

No-Regret Constrained Bayesian Optimization of Noisy and Expensive Hybrid Models using Differentiable Quantile Function Approximations

Congwen Lu, Joel A. Paulson*

Department of Chemical and Biomolecular Engineering, The Ohio State University, Columbus OH, USA

Abstract

This paper investigates the problem of efficient constrained global optimization of hybrid models that are a composition of a known white-box function and an expensive multi-output black-box function subject to noisy observations, which often arises in real-world science, engineering, manufacturing, and control applications. We propose a novel method, Constrained Upper Quantile Bound (CUQB), to solve such problems that directly exploits the composite structure of the objective and constraint functions that we show leads substantially improved sampling efficiency. CUQB is a conceptually simple, deterministic approach that avoids the constraint approximations used by previous methods. Although the CUQB acquisition function is not available in closed form, we propose a novel differentiable sample average approximation that enables it to be efficiently maximized. We further derive bounds on the cumulative regret and constraint violation under a non-parametric Bayesian representation of the black-box function. Since these bounds depend sublinearly on the number of iterations under some regularity assumptions, we establish explicit bounds on the convergence rate to the optimal solution of the original constrained problem. In contrast to most existing methods, CUQB further incorporates a simple infeasibility detection scheme, which we prove triggers in a finite number of iterations when the original problem is infeasible (with high probability given the Bayesian model). Numerical experiments on several test problems, including environmental model calibration and real-time optimization of a reactor system, show that CUQB significantly outperforms traditional Bayesian optimization in both constrained and unconstrained cases. Furthermore, compared to other state-of-the-art methods that exploit composite structure, CUQB achieves competitive empirical performance while also providing substantially improved theoretical guarantees.

Key words: Constrained Bayesian optimization, Hybrid modeling, Multi-output Gaussian process regression, Quantile functions

1. Introduction

In many real-world science and engineering applications, we are tasked with zeroth order derivative-free optimization (DFO) of an expensive-to-evaluate functions. For example, closed-loop performance optimization [1, 2, 3, 4], hyperparam-

*This work was supported by the National Science Foundation (NSF) under award numbers 2029282 and 2237616.

*Corresponding author

Email addresses: lu.2318@osu.edu (Congwen Lu), paulson.82@osu.edu (Joel A. Paulson)

ter tuning in machine learning algorithms [5], the choice of laboratory experiments in material and drug design [6, 7, 8], and calibration of computationally intensive simulators [9, 10, 11] can all be formulated as derivative-free optimization problems. In such applications, we can only learn about the unknown functions, which might be highly nonconvex and/or multi-modal, by sequentially querying them at specific input points. Additionally, when each function evaluation costs a large amount of energy, money, time, and/or computational resources, then we want to use as few evaluations as possible to find a globally optimal solution. For example, optimizing the hyperparameters of a deep neural network requires the weight and bias parameters to be internally optimized for every tested configuration, which can take substantial amounts of wall-clock time and/or GPU computation.

DFO methods can be broadly categorized as either stochastic or deterministic [12]. Stochastic DFO methods, such evolutionary and swarm-based algorithms [13, 14, 15], typically require a large number of high precision function evaluations and therefore cannot be applied to expensive noisy black-box functions. Deterministic DFO approaches, which are often motivated by optimization of expensive functions, are typically further subdivided into direct search and model-based methods. Direct search methods propose sample points that evaluate the function over a particular geometry such as Nelder-Mead (NM) [16] and generalized pattern search (GPS) [17]. Many other direct DFO algorithms based on box subdivision principles have been developed, which are often inspired from the DIviding RECTangle (DIRECT) algorithm [18]. Model-based DFO methods, on the other hand, construct surrogate models and propose sample points by optimizing the model. The choice of surrogate model plays a big role on the performance and convergence of the method. Parametric function approximators may not be sufficiently expressive enough to capture the complexity of the unknown function and can be inaccurate when a small number of noisy observations are available. Although some methods that can handle observation noise in the DFO setting have been developed, e.g., [19, 20], they often focus on finding a local optimum.

The common theme amongst the previously described DFO approaches is that they do not directly address the tradeoff between exploration (learning for the future) and exploitation (immediately advancing toward the goal given current knowledge), which implies they are prone to sample the unknown functions more than required. Bayesian optimization (BO) [21, 22], on the other hand, refers to a collection of data-efficient global optimization methods for expensive black-box functions subjected to noisy observations. BO’s effectiveness is derived from its probabilistic representation of the unknown functions, which allows it to characterize the expected information content that future sample points show with respect to potential global optima. As such, BO can systematically address the exploration-exploitation tradeoff by taking an information-theoretic view of the global black-box optimization. BO consists of two key steps: (i) the construction of some “surrogate model” given by a Bayesian posterior distribution equipped with uncertainty estimates and (ii) the specification of an “acquisition function” (that depends on the posterior distribution) whose value at any point is a metric for the value/utility of evaluating the unknown functions at this point.

In addition to expensive objective evaluations, many optimization problems also have expensive black-box constraint functions. For example, when designing a pharmaceutical drug, one may wish to maximize the drug’s activity while constraining its toxicity to be below acceptable limits. When deploying a machine learning algorithm on low-cost embedded hardware, one often wants to maximize performance subject to constraints on the model’s memory and power usage. Similarly, when tuning the parameters of a temperature controller in a building application, one may want to minimize closed-loop energy consumption subject to comfort constraints. The same challenge is present in each of these examples: both the performance and

constraint functions have an unknown (black-box) relationship to the design/tuning parameters. There has been a significant amount of work on extending BO to handle black-box constraints. All methods are based on the idea that separate probabilistic surrogate models for the black-box objective and constraint functions can be constructed. Thus, they mainly differ in terms of how they incorporate the constraint models into the acquisition function.

Roughly speaking, all constrained BO methods fall into one of two groups, i.e., those that do and those that do not allow constraint violation during the optimization process. The setting wherein no violations are tolerated is often referred to as “safe BO” [23, 24, 25, 26] and are motivated by online safety-critical applications (e.g., tuning an autonomous vehicle, robot, or personalized medicine control system). The enforcement of such hard constraints under uncertainty, however, implies that these algorithms may get stuck in a local (suboptimal) solution. This setting is not relevant in this work since we are interested in developing an algorithm that is guaranteed to find the true global solution. The other class of constrained BO methods (that do allow violation at each iteration) can be further divided into either *implicit* or *explicit* methods. Implicit methods encode the constraint information as a type of merit function in the original unconstrained acquisition function – in other words they do not directly enforce any constraint on the allowable violation at every iteration. Two examples of merit functions that have been proposed are the expected improvement with constraints (EIC) function [27] and the augmented Lagrangian BO (ALBO) [28]. However, to the best of our knowledge, no theoretical guarantees on convergence, optimality, or constraint violations incurred during the optimization process have been established for this class of methods.

