Procrastinated Tree Search: Black-box Optimization with Delayed, Noisy, and Multi-fidelity Feedback

Junxiong Wang,[1] **Debabrota Basu,**[2] **Immanuel Trummer**[1]

[1] Dept. of Computer Science, Cornell University, Ithaca, NY, USA 14850
[2] Équipe Scool, Inria, UMR 9189 - CRIStAL, CNRS, Univ. Lille, Centrale Lille, Lille, France 59000

Abstract

In black-box optimization problems, we aim to maximize an unknown objective function, where the function is only accessible through feedbacks of an evaluation or simulation oracle. In real-life, the feedbacks of such oracles are often noisy and available after some unknown delay that may depend on the computation time of the oracle. Additionally, if the exact evaluations are expensive but coarse approximations are available at a lower cost, the feedbacks can have multi-fidelity. In order to address this problem, we propose a generic extension of hierarchical optimistic tree search (HOO), called ProCrastinated Tree Search (PCTS), that flexibly accommodates a delay and noise-tolerant bandit algorithm. We provide a generic proof technique to quantify regret of PCTS under delayed, noisy, and multi-fidelity feedbacks. Specifically, we derive regret bounds of PCTS enabled with delayed-UCB1 (DUCB1) and delayed-UCB-V (DUCBV) algorithms. Given a horizon T, PCTS retains the regret bound of non-delayed HOO for expected delay of $O(\log T)$ and worsens by $O(T^{\frac{1-\alpha}{d+2}})$ for expected delays of $O(T^{1-\alpha})$ for $\alpha \in (0, 1]$. We experimentally validate on multiple synthetic functions and hyperparameter tuning problems that PCTS outperforms the state-of-the-art black-box optimization methods for feedbacks with different noise levels, delays, and fidelity.

1 Introduction

Black-box optimization (Munos 2014; Sen, Kandasamy, and Shakkottai 2019), alternatively known as zeroth-order optimization (Xu et al. 2020) or continuous-arm multi-armed bandit (Bubeck et al. 2011), is a widely studied problem and has been successfully applied in reinforcement learning (Munos 2014; Grill et al. 2020), neural architecture search (Wang et al. 2019), large-scale database tuning (Pavlo et al. 2017; Wang, Trummer, and Basu 2021), robotics (Martinez-Cantin 2017), AutoML (Fischer, Gao, and Bernstein 2015), material science (Xue et al. 2016; Kajita, Kinjo, and Nishi 2020), and many other domains. In black-box optimization, we aim to maximize an unknown function $f : \mathcal{X} \to \mathbb{R}$, i.e. to find

$$x^* \triangleq \arg\max_{x \in \mathcal{X}} f(x). \tag{1}$$

In this setting, the optimizer does not have access to the derivatives of f, rather can access f only by sequentially querying a simulation or evaluation oracle (Jamieson, Nowak, and Recht 2012). The goal is to minimize the expected error in optimization, i.e. $\mathbb{E}[f(x^*) - f(x_T)]$, after T queries (Munos 2014), or to reach a fixed error threshold with as few queries as possible (Jamieson, Nowak, and Recht 2012). We adopt the first approach in this paper.

Approaches to Black-box Optimization. (Jamieson, Nowak, and Recht 2012) have shown that black-box optimization for convex functions is in general efficient. For convex functions, typically a Zeroth-order (ZO) Gradient Descent (GD) framework is used that replaces the gradient with the difference between functional evaluations (Jamieson, Nowak, and Recht 2012; Kumagai 2017; Liu et al. 2018). This approach requires double evaluation queries per-step and also multiple problem-specific hyperparameters to be tuned to obtain reasonable performance. Still, these methods are less robust to noise and stochastic delay (Li, Chen, and Giannakis 2019) than the next two other approaches, i.e. Bayesian Optimization (BO) and Hierarchical Tree Search.

For an objective function with no known structure except local smoothness, solving the black-box optimization problem is equivalent to estimating f almost everywhere in its domain $\mathcal{X}$ (Goldstein 1977). This can lead to an exponential complexity in the dimensionality of the domain (Chen 1988; Wang, Balakrishnan, and Singh 2019). Thus, one approach for this problem is to learn a surrogate $\hat{f}$ of the actual function f, such that $\hat{f}$ is a close approximation of f and $\hat{f}$ can be learned and optimized with fewer samples. This has led to research in Bayesian Optimization (BO) and its variants (Srinivas et al. 2010; Jones, Schonlau, and Welch 1998; Huang et al. 2006; Kandasamy et al. 2016), where specific surrogate regressors are fitted to the Bayesian posterior of f. However, if f is highly nonlinear or high dimensional, the Bayesian surrogate, namely Gaussian Process (GP) (Srinivas et al. 2010) or Bayesian Neural Network (BNN) (Springenberg et al. 2016), requires many samples to fit and generalize well. Also, there are two other issues. Firstly, often myopic acquisition used in BO algorithms leads to excessive exploration of the boundary of the search domain (Oh, Gavves, and Welling 2018). Secondly,

Table 1: Comparison of existing tree search, BO, and zeroth-order GD optimizers.

Algorithm	Expected Simple Regret	Delay	Noise	Fidelity	Assumptions
PCTS	$T^{-1/(d+2)}(\log T + \frac{\mathbb{E}[Delay]}{\sigma^2+2b})^{1/(d+2)}$	Stochastic	Unknown	Yes	Local Lip.
HOO (Bubeck et al. 2011)	$T^{-1/(d+2)}(\log T)^{1/(d+2)}$	x	Known	MF-HOO (Sen, Kandasamy, and Shakkottai 2019)	Local Lip.
GP-UCB (Srinivas et al. 2010)	$T^{-1/2}\,\mathrm{InfoGain}(T)$	x	Known	MF-GP-UCB (Kandasamy et al. 2016)	GP surrogate
GP-EI (Jones, Schonlau, and Welch 1998; Nguyen et al. 2019)	$T^{-1/2}O((\log T)^{d/2})$	x	Known	x	GP surrogate
DBGD (Li, Chen, and Giannakis 2019)	$\sqrt{T+\sum Delay}/T$	Bounded	x	x	Convex

the error bounds of BO algorithms include the information gain term ($\mathrm{InfoGain}(T)$) that often increases with T (Srinivas et al. 2010).

Instead of fixing on to such specific surrogate modelling, the alternative is to use *hierarchical tree search* methods which have drawn significant attention and success in the recent past (Munos 2014; Bubeck et al. 2011; Grill et al. 2015; Shang, Kaufmann, and Valko 2018, 2019; Sen, Kandasamy, and Shakkottai 2018, 2019).The tree search approach explores the space using a hierarchical binary tree with nodes representing subdomains of the function domain $\mathcal{X}$. Then, it leverages a bandit algorithm to balance the exploration of the domain and fast convergence towards the subdomains with optimal values of f. This approach does not demand more than local smoothness assumption with respect to the hierarchical partition (Shang, Kaufmann, and Valko 2018, Assumption 1) and an asymptotically consistent bandit algorithm (Bubeck et al. 2011). The generic nature of hierarchical tree search motivated us to extend it to black-box optimization with delayed, noisy, and multi-fidelity feedbacks.

