A Bayesian Approach for Bandit Online Optimization with Switching Cost

Zai Shi[1] **Jian Tan**[1] **FeiFei Li**[1]

[1]Alibaba Group, Hangzhou, Zhejiang, China

Abstract

As a classical problem, online optimization with switching cost has been studied for a long time due to its wide applications in various areas. However, few works have investigated the bandit setting where both the forms of the main cost function $f(x)$ evaluated at state x and the switching cost function $c(x, y)$ of transitioning from state x to y are unknown. In this paper, we consider the situation when $(f(x_t) + \varepsilon_t,\ c(x_t, x_{t-1}))$ can be observed with noise ε_t after making a decision x_t at time t, aiming to minimize the expected total cost within a time horizon. To solve this problem, we propose two algorithms from a Bayesian approach, named Greedy Search and Alternating Search, respectively. They have different theoretical guarantees of competitive ratios under mild regularity conditions, and the latter algorithm achieves a faster running speed. Using simulations of two classical black-box optimization problems, we demonstrate the superior performance of our algorithms compared with the classical method.

1 INTRODUCTION

Inspired by applications in online control [Goel et al., 2017], networking [Lin et al., 2012], video streaming [Joseph and de Veciana, 2012], power generation planning [Kim et al., 2015] and various other ones [Kim et al., 2015, Goel et al., 2017], the problem on the online optimization with switching costs has attracted increasing attention. It is a sequential decision problem where the decision-maker makes a decision x_t at each time slot t and then receives a main cost $f_t(x_t)$ associated with this decision. Meanwhile, an additional cost, called switching cost $c(x_t, x_{t-1})$, is also revealed, which measures the cost of transitioning from the last state x_{t-1} to the current state x_t. The objective is to minimize the total cost within a fixed time horizon.

The basic version of this problem was introduced in Lin et al. [2012] with applications of scheduling in data centers. Since then, many algorithms have been proposed for problems with more complex settings. For more related works, please see Section 1.1.1 for details. Previous works predominantly assume that the forms of $f_t(\cdot)$ and $c(\cdot, \cdot)$ are already known at time t before choosing x_t. In some applications, however, we can only get access to the observed values $(f_t(x_t), c(x_t, x_{t-1}))$ after making a decision to select x_t at time t, which is called a bandit setting [Hazan et al., 2016]. Particularly in this paper, we assume that $f_t(x) = f(x) + \varepsilon_t$, where $\{\varepsilon_t\}$ is a sequence of i.i.d. zero-mean additive random noises when observing $f(x)$ at time t. Our objective is to minimize the expected total cost, i.e., $\sum_{t=1}^{T} (f(x_t) + c(x_t, x_{t-1}))$ within a time horizon $[1, T]$. We will show some examples of this setup in Section 2. The black-box functions $f(\cdot)$ and $c(\cdot, \cdot)$ can be highly non-convex, and thus, zeroth-order methods using gradient estimators [Liu et al., 2020] are not suitable herein for global optimization.

To tackle this challenge, we propose two methods based on Bayesian optimization (BO) [Frazier, 2018] with theoretical guarantees for a class of functions under mild conditions. The contributions of our paper are as follows.

- Inspired by the idea of greedy algorithms, we first propose Greedy Search (GS) with a theoretical guarantee in terms of competitive ratio, a common performance metric for online optimization, shown in Table 1. The ratio approaches a dimension-free constant as $T \to \infty$ under a mild condition. Even though this algorithm is easy to implement, it needs to operate on a higher dimension than the original problem, thus resulting in increased running time. It also requires additional properties of f and c for its theoretical guarantee as shown in Table 1. GS is discussed in Section 4.
- To mitigate the above issues, we propose Alternating Search (AS) with a novel structure, which switches be-

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Algorithms	Competitive Ratio	ψ Value	Requirements
IGP-UCB (Chowdhury and Gopalan)	$1+\tilde{O}(T^{g(d)-1})$ (Without switching)	N.A.	No
Greedy Search (Our paper)	$\psi+\tilde{O}(T^{g(2d)-1})$	$\max\{\frac{1+\eta^2/\lambda}{1-\eta^2/\lambda}, \frac{\eta}{1-\eta^2/\lambda}\}$	$\eta^2/\lambda<1$
Alternating Search (Our paper)	$\psi+\tilde{O}(T^{(g(d)-1)/2})$	$\max\{1+2\eta^3/\lambda, 2\eta^2\}$	No

Table 1: Theoretical results of our proposed algorithms in terms of competitive ratio. $\tilde{O}$ means big O neglecting the $\log$ term. Here IGP-UCB is a classical BO algorithm listed as a benchmark by writing its regret bound as $\tilde{O}(T^{g(d)})$ for some function g of the domain dimension d, whose details can be found in Lemma 1. Note that the result of IGP-UCB is for single black-box function without switching cost. η and λ are two parameters related to properties of f and c shown in Assumption 2 and 3.

tween two optimization strategies at deliberately chosen time slots. It achieves a competitive ratio shown in Table 1, which also tends to a dimension-free constant as $T\to\infty$, under a condition even milder than GS. Meanwhile, each step of AS has a running time faster than GS, as is discussed in Section 5.

- Through simulations of two classical online control problems, we demonstrate the theoretical findings in our paper, as shown in Section 6.

To the best of our knowledge, our algorithms are the first ones with theoretical guarantees for bandit online optimization problems with switching costs.

1.1 RELATED WORKS

This section reviews related works on online optimization with switching cost in Section 1.1.1 and Bayesian optimization in Section 1.1.2.

1.1.1 Online Optimization with Switching Cost

The fundamental version of online optimization with switching cost is Smoothed Online Convex Optimization (SOCO), where the main cost function f_t is convex and f_t is available before making a decision at time t. It was first introduced in Lin et al. [2012] with the background of dynamic power management in data centers. Afterwards, this setup has found applications in many other areas including speech animation [Kim et al., 2015], multi-timescale control [Goel et al., 2017], video streaming [Joseph and de Veciana, 2012], and power generation planning [Kim et al., 2015].

