Bernoulli **21**(4), 2015, 2289–2307
DOI: 10.3150/14-BEJ644

Adaptive-treed bandits

ADAM D. BULL

Statistical Laboratory, Wilberforce Road, Cambridge CB3 0WB, UK. E-mail: a.bull@statslab.cam.ac.uk

We describe a novel algorithm for noisy global optimisation and continuum-armed bandits, with good convergence properties over any continuous reward function having finitely many polynomial maxima. Over such functions, our algorithm achieves square-root regret in bandits, and inverse-square-root error in optimisation, without prior information.

Our algorithm works by reducing these problems to tree-armed bandits, and we also provide new results in this setting. We show it is possible to adaptively combine multiple trees so as to minimise the regret, and also give near-matching lower bounds on the regret in terms of the zooming dimension.

Keywords: bandits on taxonomies; continuum-armed bandits; noisy global optimisation; tree-armed bandits; zooming dimension

1. Introduction

In *noisy global optimisation*, we wish to maximise a continuous function $\mu : X \to [0, 1]$ over a space $X = [0, 1]^p$, given only noisy observations of the function values $\mu(x)$. This problem arises in a wide variety of engineering applications, and has been considered by many authors (for example, see references in [11,13,19,21]).

To be precise, we suppose that at each time t, we choose a design point $x_t \in X$, and then observe a random variable $Y_t \in [0, 1]$ with mean $\mu(x_t)$, as in Figure 1. After T steps, our goal is to choose an estimated maximum $\hat{x}_T$ of μ, so as to minimise the *simple regret*,

$$S_T = \mu^* - \mu(\hat{x}_T), \tag{1}$$

where $\mu^* = \sup_{x \in X} \mu(x)$.

We would like to find a solution to this problem which achieves good rates of convergence, and can also be expected to provide good practical performance. We note that good convergence of S_T does not necessarily ensure good practical performance: for example, if μ is Lipschitz on $[0, 1]$, the optimal rate of $\tilde{\mathrm{O}}(T^{-1/3})$ can be achieved by a fixed choice of design points x_t; nonetheless, we can expect better practical performance from a choice which varies with the observations Y_t. (The result for a fixed design is given by [17]; the corresponding lower bound can be proved similarly to our Theorem 2.)

An alternative is to instead minimise the *cumulative regret*,

$$R_T = \sum_{t=1}^{T} \big(\mu^* - \mu(x_t)\big). \tag{2}$$

If an algorithm controls the cumulative regret at rate Tr_T, it can also control the simple regret at rate r_T [6]; bounding the cumulative regret is thus a stronger result. The advantage in bounding

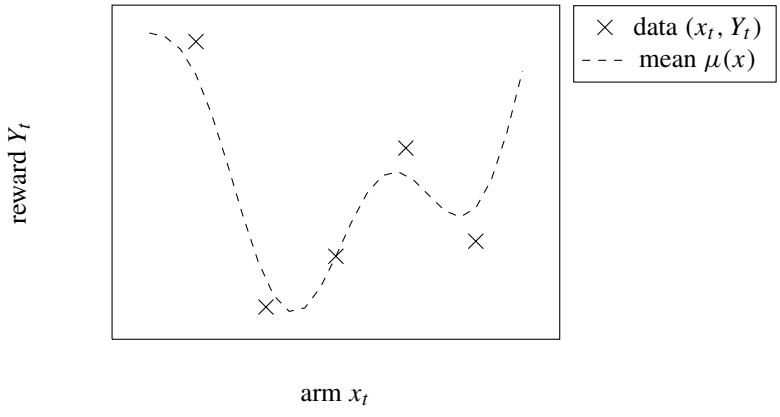

Figure 1. Noisy global optimisation: we choose design points x_t, observe data Y_t with mean $\mu(x_t)$, and wish to maximise μ.

R_T is that it also ensures our solution will place most of its design points in regions where μ is near-optimal; that few observations will be wasted.

We would thus expect algorithms which control the cumulative regret to offer improved practical performance. For example, in our Lipschitz model above, a fixed choice of design points must suffer $\Omega(T)$ cumulative regret; an algorithm which concentrates its design points in optimal regions of μ can simultaneously achieve the optimal rates of $\tilde{\mathrm{O}}(T^{2/3})$ cumulative regret, and $\tilde{\mathrm{O}}(T^{-1/3})$ simple regret [16].

In the following, we will therefore seek an algorithm for choosing the design points x_t which minimises the cumulative regret. Problems of this kind are known as *multi-armed bandits*; they can be thought of as attempting to optimally play an unknown slot machine (or 'bandit') with multiple arms.

The field of multi-armed bandits has a long history in the literature, and comprises many difficult problems even when the set X to optimise over is small and finite (see references in [5]). However, recent work has also focused on the specific problem of *continuum-armed bandits*, where $X = [0, 1]^p$, and we make some smoothness assumption on the reward μ; we discuss this work in more detail below.

Many solutions to this problem involve placing a tree structure over $[0, 1]^p$, for example as in Figure 2. The problem can thus also be thought of as lying within the more general field of *tree-armed bandits*, where the optimisation occurs over any set with a known tree structure. Such problems are of interest not only in noisy optimisation, but also in areas such as artificial intelligence and online services (see references in [12,20,23]).

In the following paper, we will describe a new algorithm for noisy global optimisation, which obtains good cumulative regret under fewer assumptions than previous results in the literature. As a consequence, we will also prove new results for continuum-armed and tree-armed bandits, which may be of wider interest.

We proceed by discussing previous work in more detail, before then outlining our contributions. The continuum-armed bandit problem was devised by Agrawal [1], and for Lipschitz

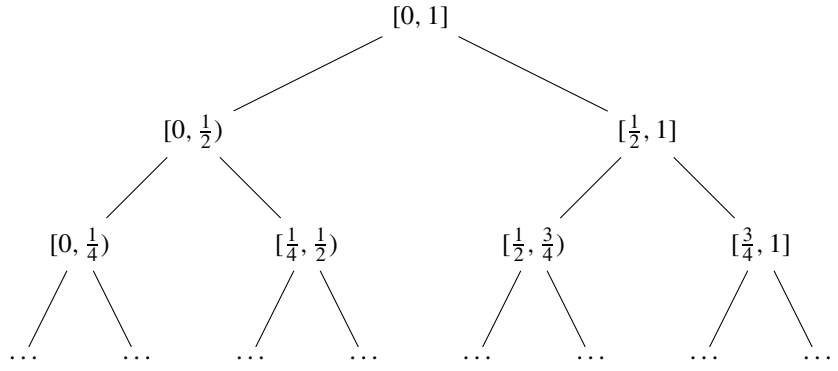

Figure 2. The dyadic tree over [0, 1].

reward functions μ, nearly tight bounds on the cumulative regret were first proved by Kleinberg [16]. Kleinberg applied the UCB1 strategy of Auer, Cesa-Bianchi and Fischer [2] to a simple fixed discretisation of the arm space [0, 1], achieving $\tilde{O}(T^{2/3})$ regret.

