Derivative-Free & Order-Robust Optimisation

Victor Gabillon,[1] **Rasul Tutunov,**[1] **Michal Valko,**[2] **Haitham Bou Ammar**[1]
Huawei R&D UK[1] Inria Lille-Nord Europe[2]

Abstract

In this paper, we formalise order-robust optimisation as an instance of online learning minimising simple regret, and propose VROOM, a zeroth order optimisation algorithm capable of achieving vanishing regret in non-stationary environments, while recovering favorable rates under stochastic reward-generating processes. Our results are the first to target simple regret definitions in adversarial scenarios unveiling a challenge that has been rarely considered in prior work.

1 Introduction

Derivative-free optimisation is a discipline by which learners attempt to determine optimal solutions while only exploiting function value information (Matyas, 1965). Such a setting is of great interest for applications in which it is either difficult to define, access or even compute first and/or second-order function information (Nesterov and Spokoiny, 2017). As such, derivative-free optimisation naturally addresses optimising over functions that are non-differentiable, non-continuous or even non-smooth.

A variety of versatile zero-order methods have been developed under minimal smoothness assumptions (Auer et al., 2007; Kleinberg et al., 2008). Though flexible, most algorithms in the literature are designed under specific assumptions on the process by which evaluation data is generated. SOO (Munos, 2011), for instance, optimises sequentially over a deterministic function, while StoSOO (Valko et al., 2013) optimises a sequence of noisy but stationary functions. No such algorithm, however, handles a sequence of non-stationary observations – a setting commonly faced in a variety of real-world problems. Consequently, in a scenario in which the process generating the data is unknown a priori, what algorithm would a practitioner employ?

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

To illustrate the above concept, consider a lifelong learning problem (Thrun and Mitchell, 1995; Ammar et al., 2014; Parisi et al., 2019) where a model is updated while interacting with a sequence of tasks. Here, the objective is to have a learner capable of performing well on average over all observed data. If the tasks are similar, learning online helps in solving novel tasks. However, when task differences are drastic, catastrophic forgetting occurs (French, 1999; Kirkpatrick et al., 2017) leading to situations where newly observed data hurts performance on earlier problems. In fact, it has been reported that the order by which tasks are streamed dramatically affects average performance. It is for this reason that recent research in lifelong learning has focused on building *order-robust* approaches (Yoon et al., 2019) that we formalise in this work as an instance of online learning with simple regret considerations.

Precisely, we formalise the above problem by optimising over elements x in a continuous set $\mathcal{X}$. n tasks are streamed sequentially allowing the learner to attempt a sequence $x_1, \ldots, x_n$ across rounds. At round t, the learner observes a reward $f_t(x_t)$ corresponding to the performance of parameter x_t on task t as represented by the mapping f_t. After n rounds, the agent recommends a parameter $x(n)$ with the objective of maximizing its average reward over all observed tasks, i.e., $\frac{1}{n}\sum_{t=1}^{n} f_t(x(n))$.

Contrary to other methods in the literature, we believe that minimal assumptions on the order by which $f_1, \ldots, f_n$ are observed have to be invoked to ensure order-robustness. Furthermore, our algorithm should also behave near optimally as if an a priori knowledge of such an order (e.g., stochastic observations) was explicitly provided. Interestingly, this motivation unveils a novel problem which we refer to as **the best of both worlds (BOB) challenge**. Here, we aim to design one simple algorithm that is unaware of the nature of the reward generating process but can acquire near-optimal regret guarantees in both stochastic and adversarial non-stationary settings. In this paper, we take the first step to resolving the aforementioned challenge by proposing VROOM a novel algorithm that optimises over f at different levels of discretisation of the input space $\mathcal{X}$. VROOM makes use of the standard importance-weighted estimates used in non-stochastic literature for efficient exploration. We realise, however, that the direct application of these

techniques to our setting suffers from two major drawbacks related to variance explosion when observation probabilities diminish with discretisation widths, and estimate unreliability due to variance disparities. Providing solutions to each of these above problem, our contributions are summarised as: 1) formally introducing simple regret minimisation in non-stochastic and order-robust optimisation, 2) analysing a uniform exploration algorithm and demonstrating state-of-the-art bounds in non-stochastic settings, and 3) introducing VROOM as a solution to order-robustness proving vanishing regrets in non-stochastic settings and $\widetilde{\mathcal{O}}\Big(n^{-\frac{1}{d+3}}\Big)$ in the stochastic case.

2 Problem Formulation and Analysis Tools

In this section, we detail our problem formulation, its novelty, the associated challenge, and our contributions.

In budgeted optimisation, a learner optimises a function $f : \mathcal{X} \to \mathbb{R}$ having access to a number of evaluations limited by n. This setting also includes the case $\mathcal{X} \subset \mathbb{R}^D$. We consider a general case where f is decomposable as,

$$f = \frac{1}{n}\sum_{t=1}^{n} f_t.$$

It is clear that f depends on n. However, since n is a fixed input parameter of the problem, we drop such dependency in our notation for ease of exposition. At each round $t \in \{1, \dots, n\}$, the learner chooses an element $x_t \in \mathcal{X}$ and observes a real number y_t, where $y_t = f_t(x_t)$ quantifying its reward. As we are concerned with order-robustness, we distinguish two feedback settings with respect to the process by which f_t's are interconnected:

Stochastic feedback In stochastic feedback, function evaluations are perturbed by a noise in the range $b \in \mathbb{R}_+$[1]: Precisely, at any round, we have $f_t = \overline{f} + \varepsilon_t$ with ε_t being a random variable that is identically and independently distributed (i.i.d.) over rounds. Further, we consider the case when $\overline{f}$ is a function that is independent of t and n, and where:

$$\mathbb{E}[\varepsilon_t] = 0 \quad \text{and} \quad |\varepsilon_t| \leq b. \tag{1}$$

Non-stochastic feedback To consider non-stationary and non-stochastic data, we minimally assume:

$$|f_{t'}(x) - f_t(x)| \leq b \text{ for all } t, t' \text{ and } x \in \mathcal{X}. \tag{2}$$

Actually we will sometimes rephrase this condition as the equivalent condition $|f_t(x)| \leq f_{max}$ for all $x \in \mathcal{X}$ and $t \in [n]$.

[1] Alternatively, we can turn the boundedness assumption into a sub-Gaussianity assumption equipped with a variance parameter equivalent to our range b.

Given these feedback laws, the learner's objective is to return an element $x(n)$ in $\mathcal{X}$ with the largest possible value $f(x)$ after the n evaluations. To that end, we allow the learner to utilise internal randomisation, i.e., sample $x(n)$ from a distribution ν_n of its choice, $x(n) \sim \nu_n$.