Imperfect Oracle: Delay, Noise, and Multi-fidelity (DNF). In real-life, the feedbacks of the evaluation oracle can be received after a delay due to the computation time to complete the simulation or evaluation (Weinberger and Ordentlich 2002), or to complete the communication between servers (Agarwal and Duchi 2012; Sra et al. 2015). Such delayed feedback is natural in different optimization problems, including the white-box settings (Wang, Trummer, and Basu 2021; Li, Chen, and Giannakis 2019; Joulani, Gyorgy, and Szepesvári 2016; Langford, Smola, and Zinkevich 2009). In some other problems, introducing artificial delays while performing tree search, may create opportunities for work sharing between consecutive evaluations, thereby reducing computation time (Wang, Trummer, and Basu 2021). This motivated us to look into the delayed feedback for black-box optimization. Additionally, the feedbacks of the oracle can be noisy or even the objective function itself can be noisy, for example simulation oracles for physical processes (Kajita, Kinjo, and Nishi 2020) and evaluation oracles for hyperparameter tuning of classifiers (Sen, Kandasamy, and Shakkottai 2019) and computer systems (Fischer, Gao, and Bernstein 2015; Wang, Trummer, and Basu 2021). On the other hand, the oracle may invoke a multi-fidelity framework. Specially, if there is a fixed computational or time budget for the optimization, the optimizer may choose to access coarse but cheaper evaluations of f than the exact and costlier evaluations (Sen, Kandasamy, and Shakkottai 2018, 2019; Kandasamy et al. 2016). Both the noisy functions and multi-fidelity frameworks are studied separately in tree search regime while assuming a known upper bound on the noise variance (Sen, Kandasamy, and Shakkottai 2019) or known range of noise (Xu et al. 2020). We propose to extend tree search to a setting where all three imperfections, delay, noise, and multi-fidelity (*DNF*), are encountered concurrently. Additionally, we remove the requirement that the noise is either known or bounded.

Our Contributions. The main contributions of this paper are as follows:

1. *Algorithmic:* We show that the hierarchical tree search (HOO) framework is extendable to delayed, noisy and multi-fidelity (DNF) feedback through deployment of the upper confidence bounds of a bandit algorithm that is immune to the corresponding type of feedback. This reduces the tree search design problem to designing compatible bandit algorithms. In Section 3.1, we describe this generic framework, and refer to it as the *Procrastinated Tree Search* (PCTS).

2. *Theoretical:* We leverage the generalities of the regret analysis of tree search and incorporate delay and noise-tolerant bandit algorithms to show the expected simple regret bounds for expected delay $\tau = O(\log T)$ and $O(T^{1-\alpha})$ for $\alpha \in (0, 1)$. We instantiate the analysis for delayed versions of UCB1-σ (Auer, Cesa-bianchi, and Fischer 2002) and UCB-V (Audibert, Munos, and Szepesvári 2007). This requires analysing a delayed version of UCB1-σ and extending UCB-V to the delayed setting. We show that we have constant loss and $T^{(1-\alpha)/(d+2)}$ loss compared to non-delayed HOO in case of the two delay models (Sec. 3.2). We also extend the analysis to unknown noise variance (Sec. 3.3) and multi-fidelity (Sec. 3.4). To the best of our knowledge, we are the first to consider DNF-feedback in hierarchical tree search, and our regret bound is more general than the existing ones for black-box optimization with either delay or known noise or multi-fidelity (Table 1).

3. *Experimental:* We experimentally and comparatively evaluate performance of PCTS on multiple synthetic and real-world hyperparameter optimization problems against the state-of-the-art black-box optimization algorithms (Sec. 4)[1]. We evaluate for different delays, noise variances (known and unknown), and fidelities. In all the experiments, PCTS with delayed-UCB1-σ (DUCB1σ) and delayed-UCB-V (DUCBV) outperform the competing tree search, BO, and zeroth-order GD optimizers.

2 Background and Problem Formulation

We aim to maximize an objective function $f : \mathcal{X} \leftarrow \mathbb{R}$, where the domain $\mathcal{X} \subseteq \mathbb{R}^D$. At each iteration, the algorithm queries f at a chosen point $x_t \in \mathcal{X}$ and gets back an evaluation $y = f(x_t) + \epsilon$, such that $\mathbb{E}[\epsilon] = 0$ and $\mathbb{V}[\epsilon] = \sigma^2$ (Jamieson, Nowak, and Recht 2012). We consider both, the case where σ^2 is known and where it is un-

[1]Link to our code: https://github.com/OVSS/PCTS

known to the algorithm. We denote x^* as the optimum and $f^* \triangleq f(x^*)$ as the optimal value.

Structures of the Objective Function. In order to prove convergence of PCTS to the global optimum f^*, we need to assume that the domain $\mathcal{X}$ of f has at least a semi-metric ℓ defined on it (Munos 2014). This allows us to define an ℓ-ball of radius ρ with $\mathcal{B}_\rho \triangleq \{x | \max_y \ell(x,y) \leq \rho \ \forall \ x, y \in \mathcal{B}_\rho \subseteq \mathcal{X}\}$. Now, we aim to define the near-optimality dimension of the function f, given semi-metric ℓ. The near-optimality dimension quantifies the inherent complexity of globally optimising a function using tree search type algorithms. Near-optimality dimension quantifies the ϵ-dependent growth in the number of ℓ-balls needed to pack this set of ϵ-optimal states: $\mathcal{X}_\epsilon \triangleq \{x \in \mathcal{X} | f(x) \geq f^* - \epsilon\}$.

Definition 1 (c-near-optimality dimension (Bubeck et al. 2011))**.** *c-near-optimality dimension is the smallest $d \geq 0$, such that for all $\epsilon > 0$, the maximal number of disjoint ℓ-balls of radius $c\epsilon$ whose centers can be accommodated in $\mathcal{X}_\epsilon$ is $O(\epsilon^{-d})$.*

This is a joint property of f and the dissimilarity measure ℓ. d is independent of the algorithm of choice and can be defined for any f and $\mathcal{X}$ with semi-metric ℓ. Additionally, we need f to be smooth around the optimum x^*, i.e. to be weak Lipschitz continuous, for the tree search to converge.

Assumption 1 (Weak Lipschitzness of f (Bubeck et al. 2011))**.** *For all $x, y \in \mathcal{X}$, f satisfies $f^* - f(y) \leq f^* - f(x) + \max\{f^* - f(x), \ell(x,y)\}$.*

Weak Lipschitzness implies that there is no sudden drop or jump in f around the optimum x^*. Weak Lipschitzness can hold even for discontinuous functions. Thus, it widens applicability of hierarchical tree search methodology and corresponding analysis to more general performance metrics and domain spaces in comparison with algorithms that explicitly need gradients or smoothness in stricter forms. In Appendix, we show that we can relax this assumption proposed in Hierarchical Optimistic Optimization (HOO) (Bubeck et al. 2011) to more local assumptions like (Shang, Kaufmann, and Valko 2018). As we develop PCTS using the HOO framework, we keep this assumption here to directly compare the effects of DNF feedbacks.