In general, the performance metric of an algorithm for online optimization with switching cost is given by competitive ratio in the form of $\frac{\sum_{t=1}^T f_t(x_t)+c(x_t,x_{t-1})}{\sum_{t=1}^T f_t(x_t^*)+c(x_t^*,x_{t-1}^*)}$, where $\{x_t\}_{t=1}^T$ are the output of the algorithm and $\{x_t^*\}_{t=1}^T$ are the optimal offline solution. In Lin et al. [2012], a 3-competitive algorithm was proposed for one-dimensional action spaces. An improved algorithm with a competitive ratio of 2 was introduced in Bansal et al. [2015] and was shown optimal in one dimension in Antoniadis and Schewior [2017]. Beyond one dimension, it was found that a dimension-free competitive ratio is possible when the main cost function has specific structures. Online Balanced Descent (OBD) introduced in Chen et al. [2018] was such an algorithm for polyhedral functions. In Goel et al. [2019], OBD was found to provide a dimension-free competitive ratio for strongly convex functions as well. The authors also gave an optimal algorithm called Regularized OBD for the strongly convex case.

All the above works considered the case where only 1-step prediction is available. When there are w-step predictions, i.e., $f_t,..,f_{t+w-1}$ are available before choosing x_t, some works proposed further improved algorithms by taking advantage of predictions. In Chen et al. [2015], the authors showed that a classical control algorithm called Receding Horizon Control cannot achieve a competitive ratio that tends to one as $w\to\infty$. Therefore, they proposed a method called Averaging Fixed Horizon Control that can fill this gap. When the main cost function is strongly convex with bounded gradients and the switch cost is quadratic, the authors of Li et al. [2020] proposed two algorithms whose competitive ratios decay exponentially in w without the need to solve sub-problems in each step. Non-convex main cost functions were considered in Lin et al. [2020], where several new algorithms were proposed that can achieve a $1+O(1/w)$ competitive ratio with different assumptions on f_t and c. When there is a feedback delay of predictions and the switching cost depends on multiple previous decisions in a nonlinear manner, the authors in Pan et al. [2022] proposed a method called Iterative Regularized OBD, which has a constant and dimension-free competitive ratio.

Note that all the above works do not consider the bandit setup. There is also a line of works [Dekel et al., 2014, Guha and Munagala, 2009, Agrawal et al., 1990, Koren et al., 2017] considering multi-armed bandits with switching cost. Different from their works, the decisions in our setup are chosen from a continuous state space instead of a finite set, which is more challenging.

1.1.2 Bayesian Optimization

Bayesian optimization (BO) is a global optimization method suitable for a black-box objective function f that is expen-

sive to evaluate the function value at a point x. It uses a Gaussian process (GP) with kernel $k(x, x')$ as a surrogate model of f to execute the optimization process. The main techniques of BO will be introduced in Section 3. Since our paper is theoretically focused, we will only introduce BO methods with theoretical results here.

Most of BO algorithms focus on the sequential optimization of a black-box function f within a compact set, where the observation values contain additive random noises. Under some regularity assumptions of f, the best known regret bounds of Upper Confidence Bound (UCB) [Chowdhury and Gopalan, 2017], Thompson Sampling (TS) [Chowdhury and Gopalan, 2017] and Expected Improvement (EI) [Gupta et al., 2022] types of BO methods scale as $\tilde{O}(\gamma_T\sqrt{T})$ within a time horizon T, where γ_T is called the maximal information gain between the noisy observation and the latent GP surrogate model given the past T observations. The details and the bounds of this term can be found in Vakili et al. [2021].

Beyond these classical algorithms, new algorithms with $\tilde{O}(\sqrt{\gamma_T T})$ have been proposed recently with far more complex implementations including SupKernelUCB [Valko et al., 2013], GP-ThreDS [Salgia et al., 2021], RIPS [Camilleri et al., 2021], BPE [Li and Scarlett, 2022] and so on, which are of more theoretical interests.

2 PROBLEM FORMULATION

Consider a class of sequential decision problems that aim to minimize

$$\sum_{t=1}^{T}[\mathbb{E}f_t(x_t) + c(x_t, x_{t-1})]$$

by choosing x_t at time t from a d-dimensional set $\mathcal{X}$ given an initial state x_0, where $f_t(x_t) = f(x_t) + \varepsilon_t$ with a zero-mean random variable ε_t implying $\mathbb{E}f_t(x_t) = f(x_t)$. We call $f(x)$ the main cost function and $c(x, y)$ the switching cost function from y to x. $f_t(x_t)$ can be regarded as the observation value of $f(x_t)$ at time t with random noise ε_t. In our setup, the forms of f and c are unknown to the decision-maker and their values at x can be observed separately after making a decision x. Particularly, we assume that

Assumption 1. *$\{\varepsilon_t\}_{t=1}^T$ are independent R-subGaussian random variables satisfying $\mathbb{E}\left[\exp\left[s(\varepsilon_t - \mathbb{E}[\varepsilon_t])\right]\right] \leq \exp(R^2 s^2/2)$ with zero mean, and $c(x, y)$ is observed without noise.*

This setup is common in online control problems of robotics, aerospace, etc.. In these problems, the relation between the main cost (often defined to be the negative value of a reward) and the corresponding controllable parameters is unknown to us. Meanwhile, the change of controllable parameters can result in a cost such as energy consumption, whose relation is also unknown to us. Both the main cost and the switching cost can be observed after a set of controllable parameters are chosen.

In Section 6, we will test our algorithms on two black-box control problems commonly used in existing BO studies, which satisfy all the assumptions of our general setup. In the robot pushing problem [Wang et al., 2018], we want to control robot hands to push objects to their goals. We can control the rotation, the pushing speed, the moving direction and the pushing time of the robot hands, and changing these parameters will consume the power of the robot, which forms the switching cost in this problem. We can set the total distances of the objects to their goals as the reward, and then the main cost is its negative value. Our objective is to minimize a weighted combination of the main cost and the switching cost within a time horizon.

In the lunar lander problem [Eriksson et al., 2019], we want to learn a controller for a lunar lander, whose controllable parameters include the positions, the angles and their time derivatives of firing booster engines. The reward is the total distance of each leg towards landing on a certain terrain of the moon. Here we need to consume the energy of the lunar lander if we change its controllable parameters. Similar to the first example, our setup can be applied to this problem.