Independently, Cope [10] found it was possible to achieve $O(\sqrt{T})$ regret given stronger assumptions on μ: Cope showed this for the stochastic approximation algorithm of Kiefer and Wolfowitz [14], applied to unimodal reward functions μ. Auer, Ortner and Szepesvári [3] obtained similar bounds by extending the method of Kleinberg [16]: Auer, Ortner and Szepesvári obtained $\tilde{O}(\sqrt{T})$ regret over any reward function μ with, say, finitely many quadratic global maxima.

Kleinberg, Slivkins and Upfal [15] described a new 'zooming' algorithm, which used an adaptive discretisation of the arm space X, and could be applied whenever X was a metric space. For Lipschitz μ, Kleinberg, Slivkins and Upfal obtained regret like $\tilde{O}(T^{1-1/(\beta+2)})$, for a parameter $\beta \geq 0$ they called the *zooming dimension*, measuring the difficulty of the bandit problem.

Bubeck *et al.* [7] described a related algorithm, HOO, which could be applied whenever X had a known tree structure. Bubeck *et al.* again obtained $\tilde{O}(\sqrt{T})$ regret over μ with, say, finitely many quadratic global maxima, while also covering more general arm spaces and reward functions.

While the above results are significant, a shared weakness is that they all require some assumptions on the shape of the reward function μ. The strongest results, providing $\tilde{O}(\sqrt{T})$ regret, require us to assume say that μ has quadratic global maxima, as in the function

$$\mu(x) = -x^2. \tag{3}$$

However, if we make such an assumption, and then try to optimise a function with maxima of a different power, such as

$$\mu(x) = -x^4, \tag{4}$$

or of mixed powers, such as

$$\mu(x, y) = -x^2 - y^4, \tag{5}$$

we will achieve worse rates of regret.

Several authors have tried to improve upon this, constructing bandit algorithms which adapt to the shape of the reward function. Under further regularity assumptions, Bubeck, Stoltz and Yu [8] extended the algorithm of Kleinberg [16] to adapt to the Lipschitz constant. In a noiseless problem, for the simple regret, Munos [18] described an algorithm based on HOO, which adapts to a wide range of reward functions μ.

In this paper, we will build upon an approach described by Slivkins [20] for tree-armed bandits. Slivkins described an algorithm, TaxonomyZoom, which can adapt to a wide range of reward functions μ, if the arm space X is given by a finite tree.

Our first contribution is to extend the TaxonomyZoom algorithm to apply in noisy global optimisation and continuum-armed bandits. We modify the algorithm to apply directly to infinite arm spaces such as $[0, 1]^p$ (rather than using a fixed discretisation, which could harm convergence). We also give an explicit estimated maximum $\hat{x}_T$ (noting that while we could derive a naive choice as in [6], ours will be more reliable in practice), and fix a gap in the proofs of Slivkins.

Our second, more significant contribution is to give a construction of TaxonomyZoom which can adaptively vary the tree it optimises over. In previous work on bandits, optimisation has proceeded either over a fixed partition of the space X, or over partitions selected from a fixed tree. However, in order to get good convergence rates over functions such as (5), we will need to use trees which adapt to the data.

When $X = [0, 1]^p$, our algorithm constructs a tree by adaptively partitioning subsets of X along the axes; we will show that this procedure achieves optimal convergence rates for a wide variety of reward functions μ. While the motivation for our algorithm comes from continuum-armed bandits, our results will apply more generally in the tree-armed setting, where the tree can be constructed adaptively from any suitable collection of sub-trees.

Our third contribution is a lower bound on the convergence rate in tree-armed bandits, given in terms of the zooming dimension β. While this result forms part of our lower bound in the continuum-armed setting, such results have also been missing from previous literature on tree-armed bandits, and may thus be of wider interest.

Our final contribution is in the interpretation of our results in noisy global optimisation and continuum-armed bandits. To apply our algorithm in these settings, we will need to assume the reward function μ is sufficiently well-behaved; essentially, that it is continuous with finitely many polynomial maxima.

The precise condition we will require is that μ be what we call *zooming continuous*. This new condition generalises assumptions previously made for example in Auer, Ortner and Szepesvári [3] or Bubeck *et al.* [7], and gives a concise description of the reward functions μ over which we can achieve good cumulative regret.

When the reward function μ is zooming continuous, we will show that our algorithm obtains $\tilde{\mathrm{O}}(\sqrt{T})$ cumulative regret, and $\tilde{\mathrm{O}}(1/\sqrt{T})$ simple regret, with computation time $\tilde{\mathrm{O}}(T)$. While the constants in these rates will depend on μ, our algorithm will attain said rates without prior knowledge of the rewards.

We note that concurrently with this work, Valko, Carpentier and Munos [22] have described another adaptive algorithm which can be applied to continuum-armed bandits, based on the approach of Munos [18]. While their results bound only the simple regret, and do not adapt to reward functions like (5), their approach may be easier to generalise, and their results are complementary to ours.

In Section 2, we will discuss the continuum-armed bandit problem, and describe the class of zooming continuous reward functions. In Section 3, we will then describe our algorithm for tree-armed bandits, and state our results. Finally, in the supplemental article [9] we will give proofs.

2. Continuum-armed bandits

In this section, we describe our results on continuum-armed bandits; we begin with a precise definition of the multi-armed bandit problem. Suppose we have a measurable arm space $(X, \mathcal{E})$, and for each $x \in X$, some unknown distribution $P(x)$ over $[0, 1]$, with mean $\mu(x)$. At time t, we are allowed to choose an arm $x_t \in X$, and then receive a reward Y_t with distribution $P(x_t)$.

Formally, we take a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with random variables $Y_t \in [0, 1]$ and $Z_t \in Z$, $t \in \mathbb{N}$, for a measurable space $(Z, \mathcal{E}')$; the variables Z_t represent a source of randomisation. At time t, we require that Z_t is distributed independently of past events, x_t is an $\mathcal{E}$-measurable function of $Y_1, \dots, Y_{t-1}, Z_1, \dots, Z_t$, and Y_t has distribution $P(x_t)$, conditionally on past events and Z_t. A *strategy* for the multi-armed bandit problem is given by the functions x_t, and the distributions of the variables Z_t.

If our goal is to optimise μ, we can additionally return an estimated maximum $\hat{x}_T$, which we require to be an $\mathcal{E}$-measurable function of $Y_1, \dots, Y_T, Z_1, \dots, Z_T$, and an independent randomisation variable $\hat{Z}_T \in Z$. Our strategy then also includes the function $\hat{x}_T$, and the distribution of the variable $\hat{Z}_T$.

Finally, we define the cumulative regret R_T as in (2), and simple regret S_T as in (1). In the following, we will first consider the arm space $X = [0, 1]^p$; our goal will then be to find a strategy which makes the regrets R_T and S_T as small as possible, for a wide variety of reward functions μ.