Structure: Non-increasing Hierarchical Partition. The Hierarchical Tree Search or $\mathcal{X}$-armed bandit family of algorithms (Bubeck et al. 2011; Shang, Kaufmann, and Valko 2019; Sen, Kandasamy, and Shakkottai 2019) grow a tree $\mathcal{T} \subseteq \cup\{(h,l)\}_{h,l=0,1}^{H,2^h}$ of depth H, such that each node (h,l) represents a subdomain $\mathcal{X}_{(h,l)}$ of $\mathcal{X}$,[2] and the corresponding upper confidence intervals partition the domain of the performance metric f. Then, it uses a UCB-type bandit algorithm to assign optimistic upper confidence values to each partition. Using these values, it chooses a node to evaluate and expand at every time step. As the tree grows deeper, we obtain a more granular hierarchical partition of the domain. As we want the confidence intervals to shrink with increase in their depth, we need to ensure certain regularity of such hierarchical partition. Though we state the hierarchical partition as an assumption, it can be considered as an artifact of the tree search algorithm.

[2]Here, (h,l) represents the l-th node at depth h.

Assumption 2 (Hierarchical Partition with Decreasing Diameter and Shape (Munos 2014))**.**

1. Decreasing diameters. *There exists a decreasing sequence $\delta(h) > 0$ and constant $\nu_1 > 0$ such that* $\text{diam}(X_{h,l}) \triangleq \max_{x \in X_{h,l}} \ell(x_{h,l}, x) \leq \nu_1 \delta(h)$,*for any depth $h \geq 0$, for any interval $X_{h,l}$, and for all $i = 1, \ldots, 2^h$. For simplicity, we consider that $\delta(h) = \rho^h$ for $\rho \in (0,1)$.*

2. Regularity of the intervals. *There exists a constant $\nu_2 > 0$ such that for any depth $h \geq 0$, every interval $X_{h,l}$ contains at least a ball $\mathcal{B}_{h,l}$ of radius $\nu_2 \rho^h$ and center $x_{h,l}$ in it. Since the tree creates a partition at any given depth h, $\mathcal{B}_{h,l} \cap \mathcal{B}_{h,l'} = \emptyset$ for all $1 \leq l < l' \leq 2^h$.*

Simple Regret: Performance Metric. While analyzing iterative or sequential algorithms, regret $\text{Reg}_T \triangleq \sum_{t=1}^T [f(x^*) - f(x_t)]$ is widely used as the performance measure (Munos 2014). For optimization algorithms, another relevant performance metric is expected error or expected simple regret incurred at time T: $\epsilon_T = \mathbb{E}[r_T] = \mathbb{E}[f(x^*) - f(x_T)] = \frac{1}{T}\mathbb{E}[\text{Reg}_T]$. Since the last equality holds for tree search (Munos 2014), we state only the expected simple regret results in the main paper. The algorithm performance is better if the expected simple regret is lower. If the upper bound on expected simple regret grows sublinearly with horizon T, the corresponding algorithm asymptotically converges to the optimum. Given the aforementioned assumptions and definitions, and choosing simple regret as the performance measure, we state the expected error bound of HOO (Bubeck et al. 2011, Thm. 6) (using UCB1).

Theorem 1 (Regret of HOO)**.** *Assume that the expected objective function f satisfies Assumption 1, and its $4\nu_1/\nu_2$-near-optimality dimension is $d > 0$. Then, under Assumption 2 and for any $d' > d$, expected simple regret of HOO*

$$\epsilon_T = \mathbb{E}[r_T] = O\left(T^{-\frac{1}{d'+2}} (\log T)^{\frac{1}{d'+2}}\right) \tag{2}$$

for $T > 1$, and $4\nu_1/\nu_2$-near-optimality dimension d of f.

3 PCTS: Procrastinated Tree Search

In this section, we first provide a generic template for our framework. Following that, we incrementally show expected error bounds under delayed, noisy with known variance, noisy with unknown variance, and multi-fidelity feedbacks.

3.1 Algorithmic Framework

PCTS adapts the HOO algorithm (Bubeck et al. 2011) to delayed, noisy, and multi-fidelity feedbacks. We illustrate the pseudocode in Algorithm 1. Thus, in PCTS, we first assign optimistic $B^{\min}$ values to each node (h,l) in the hierarchical tree $\mathcal{T}_t$. Then, we incrementally select an 'optimistic path' from the root such that the path corresponds to one node at every depth and every chosen node has larger $B^{\min}$ value than its sibling nodes. Sibling nodes are the nodes that share the same parent. Following that, we sample a point x_t randomly from the subdomain $X_{(h_t,l_t)}$ that the leaf node (h_t, l_t) of the optimistic path represents. We expand this leaf

Algorithm 1: PCTS under DNF feedback and with a compatible BANDIT algorithm

1: **Input:** Total cost budget Λ, Bias function ζ, Cost function λ, Smoothness parameters (ν_1, ρ).
2: **Initialization:** $\mathcal{T}_1 = \{(0,0)\}$ (root), $B^{\min}_{(1,1)} = B^{\min}_{(1,2)} = \infty$, $t = 0$ (iteration), $C = 0$ (cost)
3: **while** $C \leq \Lambda$ **do**
4: Compute $B^{\min}$ values for each node in $\mathcal{T}_t$ using a UCB-type algorithm BANDIT(Eq. (3))
5: Select a leaf node (h_t, l_t) by following an "optimistic path" from root such that each selected node in the path has the highest $B^{\min}$ value among its sibling nodes
6: Sample a point x_t uniformly at random in the subdomain of node (h_t, l_t)
7: Query the evaluation oracle with x_t and at fidelity z_{h_t}
8: Observe the delayed and noisy feedbacks $\mathcal{O}_t \triangleq \{f_{s|t}(x_{h_s,l_s}|z_{h_s}) + \epsilon_s : s + \tau_s = t\}$ with the timestamps of invoking these queries $\{s : s + \tau_s = t\}$
9: Expand node (h_t, l_t) and add its children to $\mathcal{T}_t$ to form $\mathcal{T}_{t+1}$
10: **end while**

node and add its children to $\mathcal{T}_{t+1}$. Lines 4-6 and 9 essentially come from the HOO algorithm. The difference is in mainly three steps. In Line 7, we query the evaluation oracle with the point x_t and fidelity z_{h_t} due to multi-fidelity evaluator. In Line 8, we observe a delayed set of noisy feedbacks $\mathcal{O}_t \triangleq \{f_{s|t}(x_{h_s,l_s}|z_{h_s}) + \epsilon_s : s + \tau_s = t\}$ that arrives with corresponding timestamps when the queries were invoked. Here, $f_{s|t}(x_{h_s,l_s}|z_{h_s})$ is the multi-fidelity feedback lower bounded by $f(x_{h_s,l_s}) - \zeta(z_{h_s})$, and ϵ_s is a noise with zero mean and bounded variance. Such DNF feedback constrains us to use an asymptotically optimal bandit algorithm, BANDIT (Line 4), that allows us to get an upper confidence bound $B_{(h,l),s,t}$, which would be immune to DNF. In the following sections, we incrementally design such BANDIT confidence bounds and derive corresponding error bounds for PCTS. Though we describe the algorithm and the analysis for given smoothness parameters (ν_1, ρ), we describe in Appendix B the details of how to extend PCTS to unknown smoothness parameters.

3.2 Adapting to Delayed Feedbacks

Observable Stochastic Delay Model. We consider the stochastic delay setting (Joulani, Gyorgy, and Szepesvári 2013, 2016). This means that the feedback $f(x_s)$ of the evaluation oracle invoked at time $s \in [0, T]$ arrives with a delay $\tau_s \in \mathbb{R}^{\geq 0}$, such that $\{\tau_s\}_{s=0}^{T}$ are random variables invoked by an underlying but unknown stochastic process $\mathcal{D}$. Here, the delays are independent of the algorithm's actions.