Explicit methods, on the other hand, solve a constrained acquisition optimization problem at every iteration and, thereby, place more direct limits on the degree of violation. One of the earliest explicit constrained BO methods is [29], which simply used the mean prediction of the constraint function to construct an approximate feasible region; however, this was shown to lead to poor sample selections in the early iterations when the model is not very accurate. The upper trust bound (UTB) method was proposed in [30] to overcome this challenge by incorporating uncertainty in the constraint predictions – the key observation in this work is that better performance is achieved when one constructs a *relaxation* of the feasible region, though no theoretical results or strategies for parameter tuning were provided. This gap was addressed by the authors in [31] wherein we developed exact penalty BO (EPBO), which is an extension of UTB that uses a specific type of penalty function to establish convergence guarantees under certain regularity assumptions. EPBO requires the selection of a single penalty weight parameter ρ ; however, it was recently shown in [32] that even stronger results can be obtained when setting $\rho = \infty$. In particular, [32] derived explicit bounds for the rate of convergence to the optimal solution and the degree of constraint violation as a function of the total number of iterations. Furthermore, by setting $\rho = \infty$, one can naturally derive an infeasibility detection scheme, which is unique to [32].

The main downside of all the previously mentioned constrained BO methods is that they assume the objective and constraint functions are fully black box in nature, meaning we have little-to-no prior information about their structure. It has been shown in [33] that the black-box assumption provides a strong limit on the practical rate of convergence, which can only be overcome by utilizing prior knowledge about the structure of these functions. Luckily, in many real-world problems, we have a substantial amount of prior knowledge that can be exploited in the development of more efficient search methods. This concept forms the basis of most “grey-box” or “hybrid” optimization algorithms that have been developed within the process systems engineering community over the past several years, e.g., [34, 35, 36, 37]. Despite the prevalence of grey-box modeling/optimization, there has been limited

work on (constrained) grey-box BO due to the complexities introduced when combining known nonlinear equations with probabilistic surrogate models. Our group recently developed a method, COBALT [38], that is applicable to such problems where the objective and constraints are represented by composite functions, i.e., $f(x) = g(x, h(x))$ where h is an expensive black-box vector-valued function and g is a known function that can be cheaply evaluated. COBALT can be interpreted as an extension of EIC to handle the composite structure of the objective and constraint functions; however, it inherits several of the aforementioned challenges with EIC, which are described in detail in Section 2.3.

In this paper, we propose the constrained upper quantile bound (CUQB) as a simple and effective algorithm for efficient constrained global optimization of expensive hybrid models (grey-box composite functions) subjected to noisy observations. We can interpret CUQB as an extension of the EPBO method [31, 32] to handle general composite functions. As such, at every iteration, CUQB requires the solution of an auxiliary constrained optimization problem with the objective and constraint functions replaced by their high-probability quantile bounds. In the standard black-box setting, we can develop closed-form expressions for the quantile bounds by exploiting the properties of Gaussian random variables, which leads to the well-known upper confidence bound (UCB) acquisition function [39]. This no longer holds in the composite function setting (since a nonlinear transformation of a Gaussian random variable is no longer Gaussian), so CUQB must be evaluated via stochastic simulation. We first develop a convergent simulation-based approximation of the quantile function using a sorting operator. We then develop a simple differentiable approximation of this estimator using the soft sorting projection operator from [40], which enables efficient maximization of the quantile functions. Similarly to [32], we theoretically analyze the performance of CUQB under several conditions. Not only do we show that CUQB has at least the same cumulative regret and cumulative constraint violation bounds as the black-box case, we also establish explicit bounds on the rate of convergence to the true constrained solution under certain standard regularity assumptions. Additionally, we also incorporate an infeasibility detection scheme in CUQB that will only trigger (in a finite number of iterations) if the original problem is infeasible with high probability. To the best of our knowledge, the proposed CUQB method is the first and only constrained grey-box BO method to provide theoretical guarantees on performance and infeasibility detection. To demonstrate the effectiveness of the proposed algorithm, we compare its performance to standard BO methods and COBALT on a variety of test problems. These results confirm that CUQB consistently finds solutions that are orders of magnitude better than that found by standard BO, and similar or better quality solutions than COBALT while simultaneously providing a guarantee of either declaring infeasibility or finding the globally optimal solution.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of the constrained optimization problems of interest in this work as well as summarize our previous method, COBALT, for solving such problems. In Section 3, we present our new approach, CUQB, that can overcome some challenges of COBALT and show how it can be efficiently implemented in practice. We perform some theoretical analysis on the cumulative regret and violation probability, convergence, and infeasibility declaration in Section 4. Section 5 describes the computational setup for the numerical experiments including the practical implementation details of CUQB. We then compare the performance of CUQB to traditional BO approaches and COBALT on a variety of benchmark problems in Section 6 and present some concluding remarks in Section 7.

2. Constrained Bayesian Optimization with Composite Functions

This section first introduces the class of optimization problems considered in this work and then provides a summary of our recent approach, called COBALT [38], for tackling such problems. We end by discussing some potential challenges exhibited by COBALT that motivate our proposed work described in Section 3.

2.1. Problem formulation

In this work, we consider constrained grey-box optimization problems over composite functions of the following form¹

$$\max_{x \in \mathcal{X}} g_0(x, h(x)), \quad (1a)$$

$$\text{subject to: } g_i(x, h(x)) \geq 0, \quad \forall i \in \mathbb{N}_1^n, \quad (1b)$$

where $x \in \mathbb{R}^d$ is a set of d decision variables that are constrained to some compact set of candidate solutions $\mathcal{X} \subset \mathbb{R}^d$; $\mathbb{N}_a^b = \{a, \dots, b\}$ denotes the set of integers from a to b ; $h : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is an expensive-to-evaluate black-box function with m outputs whose structure is completely unknown and evaluations do not provide derivative information; and $g_i : \mathbb{R}^d \times \mathbb{R}^m \rightarrow \mathbb{R}$ are known, cheap-to-evaluate functions that define the objective ($i = 0$) and constraints ($i \in \mathbb{N}_1^n$) of the problem. The functions $\{g_i(x, h(x))\}_{i \in \mathbb{N}_0^n}$ are *composite* functions of known and unknown components. The known functions $\{g_i\}_{i \in \mathbb{N}_0^n}$ can be interpreted as prior knowledge that can be exploited when developing an algorithm to solve (1).