The functions μ we consider will satisfy a new condition we call *zooming continuity*. Essentially, we will require that μ remains smooth as we 'zoom in' on its maxima; Figure 3 illustrates the concept.

As this zooming operation is a common part of algorithms for continuum-armed bandits, it is natural to require that when doing so, μ remains smooth. As such behaviour is neither necessary nor sufficient for membership in standard smoothness classes, we will thus require a new definition.

For any set $U \subseteq \mathbb{R}^p$, define its diameter along axis i,

$$\operatorname{diam}_i(U) = \sup\bigl\{|x_i - y_i| : x, y \in U\bigr\},$$

and its overall diameter,

$$\operatorname{diam}(U) = \sup\bigl\{\|x - y\| : x, y \in U\bigr\}.$$

Given also $x \in \mathbb{R}^d$, define its size, relative to U, to be

$$\|x\|_U^2 = \sum_{i=1}^{p} \left(\frac{|x_i|}{\operatorname{diam}_i(U)} \right)^2.$$

We then have the following definition.

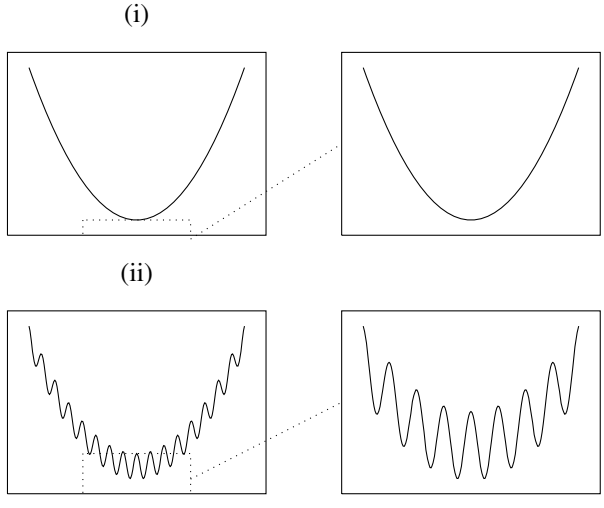

Figure 3. Function (i) remains smooth as we zoom in on its maxima; (ii) does not.

Definition 1. *Let $X \subset \mathbb{R}^p$ be a compact product of intervals. The function $f : X \to \mathbb{R}$ is* zooming continuous *if*:

(i) *f is continuous, with finitely many global maxima; and*
(ii) *for any global maximum x^* of f, and any neighbourhood U in X of x^*,*

$$\sup_{x^*, U:\mathrm{diam}(U)\leq\varepsilon} \frac{\sup_{x,y\in U:\|x-y\|_U\leq\varepsilon} |f(x) - f(y)|}{\sup_{z\in U} |f(x^*) - f(z)|} \to 0 \tag{6}$$

as $\varepsilon \to 0$.

We thus require that for any small neighbourhood U of a global maximum x^*, and any points $x, y \in U$ which are close relative to the size of U, the function f does not vary much between x and y, relative to its range over U. In other words, after 'zooming in' to f on the set U, f remains smooth.

We can show that many functions μ of interest are zooming continuous. Essentially, our definition includes any continuous function μ with finitely many maxima, each of which behaves like a suitable polynomial.

Proposition 1. *Let $X \subset \mathbb{R}^p$ be a compact product of intervals, and $f : X \to \mathbb{R}$ be continuous, with finitely many global maxima $x_1^*, \ldots, x_L^*$. For each maximum x_l^*, let f satisfy one of the following as $x \to x_l^*$.*

(i) *x_l^* is an elliptical maximum,*

$$f(x) = f\big(x_l^*\big) - \big\|A_l\big(x - x_l^*\big)\big\|^{\alpha_l}\big(1 + \mathrm{o}(1)\big),$$

for a positive-definite matrix $A_l \in \mathbb{R}^{p\times p}$, *and* $\alpha_l > 0$.

(ii) x_l^* *is a separable maximum*,

$$f(x) = f\left(x_l^*\right) - \left(\sum_{i=1}^{p} c_{l,i}\left|x_i - x_{l,i}^*\right|^{\alpha_{l,i}}\right)\left(1 + \mathrm{o}(1)\right),$$

for $c_{l,i}, \alpha_{l,i} > 0$.

Then f *is zooming continuous*.

The case of elliptical maxima includes all maxima where the function is locally a quadratic, since we may set $\alpha_l = 2$, and let A_l be the square root of the Hessian matrix. Alternatively, the case of separable maxima allows us to model functions which depend more strongly on some coordinates x_i than others.

We can thus check that zooming continuity includes functions with maxima like (3)–(5), as well as other combinations of powers. While the conditions of Lemma 1 thereby cover our motivating examples, in the following we will prefer to work directly with the more general and concise Definition 1.

When the reward function has such behaviour, the following result shows we can achieve good convergence rates for both the simple and cumulative regret. This result comes as a corollary to theorems in Section 3, where we describe a strategy for tree-armed bandits achieving such rates, and also provide a near-matching lower bound.

Corollary 1. *Let* $\varepsilon \in (0, 1)$. *There exists a strategy for continuum-armed bandits*, *depending only on* ε, *which achieves regret*

$$R_T = \tilde{\mathrm{O}}(\sqrt{T}), \qquad S_T = \tilde{\mathrm{O}}(1/\sqrt{T}),$$

on an event with probability $1 - \varepsilon$, *whenever the reward function* μ *is zooming continuous*. *Furthermore*, *on this event*, *the strategy has a total computation time of* $\tilde{\mathrm{O}}(T)$.

3. Tree-armed bandits

In this section, we will describe our results on the tree-armed bandit problem. In Section 3.1, we will give a definition of the problem, and in Section 3.2, describe the algorithm we will use to solve it. In Section 3.3, we will define a class of reward functions over which our algorithm performs well, and in Section 3.4, state our bounds on its regret and complexity.

3.1. Problem statement

In the tree-armed bandit problem, we again consider the multi-armed bandit problem described in Section 2, but now with a more general arm space X. We allow any space X on which we are

given a certain tree structure, which we define below; we will show that the continuum-armed bandit problem is a special-case of this more general setting.

To define our setting, let the arm space X be a Cartesian product $\prod_{i=1}^{p} X_i$, for coordinate spaces X_i. For $i = 1, \ldots, p$, let $\mathcal{T}_i$ be a tree with root node X_i, and whose nodes are all given by non-empty subsets of X_i. Further require that each node U is either a leaf node, or has children V which form a partition of the set U. Each non-leaf node must have at least 2 and at most q children, for a constant $q \in \mathbb{N}$.

Formally, we will also require a σ-algebra $\mathcal{E}$ on X, defined in terms of the trees $\mathcal{T}_i$. For each coordinate space X_i, let $\mathcal{E}_i$ be the sigma-algebra generated by the nodes U of $\mathcal{T}_i$. We then define $\mathcal{E}$ to be the product σ-algebra of the $\mathcal{E}_i$.