In our paper, we will use *competitive ratio* (CR) as the performance metric of algorithms in our setup. As seen in Section 1.1.1, it is commonly used in online optimization with switching cost. It is defined as

$$CR = \frac{\sum_{t=1}^{T} f(x_t) + c(x_t, x_{t-1})}{\sum_{t=1}^{T} f(x_t^*) + c(x_t^*, x_{t-1}^*)} \tag{1}$$

in our setup, where $\{x_t^*\}_{t=1}^T$ are the optimal solution for $\sum_{t=1}^{T}[f(x_t) + c(x_t, x_{t-1})]$ with $x_0^* = x_0$.

In the following, we will propose two algorithms with certain performance guarantees of CR. Particularly, both two algorithms can achieve a constant, dimension-free CR as $T \to \infty$. Before introducing our algorithms, we will first provide some background knowledge of Bayesian optimization for ease of our discussion.

3 PRELIMINARY: HOW TO DEAL WITH BLACK-BOX FUNCTIONS

The first challenge of our setup is the black-box nature of $f(\cdot)$ and $c(\cdot, \cdot)$. Some previous methods on online optimization with switching cost are based on gradients, e.g., Li et al. [2020]. Then one possible direction is to use the zeroth-order version of these methods based on gradient estimators with function observations [Liu et al., 2020]. However, since f and c may be nonconvex in our setup, this direction is not applicable because gradient-based methods could converge to the local minima.

Algorithm 1 IGP-UCB

Input: Prior $GP(0, k(x, x'))$, parameters B, R, ω, δ, T.
for $t = 1, ..., T$ **do**
  Set $\beta_t = B + R\sqrt{2(\gamma_{t-1} + 1 + \log(1/\delta))}$.
  Choose $x_t = \arg\min_{x \in \mathcal{X}} \{\mu_{t-1}(x) - \beta_t \sigma_{t-1}(x)\}$.
  Obtain the noisy observation of $f(x_t)$.
  Get $\mu_t(x)$ and $\sigma_t(x)$ using (4) and (5).
end for
Output: $x_1, ..., x_T$.

In this paper, we adopt another technique called Bayesian optimization (BO) for black-box problems, which is widely used in applications like deep learning [Wu et al., 2019], database [Zhang et al., 2021] and robotics [Berkenkamp et al., 2021]. In the following, we will introduce the intuition of BO via its basic methods aiming to minimize a single black-box function f.

First, we put a Gaussian process (GP) prior $GP(\mu_0(x), k(x, x'))$ on f with a mean function μ_0 and a kernel function k, which serves as a surrogate model of f. The kernel function k measures the similarity of two points x, x' with a certain distance $||x - x'||$. Two popular choices of k are square exponential (SE) kernel and Matérn kernel, defined as

$$k_{\text{SE}}(x, x') = \exp(-\frac{||x - x'||^2}{2u^2}) \tag{2}$$

$$k_{\text{Mat}}(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)}(\frac{||x - x'||\sqrt{2\nu}}{u})^{\nu} B_{\nu}(\frac{||x - x'||\sqrt{2\nu}}{u}), \tag{3}$$

where $u > 0$ and $\nu > 0$ are hyperparameters and $B_{\nu}(\cdot)$ is the modified Bessel function. After observing a function value, we will update the posterior distribution of f based on the new observation. Now the question is how to choose the next query point under the current posterior distribution to get a solution efficiently.

In general, a BO method will choose the next point which minimizes the expectation of some utility function $u(x)$ with regard to the current posterior distribution. Different kinds of utility functions lead to different classes of BO methods. Please refer to Frazier [2018] for details. Here we will introduce one of these methods called IGP-UCB [Chowdhury and Gopalan, 2017], which is also a basis of our algorithms in later sections. Its procedure is shown in Algorithm 1.

IGP-UCB uses a linear combination of the mean function $\mu_t(x)$ and the variance function $\sigma_t(x)$ of the current posterior distribution to choose the next point, which can be regarded as the lower confidence bound (LCB) of f (use the upper confidence bound, i.e., UCB, if it is a maximization problem) at time t. In Algorithm 1, the weight parameter β_t reflects the exploration-exploitation trade-off in the optimization process. Here δ is a positive constant that will be specified later and γ_t is called maximal information gain up to time t related to the domain $\mathcal{X}$ and the kernel k [Chowdhury and Gopalan, 2017] with bounds provided in Vakili et al. [2021]. The updates of μ_t and σ_t in IGP-UCB are as follows

$$\mu_t(x) = k_t(x)^T (K_t + \omega I)^{-1} y_{1:t}, \tag{4}$$

$$\sigma_t^2(x) = k(x, x) - k_t(x)^T (K_t + \omega I)^{-1} k_t(x), \tag{5}$$

given the past chosen t points and their corresponding observations $(x_{1:t}, y_{1:t})$. The term ωI is added due to the subGaussian observation noise, where I is an identity matrix and ω is a positive constant specified later. $K_t = [k(x, x')]_{x, x' \in \{x_1, ..., x_t\}}$ is the kernel matrix at time t and $k_t(x) = [k(x_1, x), ..., k(x_t, x)]^T$ is a vector function of x. The theoretical performance of IGP-UCB is given as follows.

Lemma 1. *[Theorem 3 of Chowdhury and Gopalan [2017]] Assume that f lies in the reproducing kernel Hilbert space (RKHS) with kernel k, $||f||_k < B$, and f is observed with independent R-subGaussian noise. Then, running IGP-UCB for f with $\omega = 1 + 2/T$ leads to*

$$\sum_{t=1}^{T} [f(x_t) - f(x^*)] = O(B\sqrt{T\gamma_T} + \sqrt{T\gamma_T(\gamma_T + \log(1/\delta))})$$

with a probability of at least $1 - \delta$, where $x^* = \arg\min_{x \in \mathcal{X}} f(x)$.

