Problem Dependent View on Structured Thresholding Bandit Problems

James Cheshire [1] **Pierre Ménard** [1] **Alexandra Carpentier** [1]

Abstract

We investigate the *problem dependent regime* in the stochastic *Thresholding Bandit problem* (*TBP*) under several *shape constraints*. In the *TBP* the objective of the learner is to output, at the end of a sequential game, the set of arms whose means are above a given threshold. The vanilla, unstructured, case is already well studied in the literature. Taking K as the number of arms, we consider the case where (i) the sequence of arm's means $(\mu_k)_{k=1}^K$ is monotonically increasing (*MTBP*) and (ii) the case where $(\mu_k)_{k=1}^K$ is concave (*CTBP*). We consider both cases in the *problem dependent* regime and study the probability of error - i.e. the probability to mis-classify at least one arm. In the fixed budget setting, we provide upper and lower bounds for the probability of error in both the concave and monotone settings, as well as associated algorithms. In both settings the bounds match in the *problem dependent* regime up to universal constants in the exponential.

1. Introduction

Stochastic multi-armed bandit problems model situations in which a learner faces multiple unknown probability distributions, or "arms", and has to sequentially sample these arms.

In this paper, we focus on the Thresholding Bandit Problem (*TBP*), a *Combinatorial Pure Exploration (CPE)* bandit setting introduced by Chen et al. (2014). The learner is presented with $[K] = \{1, \dots, K\}$ arms, each following an unknown distribution ν_k with unknown mean μ_k. We focus on the *fixed budget* variant of this problem. Given a budget $T > 0$, the learner samples the arms sequentially for a total of T times and then aims at predicting the set of arms whose mean is above a known threshold $\tau \in \mathbb{R}$. We will measure the learner's performance by the *probability of error* - i.e. the probability that the learner mis-classifies at least one arm - and consider therefore the *problem dependent regime*.

The focus of this paper is on *structured, shape constrained TBP*. More precisely, we study the influence of some classical *structures, in the form of a shape constraint* on the *sequence of means of the arms*, on the *TBP* problem. That is, we study how classical shape constraints influence the probability of error. A related study was performed by Cheshire et al. (2020) for the problem independent (overall worst-case) regime, and we aim at extending this study to the *problem dependent regime*. We will aim at finding the problem dependent quantities that have an impact on the optimal probability of error, and at providing matching upper and lower bounds.

We will discuss three structured *TBP*s in this paper; among those, we recall existing results of one, and provide results for two. Here is a short overview.

Vanilla, unstructured case *TBP* The vanilla, unstructured case is the simplest *TBP* where we only assume that the distributions of the arms are sub-Gaussian - also related to the TOP-M[1] setting. The *TBP* is already well studied in the literature - both in a fixed budget and in a fixed confidence context - and we only introduce it here to provide a benchmark for later structured problems. We recall here results in the problem dependent, fixed budget, setting, which is most relevant for this paper. Locatelli et al. (2016) prove that up to multiplicative constants, and additives $\log(TK)$ terms, in the exponential, the optimal probability of regret in this problem is $\exp(-\frac{T}{\sum_{i:\Delta_i>0}\Delta_i^{-2}})$, where $\Delta_i = |\tau - \mu_i|$. We present their results for completeness and comparison to the bounds under additional shape constraints in Table 5.3 - see also Subsection 3.1. The *TBP* in the problem dependent regime is also studied by Mukherjee et al. (2017) and Zhong et al. (2017), however they consider a problem complexity based also upon variance making their results not so relevant to our setting. The *problem independent* regime for the *TBP* is studied by Cheshire et al. (2020), we also present their results in Ta-

[1]Otto von Guericke University Magdeburg. Correspondence to: James Cheshire <james.cheschire@ovgu.de>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]In the TOP-M setting, the objective of the learner is to output the M arms with highest means. A popular version of it it is the TOP-1 or "best arm identification" problem where the aim is to find the arm that realises the maximum.

ble 5.3 for comparison across the different regimes.

Monotone constraint, *MTBP*. We then consider the problem where on top of assuming that the distributions are sub-Gaussian, we assume that the sequence of means $(\mu_k)_{k\in[K]}$ is monotone - this is problem *MTBP*. This specific instance of the *TBP* is introduced within the context of drug dosing by Garivier et al. (2017). In this paper, the authors provide an algorithm for the fixed confidence setting that is optimal asymptotically, in the fixed confidence regime. However the definition of the algorithms, as well as the provided optimal error bound, are defined in an implicit way and not so easy to relate in a simple way to the gaps Δ_i moreover it is not clear how to translate a result from the fixed confidence setting to the fixed budget one. On the other hand, the shape constraint on the means of the arms implies that the *MTBP* is related to *noisy binary search*, i.e. inserting an element into its correct place within an ordered list when only noisy labels of the elements are observed, see Feige et al. (1994). They describe an algorithm structurally similar to ours, using a binary tree with infinite extension however they consider a simpler setting where the probability of correct labeling is fixed as some $\delta > \frac{1}{2}$ and go on to show that there exists an algorithm that will correctly insert an element with probability at least $1-\delta$ in $\mathcal{O}\left(\log\left(\frac{K}{\delta}\right)\right)$ steps. For further literature on the related yet different problem of noisy binary search, see Feige et al. (1994), Ben-Or & Hassidim (2008), Emamjomeh-Zadeh et al. (2016), Nowak (2011). Again, these papers consider settings with more structural assumptions than our own and are focused on the problem independent, fixed confidence regime. The *problem independent* regime for the *MTBP* is studied by Cheshire et al. (2020), we also present their results in Table 5.3 for comparison across the different regimes.

In this work, we prove that, up to universal multiplicative constants and additive $\log(K)$ terms in the exponential, the optimal error probability is $\exp(-T\min_k \Delta_k^2)$, which highlights the somewhat surprising fact that this structured monotone *TBP* problem is akin to a one armed *TBP*- see Subsection 3.2. We provide the Problem Dependent Monotone *TBP* (PD-MTB) algorithm that matches this bound, see Section 4.

Concave constraint, *CTBP*. We next consider the problem where on top of assuming that the distributions are sub-Gaussian, we assume that the sequence of means $(\mu_k)_{k\in[K]}$ is concave - this is problem *CTBP*. Again, in the problem independent regime the *CTBP* has been studied by Cheshire et al. (2020). In the problem dependent regime however, to the best of our knowledge, the *CTBP* has not been studied in the literature. However the related problems of estimating a concave function and optimising a concave function are well studied in the literature. Both problems are considered primarily in the continuous regime which makes comparison to the K-armed bandit setting difficult. The problem of estimating a concave function has been thoroughly studied in the noiseless setting, and also in the noisy setting, see e.g. Simchowitz et al. (2018), where a continuous set of arms is considered, under Hölder smoothness assumptions. The problem of optimising a convex function in noise without access to its derivative - namely zeroth order noisy optimisation - has also been extensively studied. See e.g. Nemirovski & Yudin. (1983)[Chapter 9], and Wang et al. (2018); Agarwal et al. (2011); Liang et al. (2014) to name a few, all of them in a continuous setting with dimension d. The focus of this literature is however very different to ours and Cheshire et al. (2020), as the main difficulty under their assumption is to obtain a good dependence in the dimension d, and with this in mind logarithmic factors are not very relevant.

In this work, we prove that, up to universal multiplicative constants and additive $\log(K)$ terms in the exponential, the optimal error probability is $\exp(-T\min_k \Delta_k^2)$, which highlights the somewhat surprising fact that this structured concave *TBP* problem is also akin to a one armed *TBP*- see Subsection 3.3. We provide the Problem Dependent Concave *TBP* (PD-CTB) algorithm that matches this bound, see Section 4.

Organisation of the paper This paper is structured as follows. In Section 2 we formally introduce the *TBP* setting along with the monotone and concave shape constraints. We also describe the performance criterion - probability of error, we will be primarily using for the duration of the paper. Following this, upper and lower bounds on probability of error for all shape constraints are presented in Section 3. Descriptions of algorithms achieving said upper bounds can be found in Section 4. The results are discussed and compared to related work in Section 5. In Appendix E we conduct some preliminary experiments to explore how our theoretical results translate in practice. All proofs are found in the Appendix.

2. Setting

Problem formulation The learner is presented with a K-armed bandit problem $\underline{\nu} = \{\nu_1, \dots, \nu_K\}$, with $K \geq 3$, where ν_k is the unknown distribution of arm k.

Let $\sigma^2 \geq 0$. We remind the learner that distribution ν of mean μ is said to be σ^2-sub-Gaussian if for all $t \in \mathbb{R}$ we have,

$$\mathbb{E}_{X\sim\nu}\left[e^{t(X-\mu)}\right] \leq \exp\left(\frac{\sigma^2 t^2}{2}\right).$$

In particular the Gaussian distributions with variance smaller than σ^2 and the distributions with absolute values bounded by σ are σ^2-sub-Gaussian.

Let $\mathcal{B} := \mathcal{B}(K, \sigma^2)$ be the set of all bandit problems as presented above, i.e. where the distributions ν_k of the arms

are all σ^2 sub-Gaussian.

In what follows, we assume that all $\underline{\nu} \in \mathcal{B}$, and we write μ_k for the mean of arm k. Let $\tau \in \mathbb{R}$ be a fixed threshold known to the learner. We aim to devise an algorithm which classifies arms as above or below threshold τ based on their means. That is, the learner aims at finding the vector $Q \in \{-1,1\}^K$ that encodes the true classification, i.e. $Q_k = 2\mathbb{1}_{\{\mu_k \geq \tau\}} - 1$ with the convention $Q_k = 1$ if arm k is above the threshold and $Q_k = -1$ otherwise. The *fixed budget* bandit sequential learning setting goes as follows: the learner has a budget $T > 0$ and at each round $t \leq T$, the learner pulls an arm $k_t \in [K]$ and observes a sample $Y_t \sim \nu_{k_t}$, conditionally independent from the past. After interacting with the bandit problem and expending their budget, the learner outputs a vector $\hat{Q} \in \{-1,1\}^K$ and the aim is that it matches the unknown vector Q as well as possible.

Unstructured case *TBP* In the *problem dependent* regime, for $\bar{\Delta} \in \mathbb{R}_+^K$, we consider the following class of problems

$$\mathcal{B}^{\bar{\Delta}} = \{\nu \in \mathcal{B} : \forall k \in [K],\ |\mu_k - \tau| = \bar{\Delta}_k\}\ .$$

Monotone case *MTBP* We denote by $\mathcal{B}_m$ the set of bandit problems,

$$\mathcal{B}_m := \{\nu \in \mathcal{B} :\ \mu_1 \leq \mu_2 \leq \ldots \leq \mu_K\}\ ,$$

where the learner is given the additional information that the sequence of means $(\mu_k)_{k\in[K]}$ is a monotonically increasing sequence. We denote by $\Delta\mathcal{B}_m = \{\bar{\Delta} \in \mathbb{R}_+^K : \exists \nu \in \mathcal{B}_m, \forall k \in [K], |\mu_k - \tau| = \bar{\Delta}_k\}$ the set of possible vectors of gaps in $\mathcal{B}_m$ - i.e. the set of sequences $\bar{\Delta}$ that would correspond to at least one problem in $\mathcal{B}_m$. In the *problem dependent* regime, for $\bar{\Delta} \in \Delta\mathcal{B}_m$, we consider the following class of problems

$$\mathcal{B}_m^{\bar{\Delta}} = \{\nu \in \mathcal{B}_m : \forall k \in [K],\ |\mu_k - \tau| = \bar{\Delta}_k\}\ .$$

Concave case *CTBP* We will denote by $\mathcal{B}_c$ the set of bandit problems,

$$\mathcal{B}_c := \left\{\nu \in \mathcal{B} : \forall 1 < k < K-1, \frac{1}{2}\mu_{k-1} + \frac{1}{2}\mu_{k+1} \leq \mu_k\right\}\ ,$$

where the learner is given the additional information that the sequence of means $(\mu_k)_{k\in[K]}$ is concave. We denote by $\Delta\mathcal{B}_c = \{\bar{\Delta} \in \mathbb{R}_+^K : \exists \nu \in \mathcal{B}_c, \forall k \in [K], |\mu_k - \tau| = \bar{\Delta}_k, \exists l : \mu_l \geq \tau\}$ the set of possible vectors of gaps in $\mathcal{B}_c$ where at least one arm is above threshold - i.e. the set of sequences $\bar{\Delta}$ that would correspond to at least one problem in $\mathcal{B}_c$ where at least one arm is above threshold. In the *problem independent* regime, for $\bar{\Delta} \in \Delta\mathcal{B}_c$, we consider the following class of problems

$$\mathcal{B}_c^{\bar{\Delta}} := \left\{\nu \in \mathcal{B}_c : \forall k < K, |\mu_k - \tau| \in \left[\frac{\bar{\Delta}_k}{2}, 3\frac{\bar{\Delta}_k}{2}\right]\right\}\ .$$

Remark 1. *The classes of problems $\mathcal{B}^{\bar{\Delta}}, \mathcal{B}_m^{\bar{\Delta}}, \mathcal{B}_c^{\bar{\Delta}}$ contain bandit problems in resp. $\mathcal{B}, \mathcal{B}_m, \mathcal{B}_c$ that are 'local' around $\bar{\Delta}$ in the sense that while the sign of $\mu_k - \tau$ is arbitrary - although severely restricted by the shape constraint when it comes to $\mathcal{B}_m^{\bar{\Delta}}, \mathcal{B}_c^{\bar{\Delta}}$ - the gap of arm k is fixed to being - approximately, for the concave case set $\mathcal{B}_c^{\bar{\Delta}}$ - $\bar{\Delta}_k$. This implies that in each case and on top of the respective shape constraint, we restrict ourselves to a small class of problems whose complexity is entirely characterised by $\bar{\Delta}$, in a problem dependent sense.*

Strategy A strategy is a sequence of functions that maps the information gathered in the past to an arm and finally to a classification. Precisely, if we denote by I_t the information available to the player at time t, that is $I_t = \{Y_1, Y_2, \ldots, Y_t\}$, with the convention $I_0 = \emptyset$. Then a strategy $\pi = \left((\pi_t)_{t\in[T]}, \hat{Q}^\pi\right)$ is given by a sampling rule $\pi_t(I_{t-t}) = k_t \in [K]$ and a classification rule $\hat{Q}^\pi(I_T) = \hat{Q} \in \{-1,1\}^K$.