As before, we sequentially choose arms $x_t \in X$, and receive rewards $Y_t \in [0, 1]$; our goal remains to find a strategy minimising the regrets R_T and S_T, for a wide variety of reward functions μ. However, we must now do so for general treed spaces X, given only the trees $\mathcal{T}_i$.

Continuum-armed bandits lie within this setting, letting each coordinate space $X_i = [0, 1]$. The trees $\mathcal{T}_i$ can be chosen to be *dyadic trees* on $[0, 1]$, defined as follows. The dyadic tree on $[0, 1)$ is the tree with root node $[0, 1)$, and where each node $[a, b)$ has children $[a, \frac{1}{2}(a + b))$, $[\frac{1}{2}(a + b), b)$.

We can similarly define the dyadic tree on $[0, 1]$, instead allowing each node with upper bound 1 to contain the point 1; this tree is illustrated in Figure 2. If the trees $\mathcal{T}_i$ are dyadic trees on $[0, 1]$, then $\mathcal{E}$ is the Borel σ-algebra on $[0, 1]^p$, and we recover the setting of Section 2.

With these definitions, we can now consider continuum-armed bandits as a special-case of tree-armed bandits. In the following section, we will describe an algorithm for solving tree-armed bandits, which when applied to continuum-armed bandits, achieves the bounds in Corollary 1.

3.2. Adaptive-treed bandits

Our algorithm proceeds in much the same fashion as the TaxonomyZoom algorithm of Slivkins [20]. At time t, we partition the arm space X into a set $\mathcal{A}_{t-1}$ of *active boxes*, chosen in terms of the past rewards $Y_1, \ldots, Y_{t-1}$. For each box $B \in \mathcal{A}_{t-1}$, we compute an *index* $I_{t-1}(B) \in \mathbb{R}$, which upper bounds its maximum reward $\sup_{x \in B} \mu(x)$. We then select an active box B_t maximising the index I_{t-1}, and pull an arm x_t chosen uniformly at random from B_t.

To describe the algorithm in detail, we will need some additional definitions. We begin with the concepts which depend on the sample space X: the set of boxes $B \subseteq X$ we will use to construct our partitions, and the distribution π over X we will treat as uniform.

In the specific case of continuum-armed bandits, the boxes B will be the products of dyadic intervals in $[0, 1]^p$, and π will be the uniform distribution on $[0, 1]^p$. However, since our methods also apply to the more general tree-armed setting, we now give more general descriptions of these ideas.

We define a box B to be any product $\prod_{i=1}^{p} U_i$, where each U_i is a node in the tree $\mathcal{T}_i$; we further let $\mathcal{B}$ denote the set of all such boxes. For a fixed reward function $\mu : X \to [0, 1]$, we also define the width W of a box B to be

$$W(B) = \sup_{x \in B} \mu(x) - \inf_{x \in B} \mu(x).$$

We next define a distribution π on the measurable space $(X, \mathcal{E})$, given as the product of distributions π_i on the spaces $(X_i, \mathcal{E}_i)$. Intuitively, π_i will be the distribution of a point in X_i chosen by uniform random descent of the tree $\mathcal{T}_i$.

To be precise, we generate a random sequence of nodes U_n in $\mathcal{T}_i$, setting $U_1 = X_i$. For $n \in \mathbb{N}$, if U_n is a leaf node, we terminate the sequence at U_n; otherwise, we choose U_{n+1} uniformly at random from the children of U_n. We then define a distribution π_i on $(X_i, \mathcal{E}_i)$ by

$$\pi_i(U) = \mathbb{P}(\exists n \in \mathbb{N} : U_n = U), \qquad U \in \mathcal{T}_i. \tag{7}$$

It can be checked this uniquely defines a distribution π_i on $(X_i, \mathcal{E}_i)$.

We have thus defined the set $\mathcal{B}$ of boxes we will use to partition X, and the distribution π over X we will take as uniform. We note that for continuum-armed bandits, these definitions agree with those given above.

In the following, we will also wish to sample from π; in the case of continuum-armed bandits this is easy, as π is the uniform distribution. More generally, we will assume that π can be easily sampled from; note that we can always generate an approximate sample by random descent of the trees $\mathcal{T}_i$. Typically we will expect the σ-algebra $\mathcal{E}$ to be fine enough to define this sample to our satisfaction, but if not, we allow any sample satisfying (7).

We now move onto the definition of the index I_t. For each active box B, $I_t(B)$ will be based on the empirical mean of past rewards Y_s associated with arms x_s in B. To ensure this is an upper bound for the maximum reward over B, we will add two additional terms: one to correct for the stochastic error associated with estimating the mean reward, and one to bound the difference between mean and maximum.

Suppose that at time t, we select the active box B_t, drawing x_t from the distribution $\pi \mid B_t$. For any box B, we will say B was *hit* at time t if $x_t \in B \subseteq B_t$. Let $n_t(B)$ be the number of times $s \leq t$ at which B was hit, and if $n_t(B) > 0$, let $\mu_t(B)$ be the corresponding average reward. For fixed μ, we note that $\mu_t(B)$ is an unbiased estimate of

$$\mu(B) = \mathbb{E}_\pi\big[\mu(x) \mid x \in B\big],$$

the expected reward on B under π.

To bound the error in this estimate, we next define a confidence radius $r_t(B)$, chosen so that $|\mu_t(B) - \mu(B)| \leq r_t(B)$ with high probability. We first fix an error probability $\varepsilon \in (0, 1)$, which will control the accuracy of our bound; we will show that our results on the regret hold with probability $1 - \varepsilon$.

For any box $B = \prod_{i=1}^{p} U_i$, we then let $d(B)$ denote the *depth* of B, the maximum depth of any U_i in its corresponding tree $\mathcal{T}_i$, and define the constant

$$\rho(B) = q^{p(d(B)+1)}.$$

We also set $\tau = 4\varepsilon^{-1}$, and then define the confidence radius

$$r_t(B) = 2\sqrt{\log\big[\rho(B)\big(\tau + n_t(B)\big)\big]/n_t(B)}. \tag{8}$$

To conclude the definition of the index $I_t(B)$, we will need a term bounding the difference between the mean and maximum reward on B. This term will depend on a constant $\gamma \in (0, 1)$ called the *quality*, a concept we inherit from Slivkins [20].

The quality γ describes how difficult we expect the tree-armed bandit problem to be, and thus how conservatively our algorithm should act. In the following sections, we discuss the implications of γ in more detail. For now, we note that smaller γ corresponds to a more difficult problem, and more conservative behaviour.

Given a fixed choice of γ, we then define the index

$$I_t(B) = \mu_t(B) + (1 + 2p\nu)r_t(B),$$

where the constant

$$\nu = 8\sqrt{2\gamma^{-1}\log_2(2\gamma^{-1})};$$

if $n_t(B) = 0$, we take $I_t(B) = +\infty$. The index $I_t(B)$ is thus a sum of the empirical mean $\mu_t(B)$, the confidence radius $r_t(B)$, and an additional term $2p\nu r_t(B)$, which bounds the difference between the mean and maximum reward over B.