Here $|| \cdot ||_k$ is RKHS norm of kernel k and $||f||_k < B$ constrains the complexity of f. The reader may refer to Rasmussen [2003] for the details of the RKHS theory. It is not easy to directly examine $||f||_k < B$ in practice. However, this assumption is the basis for almost all the theoretical results of BO methods mentioned in Section 1.1.2. In Lemma 1, γ_T is called the maximal information gain between the noisy observation and the latent GP surrogate model given the past T observations. With the bounds of γ_T for different kernels shown in Vakili et al. [2021], we can see that IGP-UCB leads to a sub-linear regret for most kernel functions. For ease of comparison between IGP-UCB and our proposed methods introduced later, we write $B\sqrt{T\gamma_T} + \sqrt{T}\gamma_T$ as $T^{g(d)}$ by neglecting the $\log$ term, where d is the dimension of domain and $g(d)$ is determined by the bounds of γ_T. Then the regret bound of IGP-UCB can be written as $\tilde{O}(T^{g(d)})$, which captures the bounds for most kinds of kernels used in practice. For example, when k is a Matérn kernel with parameter ν as in (3), we have $g(d) = \frac{\nu + d}{2\nu + d}$ [Vakili et al., 2021]. Note that we need to choose kernels with $g(d) < 1$ to get a sub-linear regret for IGP-UCB. The above notation of g is important, which will also be used

in the theoretical results of our proposed algorithms in the following two sections.

If f is lower-bounded by a positive constant C, then we can transfer the bound of Lemma 1 to the bound of CR as follows:

$$CR = \frac{\sum_{t=1}^{T} f(x_t)}{\sum_{t=1}^{T} f(x^*)} \leq \frac{\sum_{t=1}^{T} f(x^*) + \tilde{O}(T^{g(2d)})}{\sum_{t=1}^{T} f(x^*)}$$
$$\leq 1 + \frac{\tilde{O}(T^{g(d)})}{T \times C} = 1 + \tilde{O}(T^{g(d)-1}).$$

The existence of $C > 0$ is needed because if $f(x^*) = 0$, a CR value in (1) is not properly defined unless we add a small value to f. Here we can see that CR of IGP-UCB approaches 1 as $T \to \infty$ if $g(d) < 1$.

4 HOW TO DEAL WITH SWITCHING COST: GREEDY SEARCH

From the last section, we can see that theoretical results of classical BO methods are for fixed functions. However, in our setup, we are faced with a time-varying objective function that couples the decisions of the past, the presence and the future via switching cost. Therefore, our setup is far more complex which needs a carefully designed algorithm.

To that end, we first propose a simple method called Greedy Search (GS), which tries to find x_t that only minimizes $h_t(x) := f(x) + c(x, x_{t-1})$ at time t. It can be regarded as a greedy algorithm because it only cares about the next switching cost. Since classical BO methods are for fixed functions while h_t is time-varying and coupled with h_{t-1}, we choose to put a Gaussian process prior on a higher-dimensional function $h(x, x') := f(x) + c(x, x')$ and regard $w = (x, x')$ as the domain. Recall that x is a d-dimensional vector, then w is a $2d$-dimensional vector, and $h_t(x)$ is equivalent to $h(w) = f(w) + c(w)$ where the last d dimensions of w are x_{t-1}. In this way, the observations of $h_1(x_1), ..., h_t(x_t)$ can be transferred to the observations of $h(w)$ which is no longer time-varying. The implementation of this algorithm based on IGP-UCB is shown in Algorithm 2.

In this algorithm, $k'(w, w')$ is a $\mathbb{R}^{4d} \to \mathbb{R}$ kernel function and γ'_{t-1} is the maximal information gain up to time $t-1$ related to k'. The updates of $\mu'_t(w)$ and $\sigma'_t(w)$ are

$$\mu'_t(w) = (k'_t(w))^T (K'_t + \omega I)^{-1} y'_{1:t} \quad (6)$$
$$(\sigma'_t)^2(w) = k'(w, w) - (k'_t(w))^T (K'_t + \omega I)^{-1} k'_t(w), \quad (7)$$

where $y'_{1:t}$ are the past noisy observations of $h(w)$, and the definitions of $k'_t(w)$ and K'_t are similar to the ones in (4) and (5), but related to k'. The theoretical guarantee of this algorithm is based on the following lemma.

Lemma 2 (Theorem 2 of Chowdhury and Gopalan [2017]). *Assume that $h(w)$ lies in the RKHS space with kernel k' and $||h||_{k'} < B$. With Assumption 1 and the updates in* (6), (7), *we have*

$$|\mu'_{t-1}(w) - h(w)|$$
$$\leq (B + R\sqrt{2(\gamma'_{t-1} + 1 + \log(1/\delta))})\sigma'_{t-1}(w) \quad (8)$$

with probability at least $1 - \delta$ for any $w \in \mathcal{X} \times \mathcal{X}$.

Algorithm 2 Greedy Search (GS)

1: **Input:** Prior $GP(0, k'(w, w'))$ on $h(w)$, parameters B, R, ω, δ, T.
2: **for** $t = 1, ..., T$ **do**
3: Set $\beta'_t = B + R\sqrt{2(\gamma'_{t-1} + 1 + \log(1/\delta))}$.
4: Choose $x_t = \arg\min_{x \in \mathcal{X}} \{\mu'_{t-1}(x, x_{t-1}) - \beta'_t \sigma'_{t-1}(x, x_{t-1})\}$.
5: Obtain a noisy observation of $h(w_t)$ where $w_t = (x_t, x_{t-1})$.
6: Perform update to get $\mu'_t(w)$ and $\sigma'_t(w)$ using (6) and (7).
7: **end for**
8: **Output:** $x_1, ..., x_T$.

From (8), we know that

$$|\mu'_{t-1}(x, x_{t-1}) - h(x, x_{t-1})|$$
$$\leq (B + R\sqrt{2(\gamma'_{t-1} + 1 + \log(1/\delta))})\sigma'_{t-1}(x, x_{t-1})$$

with probability at least $1 - \delta$, where $h(x, x_{t-1}) = f(x) + c(x, x_{t-1})$. Therefore, in each step of Algorithm 2, we are minimizing the LCB of $f(x) + c(x, x_{t-1})$, which realizes the idea of GS via the GP surrogate model.

Now we address the following three assumptions for the theoretical results of GS.

