# A Bayesian Approach for Stochastic Continuum-armed Bandit with Long-term Constraints

**Zai Shi**
The Ohio State University

**Atilla Eryilmaz**
The Ohio State University

## Abstract

Despite many valuable advances in the domain of online convex optimization over the last decade, many machine learning and networking problems of interest do not fit into that framework due to their nonconvex objectives and the presence of constraints. This motivates us in this paper to go beyond convexity and study the problem of stochastic continuum-armed bandit with long-term constraints. For noiseless observations of constraint functions, we propose a generic method using a Bayesian approach based on a class of penalty functions, and prove that it can achieve a sublinear regret with respect to the global optimum and a sublinear constraint violation (CV), which can match the best results of previous methods. Additionally, we propose another method to deal with the case where constraint functions are observed with noise, which can achieve a sublinear regret and a sublinear CV with more assumptions. Finally, we use two experiments to compare our methods with two benchmark methods in online optimization and Bayesian optimization, which demonstrates the advantages of our algorithms.

## 1 Introduction

Multi-armed bandit (MAB) is a sequential decision process where a player chooses an arm $i_t$ from a finite number of arms $1, ..., n$ and then receives a random reward $X_{i_t}$ from that arm (Lattimore and Szepesvári, 2020). In stochastic bandits, the reward of each arm is realized from a distribution with an unknown mean. Its metric for an algorithm with time horizon $T$ is called regret, which is defined as:

$$R_T = \sum_{t=1}^{T} [\mu^* - \mu_{i_t}].$$

where $\mu^* = \max_i \mu_i$ and $\mu_i$ is the mean reward of arm $i$. We hope an algorithm to produce a sublinear regret, i.e., $R_T = o(T)$, which means that we can obtain the optimal reward asymptotically in terms of time-average.

If the set of arms $\mathcal{X}$ is continuous and closed, then the problem becomes a continuum-armed bandit (Agrawal, 1995). At each round, the player chooses a decision $x_t$ from $\mathcal{X}$ and then receives an i.i.d. random reward $f^t(x_t)$ with an unknown mean function $f(x_t)$ that may be nonconvex. Same with multi-armed bandit, the objective of an algorithm with time horizon $T$ is to produce a sublinear regret, which is defined as

$$R_T = \sum_{t=1}^{T} [f(x^*) - f(x_t)].$$

where $x^* = \arg\max_{x \in \mathcal{X}} f(x)$.

There are many approaches for this problem (Agrawal, 1995; Cope, 2009; Auer et al., 2007; Kleinberg, 2004; Singh, 2021; Chowdhury and Gopalan, 2017) with different assumptions of $f$. A well-known method among these is via a Bayesian approach, which assumes that $f$ lies within some Reproducing Kernel Hilbert Space (RKHS) and then utilizes the framework of Gaussian process (GP) regression (Shi and Choi, 2011). There are several works using this approach with theoretical results. in the paper (Srinivas et al., 2009), the authors proposed a UCB-type algorithm called GP-UCB and achieved the regret of $O(\sqrt{T}(B\sqrt{\gamma_T} + \gamma_T \log^{3/2} T)$ with high probability, where $B$ is the bound of RKHS norm of the reward function, and $\gamma_T$ is a quantity related to $\mathcal{X}$ and the kernel function of GP used in the algorithm. The authors in the paper (Chowdhury and Gopalan, 2017) improved GP-UCB with a new method called IGP-UCB, which relaxed the assumption of

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

noise distribution in the paper (Srinivas et al., 2009) and achieved a better regret of $O(\sqrt{T}(B\sqrt{\gamma_T}+\gamma_T))$ with a high probability. The authors also proposed a method based on Thompson sampling, achieving a regret of $O\left(\sqrt{T\gamma_T((\gamma_T+\log(1/\delta))d\log(BdT))}\right)$ with probability $1-\delta$, where $d$ is the dimension of $\mathcal{X}$.

In this paper, we consider a more complex setup, where the chosen decisions in the bandit process are also required to satisfy some *long-term constraints*, i.e., $\frac{1}{T}\sum_{t=1}^{T}\mathbf{g}(x_t) \preceq 0$. Here $\mathbf{g}(x)=[g_1(x),...,g_m(x)]$ is a stacked vector of $m$ constraint functions that may be nonconvex. The form of $\mathbf{g}(x)$ is unknown to us, but its value can be observed (possibly with random noise) after making a decision. Different from previous works on constrained MAB (Badanidiyuru et al., 2013; Agrawal and Devanur, 2016, 2014; Immorlica et al., 2019; Madani et al., 2004; Xia et al., 2015), our setup assumes continuous action space with continuous reward and constraint functions that may be nonconvex. The details of the setup along with some applications will be presented in Section 3. Meanwhile, we will use two metrics to measure the performance of an algorithm solving this problem, which are called regret and constraint violation (CV). Their definitions will also be detailed in Section 3. It is noted that these two metrics were widely used in previous works on online convex optimization (OCO) with long-term constraints. We will discuss these works in Section 1.1 and compare their setup with ours. Same with OCO, we hope our algorithm to have a sublinear regret and a sublinear CV concurrently.

For the above setup, our paper is the first work to propose methods with theoretical guarantees of a sublinear regret and a sublinear CV using a Bayesian approach. We will present two methods for two cases, where the constraint function values are observed without or with noise, respectively. Both methods assume that the reward function is observed with random noise. The contributions of our papers are as follows:

- We first show that for generic penalty approaches using a Lagrangian-like transformation, it is impossible to get a sublinear regret and a sublinear CV if there is no update of multipliers or penalty functions. The details of this claim will be shown in Section 4.
- Motivated by the above observation, we explore a multiplicative form of multiplier-updates and then propose a method called *GP-UCB with Noiseless Constraints* for noiseless constraint observations. This method adopts a broad class of penalty functions described by properties that are new in penalty approaches, which enables a flexible choice based on applications. These new properties can also lead to a better understanding towards the use of penalty-based approaches in online optimization. The details of this method will be presented in Section 5.
- For the above method, we also highlight the trade-off between regret and CV in its results. By changing the total iterations of the outer-loop and the inner-loop of our design, both regret and CV can be sublinear with the regret $\tilde{O}(\sqrt{T\tilde{\gamma}_T})$ matching the lower-bound of Bayesian optimization using SE kernel up to a logarithmic factor, or the CV $\tilde{O}(\sqrt{T})$ matching the best result among previous methods for OCO with long-term constraints (c.f. Section 1.1). Here $\tilde{\gamma}_T$ will be defined in Section 5.
- For noisy constraint observations, we change the rule of multiplier-updates and switch to a linear penalty function to avoid "noise amplification" brought by the penalty function used in the first method. This method, called *GP-UCB with Noisy Constraints*, can achieve the regret of $\tilde{O}(T^{3/4}\tilde{\gamma}_{\sqrt{T}})$ and the CV of $\tilde{O}(T^{3/4})$ with more assumptions needed than the first method. The proof of this result involves how to tackle the noise in the multiplier-updates, which may be interesting in its own right. The details of this method will be presented in Section 6.
- Through simulations of two problems, we demonstrate the efficiency of our algorithms compared with benchmark methods in online convex optimization and Bayesian optimization. This part will be presented in Section 5 of Supplementary Material.

Since online convex optimization with long-term constraints and constrained Bayesian optimization (BO) are closely related to our problem, in the following subsection we will discuss previous works on these two topics and point out our advantages over these methods.

## 1.1 Related Works

### 1.1.1 Online Convex Optimization with Long-term Constraints

Similar to our setup, online convex optimization (OCO) is also a sequential decision process where a reward function value $f^t(x_t)$ is observed after a decision-maker chooses an action $x_t$. The major difference is that in OCO, $f^t$ can be arbitrary (not necessarily i.i.d. from some distribution), but must be convex (Hazan, 2019). For OCO with long-term constraints,

| | **Regret** | **CV** | $f^t$ | $\mathbf{g}^t$ |
|---|---|---|---|---|
| (Mahdavi et al., 2012) | $\tilde{O}(T^{1/2})$ | $\tilde{O}(T^{3/4})$ | Arbitrary[1], Convex | Deterministic, Convex |
| (Chen and Giannakis, 2018)[2] | $\tilde{O}(V(x^*_{1:T})T^{1/2})$ | $\tilde{O}(T^{1/2})$ | Arbitrary[1], Convex | Arbitrary[1], Convex |
| (Cao and Liu, 2018) | $\tilde{O}(T^{1/2}\Delta(T)^{1/2})$ | $\tilde{O}(T^{3/4}\Delta(T)^{1/4})$ | Arbitrary[1], Convex | Arbitrary[1], Convex |
| Our work | $\tilde{O}(T^{3/4}\tilde{\gamma}_{\sqrt{T}})$ | $\tilde{O}(T^{1/2})$ | Stochastic, Nonconvex | Deterministic, Nonconvex |
| Our work | $\tilde{O}(T^{3/4}\tilde{\gamma}_{\sqrt{T}})$ | $\tilde{O}(T^{3/4})$ | Stochastic, Nonconvex | Stochastic, Nonconvex |

Table 1: The results of our paper and previous works on OCO with long-term constraints in the bandit setting. Here $\tilde{O}$ means neglecting log terms. $V(x^*_{1:T})$, $\Delta(T)$ and $\tilde{\gamma}_{\sqrt{T}}$ are problem-related quantities defined in the corresponding papers ([1] In fact, $f$ and $g$ are required to be bounded in these papers. [2] Only $f^t$ has bandit feedback. The forms of $\mathbf{g}^t$ are known to the decision-maker).

constraint function values $\mathbf{g}^t(x_t)$ are also observed after an action $x_t$ with requirements $\frac{1}{T}\sum_{t=1}^{T}\mathbf{g}^t(x_t) \preceq 0$. $\mathbf{g}^t$ can be assumed to be deterministic (Mahdavi et al., 2012), stochastic (i.i.d. from some distribution) (Yu et al., 2017; Eryilmaz and Srikant, 2007) or arbitrary across $t$ (Chen and Giannakis, 2018; Chen et al., 2017; Cao and Liu, 2018), but must be convex as well. Meanwhile in OCO, the forms of $f^t$ and $\mathbf{g}^t$ can be assumed to be known or unknown after an action is chosen at $t$, and the later setup is referred as bandit setting (Chen and Giannakis, 2018; Cao and Liu, 2018). Obviously our setup belongs to bandit setting.

Same with our paper, two performance metrics, namely regret and constraint violation, are used for OCO with long-term constraints and the objective of an algorithm is to make both metrics sublinear. In Table 1, we summarize the results of previous works on OCO with long-term constraints in the bandit setting, and compare them with ours. Here we can see that convexity is essential for OCO algorithms because gradients are utilized to make the next decision. On the contrary, our method does not rely on convexity, which can be applied to many nonconvex problems such as machine learning (Mei et al., 2018) and network optimization (Lee et al., 2005). Meanwhile, in the bandit setting, it is hard to check convexity of $f^t$ and $g^t$ without knowing their forms. Thus our method has a broader application in practice.

### 1.1.2 Constrained Bayesian Optimization

Bayesian optimization (Frazier, 2018) is a global optimization method for blackbox problems, which also utilizes the framework of GPs. The techniques of BO will be introduced in Section 2 as a basis of our methods.

Constrained Bayesian optimization aims to solve

$$\max_{x\in\mathcal{X}} f(x)$$
$$s.t.\ g_i(x) \leq 0, i = 1, ..., m$$

where the forms of $f(x)$ and $g_i(x)$ are unknown. The objective of constrained BO is to get the global maximum of the above problem with as few evaluations of $f$ and $g_i$ as possible. Constrained BO approaches aim at finding an optimal solution to the constrained problem, regardless of how they are reached. In contrast, our setup is also concerned with the sequence of solutions that are observed in the process, which need to satisfy a sublinear regret and a sublinear CV. In fact, previous methods on constrained BO are predominantly empirical without rigorous performance metrics and theoretical results. Instead, their performance was only tested by experiments, and measured by the running time to get the optimal point in a certain computing environment.

According to different classes of utility functions used in the algorithms, previous works on constrained BO can be classified as expected-improvement type (Gardner et al., 2014; Letham et al., 2019; Gelbart et al., 2014; Picheny et al., 2016; Ariafar et al., 2019) , entropy-search type (Hernández-Lobato et al., 2016; Perrone et al., 2019) and Thompson-sampling type (Eriksson and Poloczek, 2021). To the best of our knowledge, there does not exist a UCB-type method for constrained Bayesian optimization in previous literature. We hope our work to inspire such a design with a similar Bayesian approach.