Minimax expected regret The *problem independent*, *fixed budget* objective of the learner following the strategy π is then to minimize the expected simple regret of this classification for $\hat{Q} := \hat{Q}^\pi$:

$$r_T^{\underline{\nu},\pi} = \mathbb{E}_{\underline{\nu}}\left[\max_{\{k\in[K]:\ \hat{Q}_k^\pi \neq Q_k\}} \Delta_k\right]\ ,$$

where $\Delta_k := |\tau - \mu_k|$ is the gap of arm k, and where $\mathbb{E}_{\underline{\nu}}$ is defined as the expectation on problem $\underline{\nu}$ and $\mathbb{P}_{\underline{\nu}}$ the probability. However, the focus of this paper is on the *problem dependent* regime where, as usual, we consider as a performance criterion rather the related *probability of error*

$$e_T^{\underline{\nu},\pi} = \mathbb{P}_{\underline{\nu}}\Big(\exists k \in [K] : \hat{Q}_k^\pi \neq Q_k\Big)\ .$$

When it is clear from the context we will remove the dependence on the bandit problem $\underline{\nu}$ and/or the strategy π. Note that if we denote by $\bar{\Delta}_{\min} = \min_{k\in[K]} \bar{\Delta}_k$ the minimum of the gaps then

$$r_T^{\underline{\nu},\pi} \geq \bar{\Delta}_{\min} e_T^{\underline{\nu},\pi}\ .$$

Consider a set of bandit problems $\tilde{\mathcal{B}} \subset \mathcal{B}$. The minimax optimal probability of error on $\tilde{B}$ is then

$$e_T^*(\tilde{\mathcal{B}}) := \inf_{\pi \text{ strategy}} \sup_{\underline{\nu}\in\tilde{\mathcal{B}}} e_T^{\underline{\nu},\pi}.$$

We will study this quantity over the local classes $\mathcal{B}^{\bar{\Delta}}, \mathcal{B}_m^{\bar{\Delta}}, \mathcal{B}_c^{\bar{\Delta}}$.

Remark 2. *As argued above, the classes $\mathcal{B}^{\bar{\Delta}}, \mathcal{B}_m^{\bar{\Delta}}, \mathcal{B}_c^{\bar{\Delta}}$ contain only bandit problems that satisfy their respective shape constraint and whose complexity is entirely characterised by $\bar{\Delta}$, in a problem dependent sense. Studying the minimax probability of error over these very restricted classes*

is therefore a very meaningful way of studying the problem dependent regime of structured TBP problems - and we expect this probability of error to heavily depend on $\bar{\Delta}$. The focus of this paper is to characterise this dependence in a tight manner.

3. Minimax rates

In this section we present upper and lower bounds on probability of error for all three shape constraints. Given a vector $\bar{\Delta} \in \mathbb{R}_+^K$ we denote $\bar{\Delta}_{\min} = \min_{k\in[K]} \bar{\Delta}_k$.

3.1. Problem dependent unstructured setting *TBP*

The unstructured thresholding bandit in the problem dependent regime has already been considered in the literature. We remind results from Locatelli et al. (2016), where they provide tight upper and lower bounds over $e_T^*(\mathcal{B}^{\bar{\Delta}})$, for any $\bar{\Delta} \in \mathbb{R}_+^K$. In our context they prove that

$$\exp\left(-\frac{3}{\sigma^2}\frac{T}{H} - 4\sigma^{-2}\log\left(12(\log T+1)K\right)\right) \leq e_T^*(\mathcal{B}^{\bar{\Delta}}) \leq \exp\left(-\frac{1}{64\sigma^2}\frac{T}{H} + 2\log\left((\log T+1)K\right)\right),$$

where $H = \sum_{i:\bar{\Delta}_i>0} 1/\bar{\Delta}_i^2$ - see Theorems 1 and 2 by Locatelli et al. (2016). This implies that up to multiplicative universal constants and whenever $T \geq H\sigma^2\log(\log(T)+K)$, it holds that

$$-\log\left(e_T^*(\mathcal{B}^{\bar{\Delta}})\right) \asymp \frac{1}{\sigma^2}\frac{T}{H},$$

and upper and lower bound match up to universal multiplicative constants in the exponential of the error probability. The quantity H is therefore the problem dependent quantity that characterises the difficulty of the problem. Note that of course, the APT algorithm by Locatelli et al. (2016) does not take any information on the class - $\bar{\Delta}$, but also σ^2 - as parameters, and is essentially parameter free.

In this paper, we won't therefore discuss further this unstructured setting - the reminder provided here is only to be taken as a benchmark for the rest of the paper. We will on the other hand focus on the structured problems - monotone and concave and study how the minimax error probability evolves, in particular depending on the problem-dependent quantities $\bar{\Delta}$.

3.2. Problem dependent monotone setting

Given a class of problems $\mathcal{B}_m^{\bar{\Delta}}$ for some $\bar{\Delta} \in \Delta\mathcal{B}_m$, the following theorem provides a lower bound on the probability of error for any strategy π. The proof of Theorem 3 can be found in Appendix C.

Theorem 3. *Let $\bar{\Delta} \in \Delta\mathcal{B}_m$. For any strategy π there exists a monotone bandit problem $\nu \in \mathcal{B}_m^{\bar{\Delta}}$ such that*

$$e_T^{\nu,\pi} \geq \frac{1}{4}\exp\left(-\frac{T\bar{\Delta}_{\min}^2}{\sigma^2}\right).$$

Now the following theorem gives an upper bound on the probability of error for the PD-MTB algorithm. The proof of Theorem 4 can be found in Appendix C.

Theorem 4. *Let $\nu \in \mathcal{B}_m$ associated with arm gaps Δ, and assume that $T > 36\log(K)$. The algorithm PD-MTB satisfies the following bound on error probability:*

$$e_T^{\nu,\text{PD-MTB}} \leq \exp\left(-c_{\text{mon}}\frac{T\Delta_{min}^2}{\sigma^2} + c'_{\text{mon}}\log(K)\right)$$

where $c_{\text{mon}} = 1/48$ and $c'_{\text{mon}} = 12$.

The parameter free algorithm PD-MTB is described in Sections 4 - see also Appendix C.

The assumption on T is reasonable as in the monotone setting it is clear no algorithm can gain enough information in less than $\log(K)$ pulls, see Cheshire et al. (2020). Note that combining both bounds yields that whenever $T > 36\log(K)/\bar{\Delta}_{\min}^2$:

$$-\log\left(e_T^*(\mathcal{B}_m^{\bar{\Delta}})\right) \asymp \frac{1}{\sigma^2}T\bar{\Delta}_{\min}^2,$$

and upper and lower bound match up to universal multiplicative constants in the exponential of the error probability. Perhaps surprisingly, the number of arms plays no role in this rate - as long as we assume that $T > 36\log(K)/\bar{\Delta}_{\min}^2$. Only the minimal arm gap appears, and this amounts to saying that when $T > 36\log(K)/\bar{\Delta}_{\min}^2$, this problem is not more difficult - in order, up to universal multiplicative constants in the exponential - than a one-armed *TBP* with gap $\min_k \Delta_k$! And that in a sense, even if we knew in our monotone problem the position of all means but one - the arm with minimal gap - with respect to the threshold, the problem would not be significantly easier.

3.3. Problem dependent concave setting

Given a class of problems $\mathcal{B}_c^{\bar{\Delta}}$ for some $\bar{\Delta} \in \Delta\mathcal{B}_c$ the following theorem provides a lower bound on the probability of error for any strategy π. The proof of Theorem 5 can be found in Appendix D.

Theorem 5. *Let $\bar{\Delta} \in \Delta\mathcal{B}_c$. For any strategy π there exists a problem $\nu \in \mathcal{B}_c^{\bar{\Delta}}$ such that*

$$e_T^{\nu,\pi} \geq \frac{1}{4}\exp\left(-9\frac{T\bar{\Delta}_{\min}^2}{\sigma^2}\right).$$

Now the following theorem gives an upper bound on the probability of error for the PD-CTB algorithm. The proof of Theorem 6 can be found in Appendix D.

Theorem 6. *Let $\nu \in \mathcal{B}_c$ with associated gaps Δ and assume $T > 108\log(K)$. The algorithm PD-CTB has the following bound on error,*

$$e_T^{\nu,\text{PD-CTB}} \leq 3\exp\left(-c_{\text{con}}\frac{T\Delta_{\min}^2}{\sigma^2} + c'_{\text{con}}\log(K)\right)$$

where $c_{\text{con}} = 1/576$ *and* $c'_{\text{con}} = 12$.

The parameter free algorithm PD−CTB is described in Sections 4 - see also Appendix D.

The assumption on T is reasonable as in the monotone setting it is clear no algorithm can gain enough information in less than $\log(K)$ pulls, see Cheshire et al. (2020). Note that combining both bounds yields that whenever $T > 108\frac{\log(K)}{\Delta^2_{\min}}$:

$$-\log\left(e^*_T(\mathcal{B}^{\bar{\Delta}}_m)\right) \asymp \frac{1}{\sigma^2} T\bar{\Delta}^2_{\min},$$

and upper and lower bound match up to universal multiplicative constants in the exponential of the error probability. Similar comments can be made here as in the case of the monotone *TBP* in Section 3.2: the convex *TBP* is also as difficult as a one-armed *TBP* with gap $\min_k \Delta_k$.

4. Optimal algorithms in the problem dependent regime

4.1. Monotone case *MTBP*

We assume in this section, without loss of generality, instead of considering K arms, we consider for technical reasons $K+2$ arms adding two deterministic arms 0 and $K+1$ with respective means $\mu_0 = -\infty$ and $\mu_{K+1} = +\infty$. While we assume that the distributions of the original K arms are σ^2-sub-Gaussian the addition of two such arms will not invalidate our proofs, see Appendix C. We do this to ensure that, after re-indexing of the arms and adapting the number of arms, $\tau \in [\mu_1, \mu_K]$.

To match a minimax rate as described in Section 3 we will utilise a modified version of the MTB algorithm described by Cheshire et al. (2020). The algorithm PD−MTB performs a random walk on the set of arms $[K]$ as a binary tree. We consider the binary tree as Cheshire et al. (2020) with an specific extension akin to that by Feige et al. (1994).

Binary Tree We associate to each problem $\nu \in \mathcal{B}_m$ a binary tree. Precisely we consider a binary tree with nodes of the form $v = \{L, M, R\}$ where $\{L, M, R\}$ are indexes of arms and we note respectively $v(l) = L, v(r) = R, v(m) = M$. The tree is built recursively as follows: the root is $\texttt{root} = \{1, \lfloor (1+K)/2 \rfloor, K\}$, and for a node $v = \{L, M, R\}$ with $L, M, R \in \{1, \dots, K\}$ the left child of v is $L(v) = \{L, M_l, M\}$ and the right child is $R(v) = \{M, M_r, R\}$ with $M_l = \lfloor (L+M)/2 \rfloor$ and $M_r = \lfloor (M+R)/2 \rfloor$ as the middle index between. The leaves of the tree will be the nodes $\{v = \{L, M, R\} : R = L+1\}$. If a node v is a leaf we set $R(v) = L(v) = \emptyset$. We consider the tree up to maximum depth $H = \lfloor \log_2(K) \rfloor + 1$. We note $P\big(l(v)\big) = P\big(r(v)\big)$ the parent of the two children and let $|v|$ denote the depth of node v in the tree, with $|\texttt{root}| = 0$. We adopt the convention $P(\texttt{root}) = \texttt{root}$.

Extended Binary Tree We extend the above Binary tree in the following manner. For a leaf v we replace the condition $R(v) = L(v) = \emptyset$ with the following: for any leaf $v = \{L, M, R\}$ we set $R(v) = \tilde{v}$ where $\tilde{v} = \{L, M, R\}$ and set $L(v) = \emptyset$. Note that $\tilde{v}$ is also a leaf therefore iterative application this relation will lead to an infinite extension. The result being that each leaf in our original binary tree is now the root of an infinite chain of identical nodes, see Figure 1. For practical purposes we need only consider such an extension up to depth T and can simply cut the tree at this depth.

Remark 7. *We set $L(v) = \emptyset$ for some leaf v during the extension of the binary tree as by construction all leaves of the original binary tree are of the form $\{v = \{L, M, R\} : R = L + 1 \text{ and } M = L\}$.*

In order to predict the right classification we want to find the arm whose mean is the one just above the threshold τ. Finding this arm is equivalent to inserting the threshold into the (sorted) list of means, which can be done with a binary search in the aforementioned binary tree. But in our setting we only have access to estimates of the means which can be very unreliable if the mean is close to the threshold. Because of this there is a high chance we will make a mistake on some step of the binary search. For this reason we must allow PD−MTB to backtrack and this is why PD−MTB performs a binary search *with corrections*.