Assumption 3 (Bounded Mean Delay)**.** *Delays are generated i.i.d from an unknown delay distribution $\mathcal{D}$. The expectation of delays $\tau \triangleq \mathbb{E}[\tau_s : s \geq 0]$ is bounded and observable to the algorithm.*

We observe that constant or deterministic delay with $\tau_{const} < \infty$ is a special case of this delay model.

From PCTS to Delayed Bandits. Due to the delayed setting, we observe feedback of a query invoked at time s at time $t \geq s$. Let us denote such feedback as $f_{s|t}(x_s)$. Thus, at time t, PCTS does not have access to all the invoked queries but a delayed subset of it: $\cup_{t'=1}^{t} \mathcal{O}_{t'} = \cup_{t=1}^{t} \{f_{s|t'}(x_s) : s + \tau_s = t'\}$. At time t, PCTS uses $\mathcal{O}_t$ to decide which node to extend next. Thus, making PCTS immune to unknown stochastic delays reduces to deployment of a BANDIT algorithm that can handle such stochastic delayed feedback.

Multi-armed bandits with delayed feedback is an active research area (Eick 1988; Joulani, Gyorgy, and Szepesvári 2013; Vernade, Cappé, and Perchet 2017; Gael et al. 2020; Pike-Burke et al. 2018), where researchers have incrementally studied the known constant delay, the unknown observable stochastic delay, and the unknown anonymous stochastic delay settings. In this paper, we operate in the second setting, where a delayed feedback comes with the timestamp of the query. Under delayed feedback, designing an UCB-type bandit algorithm requires defining an optimistic confidence interval around the expected value of a given node i that will consider both $T_i(t)$ and $S_i(t)$. $T_i(t)$ and $S_i(t)$ are the number of times a node i is evaluated and the number of evaluation feedbacks observed until time t.

Given such delayed statistics, any UCB-like bandit algorithm computes $B_{i,s,t}$, i.e. the optimistic upper confidence bound for action i at time t (Table 2), and chooses the one with maximum $B_{i,s,t}$:

$$i_t = \arg\max_{i \in \mathcal{A}} B_{i,s,t}.$$

We show three such confidence bounds in Table 2. Here, $\hat{\mu}_{i,s}$, $\hat{\sigma}^2_{i,s}$, and σ^2 are sample mean, sample variance, and predefined variance respectively. For the non-delayed setting, $s = T_i(t-1)$, and for delayed setting, $s = S_i(t-1)$. Representing the modifications of UCB1 (Auer, Cesa-bianchi, and Fischer 2002), UCB1-σ (Auer, Cesa-bianchi, and Fischer 2002), and UCB-V (Audibert, Munos, and Szepesvári 2007) in such a general form allows us to extend them for delayed settings and incorporate them for node selection in PCTS. Thus, given an aforementioned UCB-like optimistic bandit algorithm, the leaf node (h_t, l_t) selected by PCTS at time t is

$$\begin{aligned}(h_t, l_t) &\triangleq \arg\max_{(h,l) \in \mathcal{T}_t} B^{\min}_{(h,l)}(t) \\ &\triangleq \arg\max_{(h,l) \in \mathcal{T}_t} \min\{B_{(h,l),s,t} + \nu_1 \rho^h, \max_{(h',l') \in C(h,l)} B^{\min}_{(h,l)}(t)\}.\end{aligned} \tag{3}$$

Here, $\mathcal{T}_t$ is the tree constructed at time t, and $C(h,l)$ is the set of children nodes of the node (h,l). Equation (3) is as same as that of HOO except that $B_{(h,l),s,t}$ is replaced by bounds in Table 2. Under these modified confidence bounds for delays, we derive the bound on expected regret of PCTS + DUCB1 that extends the regret analysis of bandits with delayed feedback to HOO (Munos 2014).

Theorem 2 (Regret of PCTS + DUCB1 under Stochastic Delays)**.** *Under the same assumptions as Theorem 1 and*

Table 2: Confidence Bounds for different bandit algorithms with delayed/non-delayed feedback.

BANDIT	DUCB1 (Joulani, Gyorgy, and Szepesvári 2016)	DUCB1σ	DUCBV
$B_{i,s,t}$	$\hat{\mu}_{i,s} + \sqrt{\frac{2\log t}{s}}$	$\hat{\mu}_{i,s} + \sqrt{\frac{2\sigma^2\log t}{s}}$	$\hat{\mu}_{i,s} + \sqrt{\frac{2\hat{\sigma}^2_{i,s}\log t}{s}} + \frac{3b\log t}{s}$

upper bound on expected delay τ, PCTS using Delayed-UCB1 (DUCB1) achieves expected simple regret

$$\epsilon_T = O\left(\left(\frac{\ln T}{T}\right)^{\frac{1}{d'+2}}\left(1+\frac{\tau}{\ln T}\right)^{\frac{1}{d'+2}}\right). \quad (4)$$

Corollary 1 (Regret of PCTS + DUCB1 under Constant Delay)**.** *If the assumptions of Theorem 1 hold, and the delay is constant, i.e. $\tau_{const} > 0$, the expected simple regret of PCTS + DUCB1 is*

$$\epsilon_T = O\left(\left(\frac{\ln T}{T}\right)^{\frac{1}{d'+2}}\left(1+\frac{\tau_{const}}{\ln T}\right)^{\frac{1}{d+2}}\right). \quad (5)$$

Consequences of Theorem 2. The bound of Theorem 2 provides us with a few interesting insights.

1. *Degradation due to delay:* we observe that the expected error of PCTS + DUCB1 worsens by a factor of $\left(1+\frac{\tau}{\ln T}\right)^{\frac{1}{d+2}}$ compared to HOO, which uses the non-delayed UCB1 (Auer, Cesa-bianchi, and Fischer 2002). This is still significantly better than the other global optimization algorithm that can handle delay, such as Delayed Bandit Gradient Descent (DBGD) (Li, Chen, and Giannakis 2019) that achieves expected error bound $\sqrt{\frac{1}{T}+\frac{D}{T}}$. Here D is the total delay. Also, appearance of delay as an additive term in our analysis resonates with the proven results in bandits with delayed feedback,where an additive term appears due to delay. For $d = 0$, our bound matches in terms of T and τ with the problem-independent lower bound of bandits with finite K-arms and constant delay, i.e. $\sqrt{(K/T + \tau/T)}$ (Cesa-Bianchi et al. 2016, Cor. 11), up to logarithmic factors.

2. *Wait-and-act vs.* PCTS + DUCB1. A naïve way to handle *known constant delay* is to wait for the next τ_{const} time steps and to collect all the feedbacks in that interval to update the algorithm. In that case, the effective horizon becomes $\frac{T}{\tau_{const}}$. Thus, the corresponding error bound will be $O\left(T^{-\frac{1}{d'+2}}\left(\tau_{const}\ln T\right)^{\frac{1}{d+2}}\right)$. This is still higher than our error bound in Equation 5 for *unknown constant delay* $\tau_{const} > 1$ and $T \geq 3$.