We next describe our set $\mathcal{A}_t$ of active boxes. Our goal will be to choose as few active boxes as possible, while still ensuring that for each active box B, the index $I_t(B)$ is an upper bound for the maximum reward over B. To do so, we will aim to select a set of active boxes B satisfying the inequality $W(B) \leq 2p\nu r_t(B)$; we will thus need to find estimates of the widths $W(B)$.

The estimates will work on the principle that, if the reward function μ is well-behaved, we will be able to find large enough sub-boxes $C_1, C_2 \subseteq B$ for which $\mu(C_1) - \mu(C_2) \approx W(B)$. Since we always have $\mu(C_1) - \mu(C_2) \leq W(B)$, we may thus estimate $W(B)$ by a suitable maximum of these differences, taken over many pairs C_1, C_2.

Since we will not have access to the means $\mu(C_k)$ themselves, we will need to bound them using the data. We therefore define the lower and upper bounds on the mean reward,

$$\underline{\mu}_t(B) = \mu_t(B) - r_t(B), \qquad \overline{\mu}_t(B) = \mu_t(B) + r_t(B).$$

We may then define our width estimate

$$W_t(B) = \max_{(C_1,C_2)\in\mathcal{M}(B)} \underline{\mu}_t(C_1) - \overline{\mu}_t(C_2).$$

The maximum is taken over the set $\mathcal{M}(B)$ of all pairs (C_1, C_2) of boxes $C_1, C_2 \subseteq B$, which for $k = 1, 2$ satisfy:

(i) $\pi(C_k \mid B) \geq \gamma$; and
(ii) for some $i = 1, \ldots, p$, we have $U_{i,k} \in \mathcal{T}_i$, and $U_j \in \mathcal{T}_j$, $j \neq i$, satisfying

$$C_k = U_1 \times \cdots \times U_{i-1} \times U_{i,k} \times U_{i+1} \times \cdots \times U_p. \tag{9}$$

In other words, $\mathcal{M}(B)$ contains all pairs (C_1, C_2) of boxes in B which are not much smaller than B, and agree except along one axis; one such pair is illustrated in Figure 4.

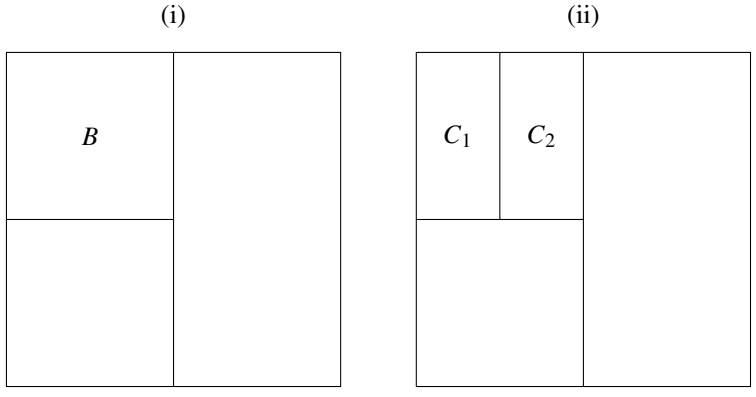

Figure 4. Plot (i) shows a partition $\mathcal{A}_t$ of the arm space $X = [0, 1]^2$ into active boxes; (ii) shows $\mathcal{A}_t$ after the box B has been split to maintain Invariant 1. The boxes C_1, C_2 satisfy condition (9).

Having defined our width estimates $W_t(B)$, we now return to the set $\mathcal{A}_t$ of active boxes. We first state that at the beginning of the algorithm, only the root box X is active: $\mathcal{A}_0 = \{X\}$. At later times t, we define each $\mathcal{A}_t$ in terms of $\mathcal{A}_{t-1}$, so as to maintain the following invariant.

Invariant 1. *Either*:

(i) $n_t(B) = 0$ *for some* $B \in \mathcal{A}_t$; *or*
(ii) $W_t(B) < \nu r_t(B)$ *for all* $B \in \mathcal{A}_t$.

We start by setting with $\mathcal{A}_t = \mathcal{A}_{t-1}$. Suppose this violates Invariant 1, so we have $W_t(B) \geq \nu r_t(B)$ for some box $B = \prod_{j=1}^{p} U_j$; then let $W_t(B)$ be maximised by boxes C_1, C_2 differing only along axis i. We remove B from $\mathcal{A}_t$, and replace it with the boxes $U_1 \times U_{i-1} \times V \times U_{i+1} \times U_p$, for all children V of U_i in $\mathcal{T}_i$.

We repeat this process until $\mathcal{A}_t$ satisfies Invariant 1; we note the process must terminate, as each step increases the number of active boxes B, without creating additional design points x_s. The process is illustrated in Figure 4.

We have thus described how we choose the set $\mathcal{A}_t$ of active boxes. Finally, we define our estimate $\hat{x}_T$ of a global maximum of μ; we set $\hat{x}_T = x_{T^*}$, where the optimal time

$$T^* = \operatorname*{arg\,min}_{t=1}^{T} r_t(B_t),$$

breaking ties arbitrarily.

We have then described in full our algorithm ATB, given by Algorithm 1. We note that our algorithm is closely related to the TaxonomyZoom algorithm of Slivkins [20]; we briefly describe the changes.

First, to allow us to work with infinite trees, we have altered the confidence radius $r_t(B)$ and constant ν. Second, to give an explicit algorithm for noisy global optimisation, we have included

Algorithm 1: Adaptive-treed bandits (ATB)

Data: space X, trees $\mathcal{T}_i$, error rate ε, quality γ
set $\mathcal{A}_0 = \{X\}$;
for $t = 1, \ldots, T$ **do**
  select a box $B_t \in \mathcal{A}_{t-1}$ maximising I_{t-1};
  play an arm x_t drawn at random from $\pi \mid B_t$;
  set $\mathcal{A}_t = \mathcal{A}_{t-1}$;
  while *Invariant 1 is violated, by* $B = \prod_{j=1}^{p} U_j$, *and* C_1, C_2 *differing along axis* i **do**
    remove B from $\mathcal{A}_t$;
    for V *a child of* U_i *in* $\mathcal{T}_i$ **do**
      add $U_1 \times \cdots \times U_{i-1} \times V \times U_{i+1} \times \cdots \times U_p$ to $\mathcal{A}_t$;
    end
  end
end
return x_{T^*}

a rule for choosing an optimal point $\hat{x}_T$. Third, we have altered Invariant 1 to allow an easier bound on the computational complexity.

Last, we have made a number of changes which allow us to work with multiple trees $\mathcal{T}_i$. The first of these is that we partition the arm space X into boxes $B \in \mathcal{B}$ given by a product of nodes in trees, rather than the nodes themselves. The second is that we have altered the width estimate $W_t(B)$ to require that the boxes C_1, C_2 agree except in one axis; this allows us to detect not only the width of a box B, but also an axis i along which it varies.

