $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$ that models the reward. Assuming the reward is distributed according to $\mathcal{N}(\mu_{ia_t}, 1/4)$, we give its explicit expression for any $B \in \mathfrak{B}(\mathbb{R})$ as:

$$p_{i,t}\big((a_1, x_1, \dots, a_t), B\big) = \int_B \sqrt{\frac{2}{\pi}} \exp\big(-2(x - \mu_{ia_t})\big) dx.$$

The probability measure over $\mathbb{P}_{i,t}$ over $(\Omega_t, \mathcal{F}_t)$ could then be define recursively as $\mathbb{P}_{i,t} = p_{i,t}\big(\pi_t \, \mathbb{P}_{i,t-1}\big)$. We use $\mathbb{E}_i$ to denote the expectation taken with respect to $\mathbb{P}_{i,T}$. Apply the same analysis as on page 21 of [17], we obtain the following proposition on KL decomposition.

Proposition 2.

$$\mathrm{KL}\left(\mathbb{P}_{0,T}, \mathbb{P}_{i,T}\right) = \mathbb{E}_0\left[\sum_{t=1}^T \mathrm{KL}\left(\mathcal{N}(\mu_{0A_t}, 1/4), \mathcal{N}\left(\mu_{iA_t}, 1/4\right)\right)\right].$$

With respect to notations and constructions described above, we now prove Theorem 2.

Proof. (Theorem 2) Let $N_{S_i}(T) = \sum_{t=1}^T \mathbb{1}\left(A_t \in S_i\right)$ denote the number of times the algorithm π selects an arm in S_i up to time T. Let $R_{i,T}$ denote the expected (pseudo) regret achieved by the algorithm π interacting with the bandit instance $\underline{\nu}_i$. Based on the construction of bandit instance in Eq. (25), we have

$$R_{0,T} \geq \frac{\Delta}{2} \sum_{i=1}^K \mathbb{E}_0\left[N_{S_i}(T)\right], \tag{26}$$

and $\forall i \in [K]$,

$$R_{i,T} \geq \frac{\Delta}{2}\left(T - \mathbb{E}_i[N_{S_i}(T)]\right) = \frac{T\Delta}{2}\left(1 - \frac{\mathbb{E}_i[N_{S_i}(T)]}{T}\right). \tag{27}$$

According to Proposition 2 and the calculation of KL-divergence between two Gaussian distributions, we further have

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}_{0,T}, \mathbb{P}_{i,T}) &= \mathbb{E}_0\left[\sum_{t=1}^T \mathrm{KL}\left(\mathcal{N}(\mu_{0A_t}, 1/4), \mathcal{N}\left(\mu_{iA_t}, 1/4\right)\right)\right] \\
&= \mathbb{E}_0\left[\sum_{t=1}^T 2\left(\mu_{0A_t} - \mu_{iA_t}\right)^2\right] \\
&= 2\, \mathbb{E}_0\left[N_{S_i}(T)\right] \Delta^2,
\end{aligned} \tag{28}$$

where Eq. (28) comes from the fact that μ_{0j} and μ_{ij} only differs for $j \in S_i$ and the difference is exactly Δ.

We now consider the average regret over $i \in [K]$:

$$
\begin{aligned}
\frac{1}{K}\sum_{i=1}^{K} R_{i,T} &\geq \frac{T\Delta}{2}\left(1 - \frac{1}{K}\sum_{i=1}^{K}\frac{\mathbb{E}_i[N_{S_i}(T)]}{T}\right) \\
&\geq \frac{T\Delta}{2}\left(1 - \frac{1}{K}\sum_{i=1}^{K}\left(\frac{\mathbb{E}_0[N_{S_i}(T)]}{T} + \sqrt{\frac{\mathrm{KL}(\mathbb{P}_{0,T},\mathbb{P}_{i,T})}{2}}\right)\right) && (29) \\
&= \frac{T\Delta}{2}\left(1 - \frac{1}{K}\frac{\sum_{i=1}^{K}\mathbb{E}_0[N_{S_i}(T)]}{T} - \frac{1}{K}\sum_{i=1}^{K}\sqrt{\mathbb{E}_0\left[N_{S_i}(T)\right]\Delta^2}\right) && (30) \\
&\geq \frac{T\Delta}{2}\left(1 - \frac{1}{K} - \sqrt{\frac{\sum_{i=1}^{K}\mathbb{E}_0\left[N_{S_i}(T)\right]\Delta^2}{K}}\right) && (31) \\
&\geq \frac{T\Delta}{2}\left(1 - \frac{1}{K} - \sqrt{\frac{2\Delta R_{0,T}}{K}}\right) && (32) \\
&\geq \frac{T\Delta}{2}\left(\frac{1}{2} - \sqrt{\frac{2\Delta B}{K}}\right), && (33)
\end{aligned}
$$