3. *Deeper trees.* While proving Theorem 2, we observe that the depth $H > 0$ achieved by PCTS + DUCB1 till time T is such that $\rho^{-H(d'+2)} \geq \frac{T}{\tau+\ln T}$. This implies that for a fixed horizon T, the achieved depth should be $H \geq \frac{1}{d'+2}\frac{\tau+\ln T}{\ln(1/\rho)} = \Omega(\tau+\ln T)$. In contrast, HOO grows a tree of depth $H = \Omega(\ln(T/\tau))$. This shows that PCTS + DUCB1 constructs a deeper tree than HOO.

4. *Benign and adversarial delays.* If the expected delay is $O(\ln T)$, the expected simple regret is practically of the same order as that of non-delayed feedbacks. Thus, in cases of applications where introducing artificial delays helps in improving the computational cost (Wang, Trummer, and Basu 2021), we can tune the delays to $O(\ln T)$ for a given horizon T without harming the accuracy. We refer to this range of delays as *benign delay*. In contrast, one can consider delay distributions that have tails with α-polynomial decay, i.e. the expected delay is $O(T^{1-\alpha})$ for $\alpha \in (0, 1)$. In that case, the expected error is at least $\tilde{O}(T^{-\frac{\alpha}{d+2}})$. Thus, it worsens the HOO bound by a factor of $T^{\frac{1-\alpha}{d+2}}$. This observation in error bound resonates with the impossibility result of (Gael et al. 2020) that, in case of delays with α-polynomial tails, a delayed bandit algorithm cannot achieve total expected regret lower than $(T^{1-\alpha})$. Thus, it is unexpected that any hierarchical tree search with such delays achieves expected error better than $O(T^{-\frac{\alpha}{d+2}})$.

3.3 Adapting to Delayed and Noisy Feedback

Typically, when we evaluate the objective function at any time step t, we obtain a noisy version of the function as feedback such that $\tilde{f}(X_t) = f(X_t) + \epsilon_t$. Here, ϵ_t is a noise sample independently generated from a noise distribution $\mathcal{N}$ with mean 0. Till now, we did not explicitly consider the noise for the action selection step. In this section, we provide analysis for both known and unknown variance cases. In both cases, we assume that the noise has bounded variance σ^2, i.e. sub-Gaussian. In general, this assumption can be imposed in the present setup as any noisy evaluation can be clipped in the range of the evaluations where we optimize the objective function. It is known that a bounded random variable is sub-Gaussian with bounded mean and variance.

Case 1: Known Variance. Let us assume that the variance of the noise is known, say $\sigma^2 > 0$. In this case, the optimistic B-values can be computed using a simple variant of delayed-UCB1, i.e. delayed-UCB-σ (DUCB1σ), where

$$B_{(h,i),S_{(h,i)}(t-1),t} \triangleq \hat{\mu}_{(h,i),S_{(h,i)}(t-1)} + \sqrt{\frac{2\sigma^2\log t}{S_{(h,i)}(t-1)}}. \quad (6)$$

Here, $\hat{\mu}_{(h,i),S_{(h,i)}(t-1)}$ is the empirical mean computed using noisy evaluations obtained till time t, i.e. $\cup_{t'=0}^{t}\mathcal{O}_t$, and for node (h, i). In multiple works, this known noise setup and $UCB - \sigma^2$ algorithm has been considered in tree search algorithms without delays.

Theorem 3. *Let us assume that the variance of the noise in evaluations is σ^2 and is available to the algorithm. Then, under the same assumptions as Theorem 1 and upper bound on expected delay τ, PCTS using DUCB1σ for node selection achieves expected simple regret*

$$\epsilon_T = O\left(T^{-\frac{1}{d'+2}}\left((\sigma/\nu_1)^2\ln T + \tau\right)^{\frac{1}{d+2}}\right). \quad (7)$$

Effect of Known Noise. We observe that even with no delay, i.e. $\tau = 0$, noisy feedback with known variance σ^2 worsens the bound of HOO with noiseless evaluations by $\sigma^{2/(d+2)}$.

Table 3: Per-step cost $\lambda(Z_h)$ and total number of iterations $H(\Lambda)$ for different Fidelity models.

Fidel. Model	Linear Growth	Constant	Polynomial Decay	Exponential Decay
$\lambda(Z_h)$	$\min\{\beta h, \lambda(1)\}, \beta > 0$	$\min\{\beta, \lambda(1)\}, \beta > 0$	$\min\{h^{-\beta}, \lambda(1)\}, \beta > 0, \neq 1$	$\min\{\beta^{-h}, \lambda(1)\}, \beta \in (0, 1]$
$H(\Lambda)$	$\sqrt{2(2\Lambda - \lambda(1))/\beta}$	$2(2\Lambda - \lambda(1))/\beta$	$(1 + (1-\beta)(2\Lambda - \lambda(1)))^{1/(1-\beta)}$	$\log_{1/\beta}(1 + (1-\beta)(2\Lambda - \lambda(1)))$

Case 2: Unknown Variance. If the variance of the noise is unknown, we have to estimate the noise variance empirically from evaluations. Given the evaluations $\{\tilde{f}(X_1)\}_{t=0}^{T}$ and the delayed statistics $S_{(h,i)}(t-1)$ of node (h,i), the empirical noise variance at time t is $\hat{\sigma}^2_{(h,i),S_{(h,i)}(t-1)} \triangleq \frac{1}{S_{(h,i)}(t-1)} \sum_{j=1}^{S_{(h,i)}(t-1)} (\tilde{f}(X_j)\mathbb{1}[(H_j, I_j) = (h,i)] - \hat{\mu}_{(h,i),S_{(h,i)}(t-1)})^2$, where empirical mean $\hat{\mu}_{(h,i),S_{(h,i)}(t-1)} \triangleq \frac{1}{S_{(h,i)}(t-1)} \sum_{j=1}^{S_{(h,i)}(t-1)} \tilde{f}(X_j)\mathbb{1}[(H_j, I_j) = (h,i)]$. Using the empirical mean and variance of functional evaluations for each node (h,i), we now define a delayed-UCBV (DUCBV) confidence bound for selecting next node:

$$B_{(h,i),S_{(h,i)}(t-1),t} \triangleq \hat{\mu}_{(h,i),S_{(h,i)}(t-1)} + \sqrt{\frac{2\hat{\sigma}^2_{(h,i),S_{(h,i)}(t-1)} \log t}{S_{(h,i)}(t-1)}} + \frac{3b \log t}{S_{(h,i)}(t-1)}. \quad (8)$$

In practice, we do not need an exact value of b. We can use a large proxy value such that the feedback is bounded by it.

Theorem 4. *Let us assume that the upper bound on variance of the noise in evaluations is σ^2, which is unknown to the algorithm. If $[0, b]$ is the range of f, under the same assumptions as of Theorem 1, PCTS using DUCBV achieves expected simple regret*

$$\epsilon_T = O\left(T^{-\frac{1}{d'+2}} \left(((\sigma/\nu_1)^2 + 2b/\nu_1) \ln T + \tau\right)^{\frac{1}{d'+2}}\right).$$

Effect of Unknown Noise. *Adapting UCB-V in the stochastic delay setting and using the corresponding bound in PCTS allows us to extend hierarchical tree search for unknown noise both in presence and absence of delays.* To the best of our knowledge, this paper is the first to extend HOO framework for unknown noise, and also UCB-V to stochastic delays. This adaptation to unknown noise comes at a cost of $(((\sigma/\nu_1)^2 + 2b/\nu_1) \ln T)^{1/(d+2)}$ in expected error, whereas for known noise variance, it is $((\sigma/\nu_1)^2 \ln T)^{1/(d+2)}$.