The final change is in the procedure for ensuring that Invariant 1 holds. When the invariant is violated by a box B, we split that box only along the axis i; this process allows us to adapt the shape of the active boxes B to the shape of the reward function μ.

3.3. Well-behaved rewards

We now describe the conditions we will require on the reward function μ. Our conditions will be motivated by Definition 1, and we will see that they hold in continuum-armed bandits whenever μ is zooming continuous. We will state the conditions more generally for the tree-armed case, however, as this allows us to both argue more directly, and also compare our conditions with those in previous work.

To begin, we will need some preliminary definitions. In the following, we will consider collections $\mathcal{C}$ of disjoint boxes $B \in \mathcal{B}$. We will say a box B is *on* $\mathcal{C}$, if it is a union of boxes in $\mathcal{C}$. We will further say $\mathcal{C}$ is a *refinement* of $\mathcal{C}'$, if this is true for all $B \in \mathcal{C}'$.

A specific type of collection $\mathcal{C}$ we will consider is the *grid*. A grid $\mathcal{G}$ is any set of boxes $\{\prod_{i=1}^{p} U_i : U_i \in \mathcal{S}_i\}$, where for each $i = 1, \ldots, p$, $\mathcal{S}_i$ is a collection of disjoint nodes in $\mathcal{T}_i$. We will say grids $\mathcal{G}_1, \ldots, \mathcal{G}_L$ are *separated*, if for any box B on $\bigcup_{l=1}^{L} \mathcal{G}_l$, B is on a single $\mathcal{G}_l$.

Finally, for a fixed reward function μ, given $\delta > 0$ we define the level set

$$\mathcal{X}_\delta = \{x \in X : \mu^* - \mu(x) \leq \delta\},$$

and for any box B, we define its maximum and average badness,

$$\delta(B) = \mu^* - \min_{x \in B} \mu(x), \qquad \Delta(B) = \mu^* - \mu(B). \tag{10}$$

We are now ready to state our conditions on μ.

Definition 2. *Let $\mu : X \to [0, 1]$ be $\mathcal{E}$-measurable. We will say μ is* well-behaved *if for each $m \in \mathbb{N}$, we have a partition $\mathcal{B}_m$ of X, made up of boxes $B \in \mathcal{B}$, and a subset $\mathcal{C}_m \subseteq \mathcal{B}_m$, satisfying the following.*

(i) *For each $m \in \mathbb{N}$, letting $\delta_m = 2^{1-m}$, the level set $\mathcal{X}_{\delta_m}$ is covered by $\mathcal{C}_m$.*
(ii) *Each $\mathcal{C}_m$ has cardinality at most $\kappa \delta_m^{-\beta}$, for constants $\kappa > 0$, $\beta \geq 0$.*
(iii) *For each $m \in \mathbb{N}$, the boxes $B \in \mathcal{C}_m$ satisfy*:
  (a) *$W(B) \leq \delta_m / 12p$, and*
  (b) *$d(B) \leq \lambda m$, for a constant $\lambda > 0$.*
(iv) *For each box B on some $\mathcal{C}_m$, there exist two sub-boxes $C_1, C_2 \subseteq B$ satisfying condition (9), with*:
  (a) *$\pi(C_k \mid B) \geq \gamma$, $k = 1, 2$, for a constant $\gamma \in (0, 1)$, and*
  (b) *$\mu(C_1) - \mu(C_2) \geq \frac{1}{p}(W(B) - \frac{1}{4}\delta(B))$.*
(v) *For each $m \in \mathbb{N}$, we have some $L_m \in \mathbb{N}$, and separated grids $\mathcal{G}_{1,m}, \ldots, \mathcal{G}_{L_m,m}$, such that $\mathcal{C}_m \subseteq \bigcup_{l=1}^{L_m} \mathcal{G}_{l,m}$.*
(vi) *Each $\mathcal{B}_{m+1}$ is a refinement of $\mathcal{B}_m$.*

We will call β the zooming dimension, *and γ the* quality.

We next discuss the implications of our definition, which is illustrated in Figure 5. Firstly, we note that the conditions are all satisfied when the reward function μ is zooming continuous.

Theorem 1. *Let the arm space $X = [0, 1]^p$, given as the product of coordinate spaces $X_i = [0, 1]$, $i = 1, \ldots, p$, with dyadic trees $\mathcal{T}_i$ over each X_i. If $\mu : X \to [0, 1]$ is zooming continuous, then μ is well-behaved, with zooming dimension $\beta = 0$.*

Second, we note that the conditions of Definition 2 are related to other conditions previously studied in the literature. The zooming dimension $\beta \geq 0$, and quality $\gamma \in (0, 1)$, are related to similar concepts defined by Kleinberg, Slivkins and Upfal [15] and Slivkins [20], and measure the difficulty of solving a bandit problem with reward function μ, when subdividing the arm space X using the trees $\mathcal{T}_i$.

We will discuss in more detail the meaning of these quantities below; for now, we note that they are a function both of the reward function μ, and the trees $\mathcal{T}_i$. In the following, we will assume that we have some natural choice of trees $\mathcal{T}_i$ we may treat as fixed, as is the case in continuum-armed bandits; we may thus consider these quantities primarily as a function of μ.

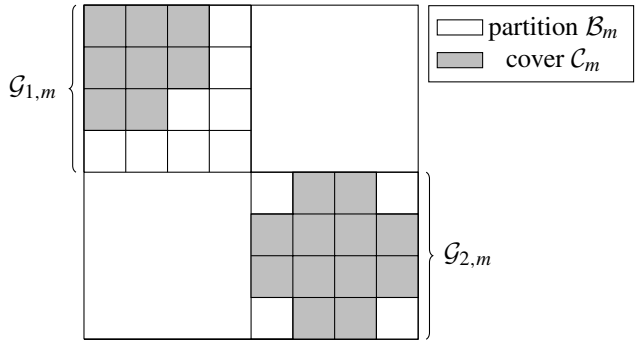

Figure 5. A partition $\mathcal{B}_m$ of the arm space $X = [0, 1]^2$, together with the cover $\mathcal{C}_m$, and grids $\mathcal{G}_{l,m}$.

Intuitively, conditions (i)–(iii)(a) state that μ has zooming dimension $\beta \geq 0$. This concept was introduced by Kleinberg, Slivkins and Upfal [15], and bounds the number of near-maximal boxes B we must evaluate to find the global maxima of μ. The larger β is, the more alternatives we must consider, and the worse our regret rates will be.

Kleinberg, Slivkins and Upfal [15] defined the zooming dimension relative to a fixed metric, with respect to which μ is assumed to be Lipschitz. Our formulation is more closely related to that of Slivkins [20], who did not fix a metric, but instead used the strongest metric which μ is Lipschitz with respect to.