PD−MTB algorithm First, define the following integers

$$T_1 := \lceil 6\log(K) \rceil \qquad T_2 := \left\lfloor \frac{T}{3T_1} \right\rfloor . \tag{1}$$

The algorithm PD−MTB is then essentially a random walk on said binary tree moving one step per iteration for a total of T_1 steps. Let $v_1 = \texttt{root}$ and for $t < T_1$ let v_t denote the current node, the algorithm samples arms $\{v_t(j) : j \in \{l, m, r\}\}$ each T_2 times. Let the sample mean of arm $v_t(j)$ be denoted $\hat{\mu}_{j,t}$. PD−MTB will use these estimates to decide which node to explore next. If an error is detected - i.e. the interval between left and rightmost sample mean does not contain the threshold, then the algorithm backtracks to the parent of the current node, otherwise PD−MTB acts as the deterministic binary search for inserting the threshold τ in the sorted list of means. More specifically, if there is an anomaly, $\tau \notin [\hat{\mu}_{l,t}, \hat{\mu}_{r,t}]$, then the next node is the parent $v_{t+1} = P(v_t)$, otherwise if $\tau \in [\hat{\mu}_{l,t}, \hat{\mu}_{m,t}]$ the the next node is the left child $v_{t+1} = L(v_t)$ and if $\tau \in [\hat{\mu}_{m,t}, \hat{\mu}_{r,t}]$ the next node is the right child $v_{t+1} = R(v_t)$. If at time t, $\tau \in [\hat{\mu}_{l,t}, \hat{\mu}_{r,t}]$ and the node v_t is a leaf, that is $v(r) = v(l) + 1$, then due to the extension of our binary tree $R(v_t) = L(v_t) = \tilde{v}_t$ where $\tilde{v}$ is a duplicate of v_t. Hence $v_{t+1} = \tilde{v}_t$. Via this mechanism the PD−MTB algorithm essentially gives additional preference the the node v_t. See PD−MTB for details. We now formally

state the parameter free PD-MTB algorithm (Problem Dependent Monotone Thresholding Bandit Algorithm). We rely on the assumption $T > 36\log(K)$, see Theorem 4 to ensure $T_2 \geq 1$.

Algorithm 1 PD-MTB

Initialization: $v_1 = \texttt{root}$
for $t = 1 : T_1$ **do**
  sample T_2 times each arm in v_t
  if $\tau \notin [\hat{\mu}_{l,t}, \hat{\mu}_{r,t}]$ **then**
    $v_{t+1} = P(v_t)$
  else if $\hat{\mu}_{m,t} \leq \tau \leq \hat{\mu}_{r,t}$ **then**
    $v_{t+1} = R(v_t)$
  else if $\hat{\mu}_{l,t} \leq \tau \leq \hat{\mu}_{m,t}$ **then**
    $v_{t+1} = L(v_t)$
  end if
end for
Set $\hat{k} = v_{T_1+1}(r)$
return $(\hat{k}, \hat{Q}) : \quad \hat{Q}_k = 2\mathbb{1}_{\{k \geq \hat{k}\}} - 1$

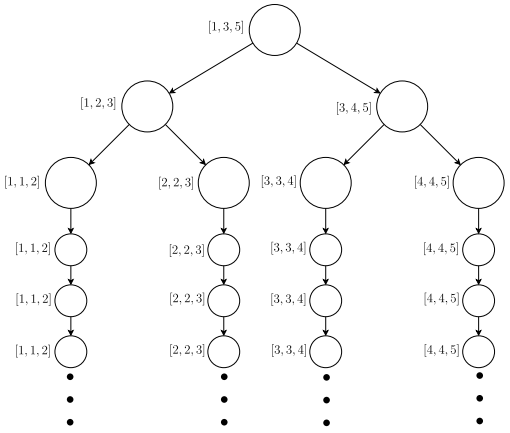

Figure 1. Extended binary tree for $K = 5$

Remark 8 (Adaptation of PD-MTB to a non-increasing sequence, PD-DEC-MTB)**.** *PD-MTB is applied for a monotone non-decreasing sequence $(\mu_k)_{k\in[K]}$, and it is easy to adapt it to a monotone non-increasing sequence $(\mu_k)_{k\in[K]}$. In this case, we transform the label of arm k into $K - k$, and apply PD-MTB to the newly labeled problem - where the mean sequence in now non-decreasing. We refer to this modification as PD-DEC-MTB.*

Remark 9 (Relaxing the monotone assumption)**.** *By inspecting the proof of Theorem 4 in Appendix C we can obtain the same guarantee for a larger class of problem than one with increasing means. Indeed we only need that there exists an arm for which all the arms before it have a mean below the threshold and all arm after have a mean above the threshold. Precisely the bound of Theorem 4 holds also for problems that belongs to*

$$\mathcal{B}_{rm} := \{\nu \in \mathcal{B} : \exists k \in [1, K], \forall j \leq k\ \mu_j \leq \tau, \\ \forall j \geq k+1\ \mu_j \geq \tau\}\ .$$

Note the same remark also applies for problems with monotone non-increasing sequence.

4.2. Concave case *CTBP*

We assume in this section, without loss of generality, instead of considering K arms, we consider for technical reasons $K + 2$ arms adding two deterministic arms 0 and $K + 1$ with respective means $\mu_0 = \mu_{K+1} = -\infty$. While we assume that the distributions of the original K arms are σ^2-sub-Gaussian the addition of two such arms will not invalidate our proofs, see Appendix D. We do this to ensure that after re-indexing $\tau > \mu_1, \mu_K$.

As in the monotone case we construct a binary tree to span the arms of the bandit problem. The construction of this tree is identical to that described in Section 4.1 but without the infinite extension. We will use a variant off the PD-MTB Algorithm, Grad-Explore to move around the tree. The difference is that Grad-Explore bases its movement off the estimated gradients of the arms as opposed to their sample means. The objective of Grad-Explore is to find an arm with corresponding mean above threshold. Once such an arm has been identified we split our problem into two "relaxed monotone" bandit problems - see Remark 9, one increasing and one decreasing. We then run PD-MTB and PD-DEC-MTB respectively. We split our budget evenly across the three algorithms: Grad-Explore, PD-MTB and PD-DEC-MTB.

Grad-Explore algorithm As with PD-MTB the algorithm Grad-Explore is essentially a random walk on the said binary tree moving one step per iteration for a total of T_1 steps. Let $v_1 = \texttt{root}$ and for $t < T_1$ let v_t denote the current node, the algorithm samples arms $\{v_t(l), v_t(l) + 1, v_t(m), v_t(m) + 1, v_t(r), v_t(r) + 1\}$ each T_2 times. As in Section 4.1, we adopt the convention that the arm $K + 1$ is a Dirac distribution at $-\infty$. Let the sample mean of arm $v_t(j)$ be denoted $\hat{\mu}_{j,t}$ and the sample mean of arm $v_t(j)+1$ be denoted $\hat{\mu}_{j+1,t}$. Let the estimated local gradient at arm j, that is $\hat{\mu}_{j,t} - \hat{\mu}_{j+1,t}$ denote $\hat{\nabla}_{j,t}$. Grad-Explore will use these estimates to decide which node to explore next. If an error is detected - i.e. the left most or right most gradient is negative or positive respectively, then the algorithm backtracks to the parent of the current node, otherwise Grad-Explore acts as the deterministic binary search for the maximum mean, $\max_{i\in[K]} \mu_i$. More specifically, if there is an anomaly, $\left(\hat{\nabla}_{l,t}, \hat{\nabla}_{r,t}\right) \notin (\mathbb{R}_+, \mathbb{R}_-)$, then the next node is the parent $v_{t+1} = P(v_t)$, otherwise if $\hat{\nabla}_{m,t} < 0$ the next node is the

left child $v_{t+1} = L(v_t)$ and if $\hat{\nabla}_{m,t} \geq 0$ the next node is the right child $v_{t+1} = R(v_t)$. See Algorithm 2 for details.

Algorithm 2 Grad-Explore

Initialization: $v_1 = \texttt{root}$
for $t = 1 : T_1$ **do**
  $S_{t+1} = S_t$
  for each $k \in v_t$ sample $\frac{T_2}{12}$ times the arms $k, k+1$
  if $\exists k \in \{l, m, r\} : \hat{\mu}_k > \tau$ **then**
    Append arm k to the list S_{t+1}
    $v_{t+1} = v_t$
  else if $\left(\hat{\nabla}_{l,t}, \hat{\nabla}_{r,t}\right) \notin (\mathbb{R}_+, \mathbb{R}_-)$ **then**
    $v_{t+1} = P(v_t)$
  else if $\hat{\nabla}_{m,t} \geq 0$ **then**
    $v_{t+1} = R(v_t)$
  else if $\hat{\nabla}_{m,t} < 0$ **then**
    $v_{t+1} = L(v_t)$
  end if
end for

Algorithm 3 PD-CTB

run Grad-Explore
output list S_{T_1}
if $|S_{T_1}| \leq \frac{T_1}{4}$ **then**
  return $\hat{Q} = \{-1\}^K$
else
  $\hat{k} = \text{Median}(S_{T_1})$
  l = output of PD-DEC-MTB on set of arms $[1, \hat{k}]$ budget: $\frac{T}{3}$
  r = output of PD-MTB on set of arms $[\hat{k}, K]$ budget: $\frac{T}{3}$
  return $\hat{Q}: \quad \hat{Q}_k = 1 - 2\mathbb{1}_{k<l} - 2\mathbb{1}_{k>r}$
end if

For the arms whose means are below threshold, due to the concave property gradients are essentially greater than $\bar{\Delta}_{\min}$ and can easily be estimated. Above threshold however gradients are less than $\bar{\Delta}_{\min}$ and are relatively hard to estimate. Therefore, although on the face Grad-Explore is in part a binary search for the arm with maximum mean, in reality this is not feasible. The true utility of Grad-Explore to the learner is to act as a binary search for the "set" of arms above threshold. If we refer to nodes containing an arm $k : \mu_k > \tau$ as "good nodes" the idea behind Grad-Explore is to spend a sufficient amount of time in exploring this set of nodes and adding "good arms" - i.e ones with a corresponding mean above threshold, to the list S. We can then output such an arm with high probability when outputting the median of S_{T_1}.

Once we have identified our arm above threshold we split our problem into two bandit problems where the classification can be done by binary search, see Remark 9 and 8. We can thus then apply PD-MTB and PD-DEC-MTB. Precisely, the complete procedure, namely PD-CTB (Problem Dependent- Concave Threshold Bandits), is detailed in Algorithm 3.

5. Discussion

5.1. Algorithms PD-MTB and PD-CTB

Both the PD-MTB and PD-CTB are based upon a binary search with corrections, this allows them to exploit the structure of the shape constraints reducing the problems to sets of arms with cardinally of order $\log(K)$, something in sharp contrast to existing algorithms for the vanilla setting. The difference between PD-MTB and PD-CTB is that while PD-MTB works exclusively on a binary tree based upon the classification of an arms mean above or below threshold, the sub algorithm Grad-Explore of PD-CTB bases a binary tree on positive or negative gradient. Therefore PD-MTB acts as a search for the point the arms cross threshold while Grad-Explore acts as a search for the arm $k^* = \arg\max_k(\bar{\Delta}_k)$. Another more subtle difference is that on a "good decision" at time t - i.e when the sample means are well concentrated up to $\bar{\Delta}_{\min}$, PD-MTB will make a step in the right direction. The same cannot be said for Grad-Explore as we can only guarantee that the increments between arms are greater than $\bar{\Delta}_{\min}$ for arms below threshold, this is a direct result of the concave property. Therefore the true utility of Grad-Explore is not to find k^* but to find any arm $k : \mu_k > \tau$.

It is worth noting that both algorithms described in this paper are parameter free, being adaptive not only to the hardness of the problem characterised by the gaps $\bar{\Delta}$, but also to the underlying sub-Gaussian assumption parameter σ^2.

5.2. Problem classes and optimality

In the monotone and concave settings we consider a very narrow class of problems and argue our classes are relevant for characterising the problem dependent regime - i.e. are narrow enough.

- In the monotone setting this is obvious as the class of problems is defined by a specific vector $\bar{\Delta} \in \mathbb{R}_+^K$, so that all problems in this class have a similar complexity, bear in mind that our algorithms do not need to know $\bar{\Delta}_{\min}$ or any aspect of $\bar{\Delta}$. In fact, when constructing our lower bound, we just need a class with two problems where, given a first problem, we simply switch the arm with minimal gap $\bar{\Delta}_{\min}$ from below to above threshold in order to obtain the second problem - see the proof of Theorem 3.
- In the concave setting this approach is unfeasible as under the concave constraints the class of problems defined by a specific vector of gaps $\bar{\Delta} \in \mathbb{R}_+^K$ has very often cardinality 1 which is nonsensical for a lower bound. Instead, given a specific vector $\bar{\Delta} \in \mathbb{R}_+^K$ we consider a class of problems with gaps within a proportional tolerance of $\bar{\Delta}$. This class is designed to be

as narrow as possible while still containing multiple problems which disagree on the placement of certain arms above or below threshold. In fact, when constructing our lower bound, we just need a class with two problems where, starting from a first problem, we simply flip the arm with minimal gap and translate other means vertically in such a way to preserve concavity - see the proof of Theorem 3.

In both cases, we prove that for T large enough, the problem dependent optimal probability of error is of order

$$\exp(-T\bar{\Delta}^2_{\min}/\sigma^2),$$

up to universal multiplicative constants inside and outside the exponential. This implies that from a problem dependent perspective, both problems are as difficult as a one armed bandit problem where we just want to decide whether the arm with minimal gap $\bar{\Delta}_{\min}$ is up or down the threshold, which is quite surprising - as the number of arms plays therefore no role asymptotically. While the lower bounds are relatively simple, the upper bounds are more interesting and challenging.