Since the function h is unknown, we must learn it from data. We focus on the so-called bandit feedback setting wherein, at every iteration t , an algorithm selects a query point $x_t \in \mathcal{X}$ for which noisy evaluations of $h(x_t)$ can be observed. That is, we receive an observation $y_t = h(x_t) + \epsilon_t$ where $\epsilon_t \sim \mathcal{N}(0, \sigma^2 I_m)$ is a zero-mean Gaussian noise term. To make progress toward an efficient sampling procedure, we need to make some “regularity” assumption for h . Here, we will focus on the case that h satisfies some underlying smoothness properties such that the sequence of data points $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^t$ can be correlated to future predictions. In particular, we assume h is a sample from a multi-output Gaussian process (GP) distribution, $\mathcal{GP}(\mu_0, K_0)$, where $\mu_0 : \mathcal{X} \rightarrow \mathbb{R}^m$ is the prior mean function and $K_0 : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{S}_{++}^m$ is the prior covariance (or kernel) function with \mathbb{S}_{++}^m denoting the set of positive definite matrices of dimension m [41]. Given the current set of t observations \mathcal{D}_t , the posterior remains a GP with closed form expressions for updated mean μ_t and covariance K_t functions [42]. As such, the posterior predictions of the unknown function $h(x)$ can be described by a multivariate normal distribution of the form

$$h(x) \mid \mathcal{D}_t \sim \mathcal{N}(\mu_t(x), \Sigma_t(x)), \quad (2)$$

where $\Sigma_t(x) = K_t(x, x)$ is the (marginal) covariance matrix at the test point x . Since (1) is capable of flexibly fusing physical knowledge with data, it is a particularly effective formulation for optimizing over “hybrid” models.

Remark 1. *There has been significant work on multi-output GPs, with the most straightforward method being to model each output as an independent GP with or without shared hyperparameters, e.g., [43]. There are two main drawbacks of such*

¹Throughout the paper, we use standard BO convention to define our original optimization problem in (1), which uses a reward (max) perspective subject to ≥ 0 inequalities. This is opposite to the standard convention in the optimization literature, which uses a cost (min) perspective subject to ≤ 0 inequalities. Note that we can always convert between these representations by introducing a minus sign in the objective and constraints.

approaches: (i) it neglects potentially important correlation between the outputs and (ii) constructing and storing a separate GP can be inefficient (from both a memory and computation point of view), especially when the number of outputs is large. As such, more recent work has focused on so-called multi-task GP (MTGP) models that use structured covariance functions of the form $k([x, i], [x', j]) = k_{\text{data}}(x, x')k_{\text{task}}(i, j)$ where x is the input to task $i \in \mathbb{N}_1^n$ and x' is the input to task $j \in \mathbb{N}_1^n$. This induces a Kronecker structure in the covariance matrix, which greatly simplifies the equations for $\mu_t(x)$ and $\Sigma_t(x)$, as shown in [44]. Our proposed algorithm applies in all of these settings, as it only requires the ability to evaluate (2).

Remark 2. The noise term ϵ_t is used to model errors associated with the measurement process used to obtain values of the unknown function $h(x)$, which is never perfect in practice. For a physical system, this noise would likely come from unmeasured fluctuations in the sensor. For a simulation, this noise could be due to stochastic fluctuations in the underlying model (e.g., random motion of particles in a molecular dynamics simulation) or other unknown sources of numerical error. It is worth noting that we assume that this noise follows a zero-mean Gaussian distribution to simplify the construction of the posterior distribution in (2). Although our theoretical results in Section 4 are based on this assumption, there are strategies that can be used to replace it with the more agnostic setting considered in [39] in which the noise only needs to be i.i.d. σ -sub Gaussian. The proposed algorithm can be straightforwardly applied in this more general setting and we plan to study the theoretical implications of more general noise settings in future work.

2.2. The COBALT algorithm

In principle, we could tackle (1) with Bayesian optimization (BO) algorithms by neglecting the composite structure and treating it as a black-box problem

$$\max_{x \in \mathcal{X}} f_0(x), \quad (3a)$$

$$\text{subject to: } f_i(x) \geq 0, \quad \forall i \in \mathbb{N}_1^n, \quad (3b)$$

where $f_i(x) = g_i(x, h(x))$ for all $i \in \mathbb{N}_0^n$. Given GP surrogate models for the unknown functions, the second major component of BO is to choose an acquisition function $\alpha_t : \mathcal{X} \rightarrow \mathbb{R}$ that depends on the posterior distributions. When properly selected, the value of $\alpha_t(x)$ at any $x \in \mathcal{X}$ should be a good measure of the potential benefit of querying the unknown function at this point in the future [45]. We can then define the BO framework as the sequential learning process that corresponds to selecting x_{t+1} as the maximizer of α_t , i.e., $x_{t+1} \in \operatorname{argmax}_{x \in \mathcal{X}} \alpha_t(x)$. The challenge is that, although we can place GP priors on $\{f_i\}_{i \in \mathbb{N}_0^n}$ to derive α_t , this is not an effective model in general for the case of composite functions. Consider the simple example $f_0(x) = h(x)^2$ – we clearly see that f_0 must be positive for all x but this is not enforced when f_0 is directly modeled as a GP. This observation was the main motivation for our previous algorithm, COBALT, that constructs an improved acquisition function given proper statistical representations of the posterior predictive distributions $\{f_i(x) \mid \mathcal{D}_t\}_{i \in \mathbb{N}_0^n}$.

COBALT can be derived using a similar thought experiment to the expected improvement (EI) acquisition function [46, 47] commonly used in the black box case. Suppose we have sampled at a set of t points, the observations are noise-free, and we are only willing return an evaluated point as our final solution. If we have no evaluations left to make, the optimal choice must correspond to the point with the largest feasible objective value. Let $f_{0,t}^*$ denote the objective value of this point

where t is the number of times we have evaluated h so far. That is,

$$f_{0,t}^* = \begin{cases} \max_{j \in S_t} f_0(x_j), & \text{if } |S_t| > 0, \\ M, & \text{otherwise.} \end{cases} \quad (4)$$