3.4 Adapting to Delayed, Noisy, and Multi-fidelity (DNF) Feedback

Now, let us consider that we do not only have a delayed and noisy functional evaluator at each step but also an evaluator with different fidelity at each level h of the tree. This setup of multi-fidelity HOO without unknown noise and delay was first considered in (Sen, Kandasamy, and Shakkottai 2019). We extend their schematic to the version with delayed and noisy feedback with unknown delays and noise. Following the multi-fidelity formulation of (Sen, Kandasamy, and Shakkottai 2018, 2019), we consider the mean of the multi-fidelity query, $f_z(x)$, as biased, and progressively smaller bias can be obtained but with varying costs. The cost of selecting a new node at level $h > 0$ is $\lambda(Z_h) \in \mathbb{R}^+$ and the bias added in the decision due to the limited evaluation is $\zeta(Z_h) \in \mathbb{R}^+$. Here, the bias function is monotonically decreasing, and $Z_h \in \mathcal{Z}$ is the state of fidelity of the multi-fidelity evaluator, which influences both the cost of and the bias in evaluation. Thus, the evaluation at x_s is $\tilde{f}(x_{(h_s,l_s)}|z_{h,s}) \triangleq f(x_{(h_s,l_s)}) + \epsilon_s + \zeta(z_{h_s})$. Hence, under DNF feedback, the DUCBV selection rule becomes

$$B_{(h,i),S_{(h,i)}(t-1),t} \triangleq \hat{\mu}_{(h,i),S_{(h,i)}(t-1)} + \sqrt{\frac{2\hat{\sigma}^2_{(h,i),S_{(h,i)}(t-1)} \log t}{S_{(h,i)}(t-1)}} + \frac{3b \log t}{S_{(h,i)}(t-1)} + \zeta(Z_h).$$

Here, the empirical mean and variance are computed using the multi-fidelity and delayed feedbacks. We do not need to know ζ for the algorithm but we assume it to be known for the analysis. Given this update rule and the multi-fidelity model, we observe that the Lemma 1 of (Sen, Kandasamy, and Shakkottai 2019) holds. Given a total budget Λ and the multi-fidelity selection rule, the total number of iterations that the algorithms runs for is $T(\Lambda) \geq H(\Lambda) + 1$, where $H(\Lambda) \triangleq \max\{H : \sum_{h=1}^{H} \lambda(Z_h) \leq \Lambda\}$ Thus, we can retain the previously derived bounds of Theorem 2 and 4 by substituting $T = H(\Lambda)$.

Corollary 2 (PCTS + DUCBV under DNF Feedback)**.** *If the function under evaluation has h-dependent fidelity such that $H(\Lambda) \triangleq \max\{H : \sum_{h=1}^{H} \lambda(Z_h) \leq \Lambda\}$ and the induced bias $\zeta(Z_h) = \nu_1 \rho^h$, then under the same assumptions as of Theorem 1, PCTS using DUCB1 achieves*

$$\epsilon_\Lambda = O\left((H(\Lambda))^{-\frac{1}{d'+2}} (\ln H(\Lambda) + \tau)^{\frac{1}{d'+2}}\right),$$

and PCTS using DUCBV achieves expected simple regret

$$\epsilon_\Lambda = O\left((H(\Lambda))^{-\frac{1}{d'+2}} \left(\ln H(\Lambda) + \frac{\tau}{(\sigma/\nu_1)^2 + 2b/\nu_1}\right)^{\frac{1}{d'+2}}\right).$$

Models of Multi-fidelity. Depending on the evaluation problem, we may have different cost functions. In Table 3, we instantiate the cost model, bias model, and total number of iterations for four multi-fidelity models with linear growth, constant, polynomially decaying, and exponentially decaying costs of evaluations. The linear growth, polynomial decays, and exponential decays are observed in the cases of hyperparameter tuning of deep-learning models, database optimization, and tuning learning rates of convex optimization respectively. Further details are in Appendix C.

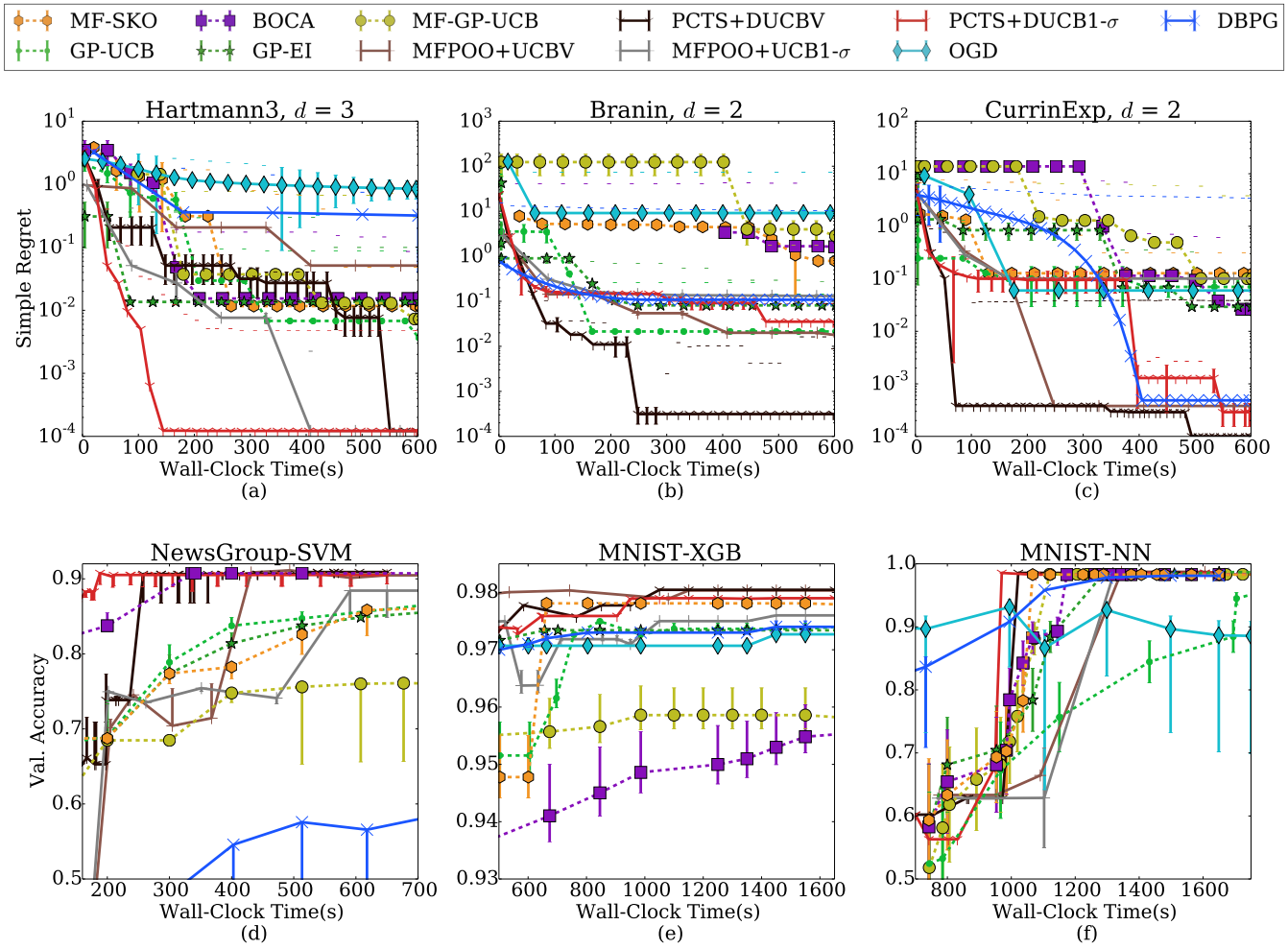

Figure 1: Figures (a) to (c) show simple regret (median of 10 runs) of different algorithms on synthetic functions with DNF feedbacks. Figures (d) to (f) show the cross-validation accuracy (median of 5 runs) achieved on the hyperparameter tuning of classifiers on datasets with DNF feedbacks.