Assumption 2. *For all $x \in \mathcal{X}$, there exists a positive constant λ such that $f(x) - f(x^*) \geq \lambda c(x, x^*)$, where x^* is the global minimum of f. $f(x^*) \geq 0$.*

Assumption 3. $c(x, z) \leq \eta(c(x, y) + c(y, z))$ *and* $c(x, z) \leq \eta(c(x, y) + c(z, y))$ *for some constant $\eta > 0$, where $x, y, z \in \mathcal{X}$.*

Assumption 4. $0 \leq c(x, y) \leq D$ *where $x, y \in \mathcal{X}$.*

Assumption 2 means that f is not flat around x^* (and it is allowed to tend to be flat as λ approaches 0). It also measures the benefit of finding the global minimum of f compared with the incur of switching cost, thus will impact the performance of our algorithms. One example of f satisfying Assumption 2 is the highly nonconvex Ackley function [Ackley, 1987] widely used as a test function for global optimization methods. Its two-dimensional shape is shown in Figure 1, where $x_i \in [-32.768, 32.768]$, $x^* = (0, 0)$ and $f(x^*) = 0$. If $c(x, x^*) = ||x - x^*||_2$, then we can set $\lambda = 0.1$ in Assumption 2.

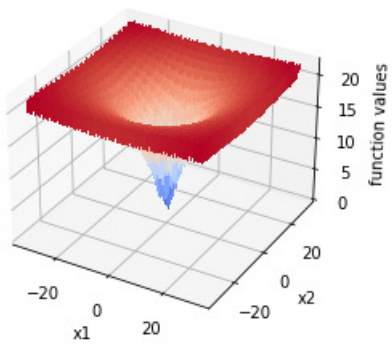

Figure 1: Two-dimenisonal Ackley function.

Assumption 3 is the generalized triangle inequality, and it becomes the standard one when $\eta = 1$. Particularly, this assumption can be satisfied with the following condition. Suppose that $c(z, y) \geq \alpha c(y, z)$ for some $\alpha > 0$ and $c(x, z) \leq c(x, y) + c(y, z)$. Then we can always find a η' such that $c(x, z) \leq \eta'(c(x, y) + c(z, y))$, which means that Assumption 3 is satisfied with $\eta = \max\{1, \eta'\}$. In the lunar lander problem mentioned in Section 2, the condition $c(z, y) \geq \alpha c(y, z)$ means that the energy consumed by some movement is at least α portion of the opposite movement, which is not a limiting condition in practice.

Assumption 4 is a common property of switching cost in practice. With these assumptions, we have the following theorem.

Theorem 1. *Assume that $h(w)$ lies in the RKHS space with kernel k' and $||h||_{k'} < B$. If Assumption 1-4 are satisfied with $\eta^2/\lambda < 1$, then Algorithm 2 with $\omega = 1 + 2/T$ gives*

$$\sum_{t=1}^{T} (f(x_t) + c(x_t, x_{t-1})) - \sum_{t=1}^{T} \psi \left(f(x_t^*) + c(x_t^*, x_{t-1}^*)\right) = \tilde{O}(T^{g(2d)})$$

with probability at least $1 - \delta$, where $\psi = \max\{\frac{1+\eta^2/\lambda}{1-\eta^2/\lambda}, \frac{\eta}{1-\eta^2/\lambda}\}$.

Proof. Please refer to Section 1 of Supplementary Material. □

From the above theorem, we can see that Algorithm 2 needs to know T beforehand. In fact, if T is unknown, the doubling trick can bu used to convert our algorithm into an anytime algorithm. Please refer to Besson and Kaufmann [2018] for details.

Now similar to the discussion at the end of Section 3, we can transfer the above bound into CR shown as follows:

Corollary 1. *If f is lower-bounded by some positive constant C, then under assumptions of Theorem 1, we have*

$$CR = \psi + \tilde{O}(T^{g(2d)-1})$$

for Algorithm 2, where $\psi = \max\{\frac{1+\eta^2/\lambda}{1-\eta^2/\lambda}, \frac{\eta}{1-\eta^2/\lambda}\}$.

Proof. Please refer to Section 2 of Supplementary Material. □

Here we can see that Algorithm 2 achieves a constant, dimension free CR ψ as $T \to \infty$ if $g(2d) < 1$. Compared with IGP-UCB that requires $g(d) < 1$, it is a stricter condition, thus the choices of kernel functions are narrowed for Algorithm 2. It is due to the fact that this algorithm puts a prior on a $2d$-dimensional function. Meanwhile, when $\eta = 0$, we have $\psi = 1$. In this case, $c(x, y)$ is always equal to 0 from Assumption 3, which means that there is no switching cost. Then same with IGP-UCB, CR of Algorithm 2 approaches 1 as $T \to \infty$. Therefore, it is the existence of switching cost that degrades the performance of the algorithm.

Despite its simplicity, Algorithm 2 has two disadvantages. First, the requirement $\eta^2/\lambda < 1$ restricts the applicability of the algorithm. Second, the updates of posterior distribution, i.e., (6) and (7), are on a higher dimension than the original problem ($2d$ instead of d), which leads to inefficiency since it involves matrix inverse. In the next section, we will propose another algorithm that can mitigate these issues while still possessing a theoretical guarantee of CR.

5 HOW TO DEAL WITH SWITCHING COST: ALTERNATING SEARCH

The biggest challenge to reduce the operational dimension of GS is that we cannot avoid the time-varying c in the original dimension. To tackle it, we design an epoch-based structure that alternatively changes c across epochs, but fixes it within the epoch. Due to this characteristic, we call this new algorithm Alternating Search (AS). In the following, we will illustrate this algorithm using Figure 2.

Before the start of each epoch, e.g., epoch m, we choose a point that has the minimal observation value of f among all the past points. We call it the pivot point of epoch m denoted by v_m. Then during this epoch except its last iteration, we choose the next point by minimizing the GP surrogate model of $f(x) + c(x, v_m)$, where $c(x, v_m)$ is no longer time-varying within this epoch. We call it limited exploration (LE) because the search is constrained around v_m due to the existence of $c(x, v_m)$. In the last iteration of this epoch, we choose the point minimizing the GP surrogate model of $f(x)$, which is called unlimited exploration (UE) since the switching cost is not considered. Meanwhile, we set the length of epochs to be linearly increasing.

Algorithm 3 shows the IGP-UCB version of this idea. In the whole process of this algorithm, the posterior distribution of $f(x)$ is always updated after a point is chosen. The one of $c(x, v_m)$, however, is only updated within epoch m because $c(x, v_m)$ will be changed to $c(x, v_{m+1})$ in the next epoch and thus its GP model will be reset.