where $S_t = \{j \in \mathbb{N}_1^t : f_i(x_j) \geq 0, \forall i \in \mathbb{N}_1^n\}$ is the set of feasible observations and $M \in \mathbb{R}$ is a penalty for not having a feasible solution (typically set to be smaller than the lowest possible objective value $M \leq \min_{x \in \mathcal{X}} f_0(x)$). Now suppose that we have one additional evaluation to perform and can perform it at any point $x \in \mathcal{X}$ for which we observe $h(x)$. After this new evaluation, the value of the best point will either be $f_0(x) = g_0(x, h(x))$ or $f_{0,t}^*$. The former will occur if the point was feasible ($f_i(x) = g_i(x, h(x)) \geq 0, \forall i \in \mathbb{N}_1^n$) and it leads to an increase in the objective value $f_0(x) \geq f_{0,t}^*$. The latter will occur if the point was infeasible ($\exists i \in \mathbb{N}_1^n : f_i(x) < 0$) or there was no improvement in the objective value $f_0(x) \leq f_{0,t}^*$. The increase/improvement in the best observed point due to this new evaluation can then be compactly represented as $[f_0(x) - f_{0,t}^*]^+ \Delta(x)$ where $[a]^+ = \max\{a, 0\}$ is the positive part and $\Delta(x)$ is a feasibility indicator that is equal to 1 if $f_i(x) \geq 0, \forall i \in \mathbb{N}_1^n$ or 0 otherwise. We would like to choose x to make this improvement as large as possible; however, $h(x)$ is unknown until after the evaluation. A simple way to deal with this uncertainty is to take the expected value of the improvement, which is the basis for the COBALT acquisition function [38]

$$\text{COBALT}_t(x) = \mathbb{E}_t \{ [g_0(x, h(x)) - f_{0,t}^*]^+ \Delta(x) \}, \quad (5)$$

where $\mathbb{E}_t \{\cdot\} = \mathbb{E}_t \{\cdot \mid \mathcal{D}_t\}$ is the expectation under the posterior distribution of h given the data \mathcal{D}_t . The posterior distribution is given in (2), which is a multivariate normal with mean $\mu_t(x)$ and covariance $\Sigma_t(x)$.

2.3. Current challenges with COBALT

The COBALT acquisition function (5) is constructed to be one-step Bayes optimal under the two previously mentioned assumptions: (i) the function evaluations are noise-free and (ii) the final recommended point is restricted to a previously sampled point. However, there are some theoretical and computational challenges with COBALT that we briefly summarize below.

2.3.1. Noisy observations

In many applications, such as those involving physical experiments and/or stochastic simulations, the function evaluations are inherently noisy such that (5) is not directly applicable because $f_{0,t}^*$ is not well-defined under noisy observations. In practice, one can replace $\max_{j \in S_t} f_0(x_j)$ in (4) with its noisy counterpart defined in terms of y_t , but this results in a loss in the Bayes optimal property as well as does not properly account for uncertainty in the prediction. Another alternative is to use a “plug-in” estimate by maximizing the mean objective function subject to the mean constraint function [27], however, this has a tendency to overestimate the incumbent, leading to insufficient exploration in practice.

2.3.2. Computation and constraint handling

Another important challenge with (5) is that it does not have an analytic expression for general nonlinear functions $\{g_i\}_{i \in \mathbb{N}_0^n}$. Instead, we must resort to a Monte Carlo (MC) sampling procedure to approximate it by performing the following “whitening transformation” [48, 49] (derived from the closure of Gaussian distributions under linear transformations)

$$h(x) \mid \mathcal{D}_t = \mu_t(x) + C_t(x)Z, \quad (6)$$

where $C_t(x)$ is the lower Cholesky factor of $\Sigma_t(x) = C_t(x)C_t(x)^\top$ and $Z \sim \mathcal{N}(0, I_m)$ is an m -dimensional standard normal random vector. Using (6), the MC approximation of (5) is given by

$$\text{COBALT}_t(x) \approx \frac{1}{L} \sum_{l=1}^L \left\{ \left[g_0(x, \mu_t(x) + C_t(x)Z^{(l)}) - f_{0,t}^* \right]^+ \Delta^{(l)}(x) \right\}, \quad (7)$$

where $Z^{(1)}, \dots, Z^{(L)} \sim \mathcal{N}(0, I_m)$ are a set of L i.i.d. random samples and $\Delta^{(l)}(x)$ is the evaluation of the feasibility function for the l^{th} sample

$$\Delta^{(l)}(x) = \mathbb{I} \left(\min_{1 \leq i \leq m} g_i(x, \mu_t(x) + C_t(x)Z^{(l)}) \geq 0 \right), \quad (8)$$

for indicator function $\mathbb{I}(A)$ that is equal to 1 if logical proposition A is true and 0 otherwise. The presence of the indicator function in (7) results in a highly non-smooth function that is difficult to maximize using standard gradient-based methods. As such, we previously suggested to apply an explicit linearization technique to overcome this challenge in practice (see [38] for details). Not only can this linearization technique degrade when $\{g_i\}_{i=1}^n$ are highly nonlinear, this requires us to select so-called “trust level” parameters related to the desired satisfaction probability. In [38], we develop a heuristic procedure to design these trust levels, which performed well in practice, though their impact on performance remains an open question.

2.3.3. Theoretical guarantees

Due to the practical approximations mentioned in the previous section, no theoretical performance guarantees have been established for the current implementation of COBALT. The heuristic noise and constraint handling methods further complicate this task. It is worth noting that, in the absence of constraints, one could follow a similar approach to [50, Proposition 2] to establish convergence of the exact COBALT acquisition function, i.e., $f_{0,t}^* \rightarrow \max_{x \in \mathcal{X}} f_0(x)$ when $x_{t+1} \in \text{argmax}_{x \in \mathcal{X}} \text{COBALT}_t(x)$ as $t \rightarrow \infty$. However, this result only provides a guarantee of convergence in the limit of infinite samples and does not establish any sort of bounds on the rate of convergence for a finite t . This serves as a key source of inspiration for this work, as we would like to come up with an algorithm for which we can derive convergence rate bounds. We present such an algorithm in the next section. It also turns out that our proposed approach can straightforwardly handle noisy observations and does not require any constraint linearization.

3. Proposed Constrained Upper Quantile Bound (CUQB) Method

This section summarizes our proposed approach, called constrained upper quantile bound (CUQB), for solving (1). The idea is a generalization of the upper confidence bound (UCB) method [39, 51, 52] for solving black-box optimization problems of the form (3) when all functions $\{f_i\}_{i \in \mathbb{N}_0^n}$ are fully unknown. UCB relies on the ability to calculate explicit expressions for the “confidence bounds” for GP models that give us a range of possible observations. As we show next, we can still compute such bounds for composite functions $\{g_i(x, h(x))\}_{i \in \mathbb{N}_0^n}$ (with non-Gaussian predictions) by determining quantiles of the implied random variable.