5.3. Comparison of rates between settings

Table 5.3 presents a comparison of results across the problem independent and dependent regimes. Although the results are not immediately comparable between the regimes, of particular interest is the difference in rates across the monotone and concave settings in the problem independent regime compared to the lack of difference between said rates in the problem dependent regime.

problem:	independent	dependent
Unconstrained	$\sqrt{\frac{K\log K}{T}}$	$\exp\left(-\frac{T}{H}\right)$
Monotone	$\sqrt{\frac{\log K \vee 1}{T}}$	$\exp\left(-T\bar{\Delta}^2_{\min}\right)$
Concave	$\sqrt{\frac{\log\log K \vee 1}{T}}$	$\exp\left(-T\bar{\Delta}^2_{\min}\right)$

Table 1. Order of the optimal problem dependent probability of error, and of the problem independent expected simple regret for the three structured *TBP*, in the case of all four structural assumptions on the means of the arms considered in this paper. All results are given up to universal multiplicative constants both in and outside the exponential. The first line concerns the problem independent setting and the simple regret, see Cheshire et al. (2020). The second line concerns the problem dependent setting and the probability of error, the main focus of this paper. The results for the monotone and concave are novel and can be found in this paper, see Section 3. The results for the unstructured setting are by Locatelli et al. (2016), where they take $H = \sum_{i=1}^{K} \bar{\Delta}_i^{-2}$

In both the monotone and concave setting an initial lower bound is one which does not depend upon K - imagine the setting in which a learner places their entire budget on the two arms either side of the threshold. We show that in the problem dependent regime a binary search with corrections can match this bound, up to a $\log(K)$ term which disappears for large T. The intuition behind this is that as the depth of the tree is only $\log(K)$ the binary search can quickly find the point of interest and spend the majority of its time there. As both the concave and monotone problems can be solved with a binary search they therefore have the same rate.

In the problem independent regime the situation is slightly more nuanced. In terms of lower bounds one is no longer restricted to a narrow class of problems and can consider a number of different problems, all close in terms of distributional distance but nevertheless disagreeing on the classification of certain arms above or below threshold. The cardinality of these sets differs between the monotone and concave setting - being $\log(K)$ and $\mathrm{loglog}(K)$ respectively. This then leads to a difference in the lower bound. Upper bounds naturally must follow suit, while an adaptation of the standard binary search is still optimal in the monotone case in the concave case an algorithm using a binary search on a log scale is required. The above is by no means a rigorous explanation but hopefully gives the reader some intuition behind the differences in rates between the problem dependent and independent regimes, for more detail refer to Cheshire et al. (2020).

Acknowledgements

The work of J. Cheshire is supported by the Deutsche Forschungsgemeinschaft (DFG) DFG - 314838170, GRK 2297 MathCoRe. The work of P. Ménard is supported by the SFI Sachsen-Anhalt for the project RE-BCI ZS/2019/10/102024 by the Investitionsbank Sachsen-Anhalt. The work of A. Carpentier is partially supported by the Deutsche Forschungsgemeinschaft (DFG) Emmy Noether grant MuSyAD (CA 1488/1-1), by the DFG - 314838170, GRK 2297 MathCoRe, by the DFG GRK 2433 DAEDALUS (384950143/GRK2433), by the DFG CRC 1294 'Data Assimilation', Project A03, and by the UFA-DFH through the French-German Doktorandenkolleg CDFA 01-18 and by the UFA-DFH through the French-German Doktorandenkolleg CDFA 01-18 and by the SFI Sachsen-Anhalt for the project RE-BCI.

References

Agarwal, A., Foster, D. P., Hsu, D. J., Kakade, S. M., and Rakhlin, A. Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems*, pp. 1035–1043, 2011.

Ben-Or, M. and Hassidim, A. The bayesian learner is optimal for noisy binary search (and pretty good for quantum as well). In *2008 49th Annual IEEE Symposium on*

Foundations of Computer Science, pp. 221–230. IEEE, 2008.

Chen, S., Lin, T., King, I., Lyu, M. R., and Chen, W. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems*, pp. 379–387, 2014.

Chen, W., Hu, W., Li, F., Li, J., Liu, Y., and Lu, P. Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*, pp. 1659–1667, 2016.

Cheshire, J., Menard, P., and Carpentier, A. The influence of shape constraints on the thresholding bandit problem. In Abernethy, J. and Agarwal, S. (eds.), *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 1228–1275. PMLR, 09–12 Jul 2020.

Combes, R. and Proutiere, A. Unimodal bandits without smoothness. *arXiv preprint arXiv:1406.7447*, 2014a.

Combes, R. and Proutiere, A. Unimodal bandits: Regret lower bounds and optimal algorithms. In *International Conference on Machine Learning*, pp. 521–529, 2014b.

Emamjomeh-Zadeh, E., Kempe, D., and Singhal, V. Deterministic and probabilistic binary search in graphs. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 519–532. ACM, 2016.

Feige, U., Raghavan, P., Peleg, D., and Upfal, E. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

Garivier, A., Ménard, P., Rossi, L., and Menard, P. Thresholding bandit for dose-ranging: The impact of monotonicity. *arXiv preprint arXiv:1711.04454*, 2017.

Garivier, A., Ménard, P., and Stoltz, G. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 44(2):377–399, 2019.

Liang, T., Narayanan, H., and Rakhlin, A. On zeroth-order stochastic convex optimization via random walks. *arXiv preprint arXiv:1402.2667*, 2014.

Locatelli, A., Gutzeit, M., and Carpentier, A. An optimal algorithm for the thresholding bandit problem. In *International Conference on Machine Learning*, pp. 1690–1698. PMLR, 2016.

Mukherjee, S., Purushothama, N. K., Sudarsanam, N., and Ravindran, B. Thresholding bandits with augmented ucb. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2515–2521. AAAI Press, 2017.

Nemirovski, A. and Yudin., D. Problem complexity and method efficiency in optimization. *Wiley, New York*, 1983.

Nowak, R. D. The geometry of generalized binary search. *IEEE Transactions on Information Theory*, 57 (12):7893–7906, 2011.

Paladino, S., Trovo, F., Restelli, M., and Gatti, N. Unimodal thompson sampling for graph-structured arms. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Simchowitz, M., Jamieson, K., Suchow, J. W., and Griffiths, T. L. Adaptive sampling for convex regression. *arXiv preprint arXiv:1808.04523*, 2018.

Wang, Y., Du, S., Balakrishnan, S., and Singh, A. Stochastic zeroth-order optimization in high dimensions. In Storkey, A. and Perez-Cruz, F. (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1356–1365. PMLR, 09–11 Apr 2018.

Yu, J. Y. and Mannor, S. Unimodal bandits. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 41–48, 2011.

Zhong, J., Huang, Y., and Liu, J. Asynchronous parallel empirical variance guided algorithms for the thresholding bandit problem. *arXic preprint arXiv:1704.04567*, 2017.

A. Related work

Unstructured *TBP* As mentioned in Section 3 and demonstrated in Table 5.3 the unstructured problem dependent *TBP* is already well studied in the literature, see (Chen et al., 2014; 2016) for the fixed confidence setting and Chen et al. (2014); Locatelli et al. (2016); Mukherjee et al. (2017), Zhong et al. (2017) for the fixed budget. As mentioned in Section 1, (Locatelli et al., 2016) is most relevant to our setting as they consider the fixed budget case. Their rate for the unstructured case depends upon the distribution of gaps across all the arms, which is of course to be expected. This again highlights the fact that the rate for the monotone setting depends only upon the minimum gap - that is the one adjacent to the threshold.

Monotone constraint *MTBP* The *MTBP* problem was first introduced by Garivier et al. (2017) in the context of drug dosing. Their results are in contrast to ours as they consider the *fixed confidence* setting. Furthermore the algorithm proposed is shown to be optimal only in the asymptotic case, i.e when the confidence $1-\delta$ converges to 1. The monotone shape constraint of the *MTBP* implies it is related to a noisy binary search i.e. inserting an element into its correct place within an ordered list when only noisy labels of the elements are observed. A naive approach to the *MTBP* would be a binary search with $\frac{T}{\log(K)}$ samples at each step of the binary search. However for our setting this is not optimal, even in the problem independent case, see (Cheshire et al., 2020). In (Feige et al., 1994) this issue is solved by introducing a binary search with corrections. They describe an algorithm structurally similar to PD-MTB, using a binary tree with infinite extension however they consider a simpler setting where the probability of correct labeling is fixed as some $\delta > \frac{1}{2}$ and go on to show that there exists an algorithm that will correctly insert an element with probability at least $1-\delta$ in $\mathcal{O}\left(\log\left(\frac{K}{\delta}\right)\right)$ steps. For further literature on the related yet different problem of noisy binary search see, (Feige et al., 1994), Ben-Or & Hassidim (2008), Emamjomeh-Zadeh et al. (2016), (Nowak, 2011). Again, these papers consider settings with more structural assumptions than our own and are focused on the problem independent, fixed confidence regime. The minimax rate on expected regret for the problem independent MTB is presented by Cheshire et al. (2020).

For us the adaptation of the algorithm in Cheshire20 to the problem dependent regime is not obvious. An important fact in our problem dependent regime is that the number of arms K stops appearing in the error bound which is of order $\exp(-cT\Delta^2_{\min})$ whenever T is large enough, i.e. larger than $\log(K)/\Delta^2_{\min}$. In Cheshire et al. (2020), the number of arms appeared in all bounds and was the main topic of study therein - the bound for the monotone problem was $\sqrt{\log(K)/T}$. A key interesting phenomenon here is that somewhere between the problem independent and problem dependent regime, K stops playing a role. This implies that a very different dynamic is happening in the problem dependent regime, as compared to the problem independent regime.

Precisely in Cheshire et al. (2020) they consider a sequence of events ξ_t that depend on K and occur with constant probability - which is the target probability of error in the worst case. Lemma 15 therin then applies Hoeffding's-Azuma to the summation of the indicator functions of said events to achieve a bound on the probability of making too many bad decisions in the tree. In order to achieve a problem dependent bound, we consider events ξ_t which are problem dependent - they depend on $\Delta_{\min}$ - but NOT on K. This event is now problem dependent and the probability of its complement depends on both $\Delta_{\min}$ and $T/\log(K)$ (the number of times we sample each arm), i.e. is of order $\exp(-cT\Delta_{\min}/\log K)$, which, interestingly, is NOT the target probability of error in the problem dependent regime, but is quite larger. Our Lemma 22 is then substantially more than just a problem dependent adaptation of Lemma 15 of Cheshire et al. (2020), as we need to leverage the fact that there are many events ξ_t - here $\log K$ - in order to bypass the fact that the probability of each individual ξ_t depends on K in our setting. We use a Chernoff bound to bound the sum of the indicator functions of said events - and then in turn the probability of error - by $\exp(-T\Delta^2_{\min})$ - which is much smaller than the probability of each individual ξ_t. This phenomenon is not needed in Cheshire et al. (2020).

Another point in favour of the PD-MTB is that it is significantly simpler than that of the MTB of Cheshire et al. (2020). We use an infinite extension to the binary tree which allows it to take the final node as output. This means we don't require an additional subroutine to choose from a list of arms the algorithm has collected.

Concave constraint To the best of our knowledge the *CTBP* was first introduced in (Cheshire et al., 2020) in the *problem independent regime*. However the related problems of estimating a concave function and optimising a concave function are well studied in the literature. Both problems are considered primarily in the continuous regime which makes comparison to the K-armed bandit setting difficult. The problem of estimating a concave function has been thoroughly studied in the noiseless setting, and also in the noisy setting, see e.g. (Simchowitz et al., 2018), where a continuous set of arms is considered, under Hölder smoothness assumptions. The problem of optimising a convex function in noise without access to its derivative - namely zeroth order noisy optimisation - has also been extensively studied. See e.g. (Nemirovski &

Yudin., 1983)[Chapter 9], and (Wang et al., 2018; Agarwal et al., 2011; Liang et al., 2014) to name a few, all of them in a continuous setting with dimension d. The focus of this literature is however very different to ours and (Cheshire et al., 2020), as the main difficulty under their assumption is to obtain a good dependence in the dimension d, and with this in mind logarithmic factors are not very relevant.

B. Potential further work: Algorithms that are problem dependent and minimax-optimal simultaneously: Unimodal shape constraint

As described earlier after the related theorems, our algorithm PD−MTB is optimal for minimising the probability of error, in a problem dependent sense, and up to universal multiplicative constants in the exponential. A relevant question is on whether it is possible to construct a strategy that is optimal both in this problem dependent sense, but also in a problem independent sense - i.e. global minimax - when it comes to the simple regret.

While designed for the problem independent regime - and reaching in this regime the minimax optimal simple regret of order $\sqrt{\frac{\log K}{T}}$ - we conjecture the MTB algorithm, described by Cheshire et al. (2020) is optimal also in the problem dependent regime, i.e. that it achieves an upper bound on the probability of error of same order as that of PD−MTB in Theorem 4. However note that to prove such an opitmaility, at least for us, would be none trivial, see the above Section A.

As with PD−MTB the MTB algorithm takes a monotone bandit problem mapped to a binary tree - although without the infinite extension, as input. The MTB algorithm then consists of two sub algorithm. The first, Explore is an exploration phase, identical to our algorithm PD−MTB. However, as opposed to simply outputting the end node the history of the random walk is passed to the second algorithm, Choose. The algorithm Choose selects all arms whose sample mean is within a certain tolerance of the threshold - chosen to be as small as possible while still producing a none empty set, and then takes the median of said set. This additional step is required as the MTB algorithm aims to achieve the minimax rate on *expected regret* - that is $\sqrt{\frac{\log(K)}{T}}$, and therefore wishes to output any arm $k : |\mu_k - \tau| \lesssim \sqrt{\frac{\log(K)}{T}}$. The idea being that during the explore phase enough time will be spent on nodes containing such arms.

If we consider the problem dependent regime, and whenever we are not in the trivial regime where $\bar{\Delta}_{\min} \lesssim \sqrt{\frac{\log(K)}{T}}$, we conjecture that the MTB algorithm will spend sufficient time on the unique node $\tilde{v} : \mu_{\tilde{v}(l)} < \tau < \mu_{\tilde{v}(r)}$ with high probability matching the bound of Theorem 4. The algorithm Choose will then output arm $\tilde{v}(r)$. The problem dependent regime allows for a less convoluted approach - indeed PD−MTB is very simple in comparison to MTB. However, it is nevertheless important to note that for the monotone setting there exists an algorithm that is optimal in both problem dependent and problem independent regimes.

In regards to the concave case it is not as immediate that the CTB algorithm by Cheshire et al. (2020) will also be optimal in the problem dependent concave setting. The CTB algorithm is significantly more complex than the MTB as it successively applies a noisy binary search on a log scale to find arms increasingly close to threshold at a geometric rate. We however conjecture that it will be the case the CTB is also optimal in the problem dependent regime.