where Eq. (29) comes from applying Lemma 3 with $Z = N_{S_i}(T)/T$ and $\mathbb{P} = \mathbb{P}_{0,T}$ and $\mathbb{Q} = \mathbb{P}_{i,T}$; Eq. (30) comes from applying Eq. (28); Eq. (31) comes from concavity of $\sqrt{\cdot}$ and the fact that $\sum_{i=1}^{K}\mathbb{E}_0[N_{S_i}(T)] \leq T$; Eq. (32) comes from applying Eq. (26); and finally Eq. (33) comes from the fact that $K \geq 2$ by construction and the assumption that $R_{0,T} \leq B$.

To obtain a large value for Eq. (33), one could maximize Δ while still make $\sqrt{2\Delta B/K} \leq 1/4$. Set $\Delta = 2^{-5}KB^{-1}$, following Eq. (33), we obtain

$$
\begin{aligned}
\frac{1}{K}\sum_{i=1}^{K} R_{i,T} &\geq 2^{-8}TKB^{-1} \\
&= 2^{-8}T\left(\lfloor T^{\alpha}\rfloor - 1\right)B^{-1} && (34) \\
&\geq 2^{-10}T^{1+\alpha}B^{-1}, && (35)
\end{aligned}
$$

where Eq. (34) comes from the construction of K; and Eq. (35) comes from the assumption that $\lfloor T^{\alpha}\rfloor - 1 \geq T^{\alpha}/4$.

Now we only need to make sure $\Delta = 2^{-5}KB^{-1} \leq 1$. Since we have $K = \lfloor T^{\alpha}\rfloor - 1 \leq T^{\alpha}$ by construction and $T^{\alpha} \leq B$ by assumption, we obtain $\Delta = 2^{-5}KB^{-1} \leq 2^{-5} < 1$ as desired. $\square$

B.2 Proof of Theorem 3

Lemma 4. *Suppose an algorithm achieves rate function θ, then for any $0 < \alpha \leq \theta(0)$, we have*

$$\theta(\alpha) \geq 1 + \alpha - \theta(0). \tag{36}$$

Proof. Fix $0 < \alpha \leq \theta(0)$. For any $\epsilon > 0$, there exists constant c_1 and c_2 such that for sufficiently large T,

$$\sup_{\omega \in \mathcal{H}_T(0)} R_T \leq c_1 T^{\theta(0)+\epsilon} \quad \text{and} \quad \sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq c_2 T^{\theta(\alpha)+\epsilon}.$$

Let $B = \max\{c_1, 1\} \cdot T^{\theta(0)+\epsilon}$, we could see that $T^{\alpha} \leq T^{\theta(0)} \leq B$ holds by assumption. For T large enough, the condition $\lfloor T^{\alpha}\rfloor - 1 \geq \max\{T^{\alpha}/4, 2\}$ of Theorem 2 holds. We then have

$$c_2 T^{\theta(\alpha)+\epsilon} \geq 2^{-10}T^{1+\alpha}\left(\max\{c_1,1\}\cdot T^{\theta(0)+\epsilon}\right)^{-1} = 2^{-10}T^{1+\alpha-\theta(0)-\epsilon}/\max\{c_1,1\}.$$

For T sufficiently large, we then must have

$$\theta(\alpha) + \epsilon \geq 1 + \alpha - \theta(0) - \epsilon.$$

Let $\epsilon \to 0$ leads to the desired result. $\square$

Lemma 5. *Suppose an rate function θ is achieved by an algorithm, then we must have*

$$\theta(\alpha) \geq \min\{\max\{\theta(0), 1+\alpha-\theta(0)\}, 1\}, \tag{37}$$

with $\theta(0) \in [1/2, 1]$.