3.1. Composite quantile functions

Let $Y_{i,t}(x)$ denote the conditional random variable corresponding to the prediction of the i^{th} function in (1) at point x given data \mathcal{D}_t for all $i \in \mathbb{N}_0^n$. Using the

transformation (6), we can express the target random variable as follows

$$Y_{i,t}(x) = g_i(x, \mu_t(x) + C_t(x)Z), \quad \forall i \in \mathbb{N}_0^n, \forall t \geq 0. \quad (9)$$

In the remainder of this section we will drop subscripts i, t for notational simplicity and refer to $Y_{i,t}(x)$ as simply $Y(x)$, which is a random variable parametrized by the test point location x . Let $F_{Y(x)}(y) = \Pr\{Y(x) \leq y\}$ denote the cumulative distribution function (CDF) of $Y(x)$. The quantile function $Q_{Y(x)} : [0, 1] \rightarrow \mathbb{R}$ then maps its input p (a probability level) to a threshold value y such that the probability of $Y(x)$ being less than or equal to x is p . Therefore, we can think of the quantile as simply the generalized inverse CDF, i.e.,

$$Q_{Y(x)}(p) = F_{Y(x)}^-(p) = \inf\{y \in \mathbb{R} : F_{Y(x)}(y) \geq p\}. \quad (10)$$

Note that the generalized inverse exists for any transformation g_i . When the function $g_i(x, y)$ is differentiable in its second argument y , the CDF will have a unique inverse though we will not require this assumption in our analysis. The quantile function can be derived in closed form for a Gaussian distribution but in general will have a more complicated representation. The main value of the quantile function in our context is that it can be immediately used to bound the range of possible predictions by properly selecting the probability levels p . For example, the following bound holds by construction for any $\alpha \in (0, 1)$

$$\Pr\{Y(x) \in [Q_{Y(x)}(\frac{\alpha}{2}), Q_{Y(x)}(1 - \frac{\alpha}{2})]\} = 1 - \alpha. \quad (11)$$

The upper quantile bound $Q_{Y(x)}(1 - \frac{\alpha}{2})$ and lower quantile bound $Q_{Y(x)}(\frac{\alpha}{2})$, respectively, represent our optimistic and pessimistic estimate of the composite function $g(x, h(x))$ given all available data. We will show later that both bounds are important: the upper bound is important for search while the lower bound is important for recommendation in the presence of noise.

3.2. The CUQB sequential learning algorithm

We are now in a position to state our proposed algorithm, which we refer to as constrained upper confidence bound (CUQB). To facilitate the description of CUQB, we formally define the upper and lower quantile bound functions next.

Definition 1 (Quantile bound functions). *Let $Y_{i,t}(x)$ refer to the random variable in (9) corresponding to the posterior prediction of the objective ($i = 0$) or a constraint ($i \in \mathbb{N}_1^n$) function given data \mathcal{D}_t . The lower and upper quantile bounds for $Y_{i,t}(x)$ are then denoted by*

$$l_{i,t}(x) = Q_{Y_{i,t}(x)}\left(\frac{\alpha_{i,t}}{2}\right), \quad (12a)$$

$$u_{i,t}(x) = Q_{Y_{i,t}(x)}\left(1 - \frac{\alpha_{i,t}}{2}\right), \quad (12b)$$

where $\{\alpha_{i,t}\}_{i \in \mathbb{N}_0^n, t \geq 0}$ is a known sequence of probability values $\alpha_{i,t} \in (0, 1)$.

A complete description of the proposed CUQB method is provided in Algorithm 1 in terms of these quantile bounds. This algorithm is conceptually simple and only requires one to solve a single auxiliary problem with the grey-box objective and constraint functions replaced by their upper quantile bound functions (Line 5). Since we have assumed that $h(x)$ is expensive to query, the cost of solving this auxiliary problem should be much smaller than the original problem (1), even though it is generally nonconvex. A practical approach to solving this optimization

Algorithm 1 The Constrained Upper Quantile Bound (CUQB) Grey-Box Optimization Algorithm for Noisy and Expensive Functions

Input: Compact domain \mathcal{X} ; GP prior mean function $\mu_0(x)$ and covariance function $K_0(x, x')$; probability values $\{\alpha_{i,t}\}_{i \in \mathbb{N}_0^n, t \geq 0}$; recommendation penalty parameter $\rho > 0$; and total number of iterations T .

- 1: **for** $t = 0$ to $T - 1$ **do**
- 2: **if** $\exists i \in \mathbb{N}_1^n$ such that $\max_{x \in \mathcal{X}} u_{i,t}(x) < 0$ **then**
- 3: Declare that the original problem (1) is infeasible and stop.
- 4: **end if**
- 5: Solve the following auxiliary constrained optimization problem to determine the next best sample point x_{t+1} given our current posterior distribution

$$x_{t+1} \in \operatorname{argmax}_{x \in \mathcal{X}} \{u_{0,t}(x) \text{ s.t. } u_{i,t}(x) \geq 0, \forall i \in \mathbb{N}_1^n\}, \quad (13)$$

- 6: Query unknown function h at x_{t+1} to get noisy observation y_{t+1} .
- 7: Update data $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}$.
- 8: Update multi-output GP model for h in (2) with new data \mathcal{D}_{t+1} .
- 9: **end for**
- 10: Return the point x_{t^*} with the largest penalized lower quantile bound as our best guess for the optimal solution

$$t^* = \operatorname{argmax}_{t \in \{1, \dots, T\}} \left\{ l_{0,t-1}(x_t) - \rho \sum_{i=1}^n [-l_{i,t-1}(x_t)]^+ \right\}. \quad (14)$$

problem is discussed in the next section.²

There are two other features of Algorithm 1 that deserve further elaboration. First, in Lines 1-4, we have included an infeasibility detection scheme. Due to the consideration of potentially unbounded observation noise and finite quantile bounds, it is possible that infeasibility is declared even when the original problem (1) is feasible. Luckily, we can control the probability of a false declaration through a proper selection scheme for $\{\alpha_{i,t}\}_{i \in \mathbb{N}_0^n, t \geq 0}$, meaning we can ensure this is triggered only when the actual problem is infeasible with high probability, which is discussed further in Section 4. Second, in Line 10, we have incorporated a recommendation procedure for returning the best point in the sequence $\{x_1, \dots, x_T\}$. Although there are many potential choices (e.g., simply return the final point x_T), this approach is particularly robust to the noise in the observations and allows for convergence rate bounds to be established, which is also discussed in detail in Section 4.