Since we consider two feedback laws (i.e., stochastic and non-stochastic), we quantify the agent's performance using two notions of simple regrets. In the first, we consider regret as a random variable induced by $\varepsilon_1, \dots, \varepsilon_n$ and bound its expectation over the random sequence $f_1, \dots, f_n$:

$$\begin{aligned}
\mathbb{E}_f[r_n] &\triangleq \mathbb{E}_{f_1,\dots,f_n}\left[\sup_{x\in\mathcal{X}} f(x) - \mathbb{E}_{x(n)}[f(x)]\right] \\
&= \sup_{x\in\mathcal{X}} \overline{f}(x) - \mathbb{E}_{f_1,\dots,f_n}\left[\mathbb{E}_{x(n)}\left[\overline{f}(x(n)) + \sum_{t=1}^{n}\varepsilon_t\right]\right] \\
&= \sup_{x\in\mathcal{X}} \overline{f}(x) - \mathbb{E}_{x(n)}\left[\overline{f}(x(n))\right] - \mathbb{E}_{f_1,\dots,f_n}\left[\sum_{t=1}^{n}\varepsilon_t\right] \\
&= \sup_{x\in\mathcal{X}} \overline{f}(x) - \mathbb{E}_{x(n)}\left[\overline{f}(x)\right],
\end{aligned}$$

where the expectation with respect to ν_n. When it comes to the non-stochastic setting, the situation is simpler where for a given sequence of function observations, we define:

$$r_n \triangleq \sup_{x\in\mathcal{X}} f(x) - \mathbb{E}_{x(n)}[f(x)], \tag{3}$$

We further consider the case when evaluation is costly. Therefore, we minimise r_n as a function of n assuming that for any given sequence $f_1, \dots, f_n$, there exists at least one point $x^\star \in \mathcal{X}$ such that $f(x^\star) = \sup_{x\in\mathcal{X}} f(x)$.

Before commencing with our solution, it is worth noting that optimising simple regret with non-stochastic data generating processes has not been studied as a stand-alone problem in literature so far[2]. It is viewed by some authors as an ill-defined problem (Hazan et al., 2016, Chapter 3) as the objective varies at each round t. Moreover, if the simple regret is formulated as in Equation 3, one can, in some cases, derive bounds for such a quantity from the analysis of cumulative regret, $\sup_{x\in\mathcal{X}} \frac{1}{n}\sum_{t=1}^{n} f_t(x) - \frac{1}{n}\sum_{t=1}^{n} f_t(x_t)$ – a notion extensively studied in (Auer, 2002; Zinkevich, 2003; Bubeck et al., 2017). In the stochastic setting or when $f_t = f_1$ for $t \in [n]$, obtaining an upper bound, R_n, on the cumulative regret leads to an upper bound $r_n = R_n/n$ on the simple regret as noted in Hazan et al. (2016); Bubeck et al. (2011). It is worth noting that though a bound can be attained, these two objectives are not equivalent. Precisely, a bound obtained in the cumulative regret case is often sub-optimal from a simple regret point of view (Bubeck et al., 2009). Furthermore, contrary to simple-regret algorithms, cumulative-regret learners find it challenging to adapt function smooth-

[2] Section 4 extensively reviews the long history of existing results for stochastic and deterministic feedback laws.

ness without extra information on f (Locatelli and Carpentier, 2018). In fact, it is intuitive to realise that minimising cumulative regret aims at accumulating rewards (see the term $\frac{1}{n}\sum_{t=1}^{n} f_t(x_t)$), as opposed to identifying the optimum (the term $\frac{1}{n}\sum_{t=1}^{n} f_t(x(n))$); a property dictated through simple regret considerations. Finally, note that to the best of our knowledge no upper bound on the cumulative regret exists in non-stochastic settings under minimal assumptions on f used in this paper and that the connection between cumulative and simple regret is unclear in the non-stochastic setting.

2.1 Mathematical Tools

During the remainder of this paper, we will make use of mathematical tools that we briefly survey in this section. Firstly, we describe partitioning assumptions facilitating our search for an optimal solution of our optimisation problem, and then detail tree-based learners that we build on in developing VROOM.

2.1.1 Partitioning & Near-Optimality Dimension

During our exploration for an optimum, we discretise the search space into cells (nodes) allowing us to consider tree-like learners. To do so, we follow a hierarchical partitioning $\mathcal{P} = \{\{\mathcal{P}_{h,i}\}_{i=1}^{I_h}\}_{h=0}^{\infty}$ previously introduced in (Munos, 2011; Valko et al., 2013; Grill et al., 2015). For any depth $h \geq 0$ (which we think of as a tree representation), the set $\{\mathcal{P}_{h,i}\}_{1\leq i\leq I_h}$ of *cells* (or nodes) forms a partition of $\mathcal{X}$, where I_h is the number of cells at depth h. At depth 0, the root of the tree, there is a single cell $\mathcal{P}_{0,1} = \mathcal{X}$. A cell $\mathcal{P}_{h,i}$ of depth h is split into children sub-cells $\{\mathcal{P}_{h+1,j}\}_j$ of depth $h+1$. The objective of many algorithms is to explore the value of f in the cells of the partition and to identify at the deepest possible depth a cell containing a global maximum. For simplicity and without loss of generality we assume all cells have K children sub-cells.

Given a global maximum $x^\star$ of f, $i_h^\star$ denotes the index of the unique cell of depth h containing $x^\star$, i.e., such that $x^\star \in \mathcal{P}_{h,i_h^\star}$. We follow the work of Grill et al. (2015) and state a *single* assumption on both the partitioning $\mathcal{P}$ and the function f.

Assumption 1. *For any global optimum $x^\star$, there exists $\nu > 0$ and $\rho \in (0,1)$, where the values of ν and ρ depend on $x^\star$, such that $\forall h \in \mathbb{N}$, $\forall x \in \mathcal{P}_{h,i_h^\star}$, $f(x) \geq f(x^\star) - \nu\rho^h$.*

Assumption 1 is weaker than global or local Lipschitzness. Smooth functions being locally Lipschitz, assumption 1 is implied by (and therefore weaker than) smoothness.

The notion of a near-optimality dimension d aims at capturing the smoothness of the function and characterises the complexity of the optimisation task. We adopt the definition of near-optimality dimension given recently by Grill et al. (2015) that unlike Bubeck et al. (2011), Valko et al. (2013), Munos (2011), and Azar et al. (2014), avoids topological notions and does not artificially attempt to separate the difficulty of the optimisation from the partitioning. For each depth h, it simply counts the number of near-optimal cells $\mathcal{N}_h$, i.e., those whose value is close to $f(x^\star)$, and determines how this number evolves with the depth h. The smaller d, the more accurate is the optimisation.

Definition 1. *For any $\nu > 0$, $C > 1$, and $\rho \in (0,1)$, the **near-optimality dimension**[3] $d(\nu, C, \rho)$ of f with respect to the partitioning $\mathcal{P}$, is*

$$d(\nu, C, \rho) \triangleq \inf\Big\{d' \in \mathbb{R}^+ : \forall h \geq 0,\ \mathcal{N}_h(3\nu\rho^h) \leq C\rho^{-d'h}\Big\},$$

where $\mathcal{N}_h(\varepsilon)$ is the number of cells $\mathcal{P}_{h,i}$ of depth h such that $\sup_{x\in\mathcal{P}_{h,i}} f(x) \geq f(x^\star) - \varepsilon$.