Our condition improves upon that of Slivkins [20] by allowing the cover $\mathcal{C}_m$ to be made up of boxes $B \in \mathcal{B}$, constructed not just from a single tree $\mathcal{T}_i$, but also from arbitrary combinations of them. This flexibility allows us to ensure that a wider variety of reward functions μ will have zooming dimension $\beta = 0$; in particular, it is necessary to get near-optimal rates for the separable maxima in Lemma 1.

For the continuum-armed bandit problems we will consider, we will always have zooming dimension $\beta = 0$. However, in tree-armed bandits, we will also consider the case $\beta > 0$, as this allows our results to hold in more generality. In particular, we will prove near-matching lower bounds on the regret in terms of all $\beta \geq 0$.

Condition (iii)(b) controls the depth of near-maximal boxes B; assuming this condition allows us to construct an algorithm which is more computationally efficient. A similar approach is considered by Bubeck *et al.* [7], who discuss artificially truncating trees at a certain depth.

Intuitively, condition (iv) states that μ has quality $\gamma \in (0, 1)$. This concept was introduced by Slivkins [20], and bounds the difficulty in estimating the widths $W(B)$. Our version of this condition is new, and improves upon Slivkins' in two ways.

First, we require the bound to hold for a larger collection of boxes B; we will show this change allows us to fix a gap in the argument of Slivkins [20]. Second, we require the boxes C_1, C_2 to satisfy condition (9). In the case $p = 1$, when we have a single tree over X, this condition is trivial. However, when $p > 1$, it allows our algorithm to detect the axes along which μ varies, and so adaptively combine the trees $\mathcal{T}_i$.

Conditions (v) and (vi) are new to this work, and are also required to work with multiple trees efficiently. Again, when $p = 1$ the conditions can be shown to be trivially satisfiable; when $p > 1$, they will be necessary to prove our adaptive results.

Condition (v) requires that the near-maximal boxes B lie within a grid structure; that the boxes can be created by independent subdivisions of the axes X_i. This condition will be necessary to ensure that when our algorithm subdivides the axes, it does not create too many active boxes.

Condition (vi) requires that the near-maximal boxes $B \in \mathcal{C}_m$ become smaller as m increases; that they describe consistent regions of the arm space X as $\delta_m \to 0$. This condition will be necessary to ensure that as our algorithm progresses, the active boxes created at earlier time steps do not hinder us at later ones.

While the main motivation behind Definition 2 is our application to continuum-armed bandits, our results can also be applied to other tree-armed bandit problems, including those with finite trees. We note that while our definitions do not require it, it will be easiest to satisfy Definition 2 when all leaf nodes U_i in trees $\mathcal{T}_i$ are singleton sets, a condition which should be satisfied by any reasonable choice of trees $\mathcal{T}_i$.

3.4. Results for tree-armed bandits

We now give our regret bounds for tree-armed bandits. We will prove our results uniformly over a class of reward functions μ, which we describe below.

For an arm space X, given as the product of coordinate spaces X_i, $i = 1, \ldots, p$, each equipped with tree $\mathcal{T}_i$, a zooming dimension $\beta \geq 0$, a quality $\gamma \in (0, 1)$, and constants $\kappa, \lambda > 0$, let

$$\mathcal{P} = \mathcal{P}(X, \mathcal{T}, \beta, \gamma, \kappa, \lambda)$$

denote the class of arm distributions P whose reward functions μ are well-behaved, with the above constants. We note that the class $\mathcal{P}$ is increasing in the parameters β, κ and λ, and decreasing in γ.

We also fix some notation we will use to describe our rates of regret. Given functions $f, g : \mathbb{N} \to \mathbb{R}$ satisfying $f(T) = \mathrm{O}(g(T))$ as $T \to \infty$, we write $f(T) \lesssim g(T)$, and $g(T) \gtrsim f(T)$. If both $f(T) \lesssim g(T)$ and $f(T) \gtrsim g(T)$, we write $f(T) \approx g(T)$.

We now begin by establishing a lower bound on the regret any algorithm can achieve, in our setting of the tree-armed bandit problem. Our argument works by reducing to a finite arm space, and then applying a lower bound of Bubeck [4].

Theorem 2. *Suppose the trees $\mathcal{T}_i$ have no leaf nodes, and fix $\beta \geq 0$. For large enough $\kappa, \lambda > 0$, small enough $\gamma, \varepsilon \in (0, 1)$, and any strategy for tree-armed bandits, we have events E_T and E'_T, each of probability at least ε under some $P \in \mathcal{P}$, for which*

$$R_T \geq T r_T \qquad \text{on } E_T, \qquad S_T \geq r_T \qquad \text{on } E'_T,$$

for a rate

$$r_T \gtrsim T^{-1/(\beta+2)}.$$

This rate matches, up to log factors, the rates in upper bounds which have previously been proved, for example for the zooming algorithm of Kleinberg, Slivkins and Upfal [15], or the HOO algorithm of Bubeck *et al.* [7]. In the following, we will show that it also matches upper bounds for the adaptive algorithm described in this paper.

We begin by showing that, up to log factors, Algorithm 1 achieves the same rates, given only knowledge of the quality γ. We note that a similar result was stated by Slivkins [20], in the case of a single finite tree. In the following, we use a novel argument to fix a gap in the argument of Slivkins,[1] and also extend the result to multiple, infinite trees $\mathcal{T}_i$.

Theorem 3. *Fix $\varepsilon, \gamma \in (0,1)$. Running Algorithm 1 with error rate ε and quality γ, for any $\beta \geq 0$ and $\kappa, \lambda > 0$, we have events E_T, of probability at least $1-\varepsilon$ under any $P \in \mathcal{P}$, on which*

$$R_T \leq T r_T, \qquad S_T \leq r_T,$$

for a rate

$$r_T \lesssim \left(\frac{T}{\gamma^{-1} \log(\gamma^{-1}) \log(\varepsilon^{-1} + T) \log(T)^{1(\beta=0)}} \right)^{-1/(\beta+2)},$$

uniformly in γ and ε.

We have thus shown that Algorithm 1 achieves good rates of regret, without detailed knowledge of the reward function μ. Furthermore, the algorithm adapts to the shape of μ not only within a single tree $\mathcal{T}_i$, but also by combining the trees in whichever way minimises the zooming dimension β.

In the above theorem, Algorithm 1 still required a bound γ on the quality of μ. As a corollary, however, we can achieve similar rates of regret, up to say an additional log factor, without prior knowledge of μ.

Corollary 2. *Fix $\varepsilon \in (0,1)$. Running Algorithm 1 with error rate ε and quality $\log(T)^{-1}$, for any $\beta \geq 0$, $\kappa, \lambda > 0$ and $\gamma \in (0,1)$, we have events E_T, of probability at least $1-\varepsilon$ under any $P \in \mathcal{P}$, on which*

$$R_T \leq T r_T, \qquad S_T \leq r_T,$$

for a rate

$$r_T \lesssim \left(\frac{T}{\log(\varepsilon^{-1} + T) \log(T)^{1+1(\beta=0)} \log(\log(T))} \right)^{-1/(\beta+2)},$$

uniformly in ε.