## 2 Preliminaries on Bayesian Approaches

As a basis of our paper, we first review backgrounds of Bayesian optimization (BO) in this section. Classical Bayesian optimization methods aim to solve a black box problem $\max_{x\in\mathcal{X}} f(x)$, where the form of $f$ is unknown. Instead, we can observe $f(x_s)$ (possibly with noise) after we inquire a point $x_s$. BO uses a machine learning approach, called Gaussian process (GP) regression (Shi and Choi, 2011), to find the optimal point of the above problem. First, we put a GP prior on $f$ and get its posterior distribution after an inquiry of $f$. Then, we choose the next inquiry point and update the posterior distribution of $f$. When the posterior distribution is considered to be informative enough of the optimal point, we can get the final result

**Algorithm 1** IGP-UCB($f(x), k, B, R, \lambda, \delta, S$)

```
Input: Prior $GP(0,k)$, parameters $B, R, \lambda, \delta, S$.
for $s = 1, ..., S$ do
    Set $\beta_s = B + R\sqrt{2(\gamma_{s-1} + 1 + \log(1/\delta))}$.
    Choose $x_s = \arg\max_{x\in\mathcal{X}}\{\mu_{s-1}(x) + \beta_s\sigma_{s-1}(x)\}$.
    Obtain noisy observation of $f(x_s)$.
    Perform update to get $\mu_s$ and $\sigma_s$ using (3) and (4).
end for
Output: $x_1, ..., x_S$.
```

from it. The main component of BO is how to choose the next inquiry point of $f$ based on the current posterior distribution to minimize the number of inquiries needed in the whole process.

In general, BO uses a utility function $u$ to determine the next inquiry point, i.e., $x_{s+1} = \max_{x\in\mathcal{X}} \mathbb{E}[u(x)|x_{1:s}]$, where $x_{1:s} = [x_1, ..., x_s]$ are previous inquiries and $x_{s+1}$ is the next inquiry we will choose. Here, the expectation is taken with regard to the current posterior distribution based on $x_{1:s}$. Different kinds of utility functions produce different classes of BO methods, including Expected-Improvement (EI), Upper Confidence Bound (UCB), Thompson Sampling (TS), and so on. The reader may refer to (Frazier, 2018) for more details.

IGP-UCB method proposed in the paper (Chowdhury and Gopalan, 2017) uses a similar technique to the above discussion. In this algorithm, we assume that $f$ is observed with independent $R$-subGaussian noise $\varepsilon$ with unknown forms. We also assume that $f$ belongs to some Reproducing Kernel Hilbert Space (RKHS) $H_k$ with kernel $k$ and that $f$ has a bounded RKHS norm, i.e., $||f||_k < B$ for some constant $B$. This assumption constrains the complexity of $f$. For RKHS, the reader may refer to (Hofmann et al., 2005) for more details. Two popular choices of $k$ are square exponential (SE) kernel and Matérn kernel, defined as

$$k_{\mathrm{SE}}(x, x') = \exp(-w^2/2u^2) \tag{1}$$

$$k_{\mathrm{Matérn}}(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)}(\frac{w\sqrt{2\nu}}{u})^{\nu} B_{\nu}(\frac{w\sqrt{2\nu}}{u}) \tag{2}$$

where $w = ||x - x'||$ encodes the similarity between two points $x, x'$. $u > 0$ and $\nu > 0$ are hyperparameters in these functions and $B_\nu(\cdot)$ is the modified Bessel function. Algorithm 1 shows a typical procedure of IGP-UCB. Here $\delta$ is a parameter that will be shown in Lemma 1, $\lambda$ is some positive constant and

$$\mu_s(x) = k_s(x)^T(K_s + \lambda I)^{-1}y_{1:s} \tag{3}$$

$$\sigma_s^2(x) = k(x, x) - k_s(x)^T(K_s + \lambda I)^{-1}k_s(x) \tag{4}$$

are the mean and covariance function of the posterior distribution after observing $x_{1:s}$ by regarding the distribution of observation noise as $\mathcal{N}(0, \lambda)$. $K_s = [k(x, x')]_{x,x'\in\{x_1,...,x_s\}}$ is the kernel matrix at time $s$ and $k_s(x) = [k(x_1, x), ..., k(x_s, x)]^T$.

For IGP-UCB, we can see that its utility function is a linear combination of $\mu_{s-1}(x)$ and $\sigma_{s-1}(x)$, which is inspired by the exploitation-exploration tradeoff in bandit problems (Lattimore and Szepesvári, 2020). It has the following theoretical performance:

**Lemma 1** (Theorem 3 of (Chowdhury and Gopalan, 2017))**.** *Assume that $||f||_k < B$ and $\varepsilon_s$ is independent sampled from some $R$-subGaussian distribution. Then, running IGP-UCB for a function $f$ lying in the RKHS $H_k$ leads to*

$$\begin{aligned} R_S &= \sum_{s=1}^{S}[f(x^*) - f(x_s)] \\ &= O(B\sqrt{S\gamma_S} + \sqrt{S\gamma_S(\gamma_S + \log(1/\delta))}) \end{aligned}$$

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

Here, $\gamma_s$ is called information gain with time horizon $s$ and can be bounded given the knowledge of domain $\mathcal{X}$ and kernel function $k$ (Chowdhury and Gopalan, 2017). The newest result for the bounds of $\gamma_S$ can be found in the paper (Vakili et al., 2021).

## 3 Problem Setup

In this paper, we consider a stochastic continuum-armed bandit problem with long-term constraints, where in each iteration, the decision-maker chooses an action $x_t$ from a compact set $\mathcal{X}$, and then observes a random reward value $f^t(x_t)$ and $m$ constraint values $\{g_j^t(x_t)\}_{j=1}^m$. In our setup, we assume that $f^t(x_t) = f(x_t) + \varepsilon^t$, where the randomness of $f^t$ comes from a zero-mean random variable $\varepsilon^t$ independent across $t$. For $\{g_j^t(x_t)\}_{j=1}^m$, we will consider two cases. In the first case, $g_j^t(x_t) = g_j(x_t), \forall j$ is deterministic. And in the second case, $g_j^t(x_t) = g_j(x_t) + \varepsilon_j^t, \forall j$ is stochastic, where $\varepsilon_j^t$ is a zero-mean random variable from some unknown distribution independent across $t$. Meanwhile, the forms of $f$ and $g_j, \forall j$ are unknown and possibly nonconvex. Our target is to maximize our expected reward $f(x)$ with long-term constraints satisfied in expectation, i.e., $\frac{1}{T}\sum_{t=1}^{T} g_j(x_t) \le 0$ for any $j$. We assume that $\mathcal{X}$ includes the area satisfying all constraints.

Similar to multi-armed bandit (Lattimore and Szepesvári, 2020), we need a fixed optimal point $x^*$ not related to $T$ as the benchmark, which is defined as any global optimum of

$$\max_{x\in\mathcal{X}} f(x)$$
$$s.t.\, g_j(x) \leq 0, j = 1, ..., m. \tag{5}$$

With regard to $x^*$, an algorithm of time-horizon $T$ aims for a sublinear *regret*:

$$R_T = \sum_{t=1}^{T}[f(x^*) - f(x_t)] = o(T), \tag{6}$$

and a sublinear *constraint violation* (CV):

$$V_T = ||[\sum_{t=1}^{T}\mathbf{g}(x_t)]^+|| = o(T), \tag{7}$$

where $\mathbf{g}(x)$ is $[g_1(x), .., g_m(x)]$ and $[\mathbf{a}]^+$ means element-wise $\max(0, a)$ in vector $\mathbf{a}$. Note that our definition of $x^*$ ensures that $\limsup_{t\to\infty}\frac{1}{T}\sum_{t=1}^{T} g_j(x^*) \leq 0$. Therefore, achieving both a sublinear regret and a sublinear CV implies that, asymptotically on the average, the algorithm performs as well as the best fixed strategy in hindsight. In OCO, there are also some other definitions of regret and CV. We will briefly discuss them in Section 2 and Section 4 of Supplementary Material.

Now we give two examples of our setup in the areas of machine learning and network optimization.

**Hyperparameter Tuning:** Hyperparameter tuning (Joy et al., 2016) is a hot topic recently due to the success of deep neural networks. Hyperparameters like neural network structures and stepsize choices in the optimizer can have a great impact on the final result of deep learning. We can describe this problem in our framework as follows. After choosing a hyperparameter $x_t$, we can obtain the training errors $E^t(x_t)$ from the neural network, which is stochastic due to the randomness of the optimizer such as SGD (Bottou, 2010). Meanwhile, for each hyperparameter choice, we have a completion time $g^t(x_t)$ of the training, which is also stochastic due to the optimizer. For most hyperparameters, there are no explicit forms of the mean $E(\cdot)$ and $g(\cdot)$ (Goodfellow et al., 2016). If we want to obtain the optimal hyperparameter while making the whole tuning process complete before a time threshold $T$ in expectation, we can apply our setup.

**Data Rate Allocation Problem:** Suppose a router wants to set data rates for $d$ servers from a range $[0, C_i]$ for Server $i$. After allocating a data rate $x_i(t)$ for Server $i$ at time $t$, it will give back a utility value $U_i(x_i(t)) + \varepsilon_i(t)$, such as quality of service (QoS) (Lee et al., 2005), which involves random measurement errors $\varepsilon_i(t)$. Meanwhile, define $h_i^t(x_i(t))$ as the electricity cost of Server $i$ by choosing $x_i(t)$, which may be deterministic or random due to electricity price fluctuation and link quality perturbation. Then in this example, $f^t = \sum_{i=1}^{d}(U_i(x_i(t)) + \varepsilon_i(t))$ and $g^t(x(t)) = \sum_{i=1}^{d} h_i^t(x_i(t)) - B$, where $B$ is the time-averaged electricity budget. We can apply our setup if we want to maximize the actual total utility with a time-averaged electricity cost constraint satisfied in expectation.

In the following section, we will discuss a class of strategies that can be useful for our problem. Motivated by the insights of the discussion, we will propose two algorithms for noiseless and noisy observations of $g_j$'s, respectively. Meanwhile, throughout the paper, we always assume that

**Assumption 1.** *$f$ is observed with independent $R$-subGaussian noise $\varepsilon$.*

Here the independence is defined with regard to any other noise.

## 4 Generic Penalty Approach

An intuitive strategy to deal with our setup to transform the original constrained problem into an unconstrained one and then utilize some unconstrained bandit optimization algorithm. In this paper, we focus on the strategy of appending a generic penalty function to the objective as follows:

$$f(x) - \sum_{j=1}^{m}\kappa_j\Lambda(g_j(x)), \tag{8}$$

where $\Lambda(\cdot)$ is some penalty function and $\kappa_j$ is a multiplier for constraint $j$. Despite the generality of this strategy, the main difficulty is how to choose $\kappa_j$ and $\Lambda(\cdot)$ for our setup. Previous penalty approaches are mainly used for gradient-related methods (see Chapter 5 of (Bertsekas, 2014)), thus may not be suitable for our setup without knowledge of gradients.

First, we make the following impossibility claim:

**Proposition 1.** *For any fixed choices of $\kappa_j$ and $\Lambda(\cdot)$ in (8), and any unconstrained bandit optimization algorithm $\mathcal{M}$ that can produce a sublinear regret for its objective function, applying $\mathcal{M}$ to (8) will fail to yield a sublinear regret and a sublinear CV for all forms of $f(x)$ and $\mathbf{g}(x)$ in our setup.*

*Proof.* Construct $f(x)$ and $\mathbf{g}(x)$ defined in a compact set $\mathcal{X}$ as $f(x) = \sum_{j=1}^{m}\kappa_j\Lambda(g_j(x)) - (x-c)^2$ and $g(x) = x$, where $c$ is some positive constant and the point $x = c$ is included in $\mathcal{X}$. Suppose that applying $\mathcal{M}$ to

(8) with time horizon $T$ gives us

$$\sum_{i=1}^{T}[f(x_\Lambda^*) - \sum_{j=1}^{m}\kappa_j\Lambda(g_j(x_\Lambda^*)) - f(x_t) + \sum_{j=1}^{m}\kappa_j\Lambda(g_j(x_t))] \\ = o(T),$$

where $x_\Lambda^*$ is the global optimum of (8). By our construction,

$$\sum_{i=1}^{T}[f(x_\Lambda^*) - \sum_{j=1}^{m}\kappa_j\Lambda(g_j(x_\Lambda^*)) - f(x_t) + \sum_{j=1}^{m}\kappa_j\Lambda(g_j(x_t))] \\ = \sum_{t=1}^{T}(x_t - c)^2 = o(T)$$

Since $\sum_{t=1}^{T}(x_t - c)^2 \geq -2c\sum_{t=1}^{T}x_t + c^2T$, we have $\sum_{t=1}^{T}g(x_t) = \sum_{t=1}^{T}x_t \geq cT/2 - o(T)$, which is not sublinear. □

From the above observation, we can see that that how to update $\kappa_j$ or $\Lambda(\cdot)$ in the process of the algorithm is essential to getting both sublinear regret and sublinear CV performances. In the following two sections, we will propose two algorithms based on two ways of multiplier-updates, which can deal with noiseless and noisy constraint observations, respectively.