By construction we have $d \leq \log(K)/\log(1/\rho)$. In general $d \ll \log(K)\log(1/\rho)$ as having $d = 0$ is the most common case in practice (Valko et al., 2013).

2.1.2 Tree-Based Learners

Tree-based exploration or a tree search algorithm is an approach that has been widely applied to optimisation as well as bandits or planning problems (Kocsis and Szepesvári, 2006; Coquelin and Munos, 2007; Hren and Munos, 2008); see Munos (2014) for a survey.

First we define the sampling of an element x in a cell $\mathcal{P}_{h,i}$ *with respect to* $\mathcal{P}$, denoted $x \sim U_{\mathcal{P}}(\mathcal{P}_{h,i})$ as follows: Starting from a cell $c_1 = \mathcal{P}_{h,i}$, we descend the partition until depth n by choosing at depth h' (with $h \leq h' < n$) a sub-cell $c_{h'+1}$ of $c_{h'}$ chosen uniformly at random among the K children cells of $c_{h'}$. Once at depth n in $\mathcal{P}_{n,i}$, we pick an element x uniformly at random in $\mathcal{P}_{n,i}$. [4]

At each round t, the learner selects an element $x_t \in \mathcal{X}$. First the learner selects a cell $\mathcal{P}_{h_t,i_t}$ according to the distribution $\boldsymbol{p}_t$ on $\mathcal{P}$ that associates to each cell $\mathcal{P}_{h,i}$ the probability $\boldsymbol{p}_{h,i,t} = \mathbb{P}(\mathcal{P}_{h_t,i_t} = \mathcal{P}_{h,i})$ of being the selected cell $\mathcal{P}_{h_t,i_t}$ at time t. We have $\sum_{\mathcal{P}_{h,i}\in\mathcal{P}} \boldsymbol{p}_{h,i,t} = 1$ for any given t. Then, the learner samples an element x_t in $\mathcal{P}_{h_t,i_t}$ with respect to $\mathcal{P}$, $x_t \sim U_{\mathcal{P}}(\mathcal{P}_{h_t,i_t})$, and asks for its evaluation.

We denote the value $f_{h,i} \triangleq \mathbb{E}_{x\sim U_{\mathcal{P}}(\mathcal{P}_{h,i})}[f(x)]$, $f_{h,i,t} \triangleq \mathbb{E}_{x\sim U_{\mathcal{P}}(\mathcal{P}_{h,i})}[f_t(x)]$ and, in the stochastic feedback case, $\overline{f}_{h,i} = \mathbb{E}_{x\sim U_{\mathcal{P}}(\mathcal{P}_{h,i})}\big[\overline{f}(x)\big]$. We use $T_{h,i}(t) = \sum_{s=1}^{t-1} 1_{x_s\in\mathcal{P}_{h,i}}$ to denote the total number of evaluations that have been allocated by the learner between round 1 and the beginning of round t to the cell $\mathcal{P}_{h,i}$. For the stochastic noisy case, we also define the estimated value of the

[3] Grill et al. (2015) define $d(\nu, C, \rho)$ with the constant 2 instead of 3. 3 eases the exposition of our results.

[4] Assuming that each parent cell has K children, sampling from $U_{\mathcal{P}}(\mathcal{P}_{h,i})$ is just a uniform sampling from the descendants of $\mathcal{P}_{h,i}$ at depth n. If we assume that each cell can have different number of children, then $U_{\mathcal{P}}(\mathcal{P}_{h,i})$ follows the topology of $\mathcal{P}$.

cell $\mathcal{P}_{h,i} \in \mathcal{T}$ as follows: given the $T_{h,i}(t)$ evaluations $y_1, \dots, y_{T_{h,i}(t)}$, we have

$$\widehat{f}_{h,i}(t) \triangleq \frac{1}{T_{h,i}(t)} \sum_{s=1}^{T_{h,i}(t)} y_s,$$

the empirical average of rewards obtained at this cell.

Similarly, for the non-stochastic case, we define $\widetilde{f}_{h,i}(t)$ that estimates $f_{h,i,t}$ for cell $\mathcal{P}_{h,i}$ at time t. This estimates uses the function values $f_t(x_t)$ if collected from sampling directly cell $\mathcal{P}_{h,i}$ as $x_t \sim U_{\mathcal{P}}(\mathcal{P}_{h,i})$ which corresponds to $h_t = h$ and $i_t = i$. In addition, the estimate $\widetilde{f}_{h,i}(t)$ also takes into account $f_t(x_t)$ if both $x_t \in \mathcal{P}_{h,i}$ and $h_t \leq h$. This addition improves the accuracy of our estimate while forcing $h_t \leq h$ insures that $f_t(x_t)$ is an unbiased estimate of the quantity of interest $f_{h,i,t}$ as proven below. Having a sample $x_t \sim U_{\mathcal{P}}(\mathcal{P}_{h_t,i_t})$ with $\mathcal{P}_{h_t,i_t} \sim \boldsymbol{p}_t$, possibly $\mathcal{P}_{h_t,i_t} \neq \mathcal{P}_{h,i}$, and an observation $y_t = f_t(x_t)$, we have

$$\widetilde{f}_{h,i}(t) \triangleq \frac{y_t 1_{x_t \in \mathcal{P}_{h,i}} 1_{h \geq h_t}}{\mathbb{P}(x_t \in \mathcal{P}_{h,i} \cap h \geq h_t)}. \tag{4}$$

Note that
$$\begin{aligned}\mathbb{E}_{\mathcal{P}_{h_t,i_t} \sim \boldsymbol{p}_t}[\mathbb{E}_{x_t \sim U_{\mathcal{P}}(\mathcal{P}_{h_t,i_t})}[\widetilde{f}_{h,i}(t)]] \\ = \mathbb{E}_{x_t \sim U_{\mathcal{P}}(\mathcal{P}_{h_t,i_t})}[y_t | x_t \in \mathcal{P}_{h,i} \text{ and } h \geq h_t] \\ \overset{\textbf{(a)}}{=} \mathbb{E}_{x_t \sim U_{\mathcal{P}}(\mathcal{P}_{h,i})}[y_t] = f_{h,i,t}.\end{aligned}$$

where **(a)** is by definition of $U_{\mathcal{P}}(\mathcal{P}_{h_t,i_t})$. We define $\widetilde{F}_{h,i}(t) \triangleq \sum_{s=1}^{t} \widetilde{f}_{h,i}(t)$, the sum of rewards obtained at this cell. We define $F_{h,i}(t) \triangleq \sum_{s=1}^{t} f_{h,i,s}$. Finally, let $[a : c] = \{a, a+1, \dots, c\}$ with $a, c \in \mathbb{N}$, $a \leq c$, and $[a] = [1 : a]$. $\log_d$ denotes the logarithm in base $d \in \mathbb{R}$. Without a subscript, log is the natural logarithm in base e.