Proof. For any rate function θ achieved by an algorithm, we first notice that $\theta(\alpha) \geq \theta(\alpha')$ for any $0 \leq \alpha' < \alpha \leq 1$ since $\mathcal{H}_T(\alpha') \subseteq \mathcal{H}_T(\alpha)$; this also implies $\theta(\alpha) \geq \theta(0)$. From Lemma 4, we further obtain $\theta(\alpha) \geq 1+\alpha-\theta(0)$ if $\alpha \leq \theta(0)$. Thus, for any $\alpha \in (0, \theta(0)]$, we have

$$\theta(\alpha) \geq \max\{\theta(0), 1+\alpha-\theta(0)\}. \tag{38}$$

Note that this indicates $\theta(\theta(0)) = 1$, as we trivially have $R_T \leq T$. For any $\alpha \in (\theta(0), 1]$, we have $\theta(\alpha) \geq \theta(\theta(0)) = 1$, which leads to $\theta(\alpha) = 1$ for $\alpha \in [\theta(0), 1]$. To summarize, we obtain the desired result in Eq. (37). We have $\theta(0) \in [1/2, 1]$ since the minimax optimal rate among problems in $\mathcal{H}_T(0)$ is $1/2$. ☐

Theorem 3. *The rate function achieved by* `MOSS++` *with any* $\beta \in [1/2, 1]$*, i.e.,*

$$\theta_\beta : \alpha \mapsto \min\{\max\{\beta, 1+\alpha-\beta\}, 1\}, \tag{3}$$

is Pareto optimal.

Proof. From Theorem 1, we know that the rate in Eq. (3) is achieved by Algorithm 1 with input β. We only need to prove that no other algorithms achieve strictly smaller rates in pointwise order.

Suppose, by contradiction, we have θ' achieved by an algorithm such that $\theta'(\alpha) \leq \theta_\beta(\alpha)$ for all $\alpha \in [0, 1]$ and $\theta'(\alpha_0) < \theta(\alpha_0)$ for at least one $\alpha_0 \in [0, 1]$. We then must have $\theta'(0) \leq \theta_\beta(0) = \beta$. We consider the following two exclusive cases.

Case 1 $\theta'(0) = \beta$. According to Lemma 5, we must have $\theta' \geq \theta_\beta$, which leads to a contradiction.

Case 2 $\theta'(0) = \beta' < \beta$. According Lemma 5, we must have $\theta' \geq \theta_{\beta'}$. However, $\theta_{\beta'}$ is not strictly better than θ_β, e.g., $\theta_{\beta'}(2\beta-1) = 2\beta-\beta' > \beta = \theta_\beta(2\beta-1)$, which also leads to a contradiction. ☐

C Omitted proofs for Section 5

C.1 Proof of Lemma 1

Lemma 1. *Suppose α is the true hardness parameter and $\alpha_i - 1/\lceil \log T \rceil < \alpha \leq \alpha_i$, run Algorithm 2 with time horizon T and α_i leads to the following regret bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq C \, \log T \cdot T^{(1+\alpha)/2},$$

where C is a universal constant.

Proof. According to Lemma 2, the definition of α and the assumption that $\alpha \leq \alpha_i$, we know that $\mathbb{P}(\mathcal{E}) \leq 1/\sqrt{T}$. We now upper bound the regret:

$$\begin{aligned}
R_T &\leq \Big(39 \sqrt{|S_{\alpha_i}| T} + |S_{\alpha_i}|\Big) \cdot \mathbb{P}(\neg \mathcal{E}) + T \cdot \mathbb{P}(\mathcal{E}) && (39)\\
&\leq \Big(39 \sqrt{|S_{\alpha_i}| T} + |S_{\alpha_i}|\Big) \cdot 1 + T \cdot \frac{1}{\sqrt{T}} \\
&\leq 56 \, (\log T)^{1/2} \cdot T^{(1+\alpha_i)/2} + 2 \log T \cdot T^{\alpha_i} + \sqrt{T} \\
&\leq 59 \, \log T \cdot T^{(1+\alpha_i)/2} \\
&< 59 \, \log T \cdot T^{(1+\alpha)/2} \cdot T^{1/(2\, \lceil \log T \rceil)} && (40)\\
&\leq 59 \sqrt{e} \, \log T \cdot T^{(1+\alpha)/2}, && (41)
\end{aligned}$$

where Eq. (39) comes from the regret bound of `MOSS`; Eq. (40) comes from the assumption that $\alpha_i < \alpha + 1/\lceil \log T \rceil$; and Eq. (41) comes from the fact that $T^{1/(2\, \lceil \log T \rceil)} = e^{(\log T / (2\, \lceil \log T \rceil))} \leq \sqrt{e}$.[9] ☐