B.1. Unimodal constraint

A natural additional shape constraint for the *TBP* is a Unimodal one. Indeed bandit problems with a unimodal constraint are already considered in the literature, for the problem of minimising the cumulative regret or identifying the best arm under unimodal constraints see Yu & Mannor (2011), Combes & Proutiere (2014b), Paladino et al. (2017) and Combes & Proutiere (2014a). The *TBP* in particular with a unimodal constraint is studied in Cheshire et al. (2020) in the *problem independent* regime. With the above work already in hand it is natural to consider a unimodal shape constraint on the *TBP* in the *problem dependent* regime. A possible algorithm would be one which, similar to the PD−CTB, first finds an arm above threshold and then reduces the problem to one with a monotone constraint. We conjecture that if one considers a class of problems with M arms above threshold the regret of the problem will be dominated by that of finding a single arm above threshold and will be of the order $\exp\left(-\frac{MT\bar{\Delta}_{\min}}{K}\right)$ with a matching lower bound. If one wishes to consider a narrower class based on a single vector of gaps, as in the concave or monotone setting one might hope to achieve a rate $\exp\left(-\frac{MT}{K}(\frac{1}{M}\sum_{i=1}^{M}\bar{\Delta}_i)^2\right)$ however this result, for both an upper and lower bound, appears not so straightforward.

C. Proofs relating to the Monotone setting

We first state a useful inequality. Let $\mathrm{kl}(p,q)$ be the Kullback-Leibler divergence between two Bernoulli distributions of parameter p and q,

$$\mathrm{kl}(p,q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right)$$

It holds

$$\begin{aligned}\mathrm{kl}(p,q) &= p\log\left(\frac{1}{q}\right) + (1-p)\log\left(\frac{1}{1-q}\right) + p\log(p) + (1-p)\log(1-p)\\ &\geq p\log\left(\frac{1}{q}\right) - \log(2)\,. \end{aligned} \tag{2}$$

Proof of Theorem 3. We denote by N_k^t the number of times the arm k is pulled until and included time t, i.e. $N_k^t = \sum_{s=1}^t \mathbb{1}_{k_s=k}$. Let $i = \arg\min_{k\in[K]} \bar{\Delta}_k$, that is $\bar{\Delta}_i = \bar{\Delta}_{\min}$. Consider the two bandit problems $\underline{\nu}^+$ and $\underline{\nu}^-$ where

$$\nu_k^+ = \begin{cases}\mathcal{N}(\bar{\Delta}_k,\sigma^2) & \text{if } k\geq i\\ \mathcal{N}(-\bar{\Delta}_k,\sigma^2) & \text{else}\end{cases}, \qquad \nu_k^- = \begin{cases}\mathcal{N}(\bar{\Delta}_k,\sigma^2) & \text{if } k> i\\ \mathcal{N}(-\bar{\Delta}_k,\sigma^2) & \text{else}\end{cases}.$$

Note these bandit problems belong to the class of *MTBP* $\mathcal{B}_m^{\bar{\Delta}}$. In particular we can lower bound the error by the probability to make a mistake in the prediction of the label of arm i

$$e_T^{\underline{\nu}^+} \geq \mathbb{P}_{\underline{\nu}^+}(\hat{Q}_i = -1) \qquad e_T^{\underline{\nu}^-} \geq \mathbb{P}_{\underline{\nu}^-}(\hat{Q}_i = 1)\,.$$

We can assume that $\mathbb{P}_{\underline{\nu}^+}(\hat{Q}_i = -1) \leq 1/2$ otherwise the bound is trivially true. Thanks to the chain rule then the contraction of the Kullback-Leibler divergence (e.g. see Garivier et al. (2019)) and (2), it holds

$$\begin{aligned} T\frac{\bar{\Delta}_{\min}^2}{2\sigma^2} \geq \mathbb{E}_{\underline{\nu}^+}[N_i^T]\frac{\bar{\Delta}_{\min}^2}{2\sigma^2} &= \mathrm{KL}(\mathbb{P}_{\underline{\nu}^+}^{I_T}, \mathbb{P}_{\underline{\nu}^-}^{I_T})\\ &\geq \mathrm{kl}\left(\mathbb{P}_{\underline{\nu}^+}(\hat{Q}_i=1), \mathbb{P}_{\underline{\nu}^-}(\hat{Q}_i=1)\right)\\ &\geq \mathbb{P}_{\underline{\nu}^+}(\hat{Q}_i=1)\log\left(\frac{1}{\mathbb{P}_{\underline{\nu}^-}(\hat{Q}_i=1)}\right) - \log(2)\,,\end{aligned}$$

where we denote by $\mathbb{P}_{\underline{\nu}}^{I_T}$ the probability distribution of the history I_T under the bandit problem $\underline{\nu}$. Thus, using that $\mathbb{P}_{\underline{\nu}^+}(\hat{Q}_i = 1) = 1 - \mathbb{P}_{\underline{\nu}^+}(\hat{Q}_i = -1) \geq 1/2$ we obtain

$$\mathbb{P}_{\underline{\nu}^-}(\hat{Q}_i = 1) \geq \frac{1}{4}\exp\left(-\frac{T\bar{\Delta}_{\min}^2}{\sigma^2}\right).$$

Which allows us to conclude that

$$\max(e_T^{\underline{\nu}^+}, e_T^{\underline{\nu}^-}) \geq \frac{1}{4}\exp\left(-\frac{T\bar{\Delta}_{\min}^2}{\sigma^2}\right).$$

□

Proof of Theorem 4. We assume in the proof, without loss of generality, that

$$\Delta_{\min} \geq c_{\min}\sqrt{\frac{\sigma^2\log(K)}{T}}$$

with $c_{\min} = 13$. Indeed, otherwise, the bound of Theorem 4 is trivially true.

The proof of Theorem 4 is structured in the following manner. In our *original* binary tree we know there is a unique leaf v_Δ, such that $\tau \in [\mu_{v_\Delta(l)}, \mu_{v_\Delta(r)}]$. Essentially we want to show that the explore algorithm will terminate in the subtree of

this $v_{\bar{\Delta}}$ with high probability - recall that we extend our binary tree by attaching an infinite sub tree to each leaf, the nodes of which are identical to the respective leaf. At time t we say our algorithm makes a favourable decision if all sample means are well concentrated - that is with $\Delta_{\min}$ of their true mean. On such a favourable decision we show that the explore algorithm will make a step towards the subtree of v_Δ, or go deeper if it is already in it. Therefore if overall we can make sufficient proportion of favourable events we are guaranteed to terminate in the subtree of v_Δ. We then show that this favorable event holds with high probability.

Step 1: Initial definitions and lemmas We denote by $ST(v)$ the subtree rooted at node v.

Definition 10. *The subtree $ST(v)$ of a node v is defined recursively as follows: $v \in ST(v)$ and*

$$\forall\, q \in ST(v),\ L(q), R(q) \in ST(v)\ .$$

We define $Z_{\Delta_{\min}}$, the set of good nodes, as

$$Z_{\Delta_{\min}} := \{v : \exists k \in \{l, m, r\} : |\mu_{v(k)} - \tau| \leq \Delta_{\min}\}\ ,$$

Note that $Z_{\Delta_{\min}}$ is simply the leaf v_Δ and it's sub tree attached during the infinite extension of the binary tree. At time t we define w_t as the node of maximum depth whose subtree contains both v_t and $Z_{\Delta_{\min}}$. Formally, for $t \leq T_1$, we let

$$w_t \in \underset{\{v:\tau\in[\mu_{v(l)},\mu_{v(r)}]\ \&\ v_t\in ST(v)\}}{\arg\max} |v|\ . \tag{3}$$

Lemma 11. *The node w_t is unique.*

Proof. At time t consider, a node q_t which also satisfies (3). As $v_t \in ST(w_t)$ and $v_t \in ST(q_t)$ we can assume without loss of generality $q_t \in ST(w_t)$ with $|q_t| \geq |w_t|$. This then implies, from (3), that $|q_t| = |w_t|$ and as $q_t \in ST(w_t)$, we have $q_t = w_t$. □

For $t \leq T_1$ we define D_t as the relative distance from v_t to v_Δ, it is taken as the length of the path running from v_t up to w_t and then down (or up if $v_t \in Z_{\Delta_{\min}}$) to v_Δ. Formally, we have

$$D_t := |v_t| - |w_t| + |v_\Delta| - |w_t|.$$

Note the following properties of D_t and w_t,

$$ST(v_t) \cap Z_{\Delta_{\min}} \neq \emptyset \Rightarrow v_t = w_t\ , \tag{4}$$

$$D_t \leq 0 \Rightarrow v_t = w_t \text{ and } w_t, v_t \in Z_{\Delta_{\min}}\ . \tag{5}$$

We define the favorable event where the estimates of the means are close to the true ones for all the arms in v_t, At time t we define the event

$$\xi_t := \{\forall k \in \{l, m, r\}, |\hat{\mu}_{k,t} - \mu_{v_t(k)}| \leq \Delta_{\min}\}$$

and we denote $\bar{\xi}_t$ as the complement of ξ_t.

Step 2: Actions of the algorithm on all iterations After any execution of algorithm PD−MTB note the following, for $t \leq T_1$, v_t and v_{t+1} are separated by at most one edge, i.e.

$$v_{t+1} \in \{L(v_t), R(v_t), P(v_t)\}\ . \tag{6}$$

Lemma 12. *On execution of algorithm PD−MTB for all $t \leq T_1$ we have the following,*

$$D_{t+1} \leq D_t + 1$$

Proof. As the algorithm moves at most 1 step per iteration, see (6), for $t \leq T_1$, it holds

$$|v_t| - |w_t| \geq |v_{t+1}| - |w_t| - 1\ .$$

We consider two cases. Firstly, assume we are in the event $\{v_{t+1} \neq P(v_t)\} \cup \{w_t \neq v_t\}$. Under this event note that $v_{t+1} \in ST(w_t)$. It follows

$$\begin{aligned} D_t &= |v_t| - |w_t| + |v_\Delta| - |w_t| \\ &\geq |v_{t+1}| - |w_t| + |v_\Delta| - |w_t| - 1 \\ &\geq |v_{t+1}| - |w_{t+1}| + |v_\Delta| - |w_{t+1}| - 1 \\ &= D_{t+1} - 1 \,, \end{aligned}$$

where the third line comes from the definition of w_{t+1}, see (3).
In the case where $w_t = v_t$ and $v_{t+1} = P(v_t)$ note that $w_{t+1} = v_{t+1}$ and,

$$D_{t+1} = |v_\Delta| - |w_{t+1}| = |v_\Delta| - |w_t| + 1 = D_t + 1 \,.$$

Therefore in all cases we have $D_{t+1} \leq D_t + 1$. □

Step 3: Actions of the algorithm on ξ_t

Lemma 13. *On execution of algorithm PD-MTB for all $t \leq T_1$, on ξ_t, we have the following,*

$$D_{t+1} \leq D_t - 1 \,.$$

Proof. Note that on the favorable event ξ_t, we have $\forall j \in \{l, m, r\}$,

$$\mu_{v_t(j)} \geq \tau \Rightarrow \hat{\mu}_{j,t} \geq \tau \,, \tag{7}$$
$$\mu_{v_t(j)} \leq \tau \Rightarrow \hat{\mu}_{j,t} \leq \tau \,. \tag{8}$$

We consider the following three cases:

- If $\tau \notin \left[\mu_{v_t(l)}, \mu_{v_t(r)}\right]$. From (7) and (8), under ξ_t, we get $\tau \notin [\hat{\mu}_{l,t}, \hat{\mu}_{r,t}]$, and therefore $v_{t+1} = P(v_t)$ and $w_t = w_{t+1}$. Thus thanks to Lemma 11, under ξ_t,

$$D_{t+1} = |v_{t+1}| - |w_{t+1}| + |v_\Delta| - |w_{t+1}| = |v_t| - 1 - |w_t| + |v_\Delta| - |w_t| = D_t - 1 \,.$$

- If $\tau \in \left[\mu_{v_t(l)}, \mu_{v_t(r)}\right]$ and $v_t \notin Z_{\Delta_{\min}}$. Note that in this case v_t can not be a leaf and we just need to go down in the subtree of v_t to find v_Δ, id est $w_t = v_t$. Since $v_t \notin Z_{\Delta_{\min}}$, without loss of generality, we can assume for example $\mu_{v_t(m)} > \tau$. From (7) and (8), under ξ, we then have $\tau \in [\hat{\mu}_{l,t}, \hat{\mu}_{r,t}]$ and $\hat{\mu}_{m,t} \geq \tau$. Hence algorithm PD-MTB goes to the correct subtree, $v_{t+1} = L(v_t)$. In particular we also have for this node

$$\tau \in \left[\mu_{v_{t+1}(l)}, \mu_{v_t(m)}\right] \,,$$

  therefore it holds again $w_{t+1} = v_{t+1}$. Thus combining the previous remarks we obtain thanks to Lemma 11, under ξ_t,

$$D_{t+1} = |v_\Delta| - |w_{t+1}| = |v_\Delta| - |w_t| - 1 = D_t - 1 \,.$$

- If $\tau \in \left[\mu_{v_t(l)}, \mu_{v_t(r)}\right]$ and $v_t \in Z_{\Delta_{\min}}$. Firstly note that $w_t = v_t$. Now, using the same reasoning as in the previous case, as $\tau \in [\mu_{v_t(l)}, \mu_{v_t(r)}]$ we have $v_{t+1} = L(v_t)$ or $v_{t+1} = R(v_t)$. In either case we get $v_{t+1} \in Z_{\Delta_{\min}}$ because of (7) and (8), thus it holds $w_{t+1} = v_{t+1}$. Therefore we have

$$D_{t+1} = |v_{t+1}| - |w_{t+1}| + |v_\Delta| - |w_{t+1}| = |v_\Delta| - |w_t| - 1 = D_t - 1 \,.$$