## 5 Noiseless Constraint Observations

In this section, we assume that $g_j, \forall j$ is observed without noise. We will present a penalty approach based on a multiplicative form of performing multiplier-updates. For this multiplier-update strategy (Step 4 of Algorithm 2), we find a class of penalty functions with the form $\Lambda(x) = \psi(x) - 1$ that can achieve a sublinear regret and a sublinear CV by our algorithm called GP-UCB with Noiseless Constraints. The details of the algorithm are shown in Algorithm 2, where $\psi(\cdot)$ needs to satisfy the following properties:

**Assumption 2.** *1. $\psi(x)$ is convex for $x \in \mathcal{X}$. Moreover, $\psi(x) = 1$ when $x \leq 0$ and $\psi(x)$ is strictly increasing when $x \geq 0$.*
*2. $\psi(x)\psi(y) \geq \psi(x + y)$ for $x \in \mathcal{X}$ and $y \in \mathcal{X}$.*

**Algorithm 2** GP-UCB with Noiseless Constraints

```
1: Initialize $c$ and $\kappa_j^1 = 1$ for all $j$.
2: for $l = 1, ..., L$ do
3:     Run IGP-UCB($f(x) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x)) - 1), k_l, B, R, \lambda, \delta/L, S$) for $S$ iterations to produce $\{x_l^1, ..., x_l^S\}$, while obtaining $S$ observations $\{f(x_l^s) + \varepsilon_l^s\}_{s=1}^S$ and $\{g_j(x_l^s)\}_{s=1}^S, \forall j$ sequentially with the above outputs, where $\varepsilon_l^s$ is the observation noise of $f$.
4:     Set $\kappa_j^{l+1} = \kappa_j^l\psi\left(\frac{1}{S}\sum_{s=1}^{S}g_j(x_l^s)\right), \forall j$
5: end for
6: Output: $\{\{x_l^s\}_{l=1}^L\}_{s=1}^S$.
```

Property (1) makes $\Lambda(g_j(x)) = 0$ for $g_j(x) \leq 0$ and $\Lambda(g_j(x)) > 0$ for $g_j(x) > 0$, and gives a larger penalty for a larger constraint violation. Property (2) is a new characteristic used in our paper, which does not appear in previous penalty approaches (Bertsekas, 2014) to the best of our knowledge. It is also the most important one because it will lead to the final performance results of Algorithm 2 (see the proof of Theorem 1 for details). A broad class of penalty functions can satisfy these two properties for any compact set $\mathcal{X}$, e.g,

$$\psi(x) = \begin{cases} 1 & x \leq 0 \\ (cx+1)^n & x \geq 0 \end{cases} \tag{9}$$

$$\psi(x) = \begin{cases} 1 & x \leq 0 \\ \exp(cx) & x \geq 0 \end{cases} \tag{10}$$

where $n \geq 1$ and $c > 0$ are some constants.

Algorithm 2 can be regarded as an epoch-based algorithm. For epoch $l$, we will run the IGP-UCB method for $f(x) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x)) - 1)$ to produce $S$ decisions for $S$ iterations. At the end of each epoch, we will update the multiplier for each constraint as $\kappa_j^{l+1} = \kappa_j^l\psi\left(\frac{1}{S}\sum_{s=1}^{S}g_j(x_l^s)\right)$. The updates will force the outputs of the next epoch towards the region within the constraints by enlarging the multiplier of constraints where $\frac{1}{S}\sum_{s=1}^{S}g_j(x_l^s) > 0$.

The analysis of Algorithm 2 needs the following assumption.

**Assumption 3.** *$F_l(x) := f(x) - \sum_{j=1}^{m}\kappa_j^l(\psi(cg_j(x)) - 1)$ belongs to some RKHS $H_{k_l}$ with $||F_l||_{k_l} < B$.*

This assumption constrains the complexity of $F_l(x)$ so that we can use a Bayesian approach to get a guaranteed performance. Because $\kappa_j^l$ is different for each $l$, we may need a different kernel function $k_l$ for each epoch to satisfy the above assumption. See (Picheny et al., 2016) for how to determine the kernel and its hyperparameters.

Now we can show the performance results of Algorithm 2.

**Theorem 1.** *Assume that an unconstrained maximizer $x^{**} = \arg\max_{x\in\mathcal{X}} f(x)$ exists and $f(x^{**}) < \infty$. If Assumption 1-3 are satisfied, then running Algo-*

*rithm 2 with time horizon $T = LS$ leads to*

$$R_{LS} = \sum_{l=1}^{L}\sum_{s=1}^{S}[f(x^*) - f(x_l^s)]$$

$$= O(BL\sqrt{S\tilde{\gamma}_S} + L\sqrt{S\tilde{\gamma}_S(\tilde{\gamma}_S + \log(L/\delta))}) \tag{11}$$

$$V_{LS} = ||[\sum_{l=1}^{L}\sum_{s=1}^{S}\mathbf{g}(x_l^s)]^+||$$

$$= O\left(S\psi_+^{-1}\left(L + (BL\sqrt{\frac{\tilde{\gamma}_S}{S}} + L\sqrt{\frac{\tilde{\gamma}_S^2 + \tilde{\gamma}_S\log(L/\delta)}{S}})\right)\right) \tag{12}$$

*with probability at least $1 - \delta$, where $\psi_+^{-1}(x)$ is the inverse function of $\psi(x)$ when $x \geq 0$ and $\tilde{\gamma}_S$ is the maximal information gain with time horizon $S$ among $\{k_l\}_{l=1}^{L}$.*

*Proof.* See Section 1 of Supplementary Material. □

**Remark 1.** *From the proof of Theorem 1, we can replace the IGP-UCB method in the inner loops of Algorithm 2 by other unconstrained bandit optimization algorithms (such as GP-TS (Chowdhury and Gopalan, 2017)) that produce a sublinear regret for $f(x) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x)) - 1)$, then we can get a similar result to Theorem 1. Therefore Algorithm 2 can be considered as a universal method for our setup.*

Now we turn to the discussion of Theorem 1.

### 5.1 Choice of $\psi(\cdot)$

For simplicity, we assume that $k_l$'s in Algorithm 2 are all SE kernel functions (1) as an example. The results for other kernel functions can be similarly analyzed using the bound of $\gamma_T$ in the paper (Vakili et al., 2021).

From (Vakili et al., 2021), we know that for SE kernel functions, $\gamma_S = O(\log^{d+1} S)$ where $d$ is the domain dimension. Applying it to (11), we have the bounds of regret as

$$R_{LS} = O(BL\sqrt{S\log^{d+1} S} + L\sqrt{S(\log^{2d+2}(S) + \log^{d+1}(S)\log(L/\delta))}) \tag{13}$$

For CV, from (12) we can see that to make it sublinear, we need $n > 1$ if we choose (9) as $\psi(\cdot)$. To obtain the optimal order of CV, we need to choose (10) as $\psi(\cdot)$, which gives the following result:

$$V_{LS} = O(S\log(L + (BL\sqrt{\frac{\log^{d+1} S}{S}} + L\sqrt{\frac{\log^{2d+2} S + \log^{d+1} S\log(L/\delta)}{S}}))) \tag{14}$$

From the above results, Algorithm 2 achieves a sublinear regret and a sublinear CV as long as $L = o(T)$ or $S = o(T)$. Particularly, if we choose $L = O(\log T)$, then the regret matches the lower bound of Bayesian optimization algorithms using SE kernel (Table 1 of (Vakili et al., 2021)) up to a logarithmic factor.

Meanwhile, even though $\psi(x) = \exp(cx)$ gives the optimal order of CV in our theoretical analysis, it makes the multiplier-update, i.e., $\kappa_j^{l+1} = \kappa_j^l \exp\left(\frac{1}{S}\sum_{s=1}^{S} cg_j(x_l^s)\right)$, too aggressive when $\frac{1}{S}\sum_{s=1}^{S} g_j(x_l^s)$ is a large value. As a result, it is easy to make $\kappa_j^{l+1}$ overflow by applying the above update. If so, (9) can be used with a suitable $n$ chosen based on the range of $\mathbf{g}(x)$.

### 5.2 Impacts of $L$ and $S$

In this section, we want to discuss the effects of $S$ and $L$ on the performance of Algorithm 2, which are total inner iterations and total outer iterations, respectively. From (13) and (14), we can see that $R_{LS}$ has a higher order in terms of $L$ and $V_{LS}$ has a higher order in terms of $S$. This is also true for (11) and (12) if $\tilde{\gamma}_S = o(\sqrt{S})$, which is necessary to make $R_{LS}$ sublinear. Since $T = LS$, there exists a tradeoff between the bounds of these two metrics. In the extreme case, if we let $L$ to be constant, then (11) recovers the regret bound of the original IGP-UCB (Chowdhury and Gopalan, 2017), but meanwhile the CV is no longer guaranteed to be sublinear from (12).

This tradeoff can be understood intuitively as follows. If we make $S$ larger, then by running IGP-UCB longer in Algorithm 2, we have a smaller value of $\frac{1}{S}\sum_{s=1}^{S}[f(x_l^*) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x_l^*)) - 1) - f(x_l^s) + \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x_l^s)) - 1)]$, where $x_l^* = \arg\max_{x\mathcal{X}}[f(x) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x)) - 1)]$. From the proof of Theorem 1, $\frac{1}{S}[\sum_{s=1}^{S} f(x^*) - f(x_l^s)]$ is upper-bounded by the above value, thus the bound of time-averaged regret is also smaller. On the other hand, larger $S$ leads a smaller $L$ since $T = LS$. Then the updates of multiplier are less frequent, which may make the multiplier not large enough (because multipliers can only be enlarged in the updates) and thus lead to a larger CV. See Figure 1 of Supplementary Material for our demonstration of this tradeoff via an experiment.

Particularly, if $L = S$ and $\psi(x) = \exp(x)$, then both metrics are sublinear if $\tilde{\gamma}_S + \log\frac{S}{\delta} = o(\sqrt{S})$, which is determined by $\mathcal{X}$ and the kernels used in the algorithm. The original IGP-UCB (Chowdhury and Gopalan, 2017) has the same requirement (without the log term) for a sublinear regret, and it holds when $k_l$ is an SE kernel or a Matérn kernel with a certain $\nu$ by the bounds of $\gamma_S$ shown in the paper

(Vakili et al., 2021). Particularly, we have $R_T = \tilde{O}(B\sqrt{T^{3/2}\tilde{\gamma}_{\sqrt{T}}} + T^{3/4}\tilde{\gamma}_{\sqrt{T}})$ and $V_T = \tilde{O}(\sqrt{T})$ with a high probability for Algorithm 2 by neglecting the log terms. Compared with previous methods of OCO shown in Table 1, the CV of Algorithm 2 matches the best bound without assuming that both $f$ and $g$ are convex.

When the constraint function values are observed with noise, Algorithm 2 is not applicable because the penalty function $\psi(\cdot)$ will amplify the observation noise of $g_j$ since $\psi(\cdot)$ is at least higher than 1st-order from the discussion in Section 5.1. This motivates us to propose another method in the next section.

## 6 Noisy Constraint Observations

In this section, we assume that the observations of constraints contain subGaussian noise, i.e.,

**Assumption 4.** *$g_j$ is observed with independent $R$-subGaussian noise $\varepsilon_j$ for each $j$.*

Here the independence is defined with regard to any other noise.

To deal with this case, we change the rule of multiplier updates from a multiplicative form to an additive one and adopt a linear penalty function so that the noise of $g_j$ will not be amplified. The method, called GP-UCB with Noisy Constraints, is shown in Algorithm 3. It needs the following assumption similar to Assumption 3:

**Assumption 5.** *$G_l(x) := f(x) - \sum_{j=1}^{m} \kappa_j^l g_j(x)$ belongs to some RKHS $H_{k_l}$ with $||G_l||_{k_l} < B$.*

Algorithm 3 is an epoch-based method with a similar structure to Algorithm 2. However, due to the existence of the noise in the multiplier-updates, the proof of its performance results are far more complex. Since Algorithm 3 utilizes a less sharp penalty function, it needs more assumptions to guarantee the performance, which are listed as follows.