3 VROOM: An Order-Robust Algorithm

This section details our contributions to addressing order-robustness. On a high level, we split the exposition in three parts. First, we provide a robust version of uniform exploration that sets state-of-the-art regret guarantees for non-stochastic settings. While these guarantees are believed to be unimprovable, uniform exploration is known to perform sub-optimally in stochastic scenarios. As such, we revert-back to the BOB challenge discussing achievable regret rates before presenting `VROOM`.

Before diving into details of our proposed method, it is instructive to recap the challenges faced when considering two feedback laws. Targeting only stochastic feedbacks, it is well known that `StroquOOL` and `GPO`, achieve state-of-the-art regret bounds. Unfortunately, the direct application of these methods to an adversarial setting is challenging due to the potential blunder that can be caused by feeding uninformative rewards for a deterministic learner as pointed in Bubeck and Cesa-Bianchi (2012, Section 3). Therefore, it is essential for an efficient learner to employ internal randomisation that defines a positive probability $\mathbb{P}(x_t \in \mathcal{P}_{h,i})$ for each cell during its exploration quest. Given positive probabilities, we can now target an estimator for $f(x)$ to perform meaningful updates. Clearly, the simple usage of empirical averaged rewards $\widehat{f}_{h,i}(t)$ in cell $\mathcal{P}_{h,i}$ is easily biased by an adversary. Fetching an unbiased estimate, we realise that $\widetilde{f}_{h,i}(t)$ is a meaningful alternative. Though viable, $\widetilde{f}_{h,i}(t)$ can possess high variance especially if $\boldsymbol{p}_{h,i,t}$ is small (scaling with $1/\mathbb{P}(x_t \in \mathcal{P}_{h,i})$). Two sources contribute to these high variance occurrences: 1) long uniform exploration, and 2) K^h increase in the number of cells with depth (leading to variances of K^h magnitude). Realising these problems, we present our first challenge that we tackle in this paper as:

Challenge I: How to control potentially large estimator variances (especially in the stochastic setting)?

Apart from variance control, we face another interesting problem related to the optimum recommendation, $x(n)$, made by the learner after n rounds of interaction. If we are to recommend the best cell as that with the highest estimate $\sum_{t=1}^{n} \widetilde{f}_{h,i}(t)$, we might end-up comparing estimates with widely different confidence intervals[5]. At first sight, one can attempt to follow the approaches proposed by others in the literature to tackle this issue. In `StoSOO`, for instance, $x(n)$ is chosen among the cells that have been pulled in an order of $\widetilde{\mathcal{O}}(n)$. Though appealing, this method does not fit-well the cases where we need to sample a large number of cells with a limited number of pulls such as in the low noise, deterministic feedback and/or $d = 0$ settings[6]. In `StroquOOL`, on the other hand, a separate cross-validation phase allocates $\widetilde{\mathcal{O}}(n)$ extra samples to the best cells recommended at the end of an initial exploration phase. Nonetheless, when dealing with non-stochastic reward-generating processes, there are no guarantees on the relationship between collected data in two successive phases. Hence, following such a recommendation introduces a (hard-to-control) bias typically leading to additional hyper-parameters measuring exploration lengths. Observing optimum recommendation difficulties arising from considering two feedback laws, our second challenge can be stated as:

Challenge II: How to recommend an optimum $x(n)$ capable of operating successfully in both feedback settings?

The remainder of this section provides solutions to each of the above challenges ultimately proposing `VROOM` as a simple yet effective algorithm for order-robust optimisation.

[5] Please note that this is due to the dependence on the number of pulls allocated to $\mathcal{P}_{h,i}$, as well as on the variance of the estimates.

[6] In such cases `StoSOO` lacks any theoretical guarantees.

3.1 Uniform Allocation Baselines

In this section, we derive achievable baseline simple regret rates in non-stochastic scenarios. We note that such a problem has not yet been targeted by current literature. To do so, we consider a uniform exploration strategy allowing us to achieve initial results addressing **Challenge II**[7]. We specifically discuss two optimum recommendation techniques: 1) cross-validation, and 2) lower confidence bounds (LCBs). We report how existing cross-validation techniques can be used to obtain regret rates in a stochastic case where the learner is unaware of smoothness parameters and discuss corresponding limitations in non-stochastic settings. We then demonstrate that LCB allows building a robust version of uniform allocation ROBUNI for non-stochastic environments[8].

Stochastic feedback To determine regret rates, we distinguish two scenarios depending on the knowledge of the smoothness parameters. First, the uniform strategy exploits (ν, ρ), while, second, the learner is oblivious to (ν, ρ).

$\circ$ *With knowledge of* (ν, ρ)*:* At depth h a uniform algorithm can explore all K^h cells $\lfloor n/K^h \rfloor$ times. Such a strategy recommends a valid parameter x that attains the highest observed $\widehat{f}(x)$. At depth h, errors are bounded by $\nu\rho^h$, and the estimation error is given by $\mathcal{O}\left(\sqrt{K^h/n}\right)$. Optimising over h for the sum of these two errors, we can state that by setting $H = \left\lfloor \log_{K/\rho^2}(n) \right\rfloor$, $r_n = \widetilde{\mathcal{O}}(\log(1/\delta)/n)^{\frac{1}{\log K}{\log 1/\rho}+2}$ with probability at least $1-\delta$.

$\circ$ *Without knowledge of* (ν, ρ)*:* So far, we derived a bound where the optimal choice of H is dependent on smoothness parameters (ν, ρ). When not knowing (ν, ρ), our strategy uses a budget of $n/2$ rounds to explore all depths $h \in [0 : \lfloor \log_K(n) \rfloor]$. A depth h is explored uniformly with a budget of $n/(2\lfloor \log_K(n) \rfloor)$. We define $\lfloor \log_K(n) \rfloor$ candidates x_h with the highest observed $\widehat{f}(x_h)$ among the cells of depth h. Now, the final recommendation corresponds to a choice between these $\lfloor \log_K(n) \rfloor$ candidates. However, each has been pulled $T_{x_h}(n/2) = \frac{n/(2\lfloor \log_K(n) \rfloor)}{K^h}$ number of times and as such, arrives with different confidence estimates. We can implement a *cross validation* step, as used in Bartlett et al. (2019), which only requires n to make use of the remaining $n/2$ rounds. Each candidates is sampled additionally $n/(2\lfloor \log_K(n) \rfloor)$ This leads us to obtain $r_n = \widetilde{\mathcal{O}}\left(\left(\frac{K}{n\rho^2}\right)^{\frac{1}{\log K}{\log 1/\rho}+2}\right)$. With this strategy, we recover the same results as if smoothness parameters were provided up to a logarithmic factor. An alternative to cross validation with same theoretical guaranties is, after a uniform allocation on all cells at a depth smaller than $\lfloor \log_K(n) \rfloor$, to recommend among all cells these with largest *lower confidence bound* $\widehat{f}_{h,i}(n) - b\sqrt{\frac{\log(n^2/\delta)}{T_{h,i}(n)}}$. This allows to compare candidate cells at different depths by taking into account the uncertainty $b\sqrt{\frac{\log(n^2/\delta)}{T_{h,i}(n)}}$ around their estimated averages. Though this approach requires the knowledge of b (the range of ε_t), it will come handy in the non-stochastic setting detailed next.