□

Step 4: Upper bound on D_{T_1+1}

Lemma 14. *For any execution of algorithm PD-MTB*

$$D_{T_1+1} \leq 2 \sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} - \frac{3T_1}{4} \,.$$

Proof. Combining Lemma 12 and Lemma 13 respectively we have

$$D_{t+1} \leq D_t + \mathbb{1}_{\bar{\xi}_t} - \mathbb{1}_{\xi_t} .$$

Using this inequality we obtain

$$\begin{aligned} D_{T_1+1} &= D_1 + \sum_{t=1}^{T_1} (D_{t+1} - D_t) \\ &\leq D_1 + \sum_{t=1}^{T_1} \left(\mathbb{1}_{\bar{\xi}_t} - \mathbb{1}_{\xi_t}\right) \\ &\leq D_1 + 2\sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} - T_1 \\ &\leq 2\sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} - \frac{3T_1}{4} , \end{aligned}$$

where we used in the last inequality the fact that $D_1 \leq \log_2(K)$ and that $\log_2(K) \leq T_1/4$ by definition of T_1 . □

Lemma 15. *For $c_{\mathrm{mon}} = 1/48$ and $c'_{\mathrm{mon}} = 12$ it holds*

$$\mathbb{P}\left(\sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} \geq \frac{T_1}{4}\right) \leq \exp\left(c_{\mathrm{mon}} \frac{-T\Delta_{\min}^2}{\sigma^2}\right) .$$

Proof. Let $\mathcal{F}_t$ be the information available at and including step t of algorithm PD-MTB. Thanks to the Chernoff inequality and the choice of T_2, we have for all $j \in \{l, m, r\}$,

$$\begin{aligned} \mathbb{P}\left(\left|\hat{\mu}_{j,t} - \mu_{v_t(j)}\right| \geq \Delta_{\min} | \mathcal{F}_{t-1}\right) &\leq 2\exp\left(-\frac{T_2\Delta_{\min}^2}{2\sigma^2}\right) \\ &\leq 2\exp\left(-c_{\min}^2 \frac{\log(K)}{36\log(K)+6}\right) \\ &\leq \frac{1}{24} \end{aligned}$$

as we assume $\Delta_{\min} > c_{\min}\sqrt{\frac{\sigma^2 \log K}{T}}$ and $c_{\min} \geq 13$. Therefore by a union bound

$$p_t := \mathbb{P}(\bar{\xi}_t | \mathcal{F}_{t-1}) \leq p_0 := 6\exp\left(-\frac{T_2\Delta_{\min}^2}{2\sigma^2}\right) \leq \frac{1}{8}. \tag{9}$$

We will apply the Chernoff inequality to upper bound the sum of indicator function. Thanks to the Markov inequality for $\lambda \geq 0$ we have

$$\mathbb{P}\left(\sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} \geq \frac{T_1}{4}\right) \leq \mathbb{E}\left[\exp\left(\lambda \sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t}\right)\right] e^{-\lambda\frac{T_1}{4}} . \tag{10}$$

Let $\varphi_p(\lambda) = \log(1 - p + pe^{\lambda})$ be the log-partition function of a Bernoulli of parameter $p \in [0, 1]$. Note that for $\lambda \geq 0$, since $p \mapsto \varphi_p(\lambda)$ is non-decreasing and because of (9) it holds $\varphi_{p_t}(\lambda) \leq \varphi_{p_0}(\lambda)$ for all t. Thus by induction we have

$$\begin{aligned} \mathbb{E}\left[\exp\left(\lambda \sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t}\right)\right] &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(\lambda \mathbb{1}_{\bar{\xi}_t}\right) | \mathcal{F}_{T_1-1}\right] \exp\left(\lambda \sum_{t=1}^{T_1-1} \mathbb{1}_{\bar{\xi}_s}\right)\right] \\ &= \mathbb{E}\left[e^{\varphi_{p_{T_1}}(\lambda)} \exp\left(\lambda \sum_{t=1}^{T_1-1} \mathbb{1}_{\bar{\xi}_t}\right)\right] \leq e^{\varphi_{p_0}(\lambda)}\, \mathbb{E}\left[\exp\left(\lambda \sum_{s=1}^{t-1} \mathbb{1}_{\bar{\xi}_t}\right)\right] \\ &\leq \mathbb{E}\left[e^{T_1\varphi_{p_0}(\lambda)}\right] . \end{aligned}$$

Then going back to (10) and using that $\sup_{\lambda\geq 0} \lambda q - \varphi_p(\lambda) = \mathrm{kl}(q,p)$ when $q \geq p$ we get

$$\mathbb{P}\left(\sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} \geq \frac{T_1}{4}\right) \leq \exp\left(-T_1 \sup_{\lambda\geq 0}\left(\lambda\frac{1}{4} - \varphi_{p_0}(\lambda)\right)\right) = e^{-T_1\,\mathrm{kl}(1/4,p_0)}\,.$$

It remains to conclude with (2)

$$\begin{aligned} T_1\,\mathrm{kl}(1/4,p_0) &\geq T_1\frac{1}{4}\log\left(1/p_0\right) - T_1\log(2) \\ &\geq \frac{1}{8}\frac{T_1T_2\bar{\Delta}^2_{\min}}{\sigma^2} - T_1\left(\log(2) + \frac{1}{4}\log(3)\right) \\ &\geq \frac{1}{48}\frac{T\bar{\Delta}^2_{\min}}{\sigma^2} - T_1\left(\log(2) + \frac{1}{4}\log(3)\right) \\ &\geq c_{\mathrm{mon}}\frac{T\bar{\Delta}^2_{\min}}{\sigma^2} - c'_{\mathrm{mon}}\log(K) \end{aligned}$$

where $c_{\mathrm{mon}} = 1/48$ and $c'_{\mathrm{mon}} = 12$.

□

By combination of Lemmas 14 and 15 we have that $D_{T_1+1} \leq 0$ with probability greater than $1 - \exp\left(-c_{\mathrm{mon}}\frac{T\Delta^2_{\min}}{\sigma^2} + c'_{\mathrm{mon}}\log(K)\right)$. Thus with said probability we output an arm $\hat{k}$ such that $\tau \in [\mu_{\hat{k}}, \mu_{\hat{k}+1}]$.

D. Proofs relating to the concave setting

Before proceeding with the proof of Theorem 5 we present the following structural lemma.

Lemma 16. *Let $\Delta \in \Delta\mathcal{B}_c$ and let $(\mu_k)_k$ be an associated concave sequence of means. There exists a sequence of means $(\mu'_k)_k$ - with associated gaps $\Delta' = |\mu' - \tau|$ - such that*

(a) $(\mu'_k)_k$ is concave.

(b) μ and μ' have not all the arms classified in the same way:

$$\exists k \in [K] : \mathrm{sign}(\mu_k - \tau) \neq \mathrm{sign}(\mu_k - \tau)\,.$$

(c) For all $k \in [K]$ it holds that

$$|\mu'_k - \mu_k| \leq 3\Delta_{\min}.$$

(d) For all $k \in [K]$ it holds that

$$\frac{\Delta_k}{10} \leq \Delta'_k \leq 3\Delta_k.$$

Proof. Let $k^* \in \arg\min_{k\in[K]} \Delta_k$. We proceed in two cases: either this arm is up threshold, or it is below threshold. In everything that follows we set $\Delta_{\min} := \min_{k\in[K]} \Delta_k$.

Case 1: Arm below threshold, i.e. $\mu_{k^*} \leq \tau$. Let us write k_L, k_R for the two arms that are 'just' below threshold, i.e. such that $\mu_{k_L} \leq \tau \leq \mu_{k_L+1}$ and $\mu_{k_R} \leq \tau \leq \mu_{k_R-1}$. These two arms can be defined without loss of generality since there is at least one arm above threshold, and since we can always take two virtual means μ_0, μ_{K+1} at $-\infty$ on the boundaries.

In the context where $\mu_{k^*} \leq \tau$ it is clear that we can pick $k^* \in \{k_L, k_R\}$ and so let us assume w.l.g. that $k^* = k_L$.

In this case, we define μ' either:

- if $\Delta_{k_R} \leq 3\Delta_{\min}/2$, for all $k \in [K]$,

$$\mu'_k = \mu_k + 2\Delta_{\min}.$$

- if $\Delta_{k_R} \geq 3\Delta_{\min}/2$, for all $k \in [K]$,
$$\mu'_k = \mu_k + 5\Delta_{\min}/4.$$

(a) holds as we just translated vertically the concave means. Also (b) holds since we switched the sign of arm k^* by construction. (c) holds also since we precisely added at most $2\Delta_{\min}$ to the means. And finally for (d): we have for any $k \in [K]$ that $|\bar{\Delta}_k - \Delta'_k| \leq 2\Delta_{\min}$, so that
$$\Delta'_k \leq 3\Delta_k.$$
Moreover for all arms k above threshold, it is clear that $\Delta'_k \geq \Delta_k$. On the other hand, for any arm k below threshold and that are not next to an arm up threshold - i.e. not k_L or k_R - we have by concavity that
$$\tau - \mu_k \geq 3\Delta_{\min},$$
which implies
$$\tau - \mu'_k \geq \tau - \mu_k - 2\Delta_{\min} \geq \frac{\tau - \mu_k}{3},$$
i.e. $\Delta'_k \geq \Delta_k/3$. Finally for $\{k_L, k_R\}$: it is clear that $\Delta'_{k^*} \geq \Delta_{k^*}/4$ by construction so that $\Delta'_{k_L} \geq \Delta_{k_L}/4$. And also by construction:

- if $\Delta_{k_R} \leq 3\Delta_{\min}/2$, then $\Delta'_{k_R} \geq \Delta_{\min}/2 \geq \Delta_{k_R}/3$.
- if $\Delta_{k_R} \geq 3\Delta_{\min}/2$, then $\Delta'_{k_R} \geq \Delta_{k_R} - 5\Delta_{\min}/4 \geq \Delta_{k_R}/6$.

So that in both situations (d) holds.

Case 2: Arm above threshold, i.e. $\mu_{k^*} \geq \tau$. Note first that if k^* is the only arm above threshold, we simply set for any k
$$\mu'_k = \mu_k - 2\Delta_{\min},$$
and this satisfies the requirements (a)-(d). Assume now that this case does not hold, so that $k^* \in \{k_L + 1, k_R - 1\}$ and $k_L + 1 < k_R - 1$.We now again consider several cases. Note that in any case $k^* \in \{k_L + 1, k_R - 1\}$.

Sub-case 1: μ not too flat around the threshold. Assume first that $\Delta_{k_L+2} \wedge \Delta_{k_R-2} \geq 3\Delta_{\min}/2$. Assume w.l.o.g. that $k^* = k_L + 1$. In this sub-case we define μ' either as:

- if $\Delta_{k_R-1} \geq 5\Delta_{\min}/4$ set
$$\mu' = \mu - 9\Delta_{\min}/8,$$
- otherwise if $\Delta_{k_R-1} \leq 5\Delta_{\min}/4$ set
$$\mu' = \mu - 11\Delta_{\min}/8.$$

It is clear that (a) holds (vertical translation of a concave sequence), (b) holds (arm k^* changes sides of threshold) and (c) holds since we translate at most by $11\Delta_{\min}/8$. Now for (d): it is clear in both cases that $\Delta'_k \leq \Delta_k + 11\Delta_{\min}/8 \leq 3\Delta_k$. Moreover:

- if $\Delta_{k_R-1} \geq 5\Delta_{\min}/4$, then for all $k \neq k^*$, we have $\Delta'_k \geq \Delta_k - 9\Delta_{\min}/8 \geq \Delta_k/8$ - and also by definition $\Delta_{k^*} = \Delta_{k^*}/8$. And so (d) holds in this case.
- if $\Delta_{k_R-1} \leq 5\Delta_{\min}/4$ we have for all k such that $\mu_k \leq \tau$ that $\Delta'_k \geq \Delta_k$, and for any $k \in \{k_L + 2, ..., k_R - 2\}$ that $\Delta'_k \geq \Delta_k - 11\Delta_{\min}/8 \geq \Delta_k/8$ since for such k we have $\Delta_k \geq 3\Delta_{\min}/2$. Also $\Delta'_{k_L+1} \geq \Delta'_{k_R-1} \geq \Delta_{\min}/8 \geq \Delta_{k_R-1}/10 \geq \Delta_{k_L+1}/10$. And so (d) holds in this case.

Sub-case 2: μ quite flat around the threshold. Assume now that $\Delta_{k_L+2} \wedge \Delta_{k_R-2} \leq 3\Delta_{\min}/2$. Assume w.l.o.g. that $\Delta_{k_L+2} \leq 3\Delta_{\min}/2$ and set
$$\mu'_{k_L+1} = \mu_{k_L+1} - 9\Delta_{k_L+1}/8.$$
and for $k \neq k_L + 1$
$$\mu'_k = \mu_k - \Delta_{k_L+1}/2.$$
(b) holds since $\mu'_{k_L+1} \leq \tau \leq \mu_{k_L+1}$. Since $\Delta_{k_L+1} \leq \Delta_{k_L+2} \leq 3\Delta_{\min}/2$, we know that (c) and (d) hold. Finally note that
$$\mu_{k_L+1} - \mu_{k_L} \geq \mu'_{k_L+1} - \mu'_{k_L} = 3\Delta_{k_L+1}/8 + \Delta_k$$
$$\geq \mu'_{k_L+2} - \mu'_{k_L+1} + 5\Delta_{k_L+1}/8 = \mu'_{k_L+2} - \mu'_{k_L+1} \geq \mu_{k_L+2} - \mu_{k_L+1},$$

since $\mu'_{k_L+2} - \mu'_{k_L+1} \leq \Delta_{\min}/2$ - since $\Delta_{k_L+1} \leq \Delta_{k_L+2} \leq 3\Delta_{\min}/2$ - so that $\Delta_k \geq \Delta_{\min} \geq \mu'_{k_L+2} - \mu'_{k_L+1} + \Delta_{k_L+1}/4$. So (a) holds since for any $k \notin \{k_L+1, k_L+2\}$, we have $\mu_k - \mu_{k-1} = \mu'_k - \mu'_{k-1}$.