**Assumption 6.** *$|f(x)| \leq Q$ and $|g_j(x)| \leq M$ for each $j$ and $x \in \mathcal{X}$.*

**Assumption 7.** *There is a point $y$ s.t. $g_j(y) \leq -\epsilon, \forall j$, where $\epsilon > 0$.*

Assumption 6 requires the objective and the constraint functions to be bounded within $\mathcal{X}$, otherwise linear functions cannot incur enough penalty to constraint violation. Assumption 7 is Slater's condition (Boyd et al., 2004), which is commonly-used in constrained optimization. With these assumptions, we have the following results for Algorithm 3.

**Algorithm 3** GP-UCB with Noisy Constraints

```
Initialize $\kappa_j^1 = 0$ for all $j$.
for $l = 1, ..., L$ do
    Make the following $S$ decisions sequentially:
        $\{x_l^1, ..., x_l^S\} = \text{IGP-UCB}(f(x) - \sum_{j=1}^{m} \kappa_j^l g_j(x), k_l, B, \sqrt{(1 + \sum_{j=1}^{m} (\kappa_j^l)^2) R}, \lambda, \delta/(2L), S),$
    while obtaining $S$ observations $\{f(x_l^s) + \varepsilon_l^s\}_{s=1}^S$ and $\{g_j(x_l^s) + \varepsilon_{l,j}^s\}_{s=1}^S, \forall j$ sequentially with the above outputs, where $\varepsilon_l^s$ is the observation noise of $f$ and $\varepsilon_{l,j}^s$ is the observation noise of $g_j$.
    Set $\kappa_j^{l+1} = [\kappa_j^l + \mu \sum_{s=1}^S (g_j(x_l^s) + \varepsilon_{l,j}^s)/S]^+, \forall j$
end for
Output: $\{\{x_l^s\}_{l=1}^L\}_{s=1}^S$.
```

**Theorem 2.** *If Assumption 1 and 4-7 are satisfied with $\epsilon > 4(R\sqrt{2\tilde{\gamma}_S + 2 + 2\log\frac{2L}{\delta}})\sqrt{(S+2)\tilde{\gamma}_S}/S + 2R(\sqrt{4m/S} + \sqrt{\log(2L/\delta)/S})$, then running Algorithm 3 with time horizon $T = LS$ with $\mu = O(1/\sqrt{L})$ leads to*

$$R_{LS} = O\left(L(B + \sqrt{\tilde{\gamma}_S + \log\frac{2L}{\delta}})\sqrt{S\tilde{\gamma}_S} + S\sqrt{L}\right)$$

$$V_{LS} = O(L\sqrt{S\log(L/\delta)} + S\sqrt{L})$$

*with a probability at least $1-\delta$, where $\tilde{\gamma}_S$ is the maximal information gain with time horizon $S$ among $\{k_l\}_{l=1}^L$.*

*Proof.* See Section 3 of Supplementary Material. □

**Remark 2.** *Unlike Algorithm 2, it is nontrivial to extend our results to the case where IGP-UCB in Algorithm 3 is replaced by other unconstrained methods.*

The proof of Theorem 2 is different from Theorem 1 due to the fact that the noise in the observation of $f(x) - \sum_{j=1}^{m} \kappa_j^l g_j(x)$ is $\sqrt{1 + \sum_{j=1}^{m} (\kappa_j^l)^2} R$-subGaussian. If the values of $\kappa_j^l$ are related to time horizon $l$, then we cannot directly utilize Lemma 1 in our proof, where the subGaussian parameter is assumed to be constant. Therefore, we take most efforts to prove the order of $\kappa_j^l$ to get the final result, which is the most difficult part of our proof.

Compared with Algorithm 2, Algorithm 3 suffers from a higher order of CV due to the existence of noise in the constraint observations and the fact that a linear penalty function cannot incur a penalty that is as strong as the one in Algorithm 2. Particularly, if we choose $L = S$, then we achieve $R_T =$

$\tilde{O}(B\sqrt{T^{3/2}\tilde{\gamma}_{\sqrt{T}}} + T^{3/4}\tilde{\gamma}_{\sqrt{T}})$ and $V_T = \tilde{O}(T^{3/4})$ with a high probability by neglecting the log terms, and the regret is in the same order with the one of Algorithm 2 selecting $L = S$. Similar to Algorithm 2, both metrics are sublinear if $\tilde{\gamma}_S + \log \frac{2S}{\delta} = o(\sqrt{S})$. Meanwhile with this choice, the right hand side of $\epsilon > 4(R\sqrt{2\tilde{\gamma}_S + 2 + 2\log\frac{2L}{\delta}})\sqrt{(S+2)\tilde{\gamma}_S}/S + 2R(\sqrt{4m/S} + \sqrt{\log(2L/\delta)/S})$ in Theorem 2 won't increase as $L$ and $S$ increase, which becomes $\epsilon > 4(R\sqrt{2\tilde{\gamma}_1 + 2 + 2\log\frac{2}{\delta}})\sqrt{3\tilde{\gamma}_1} + 2R(\sqrt{4m} + \sqrt{\log(2/\delta)})$.

## 7 Conclusion

In this paper, motivated by problems from diverse domains, we presented two provably good Bayesian methods for stochastic continuum-armed bandit with long-term constraints. Specifically, when the constraint value is observed without noise, we used a class of penalty functions based on a multiplicative form of multiplier-updates in our first method to get a bound of constraint violation that can match the best result of previous methods, along with a sublinear regret. For the case of noisy constraint observations, we proposed another method based on an additive way of multiplier-updates with a linear penalty function to avoid noise amplification, which can produce a sublinear CV and the same order of regret with the first method. By conducting two experiments, we demonstrated the efficiency of our algorithms compared with two methods in online convex optimization and Bayesian optimization.

## 8 Acknowledgement

We thank the NSF grants: IIS-2112471, CNS-NeTS-2106679, CNS-NeTS-2007231, CNS-SpecEES-1824337, CNS-NeTS-1717045; and the ONR Grant N00014-19-1-2621 for their support of this work. We also thank the suggestions of all the reviewers and the meta-reviewer for the improvement of this paper.

## References

Lecture 8. `http://www.stat.cmu.edu/~arinaldo/Teaching/36709/S19/Scribed_Lectures/Feb21_Shenghao.pdf`. Accessed: 2021-10-10.

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33 (6):1926–1951, 1995.

Shipra Agrawal and Nikhil Devanur. Linear contextual bandits with knapsacks. *Advances in Neural Information Processing Systems*, 29:3450–3458, 2016.

Shipra Agrawal and Nikhil R Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006, 2014.

Setareh Ariafar, Jaume Coll-Font, Dana H Brooks, and Jennifer G Dy. Admmbo: Bayesian optimization with unknown constraints using admm. *J. Mach. Learn. Res.*, 20(123):1–26, 2019.

Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *International Conference on Computational Learning Theory*, pages 454–468. Springer, 2007.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216. IEEE, 2013.

Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Xuanyu Cao and KJ Ray Liu. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Transactions on automatic control*, 64(7):2665–2680, 2018.

Tianyi Chen and Georgios B Giannakis. Bandit convex optimization for scalable and dynamic iot management. *IEEE Internet of Things Journal*, 6(1): 1276–1286, 2018.

Tianyi Chen, Qing Ling, and Georgios B Giannakis. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24):6350–6364, 2017.

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

Eric W Cope. Regret and convergence bounds for a class of continuum-armed bandit problems. *IEEE Transactions on Automatic Control*, 54(6):1243–1253, 2009.

David Eriksson and Matthias Poloczek. Scalable constrained bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 730–738. PMLR, 2021.

Atilla Eryilmaz and Rayadurgam Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control.

*IEEE/ACM transactions on networking*, 15(6): 1333–1344, 2007.

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

Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945, 2014.

Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org.

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

José Miguel Hernández-Lobato, Michael A Gelbart, Ryan P Adams, Matthew W Hoffman, and Zoubin Ghahramani. A general framework for constrained bayesian optimization using information-based search. 2016.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. A tutorial review of rkhs methods in machine learning. In *Technical Report*. 2005.

Nicole Immorlica, Karthik Abinav Sankararaman, Robert Schapire, and Aleksandrs Slivkins. Adversarial bandits with knapsacks. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 202–219. IEEE, 2019.

Tinu Theckel Joy, Santu Rana, Sunil Gupta, and Sveha Venkatesh. Hyperparameter tuning for big data using bayesian optimisation. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2574–2579. IEEE, 2016.

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. *Advances in Neural Information Processing Systems*, 17:697–704, 2004.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

J.-W. Lee, R.R. Mazumdar, and N.B. Shroff. Nonconvex optimization and rate control for multi-class services in the internet. *IEEE/ACM Transactions on Networking*, 13(4):827–840, 2005. doi: 10.1109/TNET.2005.852876.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14 (2):495–519, 2019.

Omid Madani, Daniel J Lizotte, and Russell Greiner. The budgeted multi-armed bandit problem. In *International Conference on Computational Learning Theory*, pages 643–645. Springer, 2004.

Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research*, 13(1):2503–2528, 2012.

Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774, 2018.

Valerio Perrone, Iaroslav Shcherbatyi, Rodolphe Jenatton, Cedric Archambeau, and Matthias Seeger. Constrained bayesian optimization with max-value entropy search. *arXiv preprint arXiv:1910.07003*, 2019.

Victor Picheny, Robert B Gramacy, Stefan Wild, and Sébastien Le Digabel. Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1443–1451, 2016.

Jian Qing Shi and Taeryon Choi. *Gaussian process regression analysis for functional data*. CRC Press, 2011.

Shashank Singh. Continuum-armed bandits: A function space perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 2620–2628. PMLR, 2021.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

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.

Yingce Xia, Haifang Li, Tao Qin, Nenghai Yu, and Tie-Yan Liu. Thompson sampling for budgeted multi-armed bandits. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Hao Yu, Michael J Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. *arXiv preprint arXiv:1708.03741*, 2017.

# Supplementary Material: A Bayesian Approach for Stochastic Continuum-armed Bandit with Long-term Constraints

## A Proof of Theorem 1

Within epoch $l$ of Algorithm 2, we can utilize Lemma 1 in the main paper for IGP-UCB to give the following result due to Assumption 1 and 3:

With probability at least $1-\delta/L$, we have:

$$\sum_{s=1}^{S}[f(x_l^*) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x_l^*)) - 1) - f(x_l^s) + \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x_l^s)) - 1)] = O(B\sqrt{S\gamma_S^l} + \sqrt{S\gamma_S^l(\gamma_S^l + \log(L/\delta))}) \tag{15}$$

for an $l$, where $x_l^* = \arg\max_{x\in\mathcal{X}} f(x) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x)) - 1)$ and $\gamma_S^l$ is information gain with time horizon $S$ for $k_l$.