[9] One can sharpen the $\log T$ term to $(\log T)^{1/2}$ in many cases, e.g., when $\alpha < 1$ and T is large enough (with respect to α). Again, we mainly focus on the polynomial terms here.

C.2 Proof of Theorem 4

We first provide a martingale (difference) concentration result from [29] (a rewrite of Theorem 2.19).

Lemma 6. *Let $\{D_t\}_{t=1}^{\infty}$ be a martingale difference sequence adapted to filtration $\{\mathcal{F}_t\}_{t=1}^{\infty}$. If $\mathbb{E}[\exp(\lambda D_t)|\mathcal{F}_{t-1}] \leq \exp(\lambda^2\sigma^2/2)$ almost surely for any $\lambda \in \mathbb{R}$, we then have*

$$\mathbb{P}\left(\left|\sum_{i=1}^{t} D_i\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{\epsilon^2}{2t\sigma^2}\right).$$

Theorem 4. *For any $\alpha \in [0,1]$ unknown to the learner, run* `Parallel` *with time horizon T and optimal expected reward $\mu_\star$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq C\ (\log T)^2\, T^{(1+\alpha)/2},$$

where C is a universal constant.

Proof. This proof largely follows the proof of Theorem 4 in [23]. For any $T \in \mathbb{N}_+$ and $i \in [\lceil \log T \rceil]$, recall $\texttt{SR}_i$ is the subroutine initialized with T and $\alpha_i = i/[\lceil \log T \rceil]$. We use $T_{i,t}$ to denote the number of samples allocated to $\texttt{SR}_i$ up to time t, and represent its empirical regret at time t as $\widehat{R}_{i,t} = T_{i,t} \cdot \mu_\star - \sum_{t=1}^{T_{i,t}} X_{i,t}$ where $X_{i,t} \sim \nu_{A_{i,t}}$ is the t-th empirical reward obtained *by* $\texttt{SR}_i$ and $A_{i,t}$ is the index of the t-th arm pulled *by* $\texttt{SR}_i$. We consider the corresponding regret $R_{i,t} = T_{i,t} \cdot \mu_\star - \sum_{t=1}^{T_{i,t}} \mathbb{E}[\mu_{A_{i,t}}]$ (which is random in $T_{i,t}$). We choose $\delta = 1/\sqrt{T}$ as the confidence parameter and provide $\delta' = \delta/\lceil \log T \rceil$ failure probability to each subroutine.

Notice that $R_{i,t} - \widehat{R}_{i,t} = \sum_{t=1}^{T_{i,t}} \left(X_{i,t} - \mathbb{E}[\mu_{A_{i,t}}]\right)$ is a martingale with respect to filtration $\mathcal{F}_t = \sigma\left(\bigcup_{i\in[\lceil \log T \rceil]} \{T_{i,1}, A_{i,1}, X_{i,1}, \dots, T_{i,t}, A_{i,T_{i,t}}, X_{i,T_{i,t}}\}\right)$; and $(R_{i,t} - \widehat{R}_{i,t}) - (R_{i,t-1} - \widehat{R}_{i,t-1})$ defines a martingale difference sequence. Since, no matter what value $T_{i,t}$ takes, $X_{i,T_{i,t}} - \mathbb{E}[\mu_{A_{i,T_{i,t}}}] = (X_{i,T_{i,t}} - \mu_{A_{i,T_{i,t}}}) + (\mu_{A_{i,T_{i,t}}} - \mathbb{E}[\mu_{A_{i,T_{i,t}}}])$ is $(\sqrt{2}/2)$-sub-Gaussian (following a similar analysis as in Eq. (7)), applying Lemma 6 together with a union bound gives:

$$\mathbb{P}\left(\forall i \in [\lceil \log T \rceil], \forall t \in [T] : |\widehat{R}_{i,t} - R_{i,t}| \geq \sqrt{T_{i,t} \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}\right) \leq \delta. \tag{42}$$