4 Experimental Analysis

Experimental Setup. Similar to prior work on tree search with multi-fidelity and known noise (Sen, Kandasamy, and Shakkottai 2018, 2019), we evaluate performance of PCTS on both synthetic functions and machine learning models operating on real-data but under delayed, noisy (unknown), and multi-fidelity (DNF) feedback. We compare the performance of PCTS with: BO algorithms (BOCA (Kandasamy et al. 2017), GP-UCB (Srinivas et al. 2010), MF-GP-UCB (Kandasamy et al. 2016), GP-EI (Jones, Schonlau, and Welch 1998), MF-SKO (Huang et al. 2006))[3], tree search algorithms (MFPOO (Sen, Kandasamy, and Shakkottai 2018), MFPOO with UCB-V (Audibert, Munos, and Szepesvári 2007)), zeroth-order GD algorithms (OGD, DBGD (Li, Chen, and Giannakis 2019)).

[3]We use the implementations in https://github.com/rajatsen91/MFTreeSearchCV for baselines except OGD, DBGD, and MFPOO-UCBV. For BO algorithms, this implementation chooses the best among the polynomial kernel, coordinate-wise product kernel and squared exponential kernel for each problem.

In our experiments, cost is the time required for a function evaluation. Delay is the time required to communicate the result to the PCTS algorithm. The cost-function and delay can be general but in our experiments, we focus on time-efficiency. Thus, we plot the convergence of the simple regret of competing algorithms with respect to the wall-clock time. In our experiments, we do not assume the smoothness parameters, the bias function, and the cost function to be known. The smoothness parameters are computed in a similar manner as POO and MFPOO. For comparison, we keep the delay constant and use wait-and-act versions of delay-insensitive baselines. The experiments with stochastic delay are elaborated in Appendix, where PCTS variants perform even better in comparison with the constant delay setups.

Synthetic Functions. We illustrate results for three different synthetic functions, Hartmann3 (van der Vlerk 1996), Branin (van der Vlerk 1996), and CurrinExp (Currin et al. 1988) with noise variances $\sigma^2 = 0.01$, 0.05, and 0.05 respectively. We follow the fidelity setup of (Sen, Kandasamy, and Shakkottai 2018, 2019), that modifies the synthetic

functions to incorporate the fidelity space $\mathcal{Z} = [0, 1]$. The delay time τ for all synthetic functions is set to four seconds. We choose to add noise from Gaussian distributions with variance σ^2. Note that the noise can be added from any distribution with variance $\leq \sigma^2$. This σ is passed to UCB1-σ and DUCB1-σ in MFPOO and PCTS as it assumes the noise variance is known (Sen, Kandasamy, and Shakkottai 2019). We implement all baselines in Python (version 2.7). We run each experiment ten times for 600s on a MacBook Pro with a 6-core Intel(R) Xeon(R)@2.60GHz CPU and plot the median value of simple regret, i.e. l_1 distance between the value of current best point and optimal value, for each algorithm.

Real Data: Hyperparameter Tuning. We evaluate the aforementioned algorithms on a 32-core Intel(R) Xeon(R)@2.3 GHz server for hyperparameter tuning of SVM on News Group dataset, and XGB and Neural Network on MNIST datasets. We use corresponding scikit-learn modules (Buitinck et al. 2013) for training all the classifiers. For each tuning task, we plot the median value of cross-validation accuracy in five runs for 700s, 1700s, and 1800s respectively. We set $\sigma^2 = 0.02$ for algorithms where σ is known, and $b = 1$ where UCBV and DUCBVare used.

Summary of Results. *In all of the experiments, we observe that either* PCTS + DUCB1 *or* PCTS + DUCBV *outperforms the competing algorithms in terms of convergence speed.* Also, in case of synthetic functions, they achieve approximately 1 to 3 order lower simple regret. These results empirically validate the efficiency of PCTS to adapt to DNF feedback. Due to lack of space, further implementation details, results on tree depth, performance for stochastic delays, and error statistics for other synthetic functions and hyperparamter tuning experiments are deferred to appendix.

5 Discussion and Future Work

We propose a generic tree search approach PCTS for black-box optimization problems with DNF feedbacks. We provide a generic analysis to bound the expected simple regret of PCTS given a horizon T. We instantiate PCTS with delayed-UCB1 and delayed-UCBV for observable stochastic delays, and known and unknown noises respectively. Our analysis shows that the expected simple regret for PCTS worsens by a constant factor and $T^{\frac{1-\alpha}{d+2}}$ for expected delay of $O(\log T)$ and $O(T^{1-\alpha})$ respectively. We also experimentally show that PCTS outperforms other global optimizers incompatible or individually tolerant to noise, delay, or multi-fidelity on both synthetic and real-world functions. In addition, our work extends UCB-V to stochastic delays.

In the future, we plan to consider anonymous delay feedbacks in order to develop tree search optimizers that respect privacy. It also shows the need for proving a problem-independent lower bound for hierarchical tree search with stochastic delay. The other possible direction is to deploy PCTS for planning in Markov Decision Processes with delay, where tree search algorithms have been successful.

Acknowledgement

This research project is supported by NSF grant IIS-1910830 (“Regret-Bounded Query Evaluation via Reinforcement Learning”).

References

Agarwal, A.; and Duchi, J. C. 2012. Distributed delayed stochastic optimization. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 5451–5452. IEEE.

Audibert, J.-Y.; Munos, R.; and Szepesvári, C. 2007. Tuning Bandit Algorithms in Stochastic Environments. In Hutter, M.; Servedio, R. A.; and Takimoto, E., eds., *Algorithmic Learning Theory*, 150–165. Berlin, Heidelberg: Springer Berlin Heidelberg.

Auer, P.; Cesa-bianchi, N.; and Fischer, P. 2002. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3): 235–256.

Bubeck, S.; Munos, R.; Stoltz, G.; and Szepesvári, C. 2011. X-Armed Bandits. *Journal of Machine Learning Research*, 12(5).

Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; Layton, R.; VanderPlas, J.; Joly, A.; Holt, B.; and Varoquaux, G. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

Cesa-Bianchi, N.; Gentile, C.; Mansour, Y.; and Minora, A. 2016. Delay and cooperation in nonstochastic bandits. In *Conference on Learning Theory*, 605–622. PMLR.

Chen, H. 1988. Lower rate of convergence for locating a maximum of a function. *The Annals of Statistics*, 1330–1334.