Figure 1 provides a comparison of the classical EI and UCB acquisition functions to COBALT and UQB when there are no constraints, $h(x) = \sin(x) + \cos(x)$ is scalar valued, $g(x, h(x)) = h(x)^2$, and we have evaluated $h(x)$ at five points. The left-hand column shows the hybrid (grey-box) modeling case wherein we compute a GP model for $h(x)$ and propagate the uncertainty through the known function $g(x, y)$ to compute an implied posterior for $f(x) = g(x, h(x))$. The right-hand column shows

²Throughout the manuscript, we refer to ‘‘CUQB’’ as the conceptual method in Algorithm 1 that requires the optimization problem (13) to be solved to global optimality at every iteration. In practice, it can be difficult to guarantee that one has found a global maximizer and so one must often resort to a heuristic global optimization procedure (e.g., multi-start of a local gradient-based optimization algorithm) in practice. Although the theoretical analysis does not hold for the practical implementation of CUQB, our extensive numerical results show that strong performance results can still be achieved in practice under such approximations.

the standard black-box modeling case wherein we directly compute a GP model for $f(x)$. The first main observation is that the black-box GP posterior has much larger 95% confidence intervals than the grey-box posterior including predicting the potential for negative values. Both COBALT and UQB are able to take advantage of this valuable information to more quickly hone in on good sampling regions. Note that, in addition to more naturally handling noise than COBALT, UQB does not have flat regions, which makes it a bit easier to practically optimize.³

Figure 2 demonstrates the value of CUQB from a constraint handling point of view by comparing its predicted feasible region (shown in red) to that of COBALT and EPBO (black-box version CUQB) for the following two-dimensional constraint

$$\{x = (x_1, x_2) \in [-4, 4]^2 : g(h(x_1, x_2)) \leq 100\}, \quad (15)$$

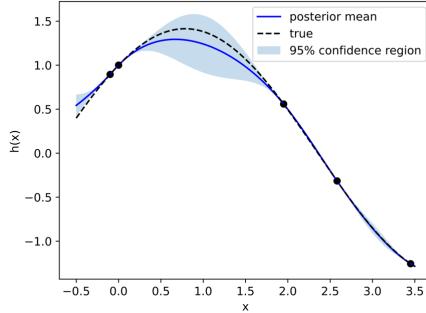
where $h(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$ is the unknown function and $g(h(x_1, x_2)) = h(x_1, x_2)^2$ is the known function. The key observation is that the feasible region for COBALT is not guaranteed to contain the true feasible region (whose boundary is shown with white lines). The feasible region for EPBO does contain the true feasible region, however, due to the black-box representation of the functions, it has a significantly larger region than CUQB as a consequence of its higher degree of prediction uncertainty. CUQB thus achieves the best of both worlds: guaranteed containment of the true feasible region (with high probability) and reduced levels of uncertainty by exploiting prior structural knowledge.

3.3. Practical optimization of CUQB using differentiable sorting

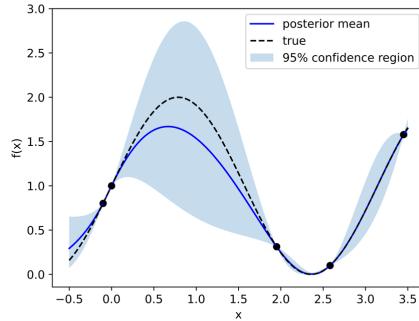
The subproblem (13) is difficult to solve since we do not even have an explicit expression for the quantile functions $\{u_{i,t}\}_{i \in \mathbb{N}_0^n}$, meaning we cannot easily evaluate them for any specific $x \in \mathcal{X}$. This presents a huge barrier when attempting to apply guaranteed global optimization methods, such as spatial branch and bound [53], that typically exploit structure of the function to compute upper and lower bounds in different domains of the search space. Instead, we resort to a more practical (heuristic) global optimization procedure. Multi-start methods that randomly sample initial conditions used to initialize a local gradient-based optimization algorithm are popular alternatives since they have been empirically shown to provide a good tradeoff between performance and computational cost. However, it turns out that multi-start is still challenging to apply in this context due to lack of a closed-form expression for the quantile functions. As such, we need to introduce two additional approximations that will allow us to build easy-to-differentiate representations of $\{u_{i,t}\}_{i \in \mathbb{N}_0^n}$. In this section, we show that these approximations are convergent in the sense that they can be made to exactly match the true quantile functions as two independent parameters are refined.

First, we must resort to some MC sampling approximation similar to that used for COBALT in (7) since the distribution of $Y_{i,t}(x)$ is not known in closed-form. Estimating quantiles of a target response, however, is fundamentally different than estimating its mean (expected value), as it provides additional information about the distribution of $Y_{i,t}(x)$. The most straightforward approach for estimating quantiles is based on the order statistics (sorted values) of a set of i.i.d. samples of $Y_{i,t}(x)$. Consider a set of L i.i.d. samples from (9), i.e., $Y_{i,t}^{(l)}(x) = g_i(x, \mu_t(x) + C_t(x)Z^{(l)})$ for all $l \in \mathbb{N}_1^L$ where $Z^{(1)}, \dots, Z^{(L)} \sim \mathcal{N}(0, I_m)$. Let $\text{Sort} : \mathbb{R}^L \rightarrow \mathbb{R}^L$ denote an

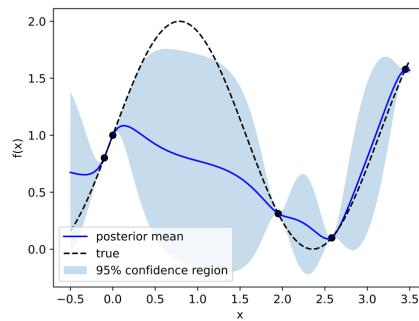
³We developed a heuristic scaling procedure in [38] to overcome the challenges with the flat regions of COBALT that have gradient equal to zero.