We use $\mathcal{E} = \left\{\forall i \in [\lceil \log T \rceil], \forall t \in [T] : |\widehat{R}_{i,t} - R_{i,t}| < \sqrt{T_{i,t} \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}\right\}$ to denote the good event that holds true with probability at least $1 - \delta$. Since the regret could be trivially upper bounded by $T \cdot \delta = \sqrt{T}$ when $\mathcal{E}$ doesn't hold, we only focus on the case when event $\mathcal{E}$ holds in the following.

Fix any subroutine $k \in [\lceil \log T \rceil]$ and consider its empirical regret $\widehat{R}_{k,T}$ up to time T. For any $j \neq k$, let $T_j \leq T$ be the last time that the subroutine $\texttt{SR}_j$ was invoked, we have

$$\begin{aligned}
\widehat{R}_{j,T_j} &\leq \widehat{R}_{k,T_j} \\
&\leq R_{k,T_j} + \sqrt{T_{k,T_j} \cdot \log\left(2T\lceil \log T \rceil/\delta\right)} \\
&\leq R_{k,T} + \sqrt{T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)},
\end{aligned} \tag{43}$$

where Eq. (43) comes from the fact that the cumulative regret $R_{k,t}$ in non-decreasing in t. Since $\texttt{SR}_j$ will only run additional $\lceil \sqrt{T} \rceil$ rounds after it was selected at time T_j, we further have

$$\begin{aligned}
\widehat{R}_{j,T} &\leq \widehat{R}_{j,T_j} + \left\lceil \sqrt{T} \right\rceil \\
&\leq R_{k,T} + \sqrt{5T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)},
\end{aligned} \tag{44}$$

where Eq. (44) comes from the combining Eq. (43) with a trivial bounding $\lceil \sqrt{T} \rceil \leq \sqrt{4T}$ for all $T \in \mathbb{N}_+$. Combining Eq. (44) with the fact that $R_{j,T} \leq \widehat{R}_{j,T} + \sqrt{T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}$ leads to

$$R_{j,T} \leq R_{k,T} + 4\sqrt{T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}. \tag{45}$$

Let $i_\star \in [\lceil \log T \rceil]$ denote the index such that $\alpha_{i_\star - 1} < \alpha \leq \alpha_{i_\star}$. As the total regret is the sum of all subroutines, we have that, for some universal constant C,

$$\sum_{i=1}^{\lceil \log T \rceil} R_{i,T} \leq \lceil \log T \rceil \cdot \left(R_{i_\star, T} + 4\sqrt{T \cdot \log\left(2T\lceil \log T \rceil / \delta\right)} \right) \tag{46}$$

$$\leq \lceil \log T \rceil \cdot \left(59\sqrt{e}\, \log T \cdot T^{(1+\alpha)/2} + 4\sqrt{T \cdot \log\left(2T^{3/2}\lceil \log T \rceil\right)} \right) \tag{47}$$

$$\leq C \, (\log T)^2 \, T^{(1+\alpha)/2},$$

where Eq. (46) comes from setting $k = i_\star$ in Eq. (45); Eq. (47) comes from applying Lemma 1 with the non-decreasing nature of cumulative regret and taking $\delta = 1/\sqrt{T}$. Integrate once more leads to the desired result. □

C.3 Anytime version

The anytime version of Algorithm 3 could be constructed as following.

Algorithm 5: Anytime version of `Parallel`

1: **for** $i = 0, 1, \ldots$ **do**
2: Run Algorithm 3 with the optimal expected reward $\mu_\star$ for 2^i rounds.
3: **end for**

Corollary 2. *For any time horizon T and $\alpha \in [0, 1]$ unknown to the learner, run Algorithm 5 with optimal expected reward $\mu_\star$ leads to the following anytime regret upper:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leq C \, (\log T)^2 \, T^{(1+\alpha)/2},$$

where C is a universal constant.

Proof. The proof is similar to the one for Corollary 1. □