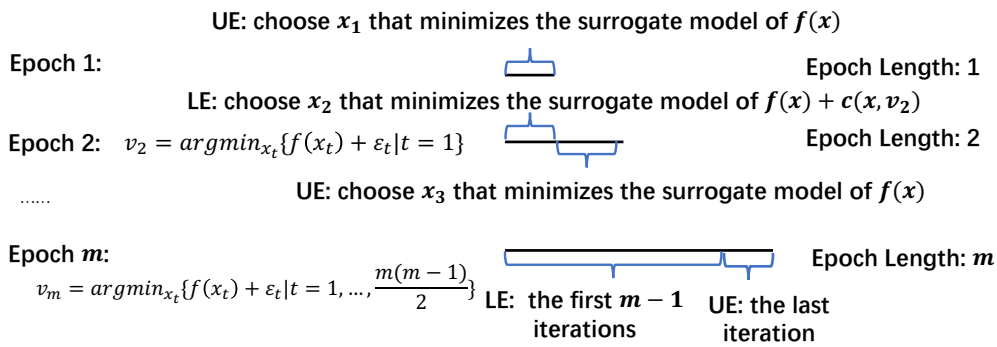

Figure 2: The structure of Alternating Search algorithm. Here $f(x_t) + \varepsilon_t$ represents the noisy observation of f at time t.

The intuition of this algorithm is as follows. From the objective $\sum_{t=1}^{T}[f(x_t) + c(x_t, x_{t-1})]$, we can see that $f(x)$ is a fixed function. Therefore, choosing a point that can sufficiently reduce $f(x)$ and searching around it will benefit the performance, which is how we choose the pivot point before the start of each epoch. We also need UE in the end of each epoch is to help find a better pivot point in the next epoch. On the other hand, UE may bring in a large switching cost since we do not incorporate the switching cost in UE. With our designed epoch lengths, this situation only happens for $O(\sqrt{T})$ times within time horizon T because it is only done once in each epoch. Meanwhile in LE, we control the switching cost $c(x_t, x_{t-1})$ via $c(x, v_m)$ in the other iterations of each epoch by utilizing Assumption 3. Combining LE and UE in such a way gives us Algorithm 3 with a theoretical guarantee of CR.

Despite Assumption 1-4, we assume that f lies in the RKHS with kernel k^f and $||f||_{k^f} < B$; $c_m(x) := c(x, v_m)$ lies in the RKHS with kernel k^{c_m} and $||c_m||_{k^{c_m}} < B$. The updates of $\mu_t^f(x)$ and $\sigma_t^f(x)$ are similar to (4) and (5) based on the past observations of $f(x)$. The updates of $\mu_s^{c_m}(x)$ and $\sigma_s^{c_m}(x)$ are slightly different by setting $\omega = 0$ in (4) and (5) since they are based on the noiseless observations of $c_m(x)$ within epoch m:

$$\mu_s^{c_m}(x) = k_s^{c_m}(x)^T (K_s^{c_m})^{-1} y_{1:s}^{c_m}$$
$$(\sigma_s^{c_m})^2(x) = k^{c_m}(x, x) - k_s^{c_m}(x)^T (K_s^{c_m})^{-1} k_s^{c_m}(x)$$

Similar to Lemma 2, we can prove that $|f(x) - \mu_{t-1}^f(x)| \leq \beta_t \sigma_{t-1}^f(x)$ with probability at least $1 - \delta$. Therefore, UE of Algorithm 3, i.e., Step 8-10, is minimizing the LCB of $f(x)$ across epoches. Meanwhile, from Lemma 11 of Lyu et al. [2019], we have $|c_m(x) - \mu_{s-1}^{c_m}(x)| \leq B\sigma_{s-1}^{c_m}(x)$. Then $|f(x) + 2\eta c_m(x) - \mu_{t-1}^f(x) - 2\eta\mu_{s-1}^{c_m}(x)| \leq 2\eta\beta_t\sigma_{t-1}^f(x) + 2\eta B\sigma_{s-1}^{c_m}(x)$. It means that LE of Algorithm 3 in epoch m, i.e., Step 11-13, is minimizing the LCB of $f(x) + 2\eta c_m(x)$. The reason why we add the coefficient of 2η is the fact that we use $c(x, v_m)$ to control $c(x_t, x_{t-1})$ by Assumption 3, whose details can be found in the proof of the following theorem.

Algorithm 3 Alternating Search (AS)

1: Initial value: $GP(0, k^f(x, x'))$ prior for $f(x)$, parameters B, R, ω, δ, T, total iterations $t = 1$.
2: **for** $m = 1, \ldots, M$ **do**
3: **if** $m > 1$ **then**
4: Choose v_m that has the smallest observation value of f among $\{x_l\}_{l=1}^{t-1}$ and set $c_m(x) = c(x, v_m)$. Put $GP(0, k^{c_m}(x, x'))$ prior on $c_m(x)$.
5: **end if**
6: **for** $s = 1, \ldots, m$ **do**
7: Set $\beta_t = B + R\sqrt{2(\gamma_{t-1}^f + 1 + \log(M/\delta))}$.
8: **if** $s = m$ **then**
9: Set $x_t = \arg\min_{x \in \mathcal{X}} \mu_{t-1}^f(x) - \beta_t \sigma_{t-1}^f(x)$.
10: Make new observations of f. Update $\mu_t^f(x)$ and $\sigma_t^f(x)$ for $f(x)$ similar to (4) and (5).
11: **else**
12: Set $x_t = \arg\min_{x \in \mathcal{X}} \mu_{t-1}^f(x) + 2\eta\mu_{s-1}^{c_m}(x) - \beta_t\sigma_{t-1}^f(x) - 2\eta B\sigma_{s-1}^{c_m}(x)$.
13: Make new observations of f and c_m respectively. Update $\mu_t^f(x)$ and $\sigma_t^f(x)$ for $f(x)$, $\mu_s^{c_m}(x)$ and $\sigma_s^{c_m}(x)$ for $c_m(x)$ similar to (4) and (5).
14: **end if**
15: $t = t + 1$
16: **end for**
17: **end for**
18: **Output:** $\{x_t\}_{t=1}^T$ where $T = 1 + 2 + \ldots + M$.