Currin, C.; Mitchell, T.; Morris, M.; and Ylvisaker, D. 1988. A Bayesian approach to the design and analysis of computer experiments. Technical report, Oak Ridge National Lab., TN (USA).

Eick, S. G. 1988. The two-armed bandit with delayed responses. *The Annals of Statistics*, 254–264.

Fischer, L.; Gao, S.; and Bernstein, A. 2015. Machines tuning machines: Configuring distributed stream processors with bayesian optimization. In *2015 IEEE International Conference on Cluster Computing*, 22–31. IEEE.

Gael, M. A.; Vernade, C.; Carpentier, A.; and Valko, M. 2020. Stochastic bandits with arm-dependent delays. In *International Conference on Machine Learning*, 3348–3356. PMLR.

Goldstein, A. 1977. Optimization of Lipschitz continuous functions. *Mathematical Programming*, 13(1): 14–22.

Grill, J.-B.; Altché, F.; Tang, Y.; Hubert, T.; Valko, M.; Antonoglou, I.; and Munos, R. 2020. Monte-Carlo tree search as regularized policy optimization. In *International Conference on Machine Learning*, 3769–3778. PMLR.

Grill, J.-B.; Valko, M.; Munos, R.; and Munos, R. 2015. Black-box optimization of noisy functions with unknown smoothness. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Huang, D.; Allen, T. T.; Notz, W. I.; and Miller, R. A. 2006. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5): 369–382.

Jamieson, K. G.; Nowak, R. D.; and Recht, B. 2012. Query complexity of derivative-free optimization. *arXiv preprint arXiv:1209.2434*.

Jones, D. R.; Schonlau, M.; and Welch, W. J. 1998. Efficient Global Optimization of Expensive Black-Box Functions. *J. of Global Optimization*, 13(4): 455–492.

Joulani, P.; Gyorgy, A.; and Szepesvári, C. 2013. Online learning under delayed feedback. In *International Conference on Machine Learning*, 1453–1461. PMLR.

Joulani, P.; Gyorgy, A.; and Szepesvári, C. 2016. Delay-tolerant online convex optimization: Unified analysis and adaptive-gradient algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Kajita, S.; Kinjo, T.; and Nishi, T. 2020. Autonomous molecular design by Monte-Carlo tree search and rapid evaluations using molecular dynamics simulations. *Communications Physics*, 3(1): 1–11.

Kandasamy, K.; Dasarathy, G.; Oliva, J. B.; Schneider, J.; and Poczos, B. 2016. Gaussian Process Bandit Optimisation with Multi-fidelity Evaluations. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Kandasamy, K.; Dasarathy, G.; Schneider, J.; and Póczos, B. 2017. Multi-fidelity bayesian optimisation with continuous approximations. In *International Conference on Machine Learning*, 1799–1808. PMLR.

Kumagai, W. 2017. Regret analysis for continuous dueling bandit. *arXiv preprint arXiv:1711.07693*.

Langford, J.; Smola, A.; and Zinkevich, M. 2009. Slow learners are fast. *arXiv preprint arXiv:0911.0491*.

Li, B.; Chen, T.; and Giannakis, G. B. 2019. Bandit online learning with unknown delays. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 993–1002. PMLR.

Liu, S.; Li, X.; Chen, P.-Y.; Haupt, J.; and Amini, L. 2018. Zeroth-order stochastic projected gradient descent for nonconvex optimization. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 1179–1183. IEEE.

Martinez-Cantin, R. 2017. Bayesian optimization with adaptive kernels for robot control. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 3350–3356. IEEE.

Munos, R. 2014. From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning. *Foundations and Trends® in Machine Learning*, 7(1): 1–129.

Nguyen, V.; Gupta, S.; Rana, S.; Li, C.; and Venkatesh, S. 2019. Filtering Bayesian optimization approach in weakly specified search space. *Knowledge and Information Systems*, 60(1): 385–413.

Oh, C.; Gavves, E.; and Welling, M. 2018. Bock: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, 3868–3877. PMLR.

Pavlo, A.; Angulo, G.; Arulraj, J.; Lin, H.; Lin, J.; Ma, L.; Menon, P.; Mowry, T. C.; Perron, M.; Quah, I.; et al. 2017. Self-Driving Database Management Systems. In *CIDR*, volume 4, 1.

Pike-Burke, C.; Agrawal, S.; Szepesvari, C.; and Grunewalder, S. 2018. Bandits with Delayed, Aggregated Anonymous Feedback. arXiv:1709.06853.

Sen, R.; Kandasamy, K.; and Shakkottai, S. 2018. Multi-fidelity black-box optimization with hierarchical partitions. In *International conference on machine learning*, 4538–4547. PMLR.

Sen, R.; Kandasamy, K.; and Shakkottai, S. 2019. Noisy blackbox optimization using multi-fidelity queries: A tree search approach. In *The 22nd international conference on artificial intelligence and statistics*, 2096–2105. PMLR.

Shang, X.; Kaufmann, E.; and Valko, M. 2018. Adaptive black-box optimization got easier: HCT only needs local smoothness. In *EWRL 2018*.

Shang, X.; Kaufmann, E.; and Valko, M. 2019. General parallel optimization without a metric. In *Algorithmic Learning Theory*, 762–788. PMLR.

Springenberg, J. T.; Klein, A.; Falkner, S.; and Hutter, F. 2016. Bayesian Optimization with Robust Bayesian Neural Networks. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Sra, S.; Yu, A. W.; Li, M.; and Smola, A. J. 2015. Adadelay: Delay adaptive distributed stochastic convex optimization. *arXiv preprint arXiv:1508.05003*.

Srinivas, N.; Krause, A.; Kakade, S.; and Seeger, M. 2010. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, 1015–1022. Madison, WI, USA: Omnipress. ISBN 9781605589077.

van der Vlerk, M. H. 1996. Stochastic programming bibliography. *World Wide Web, http://mally. eco. rug. nl/spbib. html*, 2003.

Vernade, C.; Cappé, O.; and Perchet, V. 2017. Stochastic bandit models for delayed conversions. *arXiv preprint arXiv:1706.09186*.

Wang, J.; Trummer, I.; and Basu, D. 2021. UDO: universal database optimization using reinforcement learning. In *arXiv:2104.01744*, 1–13.

Wang, L.; Zhao, Y.; Jinnai, Y.; Tian, Y.; and Fonseca, R. 2019. Alphax: exploring neural architectures with deep neural networks and monte carlo tree search. *arXiv preprint arXiv:1903.11059*.

Wang, Y.; Balakrishnan, S.; and Singh, A. 2019. Optimization of smooth functions with noisy observations: Local minimax rates. *IEEE Transactions on Information Theory*, 65(11): 7350–7366.

Weinberger, M. J.; and Ordentlich, E. 2002. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7): 1959–1976.

Xu, Y.; Joshi, A.; Singh, A.; and Dubrawski, A. 2020. Zeroth order non-convex optimization with dueling-choice bandits. In *Conference on Uncertainty in Artificial Intelligence*, 899–908. PMLR.

Xue, D.; Balachandran, P. V.; Hogden, J.; Theiler, J.; Xue, D.; and Lookman, T. 2016. Accelerated search for materials with targeted properties by adaptive design. *Nature communications*, 7(1): 1–9.