We note that in the above construction, Algorithm 1 is no longer an anytime algorithm, as its quality parameter depends on the time horizon T. If an anytime algorithm is desired, one can

[1] The proof of Slivkins' Lemma 4.4(b) incorrectly assumes that all deactivated boxes have been selected.

be constructed using the doubling trick, as in Slivkins [20]; however, we need not consider this further.

We have thus shown that Algorithm 1 can achieve near-optimal rates of regret, for the optimal combination of trees $\mathcal{T}_i$, without prior knowledge of μ. We note that, together with Theorem 1, we can use this result to deduce the first part of Corollary 1, our result establishing good rates of regret in continuum-armed bandits.

It remains to discuss the implementation of our algorithm; we will show that, for a careful implementation, it can run in almost linear time. The key idea is to store the active boxes B in a priority queue, with priority given by their index $I_t(B)$. The operation of choosing a box B_t with maximal index can then be performed in constant time.

The remaining work lies in efficiently maintaining the set $\mathcal{A}_t$ of active boxes, and their indices I_t. We note that for active boxes B, the index $I_t(B)$, width estimate $W_t(B)$, and confidence radius $r_t(B)$ are changed only when we choose an arm $x_t \in B$. We thus need ensure only that these quantities can be updated efficiently when given a new data point.

To do so, we will keep some preliminary computations stored in memory. For each active box $B \in \mathcal{A}_t$, we store a list of the past data points (x_s, Y_s), $s \leq t$, for which $x_s \in B$. For each box $C \subseteq B$ satisfying $\pi(C \mid B) \geq \gamma$, we further store the number of hits $n_t(C)$, and average reward $\mu_t(C)$. After choosing an arm $x_t \in B$, we update these stored quantities to account for the new data point, and recompute the dependent quantities $I_t(B)$, $W_t(B)$ and $r_t(B)$.

In the event that we change the active set $\mathcal{A}_t$, any newly-stored quantities can be computed directly from the past data points (x_s, Y_s), $s \leq t$. With this procedure, we can then show that our algorithm runs in almost linear time.

Theorem 4. *On the event E_T, the computational complexity of Algorithm 1 is*:

(i) *in the setting of Theorem 3,*

$$\mathrm{O}\big(\gamma^{-(1+\log_2(p))} T \log(T)\big),$$

uniformly in γ and ε; and

(ii) *in the setting of Corollary 2,*

$$\mathrm{O}\big(T \log(T)^{2+\log_2(p)}\big),$$

uniformly in ε.

Finally, we note that together with Theorem 1, we can then deduce the second part of Corollary 1, our result establishing computational efficiency in continuum-armed bandits.

Acknowledgements

We would like to thank Richard Nickl, Alexandra Carpentier, and the anonymous referees for their valuable comments and suggestions, and EPSRC for their support under Grant EP/K000993/1.

Supplementary Material

Supplement to "Adaptive-treed bandits". (DOI: 10.3150/14-BEJ644SUPP; .pdf). We provide proofs of our results.

References

[1] Agrawal, R. (1995). The continuum-armed bandit problem. *SIAM J. Control Optim.* **33** 1926–1951. MR1358102

[2] Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* **47** 235–256.

[3] Auer, P., Ortner, R. and Szepesvári, C. (2007). Improved rates for the stochastic continuum-armed bandit problem. In *Learning Theory*. *Lecture Notes in Computer Science* **4539** 454–468. Berlin: Springer. MR2397605

[4] Bubeck, S. (2010). Jeux de bandits et fondations du clustering. Ph.D. thesis, Univ. Lille 1.

[5] Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Found. Trends Mach. Learn.* **5** 1–122.

[6] Bubeck, S., Munos, R. and Stoltz, G. (2009). Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*. *Lecture Notes in Computer Science* **5809** 23–37. Berlin: Springer. MR2564216

[7] Bubeck, S., Munos, R., Stoltz, G. and Szepesvári, C. (2011). $\mathcal{X}$-armed bandits. *J. Mach. Learn. Res.* **12** 1655–1695. MR2813150

[8] Bubeck, S., Stoltz, G. and Yu, J. (2011). Lipschitz bandits without the Lipschitz constant. In *Algorithmic Learning Theory* **22** 144–158. New York: Springer.

[9] Bull, A.D. (2014). Supplement to "Adaptive-treed bandits." DOI:10.3150/14-BEJ644SUPP.

[10] Cope, E.W. (2009). Regret and convergence bounds for a class of continuum-armed bandit problems. *IEEE Trans. Automat. Control* **54** 1243–1253. MR2532613

[11] Frazier, P., Powell, W. and Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS J. Comput.* **21** 599–613. MR2588343

[12] Gelly, S., Kocsis, L., Schoenauer, M., Sebag, M., Silver, D., Szepesvári, C. and Teytaud, O. (2012). The grand challenge of computer Go: Monte Carlo tree search and extensions. *Comm. ACM* **55** 106–113.

[13] Huang, D., Allen, T.T., Notz, W.I. and Miller, R.A. (2006). Sequential kriging optimization using multiple-fidelity evaluations. *Struct. Multidiscip. Optim.* **32** 369–382.

[14] Kiefer, J. and Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.* **23** 462–466. MR0050243

[15] Kleinberg, R., Slivkins, A. and Upfal, E. (2008). Multi-armed bandits in metric spaces. In *STOC'*08 681–690. New York: ACM. MR2582691

[16] Kleinberg, R.D. (2005). Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems* **17** 697–704. Cambridge, MA: MIT Press.

[17] Müller, H.-G. (1985). Kernel estimators of zeros and of location and size of extrema of regression functions. *Scand. J. Stat.* **12** 221–232. MR0817940

[18] Munos, R. (2011). Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *Advances in Neural Information Processing Systems* **24** 783–791. Cambridge, MA: MIT Press.

[19] Parsopoulos, K.E. and Vrahatis, M.N. (2002). Recent approaches to global optimization problems through particle swarm optimization. *Nat. Comput.* **1** 235–306. MR1999724

[20] Slivkins, A. (2011). Multi-armed bandits on implicit metric spaces. In *Advances in Neural Information Processing Systems* **24** 1602–1610. Cambridge, MA: MIT Press.

[21] Srinivas, N., Krause, A., Kakade, S.M. and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the* 27*th International Conference on Machine Learning* (*ICML*-10).

[22] Valko, M., Carpentier, A. and Munos, R. (2013). Stochastic simultaneous optimistic optimization. In *Proceedings of the* 30*th International Conference on Machine Learning* (*ICML*-13) 19–27.

[23] Yu, J.Y. and Mannor, S. (2011). Unimodal bandits. In *Proceedings of the* 28*th International Conference on Machine Learning* (*ICML*-11).

Received February 2013 and revised February 2014