Theorem 2. *Assume that f lies in the RKHS with kernel k^f and $||f||_{k^f} < B$; $c_m(x)$ lies in the RKHS with kernel k^{c_m} and $||c_m||_{k^{c_m}} < B$ for any m. If Assumption 1-4 are satisfied, and $\omega = 1 + 2/T$, then Algorithm 3 gives*

$$\sum_{t=1}^{T}(f(x_t) + c(x_t, x_{t-1})) - \sum_{t=1}^{T}\psi\left(f(x_t^*) + c(x_t^*, x_{t-1}^*)\right)$$
$$= \tilde{O}(T^{(g(d)+1)/2})$$

with probability at least $1 - 3\delta$, where $\psi = \max\{1 + 2\eta^3/\lambda, 2\eta^2\}$.

Proof. Please refer to Section 3 of Supplementary Material. □

Here we can see that Algorithm 3 lifts the requirement of η and λ that exists in Algorithm 2 for its theoretical guarantee. Similar to the proof of Corollary 1, we can give CR of Algorithm 3 using the above theorem.

Corollary 2. *If f is lower-bounded by some positive constant C, then under assumptions of Theorem 2, we have*

$$CR = \psi + \tilde{O}(T^{(g(d)-1)/2}) \tag{9}$$

for Algorithm 3, where $\psi = \max\{1 + 2\eta^3/\lambda, 2\eta^2\}$.

From (9), we can see that CR of Algorithm 3 approaches ψ as $T \to \infty$ if $g(d) < 1$. This condition is the same as the one of IGP-UCB and milder than the one of Algorithm 2 which is $g(2d) < 1$. It is due to the fact that Algorithm 3 operates on a lower dimension (d instead of $2d$) than Algorithm 2, which also makes Algorithm 3 more efficient than Algorithm 2. Same with Algorithm 2, we have $\psi = 1$ when $\eta = 0$, i.e., there is no switching cost.

However, an additional challenge for Algorithm 3 is how to obtain observations of $c(x_t, v_m)$ if we can only observe $c(x_t, x_{t-1})$ at time t when $v_m \neq x_{t-1}$. Here are some cases where this problem can be solved:

- Special form of c. Here we use the drone tracking problem mentioned in Example 1 of Pan et al. [2022] for an explanation. In this problem, the switching cost from the speed x_t to x_{t-1} of the drone is expressed as:

$$c(x_t, x_{t-1}) = \frac{1}{2}(x_t - x_{t-1} + e(x_{t-1}))^2, \tag{10}$$

  where $e(x_{t-1})$ accounts for the effects of gravity and the aerodynamic drag due to x_{t-1}. In practice, only the form of e is unknown in (10) and we can obtain the value of $e(x_{t-1})$ after we observe $c(x_t, x_{t-1})$. Then we can get $c(x_t, v_m) = \frac{1}{2}(x_t - v_m + e(v_m))^2$ once $e(v_m)$ is known from the history observation involving v_m. It can be extended to other cases with a similar form of c.

- Using simulation programs. This idea is similar to the one of Kandasamy et al. [2016]. In practice, $c(x, y)$ is much easier to be simulated accurately in a program than f due to its simplicity. Then $c(x_t, v_m)$ can be obtained by running a simulation program, while the values of f are obtained from real experiments. Robot pushing problem and lunar lander problem mentioned in Section 2 are such examples, where c is just the energy cost of changing controllable parameters.

If $c(x_t, v_m)$ cannot be easily obtained, we should use Algorithm 2 instead.

6 SIMULATION RESULTS

In this section, we will use two classical black-box control problems mentioned in Section 2 to test our algorithms. Since our paper is theoretically focused, we only compare our algorithms with IGP-UCB to demonstrate our theoretical findings in this paper. The performance metric is the time-averaged total cost, which is $\frac{1}{T}\sum_{t=1}^{T}(f(x_t) + c(x_t, x_{t-1}))$. For fairness, we use the same kernel function in these algorithms, which is Matérn kernel with $\nu = 1.5$. We run each algorithm for 10 times in each problem, and plot the averaged result of these 10 tests. In both experiments, β_t's of all the three algorithms are tuned by grid search. We also add the Gaussian noise with mean 0 and variance 1 to the original observation values to satisfy our setup. More details of the experiments are reported in Section 4 of Supplementary Material.

6.1 ROBOT PUSHING PROBLEM

The original 14-dimensional robot pushing problem was first tested in Wang et al. [2018] without switching cost, where the authors implemented the simulation of pushing two objects with two robot hands in the Box2D physics engine. In our experiment, we transfer it to a minimization problem with the switching cost defined as 0.1 times the l_1 norm of the change in the first 7 parameters, and the main cost defined as the negative of the reward plus a large number so that Assumption 2 can be satisfied. The simulation results are shown in Figure 3.

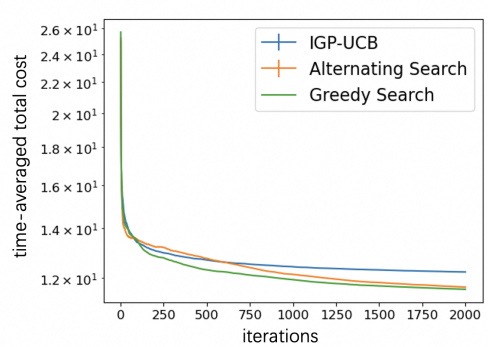

Figure 3: The simulation results of the 14-dimensional robot pushing problem. Each line is the average of 10 tests.

From the figure, we can see that our two algorithms have better performance than IGP-UCB after a few iterations. In general, GS has a slightly lower cost than AS, but their gap is disappearing as the iteration increases. Meanwhile, in Section 5 of Supplementary Material, we show the running time of AS and GS after 2000 iterations. we can see that AS is 48% faster than GS on average, thus is a better choice in this experiment.

6.2 LUNAR LANDER PROBLEM

The original 12-dimensional lunar lander problem in Eriksson et al. [2019] is to learn a controller for a lunar lander implemented in the OpenAI gym without switching cost. Similar to the above experiment, we transfer it to a minimization problem where we define the switching cost as 100 times the l_1 norm of the change in the first 6 parameters, and the main cost as the negative of the reward plus a large number to make it satisfy Assumption 2. The coefficient 100 is to make the switching cost comparable to the main cost. The simulation results are shown in Figure 4.