Parameters: $\mathcal{P} = \{\mathcal{P}_{h,i}\}$, $b, n, f_{\max}$

Set $\delta = \frac{4b}{f_{\max}\sqrt{n}}$.
For $t = 1, \ldots, n$ ◀ *Exploration* ▶
Evaluate a point x_t sampled from $U_{\mathcal{P}}(\mathcal{P}_{0,1})$.

Output $x(n) \sim \mathcal{U}(\mathcal{P}_{h(n),i(n)})$
where $(h(n), i(n)) \leftarrow \arg\max_{h,i} \widetilde{F}_{h,i}(n) - B_h^{adv}(n)$

Figure 1: The ROBUNI algorithm

Non-stochastic feedback As discussed above, in the non-stochastic setting we use a uniform allocation combined with a recommendation based lower confidence estimate of the value of cell $\mathcal{P}_{h,i}$ as $\widetilde{F}_{h,i}(n) - B_h^{adv}(n)$ where $B_h^{adv}(n) \triangleq \sqrt{2nf_{max}^2 K^h \log(n^2/\delta)} - \frac{f_{max}^2}{3} K^h \log(n^2/\delta)$. We name such an algorithm ROBUNI and detail its pseudo-code in Figure 1. ROBUNI is required knowledge of b (See Equation 2), and f_{max} that upper bounds the maximal value of the functions $f_1, \ldots, f_n$, i.e., $|f_t(x)| \leq f_{max}$ for all $x \in \mathcal{X}$ and $t \in \{1, \ldots, n\}$.

We are now ready to present the simple regret bounds attained by ROBUNI in the following theorem:

Theorem 1 (Upper bounds for ROBUNI)**.** *Consider any sequence of functions $f_1, \ldots, f_n$ such that $|f_t(x)| \leq f_{max}$ for all $x \in \mathcal{X}$ and $t \in [n]$. Let $f = \frac{1}{n}\sum_{t=1}^n f_t$, and $x^\star$ be one of the global optima of f with associated (ν, ρ). Then after n rounds, the simple regret of* ROBUNI *is bounded as:*

$$\mathbb{E}[r_n] = \mathcal{O}\left(\log(n/\delta)\left(\frac{K}{n\rho^2}\right)^{\frac{1}{\log K}{\log 1/\rho}+2}\right)$$

The above result demonstrates that using ROBUNI uniform exploration strategies can be made order-robust retaining same regret guarantees in the non-stochastic setting as those obtained in the stochastic case. However, we conjecture that this is not true for most learners, where we believe that any algorithm can only obtain, at best, the same regret rates as ROBUNI in non-stochastic cases. This is not unlike best-arm identification problems(when $\mathcal{X}$ is reduced to $\mathcal{X} = [K]$), where the authors in (Abbasi-Yadkori

[7]Note that as the above exposition considers no stochasticity. As such, answers to **Challenge I** are considered in later sections when attempting to determine a best of both worlds algorithm.

[8]We report the complete proofs in Appendix A.

	$b = 0$	stochastic ($b > 0$)	non-sto
VROOM	**open**	$\max\left(\frac{1}{d+3}, \frac{1}{\frac{\log K}{\log \frac{1}{\rho}}+2}\right)$ $\frac{1}{n}$	$\frac{1}{\frac{\log K}{\log \frac{1}{\rho}}+2}$ $\frac{1}{n}$
StroquOOL	$\left(\frac{1}{n}\right)^{\frac{1}{d}}$	$\left(\frac{1}{n}\right)^{\frac{1}{d+2}}$	✗
SequOOL	$\left(\frac{1}{n}\right)^{\frac{1}{d}}$	✗	✗
Uniform(s)	$\frac{1}{n}^{\frac{\log \frac{1}{\rho}}{\log K}}$	$1/n^{\frac{1}{\frac{\log K}{\log 1/\rho}+2}}$	

Table 1: $\widetilde{\mathcal{O}}$ rates of SOTA in deterministic, stochastic and non-stochastic settings. ✗ denotes a non-vanishing regret. Though VROOM stochastic bounds can be applied when $b = 0$, we leave a better bound open direction of future research.

et al., 2018) showed unimprovable regret rates to those obtained by uniform strategies.

3.2 Achievable Rates for BOB

Though the uniform exploration algorithm discussed above achieves order-robustness in non-stochastic settings, it can become highly sub-optimal for stochastic scenarios. In fact, it is well known that for stochastic data generating processes, StroquOOL and GPO obtain a state-of-the-art simple regret of the order $\widetilde{\mathcal{O}}\Big(\big(\frac{1}{n}\big)^{\frac{1}{d+2}}\Big)$. Yet, as detailed in Section 2, one can design a sequence of functions (i.e., non-stochastic scenario) $f_1, \dots, f_n$ with any associated parameter d, ν, ρ such that simple regret of StroquOOL, for instance, is lower bounded by a constant for any n.

Given the lack of algorithm performing well in both scenarios, we next attempt to design a learner that is unaware of the nature of the reward-generating process but simultaneously achieves near-optimal simple regret bounds, i.e.,

$$\mathbb{E}[r_n] = \widetilde{\mathcal{O}}\left(\left(\frac{1}{n}\right)^{\frac{1}{d+2}}\right) \quad \text{(stochastic feedback)}$$

$$\mathbb{E}[r_n] = \widetilde{\mathcal{O}}\left(\left(\frac{1}{n}\right)^{\frac{1}{\frac{\log(K)}{\log(1/\rho)}+2}}\right) \quad \text{(non-stochastic feedback)}$$

Rates of Optimality: To understand the optimality statements that can be considered when tackling both scenarios, we draw upon results from best-arm identification problems, i.e., when $\mathcal{X} = [K]$. There, Abbasi-Yadkori et al. (2018) showed that obtaining optimal rates in stochastic and non-stochastic cases simultaneously is impossible. We conjecture that this result carries to our setting, where we believe simultaneous optimal rates are also not achievable.

This, consequently, poses the question of what type of optimal rates can an algorithm obtain in stochastic feedback settings, while still guaranteeing vanishing regrets in non-stochastic cases. A formal lower bound guarantee of optimality is beyond the scope of this paper, and is left as an open question for future research. We do, however, demonstrate VROOM to be the first algorithm acquiring vanishing regrets in non-stochastic scenarios, while still achieving favourable rates compared to state-of-the-art stochastic algorithms, i.e., $\mathbb{E}[r_n] = \widetilde{\mathcal{O}}\Big(n^{-\frac{1}{d+3}}\Big)$.