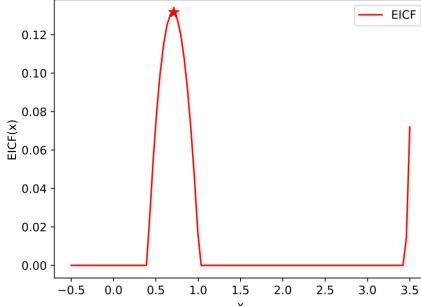
(a) Posterior for $h(x)$ used by hybrid models



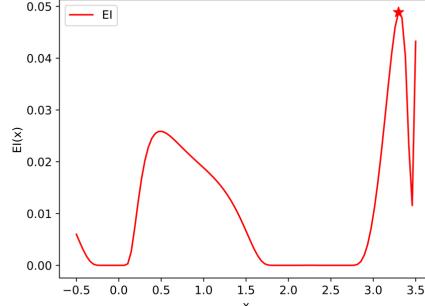
(b) Implied posterior for $f_0(x)$



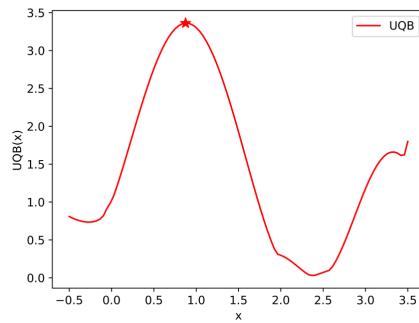
(c) Posterior for $f_0(x)$ using black-box model



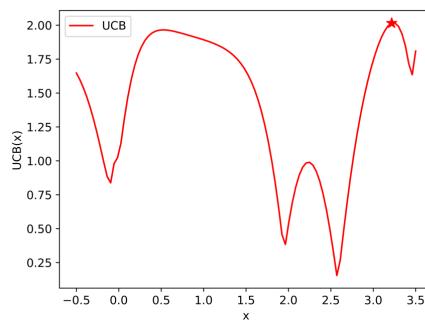
(d) COBALT acquisition function



(e) Classical EI acquisition function



(f) Proposed UQB acquisition function



(g) Classical UCB acquisition function

Figure 1: Illustrative example comparing classical EI and UCB acquisition functions to their composite (grey-box) counterparts COBALT and UQB for a problem with scalar $h(x)$ and $f(x) = g(x, h(x)) = h(x)^2$. The observations of $h(x)$ provide more useful information about the location of potential global maxima, so that COBALT and UQB choose to evaluate points much closer to the global maxima.

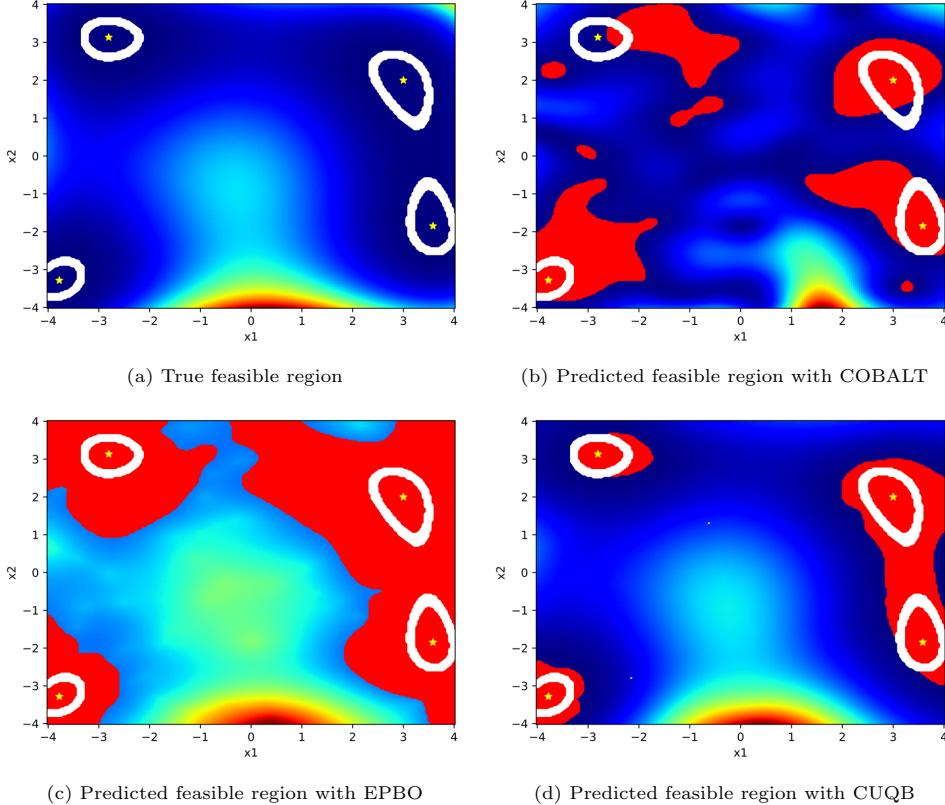


Figure 2: Illustrative example comparing the constraint handling approaches in COBALT and EPBO to the proposed CUQB method. The filled red area corresponds to the predicted feasible region while the white lines represent the boundary of the true feasible region. We see that CUQB is able to more accurately contain the true feasible region than the alternative methods.

operator that sorts these samples in ascending order

$$Y_{i,t}^{(1,L)}(x), \dots, Y_{i,t}^{(L,L)}(x) = \text{Sort}(Y_{i,t}^{(1)}(x), \dots, Y_{i,t}^{(n)}(x)), \quad (16)$$

such that

$$Y_{i,t}^{(1,L)}(x) \leq Y_{i,t}^{(2,L)}(x) \leq \dots \leq Y_{i,t}^{(L,L)}(x). \quad (17)$$

We can then define the estimated (or empirical) quantile as

$$\hat{Q}_{Y_{i,t}(x)}(p) = Y_{i,t}^{(\lceil pL \rceil, L)}(x), \quad (18)$$

where $\lceil a \rceil$ denotes the ceil operator (smallest integer greater than or equal to a). Therefore, the order p approximated quantile in (18) corresponds to the $\lceil pL \rceil^{\text{th}}$ sample in the sorted list (17).

One challenge remains when replacing the true UQB functions $\{u_{i,t}(x)\}_{i \in \mathbb{N}_0^n}$ with their empirical counterparts $\{\hat{u}_{i,t}^{(L)}(x) = \hat{Q}_{Y_{i,t}(x)}(1 - \frac{\alpha_{i,t}}{2})\}_{i \in \mathbb{N}_0^n}$; the sorting operator in (16) is not differentiable everywhere such that it cannot be directly incorporated into gradient-based solvers. To better see this, we note that the sort operator can be formulated as a linear program (LP), as shown in [40]. To keep the notation consistent, we introduce $s(\boldsymbol{\theta})$ as the sort operator in *descending order* for any sequence $\boldsymbol{\theta} = (\theta_1, \dots, \theta_L) \in \mathbb{R}^L$. This can be related to the required ascending

sort operator in (16) via $\text{Sort}(\boldsymbol{\theta}) = -s(-\boldsymbol{\theta})$. The LP formulation of $s(\boldsymbol{\theta})$ is

$$s(\boldsymbol{\theta}) = \underset{\mathbf{y} \in \mathcal{P}(\boldsymbol{\theta})}{\text{argmax}} \mathbf{y}^\top \boldsymbol{\rho}, \quad (19)$$