Meanwhile, by the definition of $x^*$ and $x_l^*$, we have

$$\begin{aligned} f(x^*) &= f(x^*) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x^*)) - 1) \\ &\leq f(x_l^*) - \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x_l^*)) - 1) \end{aligned}$$

since $g_j(x^*) \leq 0, \forall j$. Applying the above inequality to (15), we have

$$\sum_{s=1}^{S}[f(x^*) - f(x_l^s) + \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x_l^s)) - 1)] = O(B\sqrt{S\gamma_S^l} + \sqrt{S\gamma_S^l(\gamma_S^l + \log(L/\delta))}) \tag{16}$$

for an $l$ with probability at least $1-\delta/L$. Now by the union bound, with probability at least $1-\delta$, we have

$$\sum_{l=1}^{L}\sum_{s=1}^{S}[f(x^*) - f(x_l^s) + \sum_{j=1}^{m}\kappa_j^l(\psi(g_j(x_l^s)) - 1)] = O(BL\sqrt{S\tilde{\gamma}_S} + L\sqrt{S\tilde{\gamma}_S(\tilde{\gamma}_S + \log(L/\delta))}) \tag{17}$$

To bound $R_{LS} = \sum_{l=1}^{L}\sum_{s=1}^{S}(f(x^*) - f(x_l^s))$, we need to bound $\sum_{l=1}^{L}\sum_{s=1}^{S}\kappa_j^l(\psi(g_j(x_l^s)) - 1)$ for any $j$. By convexity of $\psi(\cdot)$, we have

$$\sum_{l=1}^{L}\kappa_j^l\sum_{s=1}^{S}(\psi(g_j(x_l^s)) - 1) \geq S\sum_{l=1}^{L}\kappa_j^l(\psi(\sum_{s=1}^{S}g_j(x_l^s)/S) - 1) \tag{18}$$

Since $\kappa_j^{l+1} = \kappa_j^l\psi(\sum_{s=1}^{S}g_j(x_l^s)/S)$ from Algorithm 2, we have $S\kappa_j^l(\psi(\sum_{s=1}^{S}g_j(x_l^s)/S) - 1) = S(\kappa_j^{l+1} - \kappa_j^l)$. Therefore, along with $\kappa_j^1 = 1$, we have

$$S\sum_{l=1}^{L}\kappa_j^l(\psi(\sum_{s=1}^{S}g_j(x_l^s)/S) - 1) = S(\kappa_j^{l+1} - \kappa_j^1) = S(\prod_{l=1}^{L}\psi(\sum_{s=1}^{S}g_j(x_l^s)/S) - 1) \tag{19}$$

By Assumption 2, we have

$$\prod_{l=1}^{L}\psi(\sum_{s=1}^{S}g_j(x_l^s)/S)\geq\psi(\sum_{l=1}^{L}\sum_{s=1}^{S}g_j(x_l^s)/S) \tag{20}$$

Now, since $0\leq\Delta=f(x^{**})-f(x^*)<\infty$ by the assumption, along with (18),(19) and (20), we have

$$\begin{aligned}-LS\Delta=&\sum_{l=1}^{L}\sum_{s=1}^{S}(f(x^*)-f(x^{**}))\\ \leq&\sum_{l=1}^{L}\sum_{s=1}^{S}(f(x^*)-f(x_l^s))\\ \leq&\, O\left(BL\sqrt{S\tilde{\gamma}_S}+L\sqrt{S\tilde{\gamma}_S(\tilde{\gamma}_S+\log(L/\delta))}\right)-S\sum_{j=1}^{m}(\psi(\sum_{l=1}^{L}\sum_{s=1}^{S}g_j(x_l^s)/S)-1).\end{aligned} \tag{21}$$

with probability at least $1-\delta$. Therefore,

$$S\sum_{j=1}^{m}(\psi(\sum_{l=1}^{L}\sum_{s=1}^{S}g_j(x_l^s)/S)-1)\leq LS\Delta+O\left(BL\sqrt{S\tilde{\gamma}_S}+L\sqrt{S\tilde{\gamma}_S(\tilde{\gamma}_S+\log(L/\delta))}\right) \tag{22}$$

with probability at least $1-\delta$. Since $\psi(\sum_{l=1}^{L}\sum_{s=1}^{S}g_j(x_l^s)/S)-1\geq 0$ for any $j$, we have

$$S(\psi(\sum_{l=1}^{L}\sum_{s=1}^{S}g_j(x_l^s)/S)-1)=O\left(LS+BL\sqrt{S\tilde{\gamma}_S}+L\sqrt{S\tilde{\gamma}_S(\tilde{\gamma}_S+\log(L/\delta))}\right)$$

for any $j$ with probability at least $1-\delta$, which results in

$$\sum_{l=1}^{L}\sum_{s=1}^{S}g_j(x_l^s)=O\left(S\psi_+^{-1}(L+(BL\sqrt{\tilde{\gamma}_S/S}+L\sqrt{\tilde{\gamma}_S(\tilde{\gamma}_S+\log(L/\delta))/S}))\right)$$

for any $j$ with probability at least $1-\delta$ from (22). The above bound gives the final result of CV.

Meanwhile, since $S\sum_{j=1}^{m}(\psi(\sum_{l=1}^{L}\sum_{s=1}^{S}g_j(x_l^s)/S)-1)\geq 0$, we have
$R_{LS}=O(BL\sqrt{S\tilde{\gamma}_S}+L\sqrt{S\tilde{\gamma}_S(\tilde{\gamma}_S+\log(L/\delta))})$ with a probability at least $1-\delta$ from (21).

## B Performance of Algorithm 2 for Other Definitions of Regret

We note that there are some works, such as Yu et al. (2017), which define regret as $\sum_{t=1}^{T}[f^t(x^*)-f^t(x_t)]$ with $x^*=\arg\max_{x\in\mathcal{X}}\{\sum_{t=1}^{T}f^t(x)\text{ s.t. }\mathbf{g}(x)\preceq 0\}$. In our setup, $f^t$ is a realization of $f$ at time $t$. So it is easy to extend our results to this regret by a similar proof to the one of Theorem 1, with assumptions that an unconstrained maximizer $x^{**}=\arg\max_{x\in\mathcal{X}}\sum_{l=1}^{L}\sum_{s=1}^{S}f_l^s(x)$ exists and $f_l^s(x^{**})<\infty$ for any $l$ and $s$. Here $f_l^s(x)=f(x)+\varepsilon_l^s$ is a realization at iteration $s$ of epoch $l$ in Algorithm 2. The only difference in this extension is to utilize the Hoeffding inequality for $|\sum_{s=1}^{S}f_l^s(x)-Sf(x)|$ to get a bound of $\sum_{s=1}^{S}[f_l^s(x^*)-f_l^s(x_l^s)+\sum_{j=1}^{m}\frac{\kappa_j^l}{c}\psi(cg_j(x_l^s))]$ based on (15). Then following the same procedure of the proof for Theorem 1, we can get the result which has the same order with Theorem 1.

## C Proof of Theorem 2

For epoch $l$ of Algorithm 3, we cannot directly use Lemma 1 in the main paper to give an intermediate result like (15). It is because the distribution of total observation noise for $f(x)-\sum_{j=1}^{m}\kappa_j^l g_j(x)$ is $\sqrt{1+\sum_{j=1}^{m}(\kappa_j^l)^2}R$-subGaussian and $\kappa_j^l$ can be related to $l$ due to the multiplier update. Meanwhile, from Theorem 1 of Chowdhury and Gopalan (2017), we have the following lemma.

**Lemma 2.** *With Assumption 1, 4 and 5, for a given $\eta > 0$ we have*

$$||\varepsilon_l^{1:s} + \sum_{j=1}^m \kappa_j^l \varepsilon_{l,j}^{1:s}||^2_{((K_t+\eta I)^{-1}+I)^{-1}} \leq 2(1+\sum_{j=1}^m (\kappa_j^l)^2) R^2 \log(\frac{2L\sqrt{\det((1+\eta)I+K_s)}}{\delta}) \tag{23}$$

*simultaneously over all $s > 0$ for an $l$ with probability at least $1-\delta/(2L)$, where $\varepsilon_l^{1:s} = [\varepsilon_l^1, .., \varepsilon_l^s]$ and $\varepsilon_{l,j}^{1:s} = [\varepsilon_{l,j}^1, .., \varepsilon_{l,j}^s]$ are previous observation noise in epoch $l$, $K_s$ is an $s \times s$ matrix defined in Section 2 of the main paper, $||x||_A = \sqrt{x^T A x}$ for a matrix $A$ and $I$ is an $s \times s$ identity matrix.*

Now following the same proof procedure of Lemma 1 in the main paper, we can get the following lemma:

**Lemma 3.** *With probability at least $1-\delta/(2L)$, we have*

$$\begin{aligned}&\sum_{s=1}^S [f(x_l^*) - \sum_{j=1}^m \kappa_j^l g_j(x_l^*) - f(x_l^s) + \sum_{j=1}^m \kappa_j^l g_j(x_l^s)] \\ &\leq 4\left(B + \sqrt{(1+\sum_{j=1}^m (\kappa_j^l)^2)} R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})}\right)\sqrt{(S+2)\gamma_S^l}\end{aligned} \tag{24}$$

*for an $l$, where $x_l^* = \arg\max_{x\in\mathcal{X}} f(x) - \sum_{j=1}^m \kappa_j^l g_j(x)$ and $\gamma_S^l$ is information gain with time horizon $S$ for $k_l$.*

Denote the above event as $E_l(\delta)$.

Meanwhile, by positivity of $\kappa_j^l$, the definition of $x^*$ and $x_l^*$, we have

$$f(x^*) \leq f(x^*) - \sum_{j=1}^m \kappa_j^l g_j(x^*) \leq f(x_l^*) - \sum_{j=1}^m \kappa_j^l g_j(x_l^*)$$

So using the event $E_l(\delta)$, for an $l$

$$\begin{aligned}&\sum_{s=1}^S [f(x^*) - f(x_l^s) + \sum_{j=1}^m \kappa_j^l g_j(x_l^s)] \\ &\leq 4\left(B + \sqrt{(1+\sum_{j=1}^m (\kappa_j^l)^2)} R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})}\right)\sqrt{(S+2)\gamma_S^l}\end{aligned} \tag{25}$$

with probability at least $1-\delta/(2L)$.

Now, since $\kappa_j^{l+1} = [\kappa_j^l + \mu\sum_{s=1}^S (g_j(x_l^s) + \varepsilon_{l,j}^s)/S]^+ \geq \kappa_j^l + \mu\sum_{s=1}^S (g_j(x_l^s) + \varepsilon_{l,j}^s)/S$, we have

$$\kappa_j^{L+1} \geq \mu \sum_{l=1}^L \sum_{s=1}^S (g_j(x_l^s) + \varepsilon_{l,j}^s)/S \tag{26}$$

Also define another event $G_l(\delta)$ as

$$\sqrt{\sum_{j=1}^m (\sum_{s=1}^S \varepsilon_{l,j}^s)^2} \leq 2R(\sqrt{4mS} + \sqrt{S\log(2L/\delta)}),$$

which occurs with probability at least $1-\delta/(2L)$ for an $l$ from Theorem 8.3 of lec. Using the events $\{G_l(\delta)\}_{l=1}^L$ and (26), we have

$$V_{LS} \leq \frac{S||\kappa^{L+1}||}{\mu} + \sum_{l=1}^L \sqrt{\sum_{j=1}^m (\sum_{s=1}^S \varepsilon_{l,j}^s)^2} \leq \frac{S||\kappa^{L+1}||}{\mu} + 2RL(\sqrt{4mS} + \sqrt{S\log(2L/\delta)}) \tag{27}$$

with probability at least $1-\delta/2$, where $\kappa^l = [\kappa_1^l, ..., \kappa_m^l]$. From (25) and (27), we find that to obtain the bounds of $R_{LS}$ and $V_{LS}$, it is important to bound $||\kappa^l||$. We will do it in the following.

Since $\kappa_j^{l+1} = [\kappa_j^l + \mu\sum_{s=1}^S (g_j(x_l^s) + \varepsilon_{l,j}^s)/S]^+$, we have

$$(\kappa_j^{l+1})^2 \leq (\kappa_j^l + \mu\sum_{s=1}^S (g_j(x_l^s) + \varepsilon_{l,j}^s)/S)^2$$

which leads to

$$\begin{aligned}\sum_{s=1}^S \kappa_j^l (g_j(x_l^s) + \varepsilon_{l,j}^s) &\geq S\frac{(\kappa_j^{l+1})^2 - (\kappa_j^l)^2}{2\mu} - \frac{S\mu}{2}(\sum_{s=1}^S (g_j(x_l^s) + \varepsilon_{l,j}^s)/S)^2 \\ &\geq S\frac{(\kappa_j^{l+1})^2 - (\kappa_j^l)^2}{2\mu} - \frac{\mu}{S}[(\sum_{s=1}^S g_j(x_l^s))^2 + (\sum_{s=1}^S \varepsilon_{l,j}^s)^2]\end{aligned}$$

So using the event $G_l(\delta)$ along with the assumption $|g_j(x)| \leq M$ and Cauchy-Swartz inequality, we have for an $l$

$$\begin{aligned}&\sum_{j=1}^m\sum_{s=1}^S \kappa_j^l g_j(x_l^s) \geq S\frac{||\kappa^{l+1}||^2 - ||\kappa^l||^2}{2\mu} - mS\mu M^2 - 4\mu R^2(\sqrt{4m} + \sqrt{\log(2L/\delta)})^2 \\ &- 2R||\kappa^l|| \cdot (\sqrt{4Sm} + \sqrt{S\log(2L/\delta)})\end{aligned} \tag{28}$$

with probability at least $1-\delta/(2L)$.
From Assumption 7, there is a point $y$ s.t. $g_j(y) \leq -\epsilon, \forall j$, where $\epsilon > 0$. Using this point and $E_l(\delta)$, we have

$$\begin{aligned}&\sum_{s=1}^S [f(y) - \sum_{j=1}^m \kappa_j^l g_j(y) - f(x_l^s) + \sum_{j=1}^m \kappa_j^l g_j(x_l^s)] \\ &\leq \sum_{s=1}^S [f(x_l^*) - \sum_{j=1}^m \kappa_j^l g_j(x_l^*) - f(x_l^s) + \sum_{j=1}^m \kappa_j^l g_j(x_l^s)] \\ &\leq 4\left(B + \sqrt{(1+\sum_{j=1}^m (\kappa_j^l)^2)} R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})}\right)\sqrt{(S+2)\gamma_S^l} \\ &\leq 4\left(B + (1+||\kappa^l||)R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})}\right)\sqrt{(S+2)\gamma_S^l}\end{aligned} \tag{29}$$

for an $l$ with probability at least $1-\delta/(2L)$, where the last inequality comes from positivity of $\kappa_j^l$.