Parameters: $\mathcal{P} = \{\mathcal{P}_{h,i}\}, b, n, f_{\max}$
Set $\delta = \frac{4b}{f_{\max}\sqrt{n}}$.

For $t = 1, \dots, n$ ◀ *Exploration* ▶

For each depth $h \in [\lfloor \log_K(n) \rfloor]$, rank[a] the cells by decreasing order of $\widehat{f^-_{h,i}}(t-1)$: Rank cell $\mathcal{P}_{h,i}$ as $\widehat{\langle i \rangle}_{h,t}$.

$x_t \sim \mathcal{U}_{\mathcal{P}}(\mathcal{P}_{h_t,i_t})$ where $\mathcal{P}_{h_t,i_t}$ is sampled so that for any $h \in [\lfloor \log_K(n) \rfloor]$ and any $i \in [K^h]$,

$$\boldsymbol{p}_{h,i,t} \triangleq \mathbb{P}(\mathcal{P}_{h_t,i_t} = \mathcal{P}_{h,i}) \triangleq \frac{1}{h\widehat{\langle i \rangle}_{h,t}\overline{\log}_K(n)}$$

and where $\overline{\log}_K(n) = \sum_{h=1}^{\lfloor \log_K(n) \rfloor} \sum_{i=1}^{K^h} \frac{1}{hi}$.

Output $x(n) \sim \mathcal{U}_{\mathcal{P}}(\mathcal{P}_{h(n),i(n)})$ where $(h(n), i(n)) \leftarrow \arg\max_{(h,i)} \widetilde{F}_{h,i}(n) - B_{h,i}(n)$

[a] Equalities between cells or comparisons with cells that have not been pulled yet are broken arbitrarily.

Figure 2: The VROOM algorithm

3.3 Robust optimisation

In this section, we present a new learner and analyse its theoretical performance against any i.i.d. stochastic problem or any non-stochastic environment.

This Very Robust Online Optimisation Method, VROOM, is detailed in Figure 2. Intuitively, VROOM first selects a depth h with a probability inversely proportional h. Given its depth selection, VROOM queries the best estimated cell with "probability" one, the second-best estimated cell with a "probability" of one half, and so on until pulling the worst-estimated cell with a "probability" $\frac{1}{K^h}$. To guarantee valid probabilities, we need a normalisation factor. As it is sufficient to sample depths $h \in [\lfloor \log_K(n) \rfloor]$, the normalising constant can be bounded as:

$$\overline{\log}_K(n) = \sum_{h=1}^{\lfloor \log_K(n) \rfloor} \sum_{i=1}^{K^h} \frac{1}{hi} \leq \sum_{h=1}^{\lfloor \log_K(n) \rfloor} \frac{1}{h}(\log(K^h) + 1)$$
$$\leq 2 \sum_{h=1}^{\lfloor \log_K(n) \rfloor} \log(K) \leq 2\log(n).$$

At round t, the estimate used in VROOM to rank the cell during exploration is given by $\widehat{f^-_{h,i}}(t-1) \triangleq \widehat{f}_{h,i}(t-1) - B^{iid}_h(t-1)$ for cell $\mathcal{P}_{h,i}$, where $B^{iid}_h(t-1) = \sqrt{\frac{\log(4n^3/\delta)}{2T_{h,i}(t-1)}}$

with $f^-_{h,i}(t-1)$ set to $-\infty$ if $T_{h,i}(t)=0$. Following this ranking procedure, we denote the estimated rank of cell i at depth h at time t as $\widehat{\langle i \rangle}_{h,t}$. After n rounds, VROOM recommends the element $x(n)$ sampled uniformly from the estimated best cell $\mathcal{P}_{h(n),i(n)}$. Recommendation of the best cell $\mathcal{P}_{h(n),i(n)}$ after n rounds is based on $\widetilde{F}_{h,i}(n) - B_{h,i}(n)$ where $B_{h,i}(t)$ defines the confidence bound around our estimate $\widetilde{F}_{h,i}(t)$. For all $t \in [n]$, such a bound is given by:

$$B_{h,i}(t) = f_{\max}\sqrt{2h\overline{\log}_K(n)\log 2n^2/\delta \sum_{s=1}^{t} \widehat{\langle i \rangle}_{h,s}} + f_{\max}\overline{\log}_K(n)\frac{\log 2n^2/\delta}{3}.$$

One can view the sampling strategy of VROOM as a randomised version of that introduced in StroquOOL (Bartlett et al., 2019). Essentially, it implements a Zipf exploration (Powers, 1998) meaning that it first ranks the different options (here cells), and then attempts to allocate evaluations inversely proportional to their rank. We note that such a strategy has also been used in previous algorithms, e.g., Successive Rejects (SR) of Audibert et al. (2010) and P1 of Abbasi-Yadkori et al. (2018).

To minimise simple regret in the stochastic case, it is crucial to limit the variance of the best-cell estimators. Therefore VROOM, from its very first pull, chooses with higher probability the cells that are estimated to be among the best. This comes with almost no additional cost. Indeed, at depth h, pulling the estimated best cell with probability $1/(h\overline{\log}_K(n))$ does not prevent from pulling all the cells almost uniformly. More precisely, for any $k \in [K^h]$ all cells ranked below k, i.e., $\widehat{\langle i \rangle}_{h,t} \leq k$, are pulled with a probability of at least $1/(hk\overline{\log}_K(n))$. Therefore, no suboptimal cell is actually left out in the early chase for a cell containing $x^\star$. Hence, the variances of the estimators can only increase by a factor of $\overline{\log}_K(n)$ w.r.t. the uniform strategy.

Additionally, compared to a fixed-phase algorithm, our analysis is also more flexible. In fact, we can analyse, for instance, the quality of the estimated ranking $\widehat{\langle \cdot \rangle}$ and, consequently, the adaptive sampling procedure of the arms at any round. Actually, these rounds can be chosen in a problem-dependent fashion, to minimise the final regret[9].

Remarkably, VROOM uses a lower confidence bound (LCB) to guide exploration and recommendation. As mentioned earlier, this allows us to compare cells at different and within the same depth by taking into account the uncertainty $b\sqrt{\frac{\log(n^2/\delta)}{T_{h,i}(n)}}$ around their estimated averages. For recommendation, this replaces the use of hand-coded cross-validation techniques. For exploration, the use of LCB needs to be handled carefully. For instance, implementing a *pessimism in front of uncertainty* that pulls the cell with the highest LCB would likely result in exclusively pulling one single arm as such bounds increase with the number of pulls. However, LCB are found to combine well with a Zipf sampler that guarantees the estimated k best cells are pulled with an order of $\widetilde{\mathcal{O}}(n/k)$ almost uniformly.