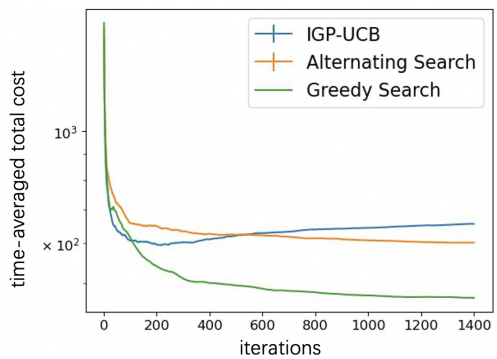

Figure 4: The simulation results of the 12-dimensional lunar lander problem. Each line is the average of 10 tests.

In this problem, the total cost of IGP-UCB is no longer constantly decreasing with regard to iterations showing the failure of this algorithm in our setup. Meanwhile, the performance of GS is much better than AS. On the other hand, the running time of AS is 43% faster than GS in average as shown in Section 5 of Supplementary Material.

7 CONCLUSION

In this paper, we investigated a bandit setting for online optimization problems with switching costs from a Bayesian perspective, which has many applications in practice but lacks algorithms with theoretical guarantees. To fill this gap, we proposed two new algorithms called Greedy Search and Alternating Search with competitive ratios approaching a constant as $T \to \infty$ under different assumptions, where the latter algorithm has a faster running time. Their superior performance was also demonstrated via two classical black-box control problems commonly tested in previous works. Meanwhile, there are still some future directions to explore. The most significant one is probably to find the lower bound of competitive ratio for bandit online optimization with switching cost. Based on this result, we can check whether our proposed algorithms are optimal. If not, then how to find the optimal algorithm for our setup is another problem that needs to be solved.

References

D.H Ackley. *A Connectionist Machine for Genetic Hill-climbing*, volume SECS28 of *The Kluwer International Series in Engineering and Computer Science.* Kluwer Academic Publishers, Boston, 1987.

R Agrawal, M Hegde, D Teneketzis, et al. Multi-armed bandit problems with multiple plays and switching cost. *Stochastics and Stochastic reports*, 29(4):437–459, 1990.

Antonios Antoniadis and Kevin Schewior. A tight lower bound for online convex optimization with switching costs. In *International Workshop on Approximation and Online Algorithms*, pages 164–175. Springer, 2017.

Nikhil Bansal, Anupam Gupta, Ravishankar Krishnaswamy, Kirk Pruhs, Kevin Schewior, and Cliff Stein. A 2-competitive algorithm for online convex optimization with switching costs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2015.

Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, pages 1–35, 2021.

Lilian Besson and Emilie Kaufmann. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*, 2018.

Romain Camilleri, Kevin Jamieson, and Julian Katz-Samuels. High-dimensional experimental design and kernel bandits. In *International Conference on Machine Learning*, pages 1227–1237. PMLR, 2021.

Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan LH Andrew. Online convex optimization using predictions. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 191–204, 2015.

Niangjun Chen, Gautam Goel, and Adam Wierman. Smoothed online convex optimization in high dimensions via online balanced descent. In *Conference On Learning Theory*, pages 1574–1594. PMLR, 2018.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. Bandits with switching costs: T 2/3 regret. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 459–467, 2014.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Gautam Goel, Niangjun Chen, and Adam Wierman. Thinking fast and slow: Optimization decomposition across timescales. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1291–1298. IEEE, 2017.

Gautam Goel, Yiheng Lin, Haoyuan Sun, and Adam Wierman. Beyond online balanced descent: An optimal algorithm for smoothed online optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Sudipto Guha and Kamesh Munagala. Multi-armed bandits with metric switching costs. In *Automata, Languages and Programming: 36th Internatilonal Collogquium, ICALP 2009, Rhodes, greece, July 5-12, 2009, Proceedings, Part II 36*, pages 496–507. Springer, 2009.

Sunil Gupta, Santu Rana, Svetha Venkatesh, et al. Regret bounds for expected improvement algorithms in gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 8715–8737. PMLR, 2022.

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

Vinay Joseph and Gustavo de Veciana. Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs. In *2012 Proceedings IEEE INFOCOM*, pages 567–575. IEEE, 2012.

Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. *Advances in neural information processing systems*, 29, 2016.

Taehwan Kim, Yisong Yue, Sarah Taylor, and Iain Matthews. A decision tree framework for spatiotemporal sequence prediction. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 577–586, 2015.

Tomer Koren, Roi Livni, and Yishay Mansour. Bandits with movement costs and adaptive pricing. In *Conference on Learning Theory*, pages 1242–1268. PMLR, 2017.

Yingying Li, Guannan Qu, and Na Li. Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit. *IEEE Transactions on Automatic Control*, 66(10):4761–4768, 2020.

Zihan Li and Jonathan Scarlett. Gaussian process bandit optimization with few batches. In *International Conference on Artificial Intelligence and Statistics*, pages 92–107. PMLR, 2022.

Minghong Lin, Adam Wierman, Lachlan LH Andrew, and Eno Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking*, 21(5):1378–1391, 2012.

Yiheng Lin, Gautam Goel, and Adam Wierman. Online optimization with predictions and non-convex losses. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):1–32, 2020.

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

Yueming Lyu, Yuan Yuan, and Ivor W Tsang. Efficient batch black-box optimization with deterministic regret bounds. *arXiv preprint arXiv:1905.10041*, 2019.

Weici Pan, Guanya Shi, Yiheng Lin, and Adam Wierman. Online optimization with feedback delay and nonlinear switching cost. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(1):1–34, 2022.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

Sudeep Salgia, Sattar Vakili, and Qing Zhao. A domain-shrinking based bayesian optimization algorithm with order-optimal regret performance. *Advances in Neural Information Processing Systems*, 34:28836–28847, 2021.

Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2021.

Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018.

Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

Xinyi Zhang, Hong Wu, Zhuo Chang, Shuowei Jin, Jian Tan, Feifei Li, Tieying Zhang, and Bin Cui. Restune: Resource oriented tuning boosted by meta-learning for cloud databases. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2102–2114, 2021.