Meanwhile from (28) and Assumption 6, we have with probability at least $1-\delta/(2L)$,

$$\begin{aligned}&\sum_{s=1}^S [f(y) - \sum_{j=1}^m \kappa_j^l g_j(y) - f(x_l^s) + \sum_{j=1}^m \kappa_j^l g_j(x_l^s)] \\ &\geq -2SQ + S\sum_{j=1}^m \kappa_j^l \epsilon + \sum_{s=1}^S\sum_{j=1}^m \kappa_j^l g_j(x_l^s) \\ &\geq -2SQ + S\epsilon||\kappa^l|| + S\frac{||\kappa^{l+1}||^2 - ||\kappa^l||^2}{2\mu} \\ &- mS\mu M^2 - 4\mu R^2(\sqrt{4m} + \sqrt{\log(2L/\delta)})^2 - 2R||\kappa^l|| \cdot (\sqrt{4Sm} + \sqrt{S\log(2L/\delta)})\end{aligned} \tag{30}$$

for an $l$, where the second inequality utilize the positivity of $\kappa_j^l$. Now from (29), (30) and the union bound, we have for an $l$

$$\begin{aligned}&-2SQ + S\epsilon||\kappa^l|| + S\frac{||\kappa^{l+1}||^2 - ||\kappa^l||^2}{2\mu} - mS\mu M^2 - 4\mu R^2(\sqrt{4m} + \sqrt{\log(2L/\delta)})^2 - \\ &2R||\kappa^l|| \cdot (\sqrt{4Sm} + \sqrt{S\log(2L/\delta)}) \leq 4\left(B + (1+||\kappa^l||)R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})}\right)\sqrt{(S+2)\gamma_S^l}\end{aligned} \tag{31}$$

with probability at least $1-\delta/L$. Meanwhile, using $G_l(\delta)$ with the multiplier-update relation, we have

$$||\kappa^l|| \geq ||\kappa^{l+1}|| - ||\kappa^{l+1}-\kappa^l|| \geq ||\kappa^{l+1}|| - (\sqrt{m}\mu M + 2\mu R(\sqrt{4m}+\sqrt{\log(2L/\delta)})/S) \tag{32}$$

with probability at least $1-\delta/(2L)$. Now we will prove the following bound using (31) and (32) by induction.

$$\begin{aligned} ||\kappa^l|| \leq{} & \sqrt{m}\mu M + 2\mu R(\sqrt{4m}+\sqrt{\log(2L/\delta)})/S) \\ & + \frac{2Q + 4(B + R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})})\sqrt{(S+2)\gamma_S^l/S} + m\mu M^2 + 4\mu R^2(\sqrt{4m}+\sqrt{\log(2L/\delta)})^2/S}{\epsilon - 4(R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})})\sqrt{(S+2)\gamma_S^l/S} - 2R(\sqrt{4m/S}+\sqrt{\log(2L/\delta)/S})}, \end{aligned} \tag{33}$$

which occurs for an $l$ with probability at least $1-\delta/L$.

- From the requirement of Theorem 2, the above bound is positive. Since $\kappa_j^1 = 0$ for all $j$, the above bound is right for $\kappa^1$.
- Suppose that $||\kappa^l||$ is less or equal to the right-hand side of (33)
- If $||\kappa^{l+1}||$ is larger than the right-hand side of (33), then from (32),

  $$||\kappa^l|| > \frac{2Q + 4(B + R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})})\sqrt{(S+2)\gamma_S^l/S} + m\mu M^2 + 4\mu R^2(\sqrt{4m}+\sqrt{\log(2L/\delta)})^2/S}{\epsilon - 4(R\sqrt{2(\gamma_S^l + 1 + \log\frac{2L}{\delta})})\sqrt{(S+2)\gamma_S^l/S} - 2R(\sqrt{4m/S}+\sqrt{\log(2L/\delta)/S})}.$$

  By (31) and the above inequality, we have $||\kappa^{l+1}||^2 < ||\kappa^l||^2$, which leads to contradiction. Therefore, $||\kappa^{l+1}||$ is also less and equal than the right-hand side of (33)

Since $\mu = O(1/\sqrt{L})$, $||\kappa^l|| = O(1)$ from (33) when both $L$ and $S$ goes to infinity. Therefore from (28) and (33), we have

$$-\sum_{j=1}^{m}\sum_{l=1}^{L}\sum_{s=1}^{S}\kappa_j^l g_j(x_l^s) = O(S\sqrt{L} + L\sqrt{S\log(L/\delta)}) \tag{34}$$

with probability at least $1-\delta$ because (28) happens due to the event $G_l(\delta)$, and (33) happens due to the two events $G_l(\delta)$ and $E_l(\delta)$. It leads to

$$R_{LS} = \sum_{l=1}^{L}\sum_{s=1}^{S}[f(x^*) - f(x_l^s)] = O\left(L(B + \sqrt{\tilde{\gamma}_S + \log\frac{2L}{\delta}})\sqrt{S\tilde{\gamma}_S} + S\sqrt{L}\right)$$

with probability at least $1-\delta$ by (25) and (34) because (25) happens due to the event $E_l(\delta)$ and (34) happens due to the events $G_l(\delta)$ and $E_l(\delta)$.

From (27) and (33),

$$V_{LS} = O(L\sqrt{S\log(L/\delta)} + S\sqrt{L})$$

with probability at least $1-\delta$ because (27) happens due to the event $G_l(\delta)$ and (33) happens due to the events $G_l(\delta)$ and $E_l(\delta)$.

## D Performance of Algorithm 3 for Other Definitions of CV

There are some works such as Chen and Giannakis (2018); Cao and Liu (2018) defining CV as $||[\sum_{t=1}^{T}\mathbf{g}^t(x_t)]^+||$ for time-varying constraint functions $\mathbf{g}^t$. Since in our setup, $g_j^t(x) = g_j(x) + \varepsilon_j$ where $\varepsilon_j$ is from a subGaussian distribution, we can get the result of this CV based on our result of Theorem 2 by using the Hoeffding inequality, which yields the same order with our result.

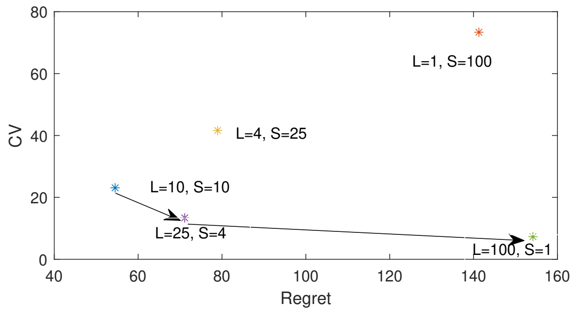


Figure 1: Mean regret and CV of Algorithm 2 with different $L$ and $S$ applied for the first experiment

# E Experiments

In this section, we will conduct two experiments to demonstrate the efficiency of our methods through comparison with two benchmarks. The first benchmark is constrained OCO with bandit feedback proposed in Cao and Liu (2018), and the second is ADMMBO proposed in Ariafar et al. (2019) as a representation of constrained BO. The section numbers mentioned in the following are referred as the ones in this appendix if there is no explicit statement.

## E.1 Test Problem with a Small Feasible Region

First, we use one synthetic function to test our methods, which were also used in previous works on constrained BO Ariafar et al. (2019); Gardner et al. (2014). Consider our setup with two-dimensional $\mathcal{X} = [0, 6]^2$, $f(x) = -\sin(x(1)) - x(2)$ and $g(x) = \sin(x(1))\sin(x(2)) + 0.95$. This is a challenging problem since both reward and constraint functions are highly nonlinear. Meanwhile, the feasible area $g(x) \leq 0$ within $\mathcal{X}$ is very small, which makes the problem more difficult.

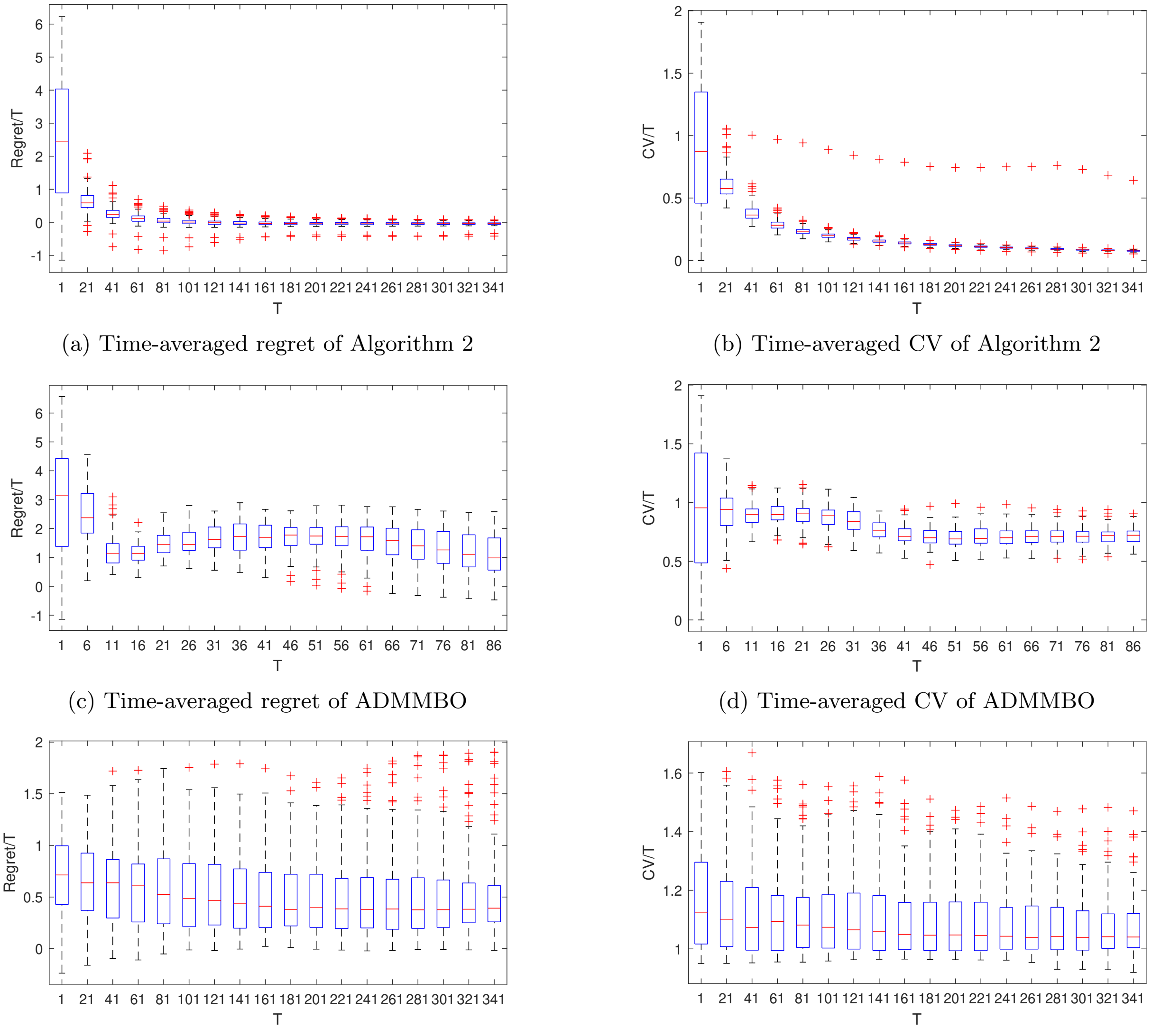
### E.1.1 Noiseless Constraint Observations

First we consider the case where $f(x)$ is observed with a Gaussian noise sampled from $\mathcal{N}(0, 0.01)$ and $g(x)$ is observed without noise. The reason why we use the noise variance of 0.01 is that the optimal value is already very small, which is around $-0.25$.