Interestingly, we demonstrate that potentially biased estimates $\widehat{f}_{h,i}$ can be used to guide exploration as long uniform exploration is guaranteed for all arms. This helps to overcome high variances (in the stochastic case) that the unbiased estimate $\sum_{t=1}^{n} \widetilde{f}_{h,i}(t)$ possess and allows us to guaranty that cells containing $x^\star$ are well ranked soon enough. After n rounds, however, we use the unbiased estimate $\sum_{t=1}^{n} \widetilde{f}_{h,i}(t)$ to recommend $x(n)$. Being unbiased, our estimates are robust to non-stationary data. Moreover, it is also possible to prove that cells containing a $x^\star$ which have been pulled enough now possess a limited variance in the stochastic setting.

Let us now present our main results for both stochastic and non-stationary data-generating process using VROOM:

Theorem 2 (Upper bounds for VROOM)**.** *In the non stochastic setting, for any sequence of functions $f_1, \ldots, f_n$ with $f = \frac{1}{n}\sum_{t=1}^{n} f_t$, we have, after n rounds, the simple regret of* VROOM *is bounded as follows:*

$$\mathbb{E}[r_n] = \mathcal{O}\left(\log(n/\delta)/n^{\frac{1}{\frac{\log K}{\log 1/\rho}+2}}\right)$$

Moreover in the stochastic setting, let $x^\star$ one of the global optimum of $\overline{f}$ with associated (ν, ρ), $C > 1$, and near-optimality dimension $d = d(\nu, C, \rho)$. Then we have,

$$\mathbb{E}[r_n] = \widetilde{\mathcal{O}}\left(\frac{1}{n}\right)^{\max\left(\frac{1}{d+3}, \frac{1}{\frac{\log K}{\log 1/\rho}+2}\right)}$$

where the expectation is taken both other ν_n and the random generation of f with respect to $\overline{f}$.

It is worth noting that the exponent obtained in the stochastic setting is $\max\left(\frac{1}{d+3}, \frac{1}{\frac{\log K}{\log 1/\rho}+2}\right)$. As mentioned in Section 2, in general we have $\frac{1}{d+2} \gg \frac{1}{\frac{\log K}{\log 1/\rho}+2}$. Therefore in most cases the exponent in the rate of VROOM is $\frac{1}{d+3}$ and is never worst than the one of uniform allocations $\frac{1}{\frac{\log K}{\log 1/\rho}+2}$.

Sketch of proof: In the non-stochastic setting, we use the fact that VROOM pulls at any depth h all the cells almost uniformly, of order $1/(hK^h)$ up to logarithmic factors, to obtain the same rate as ROBUNI.

For the stochastic case, we face **Challenge I**. Indeed, VROOM uses for recommendation the estimates $\widetilde{F}_{h,i}(n)$ for cell $\mathcal{P}_{h,i}$. Consequently, we need to carefully bound the

[9] We detail the process by which such rounds are chosen in the sketch of the proof in Section 3.2.

variance of $\widetilde{F}_{h,i}(n) = \sum_{t=1}^{n} \frac{f_{h,i,t} 1_{x_t \in \mathcal{P}_{h,i}}}{\boldsymbol{p}_{h,i,t}}$ for the cells that are near-optimal. To limit the variance, our algorithm has then two objectives, first identify a deep cell containing $x^\star$ and then pull this cell enough so that the variance of its estimate is low. Intuitively we follow the idea developed for the stochastic case in Section 3.1 that an algorithm which does not know the smoothness parameter (ν, ρ) can divide its budget of n rounds into two consecutive parts, one for each objective: First explore $\mathcal{P}$ for n^α rounds with $\alpha < 1$ in order to build a small number of good candidate cells $\mathcal{C}$ and then secondly cross validate, meaning allocate the rest of the budget, $n - n^\alpha$ rounds, to estimate better and compare this limited number of candidates in $\mathcal{C}$.

We identify two sources of errors. First the exploration error, e_r is the smallest simple regret among the candidate recommended at the end of the exploration phase after n^α rounds. Following Locatelli and Carpentier (2018) we have $e_r = \Omega\big(n^{-\frac{\alpha}{d+2}}\big)$. The second error, the cross validation error, e_c is the confidence interval of $\widetilde{F}_{h,i}(n)/n$ where our final recommendation $x(n)$ is in cell $\mathcal{P}_{h,i}$. Assuming we cannot guaranty the candidates are pulled more than uniformly during the exploration phase of n^α rounds, we obtain, at time n, $e_c = \mathcal{O}(n^\alpha/n)$.[10] Simultaneously we want large α to increase the length the exploration phase and reduce the simple regret of our candidates and small α to reduce the variance of our final estimates. Equaling both source of error we get that $n^{-\alpha/(d+2)} = n^{\alpha-1}$ gives $\alpha = (d+2)/(d+3)$ which leads to a regret $\mathcal{O}(n^{-\frac{1}{d+3}})$.

VROOM is implementing implicitly such a strategy without explicitly considering two separate phases and without the knowledge of d. In the stochastic setting, as discussed above we can study the quality of the estimated ranking at any point in time $t \in [n]$. We divide the n_α in parts $n_1 = 1, n_2, \dots, n_{\log(n_\alpha)}$ and analyse the ranking of cell $\mathcal{P}_{l,i^\star}$ at the end of round n_l for $l \in [n_{\log(n_\alpha)}]$. To analyse the ranking of the cell $\mathcal{P}_{h,i}$ we use Lemma 2 that provides conditions on h such that we can guarantee that after round n_l, $t \geq n_l$ the ranking of $\mathcal{P}_{h,i}$ verifies $\langle i^\star \rangle_{l,t} \lesssim C\rho^{-dl}$. Then Lemma 1 shows that the confidence interval around the average estimate of that cell is $n_\alpha^{\frac{1}{d+2}}$.

4 Related Work

BOB A best of *both* world question has already been addressed by Abbasi-Yadkori et al. (2018) in a more reduced optimisation problem where $\mathcal{X} = [K]$ is composed of a finite number of K elements known as the best-arm identification (BAI) problem (Bubeck et al., 2009). They propose P1, an algorithm that achieves, in the stochastic setting, the optimal simple regret rate that any algorithm, with vanishing simple regret in the non-stochastic setting, can achieve.

[10] Alternatively one can bound e_c by recommending with estimates as $\sum_{t=n_\alpha+1}^{n} \widetilde{f}_{h,i}(t)$ which bias w.r.t. $F_{h,i}(n)$ is $\mathcal{O}(n^\alpha)$.