□

Proof of Theorem 5. Consider $\bar{\Delta} \in \Delta\mathcal{B}_c$ associated with the vector of means $(\mu_k)_{k\in[K]}$. We define $\underline{\nu}$ as the Gaussian bandit problem with these means, that is, $\nu_k = \mathcal{N}(\mu_k, \sigma^2)$ for all $k \in [K]$. Thanks to Lemma 16 there exists a vector of means $(\mu'_k)_{k\in[K]}$ that verifies the conditions of Lemma 16. We denote by $\underline{\nu}'$ the Gaussian bandit problem such that $\nu'_k = \mathcal{N}(\mu_k, \sigma^2)$ for all $k \in [K]$. Thanks to (a) and (d) we know that $\underline{\nu}' \in \mathcal{B}_c$. Thanks to (b) there exists $i \in [K]$ such that, for example, $\mu_i > \tau$ and $\mu'_a < \tau$. In particular we can lower bound the error by the probability to make a mistake in the prediction of the label of arm i

$$e_T^{\underline{\nu}} \geq \mathbb{P}_{\underline{\nu}}(\hat{Q}_i = -1) \qquad e_T^{\underline{\nu}'} \geq \mathbb{P}_{\underline{\nu}}(\hat{Q}_i = 1)\,.$$

We then conclude as in the proof of Theorem 3. We can assume that $\mathbb{P}_{\underline{\nu}}(\hat{Q}_i = -1) \leq 1/2$ otherwise the bound is trivially true. Thanks to (c), the chain rule, the contraction of the Kullback-Leibler divergence and (2), it holds

$$\begin{aligned} T\frac{9\bar{\Delta}_{\min}^2}{2\sigma^2} &\geq \mathrm{KL}(\mathbb{P}_{\underline{\nu}}^{I_T}, \mathbb{P}_{\underline{\nu}'}^{I_T}) \\ &\geq \mathrm{kl}\left(\mathbb{P}_{\underline{\nu}}(\hat{Q}_i = 1), \mathbb{P}_{\underline{\nu}'}(\hat{Q}_i = 1)\right) \\ &\geq \mathbb{P}_{\underline{\nu}}(\hat{Q}_i = 1)\log\left(\frac{1}{\mathbb{P}_{\underline{\nu}'}(\hat{Q}_i = 1)}\right) - \log(2)\,, \end{aligned}$$

where we denote by $\mathbb{P}_{\underline{\nu}}^{I_T}$ the probability distribution of the history I_T under the bandit problem $\underline{\nu}$. Thus, using that $\mathbb{P}_{\underline{\nu}}(\hat{Q}_i = 1) = 1 - \mathbb{P}_{\underline{\nu}}(\hat{Q}_i = -1) \geq 1/2$ we obtain

$$\mathbb{P}_{\underline{\nu}'}(\hat{Q}_i = 1) \geq \frac{1}{4}\exp\left(-9\frac{T\bar{\Delta}_{\min}^2}{\sigma^2}\right)\,.$$

Which allows us to conclude that

$$\max(e_T^{\underline{\nu}^+}, e_T^{\underline{\nu}^-}) \geq \frac{1}{4}\exp\left(-9\frac{T\bar{\Delta}_{\min}^2}{\sigma^2}\right)\,.$$

□

Proof of Theorem 6. We assume in the proof, without loss of generality, that

$$\Delta_{\min} \geq c_{\mathrm{con-min}}\sqrt{\frac{\sigma^2\log(K)}{T}}$$

with $c_{\mathrm{con-min}} = 8064$. Indeed, otherwise, the bound of Theorem 6 is trivially true.

The proof of Theorem 6 is structured in the following manner. In our *original* binary tree we assume there is at least one arm above threshold, the contrary case is dealt with separately, see Lemma 26. We wish to show that with high probability the `Grad-Explore` algorithm will add sufficient arms above threshold to the list S_{T_1} such that when we take it's median we are guaranteed to output an arm above threshold. At time t we say our algorithm makes a favourable decision if all sample means are well concentrated - that it with $\bar{\Delta}_{\min}$ of their true mean. It is important to note that for arms below threshold this also implies the estimated gradients are close to their true values. On such a favourable decision we show that the explore algorithm will make a step towards the subtree of nodes containing an arm above threshold, or remain inside if it is already in it. We also show that upon encountering an arm above threshold, on a good decision said arm is always added to S_{T_1}. Therefore if overall we can make sufficient proportion of favourable events we are guaranteed to have a sufficient number of arms above threshold in S_{T_1}. We then show that this favorable event holds with high probability. Once we have identified an arm above threshold the problem is essentially split into two monotone problems - see Remark 9, where the point the arms cross threshold on either side can be found by applying the `PD-DEC-MTB` and `PD-MTB` algorithms in opposite directions.

Step 1: Initial definitions and lemmas We thus assume first that there is an arm $k*$ such that $\mu_{k^*} > \tau$.

Definition 17. *We define the subtree $ST(v)$ of a node v recursively as follows: $v \in ST(v)$ and*

$$\forall\, q \in ST(v),\ L(q), R(q) \in ST(v)\ .$$

Definition 18. *A consecutive tree U with root $u_{\texttt{root}}$ is a set of nodes such that $u_{\texttt{root}} \in U$ and*

$$\forall v \in U : v \neq u_{\texttt{root}},\ P(v) \in U.$$

with the additional condition,

$$\texttt{root} \in U \Rightarrow u_{\texttt{root}} = \texttt{root}$$

where `root` *is the root of the entire binary tree.*

We define Z, the set of good nodes with at least an arm with a mean above the threshold,

$$Z := \{v : \exists j \in \{l, m, r\} : \mu_{v(j)} > \tau\}\ .$$

At a given time t note the following property of Z and v_t,

$$ST(v_t) \cap Z \neq \emptyset \Leftrightarrow k^* \in [v_t(l), v_t(r)]\,. \tag{11}$$

Proposition 19. *Z is a consecutive tree with root z_{root} the unique element $v \in Z$ such that $P(v) \notin Z$ if there exists at least one, otherwise $z_{\text{root}} = \texttt{root}$.*

Proof. First, if for all $v \in Z$ we have $P(v) \in Z$ then $\texttt{root} \in Z$ and Z is a consecutive tree with root $z_{\text{root}} = \texttt{root}$. Otherwise, consider $v \in Z$, such that $P(v) \notin Z$, there is at least one such node. We first prove that v is unique. As $v \in Z$ we know that

$$\exists j \in \{l, m, r\} : \mu_{v(j)} > \tau\ . \tag{12}$$

Now since $v(l), v(r) \in P(v)$ and $P(v) \notin Z$, it follows that, thanks to (12),

$$\forall k \in \{l, r\} : \mu_{v(k)} < \tau\ .$$

For node $q \neq v$ satisfying the same properties, assume that $v(m) < q(m)$ without loss of generality. With this assumption we have,

$$v(r) \leq v(m) \leq q(l) \leq q(m)\ ,$$

however this then implies $\mu_{q(l)} > \tau$ a contradiction. Hence $v = q$, and thus v is unique which implies $\forall q \in Z :\ q \neq v,\ P(q) \in Z$. □

At time t we define w_t as the node of maximum depth whose sub tree contains both v_t and Z. Formally, for $t \leq T_1$,

$$w_t := \underset{\{ST(w) \cap Z \neq \emptyset\ \&\ v_t \in ST(w)\}}{\arg\max} |w|\,. \tag{13}$$

Lemma 20. *The node w_t is unique.*

Proof. At time t consider, a node q_t which also satisfies (13). As $v_t \in ST(w_t)$ and $v_t \in ST(q_t)$ we can assume without loss of generality $q_t \in ST(w_t)$ with $|q_t| \geq |w_t|$. This then implies, from (13), that $|q_t| = |w_t|$ and as $q_t \in ST(w_t)$, we have $q_t = w_t$. □

For $t \leq T_1$ we define D_t as the distance from v_t to Z, it is taken as the length of the path running from v_t up to w_t and then down to an good node in Z. Formally, we have

$$D_t := |v_t| - |w_t| + (|z_{\texttt{root}}| - |w_t|)^+ .$$

Note the following properties of D_t and w_t,

$$ST(v_t) \cap Z \neq \emptyset \Rightarrow v_t = w_t \, , D_t = 0 \Rightarrow v_t = w_t \text{ and } w_t, v_t \in Z \, .$$

Define at time t the counter G_t, tracking the number of good arms in S_t,

$$G_t := \left| \left\{ k \in S_t : \ \mu_k > \tau \right\} \right| . \tag{14}$$

At time t we define the following favorable event where the sampled arms at time t a well concentrated around their means,

$$\xi_t := \{ \forall j \in \{l, l+1, m, m+1, r, r+1\}, \left| \hat{\mu}_{j,t} - \mu_{v_t(j)} \right| \leq \Delta_{\min} \}.$$

Step 2: Actions of the algorithm on all iterations After any execution of algorithm `Grad-Explore` note the following,

- for $t \leq T_1$, v_t and v_{t+1} are separated by at most one edge, i.e.

$$v_{t+1} \in \{ L(v_t), R(v_t), P(v_t) \} \, , \tag{15}$$

- for $t \leq T_1$,

$$|S_t| \leq |S_{t+1}| \leq |S_t| + 1 \, . \tag{16}$$

Lemma 21. *On execution of algorithm `Grad-Explore` for all $t \leq T_1$ we have the following,*

$$D_{t+1} \leq D_t + 1, \tag{17}$$
$$G_{t+1} \geq G_t \, . \tag{18}$$

Proof. As the algorithm moves at most 1 step per iteration, see (15), for $t \leq T_1$, it holds

$$||v_t| - |w_t|| \geq ||v_{t+1}| - |w_t|| - 1 \, .$$

Noting that,

$$\begin{aligned} D_t &= ||v_t| - |w_t|| + (|z_{\texttt{root}}| - |w_t|)^+ \\ &\geq ||v_{t+1}| - |w_t|| + (|z_{\texttt{root}}| - |w_t|)^+ - 1 \\ &\geq ||v_{t+1}| - |w_{t+1}|| + (|z_{\texttt{root}}| - |w_{t+1}|)^+ - 1 \\ &= D_{t+1} - 1 \, , \end{aligned}$$

where the third line comes from the definition of w_{t+1}, see (3), we obtain $D_{t+1} \leq D_t + 1$. By (16) we have, for $t \leq T_1$,

$$|S_t| \leq |S_{t+1}| \leq |S_t| + 1 \, ,$$

hence $G_{t+1} \geq G_t$. □

Step 3: Actions of the algorithm on ξ_t We first state several properties relating to the event ξ_t. Firstly for all t we have that under event ξ_t,

$$\forall k \in \{l, m, r\}, \ \text{sign}(\hat{\mu}_{k,t} - \tau) = \text{sign}(\mu_k - \tau) \, . \tag{19}$$

Since there is at least an arm above the threshold, due to the concave property, note the following,

$$\forall k \in [K] : \ \mu_k < \tau, \ |\mu_k - \mu_{k+1}| \geq 2\Delta_{\min} \, , \tag{20}$$

thus from (20) for all t under event ξ_t, we have that,

$$\forall j \in \{l, m, r\} : \mu_{v_t(j)} < \tau, \ \text{sign}(\hat{\nabla}_{j,t}) = \text{sign}(\nabla_{v_t(j)}) \, . \tag{21}$$

Lemma 22. *On execution of algorithm* `Grad-Explore` *for all $t \leq T_1$, on ξ_t, we have the following,*

$$D_{t+1} \leq \max(D_t - 1, 0) \,, \tag{22}$$
$$G_{t+1} \geq G_t + \mathbb{1}_{\{D_t = 0\}} \,. \tag{23}$$

Proof. We first prove (23). If $D_t = 0$ then we know $v_t \in Z$. If $v_t \in Z$ then under ξ_t there exists $j \in \{l, m, r\}$ such that $\hat{\mu}_{j,t} > \tau$, see (19), and arm is added to S_{t+1}, thus $G_{t+1} \geq G_t + \mathbb{1}_{\{D_t=0\}}$.

We now prove (22). We consider the following three cases:

- If $Z \cap ST(v_t) = \emptyset$. First of all we have that $\forall j \in \{l, m, r\} : \mu_{v_t(j)} \leq \tau$. Therefore from (19) the algorithm will not add an arm to S_t. Now, we have that $k^* \notin [v_t(l), v_t(r)]$, see (11), therefore via the concave property $\nabla_{v_t(l)} < 0$ or $\nabla_{v_t(r)} > 0$. Via (21) this implies that $\hat{\nabla}_{v_t(l)} < 0$ or $\hat{\nabla}_{v_t(r)} > 0$ respectively. Thus by action of the algorithm $v_{t+1} = P(v_t)$. Since in this case we are getting closer to the set of good nodes by going up in the tree we know that $w_t = w_{t+1}$. Thus thanks to Lemma 20, under ξ_t,

$$D_{t+1} = |v_{t+1}| - |w_{t+1}| + (|z_{\texttt{root}}| - |w_{t+1}|)^+ = |v_t| - 1 - |w_t| + (|z_{\texttt{root}}| - |w_t|)^+ = D_t - 1 \,.$$

- If $k^* \in ST(v_t)$ and $v_t \notin Z$. First of all we have that $\forall j \in \{l, m, r\} : \mu_{v_t(j)} \leq \tau$. Therefore from (19) the algorithm will not add an arm to S_t. Now note that in this case v_t can not be a leaf and we just need to go down in the subtree of v_t to find an good node, id est $w_t = v_t$. Since $v_t \notin Z$, without loss of generality, we can assume for example $\hat{\nabla}_{t,m} > 0$. From (21), under ξ_t, we then have that $\nabla_{v_t(m)} > 0$ which implies $k^* \in [v_t(l), v_t(m)]$. Hence algorithm `Grad-Explore` goes to the correct subtree, $v_{t+1} = L(v_t)$. In particular we also have for this node

$$k^* \in [v_{t+1}(l),\, v_t(m)] \,,$$

  therefore it holds again $w_{t+1} = v_{t+1}$. Thus combining the previous remarks we obtain thanks to Lemma 20, under ξ_t,

$$D_{t+1} = (|w_{t+1}| - |z_{\texttt{root}}|)^+ = (|w_t| - |z_{\texttt{root}}|)^+ - 1 = D_t - 1 \,.$$

- If $k^* \in ST(v_t)$ and $v_t \in Z$. In this case there exists $j \in \{l, m, r\}$ such that $\mu_{v_t(j)} > \tau$. From 19 we have for said j that, $\hat{\mu}_{j,t} > \tau$. Hence the algorithm will not move giving $v_t = v_{t+1}$ thus $D_t = D_{t+1} = 0$.