Here we want to use this experiment to demonstrate the effects of $L$ and $S$ for Algorithm 2. We use $\psi(x) = \exp(x)$ and set $S = 20$. We use Matérn kernel with $\nu = 2.5$ as the kernel function. Figure 1 shows the average regret and CV of Algorithm 2 with different $L$ and $S$, each for 20 runs. From the figure, we can see that regret-CV tradeoff happens for the three choices connected by arrows. The reason why it does not happen for the other two choices may be due to the constant factor in the bounds. But still, we can see that more frequent multiplier-updates can help reduce the CV of our algorithm.

Now we compare Algorithm 2 with constrained OCO with bandit feedback and ADMMBO for this problem. In the following, we choose $S = 20$ while the other hyperparameters are the same with the above. The hyperparameters in the OCO method are set to achieve the best performance. The original code of ADMMBO for this problem can be accessible at https://github.com/SetarehAr/ADMMBO, where we add noise to the observation values of the reward function.

In Figure 2, we show the boxplots of the time-averaged regret and time-averaged CV for these three methods. Each method is run for 100 times starting from different initial points. For Algorithm 2 and OCO, we choose the total time horizon to be 350 and plot the boxplots for every 20 iterations, while for ADMMBO, we cannot do that because it has its own stopping criteria (the change of a parameter is less than a threshold). Therefore, the time horizon of ADMMBO is limited to be around 90 for these 100 runs and the boxplots are plotted for every 5 iterations. From the figure, we can see clearly that the regret and CV of Algorithm 2 are sublinear with a high probability, whereas the regret and CV of the other two methods are almost linear after few iterations. For OCO, the reason is that $f$ and $g$ are both nonconvex and the algorithm may be stuck at zero gradient points that are suboptimal. For ADMMBO, the method is not designed for our performance metrics even though using



(a) Time-averaged regret of Algorithm 2

(b) Time-averaged CV of Algorithm 2

(c) Time-averaged regret of ADMMBO

(d) Time-averaged CV of ADMMBO

(e) Time-averaged regret of OCO

(f) Time-averaged CV of OCO

Figure 2: Boxplots of Algorithm 2, OCO and ADMMBO over 100 runs for Section E.1.1

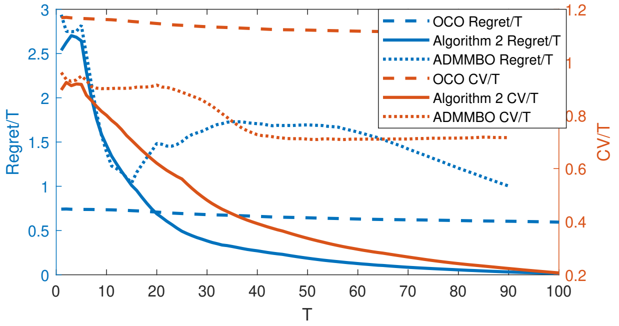


Figure 3: Comparison of Algorithm 2, OCO and ADMMBO in terms of means over 100 runs for Section E.1.1.

a Bayesian approach as well.

To make the comparison of these three methods more straightforward, we also plot the means of their performance metrics over the 100 runs and put them together in Figure 3. Since the whole time horizon of ADMMBO is only around 90, we only show the first 100 iterations of Algorithm 2 and OCO for a more clear illustration. From the figure, we can see that the mean regret and CV of Algorithm 2 are both sublinear, and their time-averaged values tend to 0 in the whole process. For OCO, its performance metrics almost never change after few iterations. It may be stuck at some suboptimal point. For ADMMBO, its performance metrics are also worse than our method.

### E.1.2 Noisy Constraint Observations

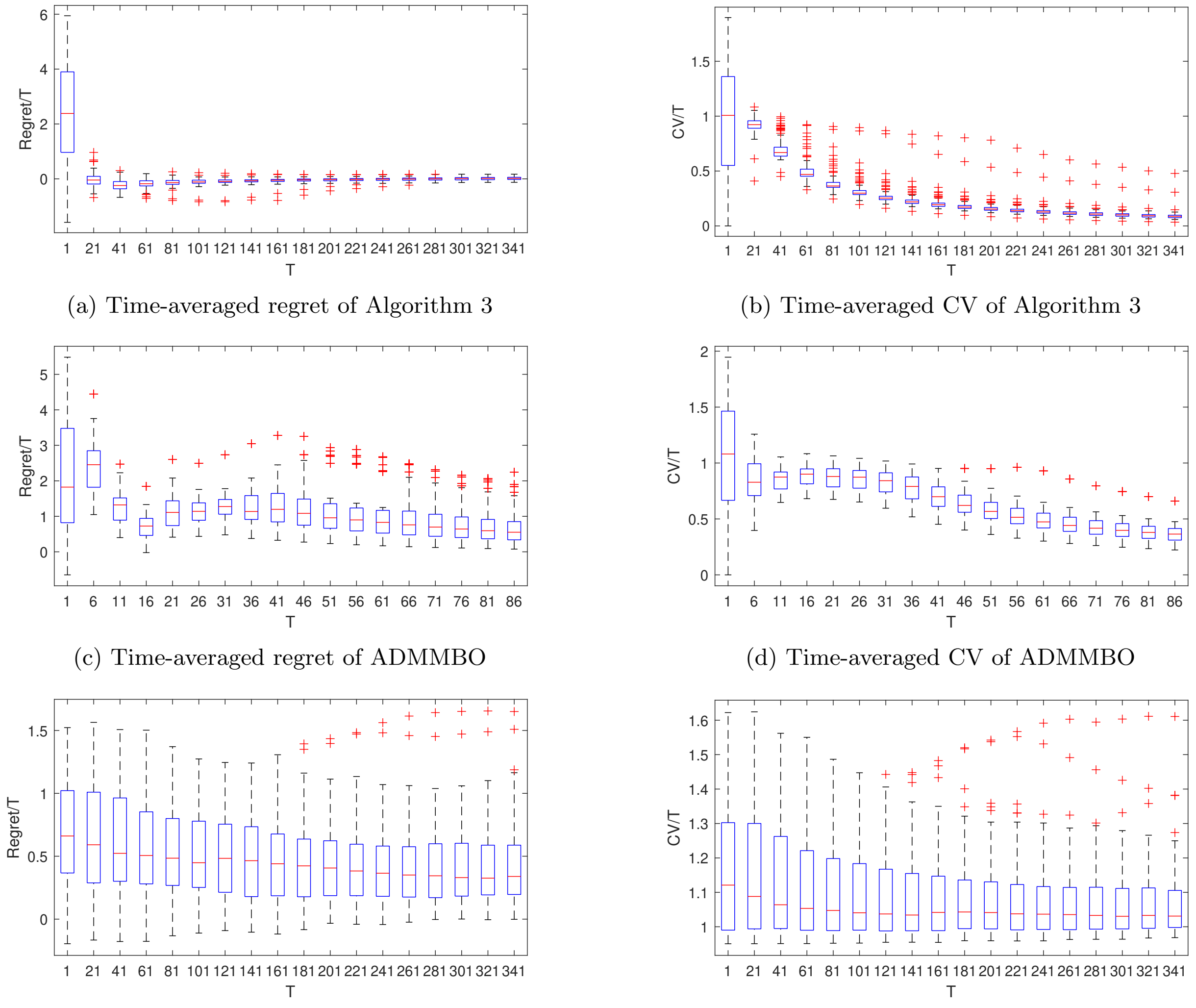
In this part, we assume that the constraint $g(x)$ is also observed with noise sampled from $\mathcal{N}(0, 0.01)$. We choose the variance to be 0.01 due to the small values of $g(x)$. Other problem settings are the same with Section E.1.1. Now we apply Algorithm 3, constrained OCO with bandit feedback and ADMMBO to this problem. Particularly in Algorithm 3, we choose $\mu = 0.5$ in its multiplier update. Other hyperparameters of Algorithm 3 are the same with Algorithm 2 in Section E.1.1. The hyperparameters in the OCO method are set to achieve the best performance.

In Figure 4, the boxplots of time-averaged regret and time-averaged CV are shown for these three methods. The experiment setup is the same with Section E.1.1, and the time horizon of ADMMBO is still limited to 90 because of its stopping criteria. From the figures, we can see that the regret and CV of Algorithm 3 are still sublinear. However, the convergence of the time-averaged CV is slower than Algorithm 2, which is predicted by our theoretical results. For the methods of OCO and ADMMBO, the reduction of the time-averaged regret and CV is much smaller compared with Algorithm 3. Compared with Figure 4c, ADMMBO produces more outliers with larger regrets than the case in Section E.1.1. On the other hand, it is surprising that its CV is better than the noiseless case in the last few iterations compared with Figure 2c. We don't know the reason since it uses a more complex utility function (expected improvement) and has no theoretical results for our metrics. For OCO, it has almost the same performance with the one shown in Figure 2, and it is achieved by using a smaller stepsize in this experiment ($\eta$ in Cao and Liu (2018)).

In Figure 5, we plot the means of the performance metrics for these three methods and put them together for comparison similar to Section E.1.1. It is much easier to see that Algorithm 3 has the best performance among these three methods, and meanwhile a slower convergence of the time-averaged CV compared with Algorithm 2.

## E.2 Data Rate Allocation Problem

In this part, we will consider the data rate allocation problem mentioned in Section 3 of the main paper. This experiment is based on the first experiment of Lee et al. (2005). Suppose that there are 4 servers with 2 classes



(a) Time-averaged regret of Algorithm 3

(b) Time-averaged CV of Algorithm 3

(c) Time-averaged regret of ADMMBO

(d) Time-averaged CV of ADMMBO

(e) Time-averaged regret of OCO

(f) Time-averaged CV of OCO

Figure 4: Boxplots of Algorithm 3, OCO and ADMMBO over 100 runs for Section E.1.2.

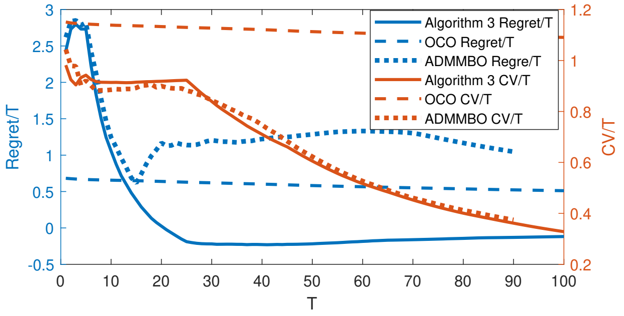


Figure 5: Comparison of Algorithm 3, OCO and ADMMBO in terms of means over 100 runs for Section E.1.2.

of utility functions:

$$U_i(x(i)) = c_i \left( \frac{1}{1+\exp(-a_i(x(i)-b_i))} + d_i \right), \text{ for } i = 1, 3 \tag{35}$$

$$U_i(x(i)) = c_i(\log(a_i x(i) + b_i) + d_i), \text{ for } i = 2, 4 \tag{36}$$

where $x(i)$ is the data rate allocated for Server $i$ from range $[0, 2]$. $a_i, b_i, c_i$ and $d_i$ are all constants, whose values are set to be the same with the first experiment of Lee et al. (2005). Each utility function can represent QoS for different classes of service Lee et al. (2005) and is observed with a measurement error sampled from $\mathcal{N}(0, 0.01)$. Meanwhile, we assume that the electricity cost for the data rates $x = [x(1), x(2), x(3), x(4)]$ is $h(x) = 1 + 0.2x(1) + 0.4x(2) + 0.8x(3) + 1.2x(4)$ and the time-averaged electricity budget is $B = 3$. Different coefficients represent the electricity pricing in different locations and $h([0, 0, 0, 0]) = 1$ represents the electricity cost for the infrastructure. For this problem $g(x) = 0.2x(1) + 0.4x(2) + 0.8x(3) + 1.2x(4) - 2$.

### E.2.1 Noiseless Constraint Observations

Still, we first consider the case where $g(x)$ is observed without noise. We apply Algorithm 2, constrained OCO with bandit feedback and ADMMBO to this problem. The hyperparameters of Algorithm 2 in this experiment are the same with Section E.1.1. The hyperparameters of OCO are set to achieve its best performance. For ADMMBO, we modify its code of a 4d example at https://github.com/SetarehAr/ADMMBO.