Prior work for stochastic and deterministic cases Among the large work on derivative-free optimisation, we focus on algorithms that perform well under *minimal* assumptions as well as minimal knowledge about the function. While some prior works assume a *global* smoothness of the function (Pintér, 1996; Strongin and Sergeyev, 2000; Hansen and Walster, 2003; Kearfott, 2013), another line of research assumes only a *weak/local* smoothness around one global maximum (Auer et al., 2007; Kleinberg et al., 2008; Bubeck et al., 2011). However, within this latter group, some algorithms require the knowledge of the local smoothness such as HOO (Bubeck et al., 2011), Zooming (Kleinberg et al., 2008), or DOO (Munos, 2011). Among the works relying on an *unknown* local smoothness, SequOOL (Bartlett et al., 2019) improves on SOO (Munos, 2011; Kawaguchi et al., 2016) and represents the state-of-the-art for the deterministic feedback. For the stochastic feedback, StoSOO (Valko et al., 2013) extends SOO for a limited class of functions. POO (Grill et al., 2015) and GPO (Shang et al., 2019) provides more general results. StroquOOL (Bartlett et al., 2019) combines up to log factors the guarantees of SequOOL and GPO for deterministic and stochastic feedback respectively without the knowledge of the range of the noise b.

5 Discussion and Future Work

Our current result holds simultaneously for stochastic and non-stochastic settings. However, it is desirable to also consider the deterministic feedback where evaluations are noiseless and stationary, that is $\forall t \in [n]$, $f_t = f_1$. Please refer to the work by de Freitas et al. (2012) for a motivation, many applications, and references on the importance of this case. The question of obtaining the best of the three worlds (BOT) which includes additionally the deterministic setting remains open. Note that StroquOOL, for instance, was able to obtain theoretical guarantees that hold for stochastic and deterministic case settings simultaneously by having a method that adapts to the level of noise b without its knowledge. However, VROOM requires the knowledge of b and f_{max} to build the lower confidence bound used for recommendation. To address the BOT question, computing higher moments of our estimates and therefore using concentration inequalities such as the one in the work by Cappé et al. (2013) is a potential direction. We also wonder if a version of VROOM that is fully using unbiased estimates can solve BOB, while VROOM uses the $\widehat{f}^-$ estimates to guide exploration. and is, therefore, over-fitting the stochastic case. Finally, fully answering the BOT question may require investigating lower bounds results.

References

Abbasi-Yadkori, Y., Bartlett, P., Gabillon, V., Malek, A., and Valko, M. (2018). Best of both worlds: Stochastic & adversarial best-arm identification. In *Conference on Learning Theory*.

Ammar, H. B., Eaton, E., Ruvolo, P., and Taylor, M. (2014). Online multi-task learning for policy gradient methods. In *International Conference on Machine Learning*, pages 1206–1214.

Audibert, J.-Y., Bubeck, S., and Munos, R. (2010). Best arm identification in multi-armed bandits. *Conference on Learning Theory*.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422.

Auer, P., Ortner, R., and Szepesvári, C. (2007). Improved rates for the stochastic continuum-armed bandit problem. In *Conference on Learning Theory*.

Azar, M. G., Lazaric, A., and Brunskill, E. (2014). Online stochastic optimization under correlated bandit feedback. In *International Conference on Machine Learning*.

Bartlett, P. L., Gabillon, V., and Valko, M. (2019). A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. In *Algorithmic Learning Theory*, pages 184–206.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122.

Bubeck, S., Lee, Y. T., and Eldan, R. (2017). Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 72–85. ACM.

Bubeck, S. and Munos, R. (2010). Open-loop optimistic planning. In *Conference on Learning Theory*.

Bubeck, S., Munos, R., and Stoltz, G. (2009). Pure exploration in multi-armed bandit problems. In *Conference on Algorithmic Learning Theory (ALT)*, pages 23–37.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. (2011). X-armed bandits. *Journal of Machine Learning Research*, 12:1587–1627.

Cappé, O., Garivier, A., Maillard, O.-A., Munos, R., Stoltz, G., et al. (2013). Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541.

Coquelin, P.-A. and Munos, R. (2007). Bandit algorithms for tree search. In *Uncertainty in Artificial Intelligence*.

de Freitas, N., Smola, A., and Zoghi, M. (2012). Exponential regret bounds for Gaussian process bandits with deterministic observations. In *International Conference on Machine Learning*.

Freedman, D. A. (1975). On tail probabilities for martingales. *The Annals of Probability*, pages 100–118.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Grill, J.-B., Valko, M., and Munos, R. (2015). Black-box optimization of noisy functions with unknown smoothness. In *Neural Information Processing Systems*.

Hansen, E. and Walster, G. W. (2003). *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press.

Hazan, E. et al. (2016). Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

Hoorfar, A. and Hassani, M. (2008). Inequalities on the lambert w function and hyperpower function. *Journal of Inequalities in Pure and Applied Mathematics (JIPAM)*, 9(2):5–9.

Hren, J.-F. and Munos, R. (2008). Optimistic Planning of Deterministic Systems. In *European Workshop on Reinforcement Learning*.

Kawaguchi, K., Maruyama, Y., and Zheng, X. (2016). Global continuous optimization with error bound and fast convergence. *Journal of Artificial Intelligence Research*, 56:153–195.

Kearfott, R. B. (2013). *Rigorous global search: continuous problems*, volume 13. Springer Science & Business Media.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Kleinberg, R., Slivkins, A., and Upfal, E. (2008). Multi-armed bandit problems in metric spaces. In *Symposium on Theory Of Computing*.

Kocsis, L. and Szepesvári, C. (2006). Bandit-based Monte-Carlo planning. In *European Conference on Machine Learning*.

Locatelli, A. and Carpentier, A. (2018). Adaptivity to Smoothness in X-armed bandits. In *Conference on Learning Theory*.

Matyas, J. (1965). Random optimization. *Automation and Remote control*, 26(2):246–253.

Maurer, A. and Pontil, M. (2009). Empirical bernstein bounds and sample variance penalization. In *Conference on Learning Theory*.

Munos, R. (2011). Optimistic optimization of deterministic functions without the knowledge of its smoothness. In *Neural Information Processing Systems*.

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):1–130.

Nesterov, Y. and Spokoiny, V. (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*.

Pintér, J. D. (1996). *Global Optimization in Action. Continous and Lipschitz Optimization: Algorithms, Implementations and Applications*. Kluwer Academic Publishers: Boston.

Powers, D. (1998). Applications and explanations of Zipf's law. In *New methods in language processing and computational natural language learning*. Association for Computational Linguistics.

Shang, X., Kaufmann, E., and Valko, M. (2019). General parallel optimization without metric. In *Algorithmic Learning Theory*.

Strongin, R. and Sergeyev, Y. (2000). *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*. Nonconvex Optimization and Its Applications. Springer.

Thrun, S. and Mitchell, T. M. (1995). Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46.

Valko, M., Carpentier, A., and Munos, R. (2013). Stochastic simultaneous optimistic optimization. In *International Conference on Machine Learning*.

Yoon, J., Kim, S., Yang, E., and Hwang, S. J. (2019). Scalable and order-robust continual learning with hierarchically decomposed networks. *arXiv preprint arXiv:1902.09432*.

Zinkevich, M. (2003). Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936.