□

Step 4: Lower bound on G_{T_1+1} We denote by $\bar{\xi}_t$ the complement of ξ_t.

Lemma 23. *For any execution of algorithm* `Grad-Explore`*,*

$$G_{T_1+1} \geq \frac{3}{4} T_1 - 2 \sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} \,.$$

Proof. Combining (22) and (17) from Lemma 21 and Lemma 22 respectively we have

$$\begin{aligned} D_{t+1} &\leq D_t + \mathbb{1}_{\bar{\xi}_t} - \mathbb{1}_{\xi_t} \mathbb{1}_{\{D_t > 0\}} \\ &= D_t + 2\mathbb{1}_{\bar{\xi}_t} - 1 + \mathbb{1}_{\xi_t} \mathbb{1}_{\{D_t = 0\}} \,. \end{aligned}$$

Using this inequality with (23) we obtain

$$\begin{aligned} G_{T_1+1} &= \sum_{t=1}^{T_1} G_{t+1} - G_t \\ &\geq \sum_{t=1}^{T_1} \mathbb{1}_{\xi_t} \mathbb{1}_{\{D_t=0\}} \\ &\geq \sum_{t=1}^{T_1} \left(D_{t+1} - D_t - 2\mathbb{1}_{\bar{\xi}_t} + 1 \right) \\ &\geq T_1 - D_1 - 2\sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t,} \\ &\geq \frac{3}{4} T_1 - 2\sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t,} \,, \end{aligned}$$

where we used in the last inequality the fact that $D_1 \leq \log_2(K)$ and that $\log_2(K) \leq T_1/4$ by definition of T_1 . □

Lemma 24. *Upon execution of algorithm* `Grad-Explore` *with budget* $\frac{T}{3}$ *we have that,*

$$\mathbb{P}\left(\sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} \leq \frac{T_1}{8} \right) \leq \exp\left(-c_{\text{con}} \frac{T\bar{\Delta}_{\min}^2}{\sigma^2} + c'_{\text{con}} \log(K) \right) .$$

where $c_{\text{con}} = \frac{1}{576}$ *and* $c'_{\text{con}} = 12$.

Proof. The proof follows as in the proof of Lemma 15, with altered constants.

□

Lemma 25. *Under the assumption* $\exists k : \mu_k > \tau$*, upon execution of algorithm* `Grad-Explore` *with output* $\hat{k}$ *we have that* $\mu_{\hat{k}} \geq \tau$ *with probability greater than*

$$1 - \exp\left(-c_{\text{con}} \frac{T\bar{\Delta}_{\min}^2}{\sigma^2} + c'_{\text{con}} \log(K) \right) .$$

Proof. By combination of Lemmas 23 and 24 we have that $G_{T_1+1} \geq \frac{1}{2} T_1$ with probability greater than $1 - \exp\left(-c_{\text{con}} \frac{T\bar{\Delta}_{\min}^2}{\sigma^2} c'_{\text{con}} \log(K) \right)$. As $|S_{T_1+1}| \leq T_1$ and as the arms G_{T_1+1} form a segment (they are all above threshold) by taking the median of S_{T_1+1} under the circumstance $G_{T_1+1} \geq \frac{1}{2} T_1$ we have that the output of `Grad-Explore` $\hat{k}$ is such that $\mu_{\hat{k}} > \tau$. This then gives the result. □

With the following lemma we deal with the special case where all arms are below threshold before finally completing the proof of Theorem 6.

Lemma 26. *Under the assumption* $\forall k \in [K] : \mu_k < \tau$*, upon execution of algorithm* `Grad-Explore` *with output* $\hat{k}$ *we have that* $\forall k \in [K], \hat{Q}_k = -1$ *with probability greater than*

$$1 - \exp\left(-c_{\text{con}} \frac{T\bar{\Delta}_{\min}^2}{\sigma^2} + c'_{\text{con}} \log(K) \right) .$$

where $c_{\text{con}} = \frac{1}{576}$ *and* $c'_{\text{con}} = 12$.

Proof. Under the assumption $\forall k \in [K] : \mu_k < \tau$, for all $t < T_1$, we have that under the event ξ_t, $S_{t+1} = S_t$, see (19). Therefore the following holds,

$$|S_{T_1}| \leq \sum_{t=1}^{T_1} \mathbb{1}_{\bar{\xi}_t} \ .$$

The proof now follows from direct application of Lemma 24. □

We are now ready to prove Theorem 6.

Proof of Theorem 6. In the case where $\mu_k < \tau, \ \forall k \in [K]$ Lemma 26 immediately gives the result. Therefore we consider the case in which $\exists k \in [K] : \mu_k > \tau$. Under this assumption the algorithm `Grad-Explore` will return an arm $\hat{k} : \mu_{\hat{k}} > \tau$ with probability greater than

$$1 - \exp\left(-\frac{1}{576}\frac{T\bar{\Delta}_{\min}^2}{\sigma^2} + 12\log(K)\right) \ ,$$

see Lemma 25. In this case we have the sets of arms $[1, \hat{k}]$, $[\hat{k}, K]$ which satisfy the assumption described in Remark 9. Therefore via Theorem 4 and a union bound we have that with probability greater than

$$1 - 2\exp\left(-\frac{1}{48}\frac{T\Delta^2}{\sigma^2} + 12\log(K)\right)$$

we will correctly classify arms on both these sets. With an additional union bound we achieve the result. □

E. Experiments

We conduct some preliminary experiments to test the performance of both `PD-MTB` and `PD-CTB` to illustrate our theoretical understanding. As a bench mark we will use both a `Uniform` algorithm and also a naive binary search - that is without back tracking, that we will term `Naive`, for an exact description of both see Appendix. Note that `Naive` essentially behaves as a uniform sampling algorithm on a bandit problem with $\log(K)$ arms. As our theoretical bounds are likely far to loose in terms of constants we also include a parameter tuned version of the `PD-MTB` where we tune the constants in the definition of T_1 and T_2, see Equation (1).

We would expect the `Naive` algorithm to have an upper bound of the order $\exp\left(-\frac{T\bar{\Delta}_{\min}^2}{\log(K)}\right)$. This is sub-optimal compared to `PD-MTB` which removes the $\log(K)$, see Theorem 4. However, `PD-MTB` must divide it's budget across several arms at each round, while `Naive` algorithm samples only one. This may out weigh the benefit of backtracking when K is not very large.

In our experiments we consider two thresholding bandit problems. In Setting 1 the gap of one arm is set to Δ, with the remaining gaps very large - i.e. 100, In Setting 2 all gaps are set to Δ, for the `PD-CTB` we modify this to a concave setting where all arms are Delta apart. The former problem should more favour `PD-MTB` as it can quickly traverse the binary tree and expend most of it's budget on the leaf in question.

In Figure 2 we consider consider the expected error in Setting 1 as a function of the gap Δ and as a function of the number of arms K.The effect of varying Δ follows our intuition. Firstly all algorithms show an increased performance for greater Δ, this should be completely expected. Secondly, in Setting 1 the `PD-MTB` algorithm decrease in probability of error faster than Naive and much faster than `Uniform`. This is also unsurprising as in this setting the `Uniform`, and to a lesser extent `Naive`, algorithms are forced to waste an unnecessary amount of their budget on arms far from threshold. In the case of varied K, on the right, `PD-MTB` appears to outperform `Naive`, showing no obvious dependency on K past a certain point, however there is considerable noise.

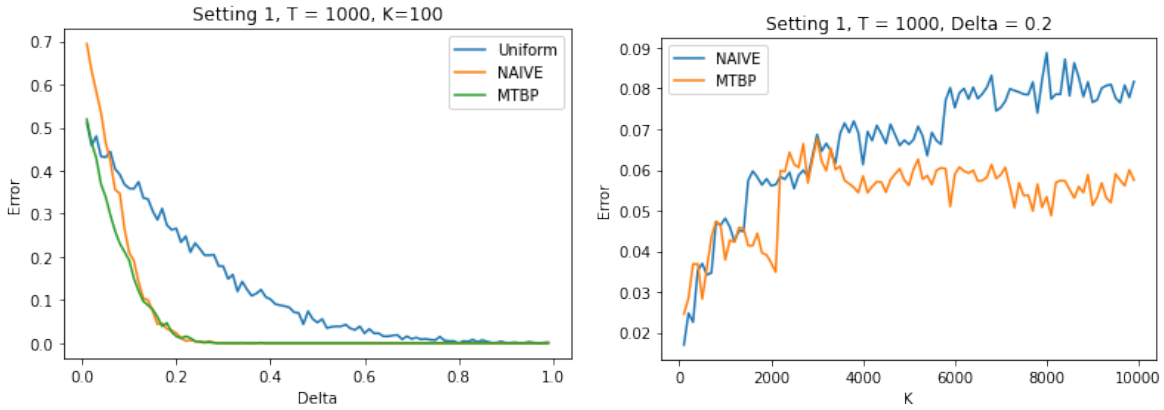

Figure 2. On the right: expected error as a function of the number of arms $K \in (100 \times i)_{i\in[100]}$ with $T = 1000$ and $\Delta = 0.2$ in Setting 1 averaged over 10000 Monte Carlo simulations. On the left: expected error as a function of the gap $\Delta \in (0.01 \times i)_{i\in[100]}$ with $T = 1000$ and $K = 100$ in setting 1 averaged over 1000 Monte Carlo simulations.

In figure 3 we consider consider the expected error in Setting 2 as a function of the gap Δ and as a function of the number of arms K. In both cases Naive out performs both PD−MTB and it's tuned version, vastly so for larger K. It would appear that here dividing our budget cancels out any gains one receives from reducing dependency on $\log(K)$. It is unfortunate that we were unable to find heuristic evidence of a lack of dependency on $\log K$, although this was perhaps expected. Based on our results, see Theorem 4, to remove such a dependency one would need $T\Delta^2 >> \log(K)$. This would lead to extremely low probabilities of error which are near impossible to detect accurately without huge numbers of Monte Carlo simulations, unfortunately beyond the scope of this paper.

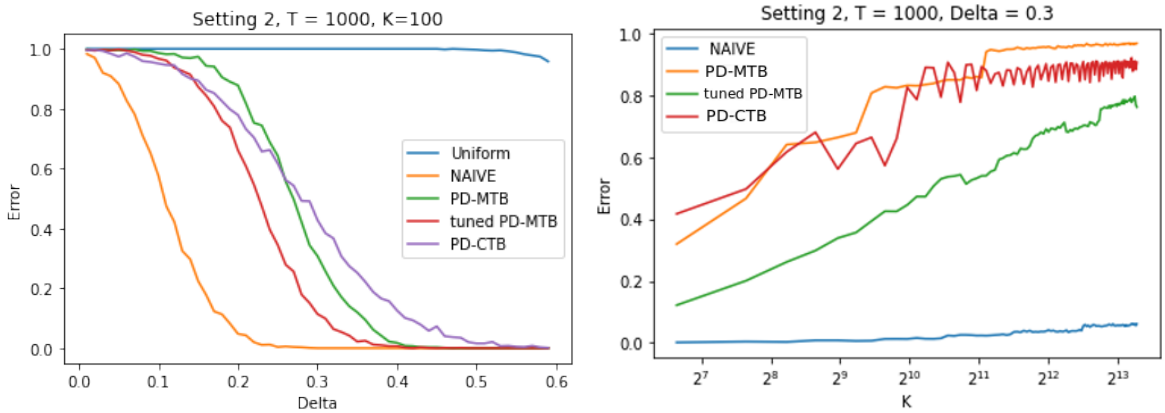

Figure 3. On the right: Expected error as a function of the number of arms $K \in (100 \times i)_{i\in[100]}$ with $T = 1000$, $\Delta = 0.3$, in Setting 2, plotted on a log scale averaged over 10000 Monte Carlo simulations. On the left: expected error as a function of the gap $\Delta \in (0.01 \times i)_{i\in[60]}$ with $K = 100$, $T = 1000$, in Setting 2, averaged over 1000 Monte Carlo simulations

Algorithm 4 `Naive`

Initialization: $v_1 = \texttt{root}$
for $t = 1 : T_1$ **do**
  sample $\lfloor \frac{T}{\log(K)} \rfloor$ times each arm in $v_t(m)$
  if $\hat{\mu}_{m,t} \leq \tau$ **then**
    $v_{t+1} = R(v_t)$
  else
    $v_{t+1} = L(v_t)$
  end if
end for
Set $\hat{k} = v_{T_1+1}(r)$
return $(\hat{k}, \hat{Q}) : \quad \hat{Q}_k = 2\mathbb{1}_{\{k \geq \hat{k}\}} - 1$

Algorithm 5 Uniform

for $k = 1 : K$ **do**
  Sample arm k a total of
  $\lfloor \frac{T}{K} \rfloor$ times.
  Compute $\hat{\mu}_k$ the sample mean of arm k.
end for
return

$$\hat{Q} : \quad \hat{Q}_k = \begin{cases} -1 & \text{if } \hat{\mu}_k < \tau \\ 1 & \text{if } \hat{\mu}_k \geq \tau \end{cases}$$