The boxplots of these three algorithms are shown in Figure 6 in terms of the time-averaged utility and CV, where the time horizon of Algorithm 2 and OCO is set to be 350 and the one of ADMMBO is limited to around 50 due to its stopping criteria. All these algorithms are run for 100 times. Since we do not know the optimal value of this problem, we only show the **expected utility** achieved by each method here. From the figures, we can see that the CV of Algorithm 2 is sublinear and its time-averaged values approach 0 as $T$ increases. Its expected utility also converges to a certain value. Compared with Algorithm 2, ADMMBO and OCO have higher utility values, but these values are achieved with a time-averaged CV away from 0 with a high probability. The CV of ADMMBO is more random because it is not designed for our performance metrics. For OCO, it can still achieve zero time-averaged CV with a certain probability, but most of the runs are stuck at suboptimal points due to nonconvexity of utility functions.

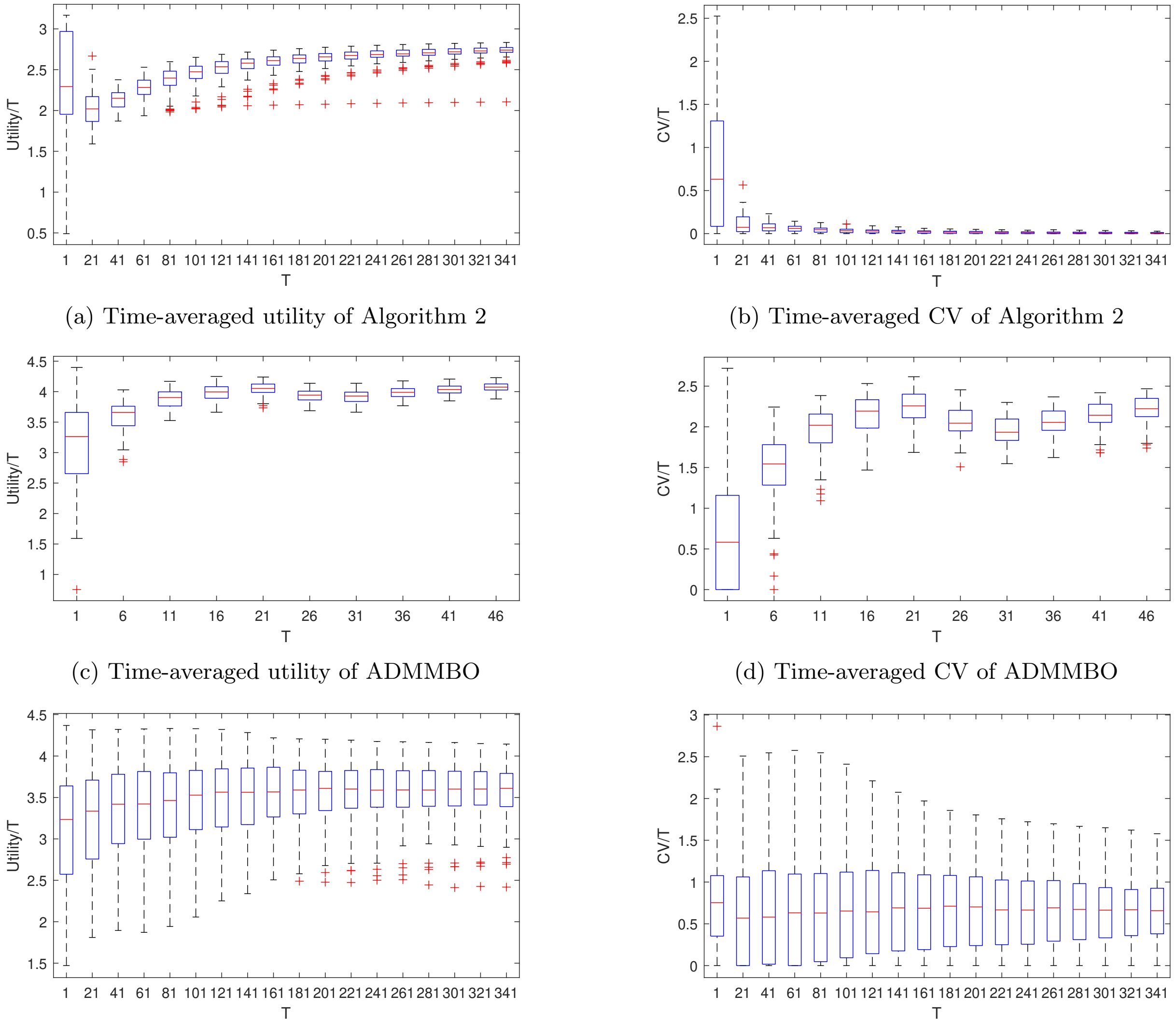
The means of their performance metrics are presented in Figure 7. We only show the first 100 iterations of Algorithm 2 and OCO for ease of comparison. This figure demonstrates the efficiency of our algorithm for satisfying the long-term constraint.

### E.2.2 Noisy Constraint Observations

In this part, $g(x)$ is also observed with random noise sampled from a uniform distribution $Unif([-0.2, 0.2])$, where the noise is due to the variation of electricity pricing. We use this problem to test Algorithm 3 in comparison with constrained OCO with bandit feedback and ADMMBO. The hyperparameters of Algorithm 3 are the same with Section E.1.2, and the ones of OCO are set to achieve the best performance.

In Figure 8, we present the boxplots of these three methods over 100 runs. Particularly, the first two figures show that the CV of Algorithm 3 is sublinear, whose time-averaged values approach 0, and that its expected utility converges to a certain value. Compared with Figure 6b, the convergence of the time-averaged CV in Algorithm 3 is slower than the one in Algorithm 2, which demonstrates our theoretical results. For the other two methods, their time-averaged CV is away from 0, which makes their higher utility values meaningless.

In Figure 9 we plot the means of the performance metrics over 100 runs for these three methods. In terms of means, only our method can let the time-averaged CV approach zero after a few iterations. Interestingly, the time-averaged CV of our method first increases and then decreases. It is because initially the multiplier is too small. After the update of the multiplier in Algorithm 3, the new value can give an appropriate penalty for constraint violations.



(a) Time-averaged utility of Algorithm 2

(b) Time-averaged CV of Algorithm 2

(c) Time-averaged utility of ADMMBO

(d) Time-averaged CV of ADMMBO

(e) Time-averaged utility of OCO

(f) Time-averaged CV of OCO

Figure 6: Boxplots of Algorithm 2, OCO and ADMMBO over 100 runs for Section E.2.1. Here ”Utility” means the value of the utility function without measurement errors.

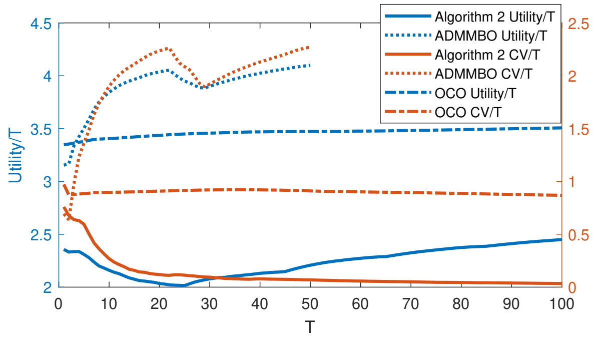
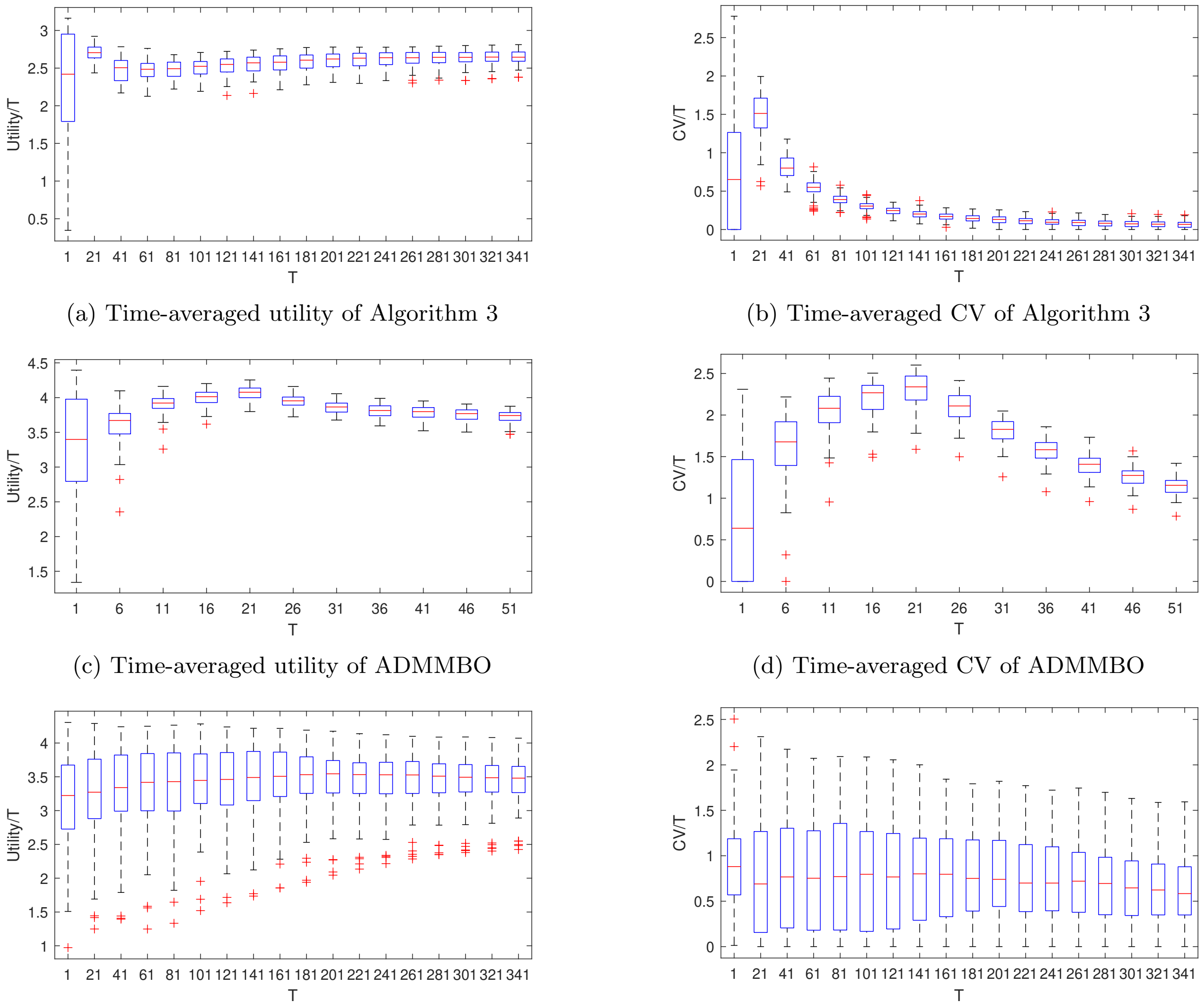


Figure 7: Comparison of Algorithm 2, OCO and ADMMBO in terms of means over 100 runs for Section E.2.1. Here ”Utility” means the values of the utility function without measurement errors.



(a) Time-averaged utility of Algorithm 3

(b) Time-averaged CV of Algorithm 3

(c) Time-averaged utility of ADMMBO

(d) Time-averaged CV of ADMMBO

(e) Time-averaged utility of OCO

(f) Time-averaged CV of OCO

Figure 8: Boxplots of Algorithm 3, OCO and ADMMBO over 100 runs for Section E.2.2. Here ”Utility” means the values of the utility function without measurement errors.

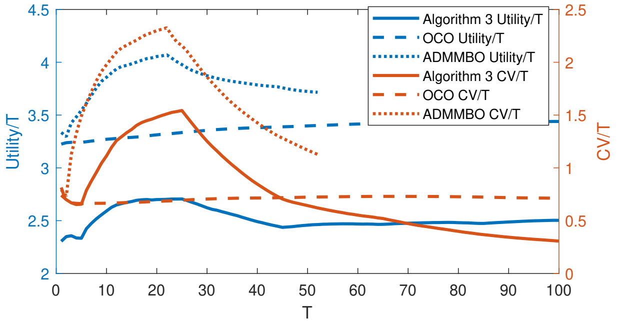


Figure 9: Comparison of Algorithm 3, OCO and ADMMBO in terms of means over 100 runs for Section E.2.2. Here ”Utility” means the values of the utility function without measurement errors.