where $\mathcal{P}(\boldsymbol{\theta})$ is the permutohedron of $\boldsymbol{\theta}$ (i.e., a convex polytope whose vertices correspond to permutations of $\boldsymbol{\theta}$) [54] and $\boldsymbol{\rho} = (L, L-1, \dots, 1)$ is a vector of descending integers from L to 1. The main idea for making a differentiable version of (19) is to introduce a quadratic regularization term leading to the following soft sort operator

$$s_\varepsilon(\boldsymbol{\theta}) = \underset{\mathbf{y} \in \mathcal{P}(\boldsymbol{\theta})}{\text{argmax}} \mathbf{y}^\top \boldsymbol{\rho} - \frac{\varepsilon}{2} \|\mathbf{y}\|^2 = \underset{\mathbf{y} \in \mathcal{P}(\boldsymbol{\theta})}{\text{argmax}} \frac{1}{2} \left\| \mathbf{y} - \frac{\boldsymbol{\rho}}{\varepsilon} \right\|^2, \quad (20)$$

which corresponds to the Euclidean projection of $\boldsymbol{\rho}/\varepsilon$ onto $\mathcal{P}(\boldsymbol{\theta})$ for a given regularization strength $\varepsilon > 0$. By replacing the exact sort s with the soft sort s_ε , we can efficiently compute derivatives of $\hat{u}_{i,t}^{(L)}(x)$ with respect to x , which enables (13) to be solved relatively quickly using local gradient-based methods. We summarize the convergence properties of $\hat{u}_{i,t}^{(L)}(x)$ in the following proposition.

Proposition 1. *The empirical quantile function $\hat{u}_{i,t}^{(L)}(x)$ in (18) converges in probability to the true quantile function $u_{i,t}(x)$ for all $x \in \mathcal{X}$ as $L \rightarrow \infty$ and $\varepsilon \rightarrow 0$.*

The proof of this result is given in Appendix A. We investigate the impact of these approximations in Section 6; empirically we found that the approximations are quite accurate for L values ~ 50 and ε values ~ 0.1 . Since the soft sort operator only requires $O(L \log L)$ complexity for the forward pass and $O(L)$ complexity for the backward differentiation pass, the cost of a single local optimization algorithm remains quite low in practice.

3.4. Simplification for linear transformations

For general nonlinear transformation functions $\{g_i\}_{i \in \mathbb{N}_0^n}$, we need to use the empirical quantile function (18) described in the previous section. However, whenever these functions are linear with respect to their second argument, we can develop closed-form solutions similar to the classical UCB and its constrained extensions. We summarize this result in the following proposition.

Proposition 2. *Suppose that $g_i(x, y) = a_i^\top(x)y + b_i(x)$ for some known functions $a_i : \mathbb{R}^d \rightarrow \mathbb{R}^m$ and $b_i : \mathbb{R}^d \rightarrow \mathbb{R}$ for all $i \in \mathbb{N}_0^n$. Then, the lower and upper quantile bound functions for $Y_{i,t}(x)$ in (12) have the following closed-form representation*

$$l_{i,t}(x) = \mu_{i,t}^Y(x) - \beta_{i,t}^{1/2} \sigma_{i,t}^Y(x), \quad (21a)$$

$$u_{i,t}(x) = \mu_{i,t}^Y(x) + \beta_{i,t}^{1/2} \sigma_{i,t}^Y(x), \quad (21b)$$

where $\mu_{i,t}^Y(x) = a_i^\top(x)\mu_t(x) + b_i(x)$, $\sigma_{i,t}^Y(x) = \sqrt{a_i^\top(x)\Sigma_t(x)a_i(x)}$, $\beta_{i,t}^{1/2} = \Phi^{-1}(1 - \frac{\alpha_{i,t}}{2})$, and Φ^{-1} is the inverse standard normal cumulative distribution function.

The proof of Proposition 2 is given in Appendix A. Proposition 2 does not imply that our approach is equivalent to constrained UCB methods, even when $\{g_i\}_{i \in \mathbb{N}_0^n}$ are linear. This is because the posterior distribution of the function $a^\top(x)h(x) + b(x)$ given observations of $h(x)$ is different from the one given only lumped observations of $a^\top(x)h(x) + b(x)$. In fact, it has been shown that the former approach necessarily leads to a reduction in variance compared to the latter approach [55, Theorem 1], such that this allows us to determine tighter confidence bounds enabling improved search performance.

4. Theoretical Analysis of CUQB

In this section, we analyze the theoretical performance of CUQB (Algorithm 1). Our goal is to establish bounds on two important performance metrics: cumulative regret and cumulative constraint violation. We then derive bounds on the cumulative performance metrics as a function of the total number of iterations T , which allow us to establish bounds on the rate of convergence to the optimal solution. Finally, we show that, with high probability (at least $1 - \delta$ where $\delta \in (0, 1)$ is a parameter chosen by the user), CUQB will declare infeasibility in a finite number of iterations if the original problem is infeasible. We note that the results presented in this section hold under the assumption that we identify a global maximizer of (13) at every iteration. In Section 6, we empirically show that strong performance results can still be achieved using effective heuristic methods to solve this problem.

Note that the proofs of all results in this section are provided in Appendix B to streamline the presentation.

4.1. Performance metrics

As in the standard bandit feedback setting, we want to minimize the gap between our queried point $f_0(x_{t+1})$ and the true solution $f_0^* = \max_{x \in \mathcal{X}, f_i(x) \geq 0, \forall i \in \mathbb{N}_1^n} f_0(x)$, which is often referred to as the *instantaneous regret*

$$r_{t+1} = f_0^* - f_0(x_{t+1}), \quad (22)$$

where x_{t+1} is the point queried by Algorithm 1 at step $t \geq 0$. In the presence of unknown constraints, we may sample infeasible points that have a larger value than f_0^* such that we further define the positive instantaneous regret as follows

$$r_{t+1}^+ = [f_0^* - f_0(x_{t+1})]^+ = \max\{f_0^* - f_0(x_{t+1}), 0\}. \quad (23)$$

Since we are also interested in finding points that satisfy the unknown constraints, we define the violation of constraint $i \in \mathbb{N}_1^n$ at step $t \geq 0$ as

$$v_{i,t+1} = [-f_i(x_{t+1})]^+ = \max\{-f_i(x_{t+1}), 0\} = -\min\{f_i(x_{t+1}), 0\}. \quad (24)$$

Ideally, we would achieve zero regret and constraint violation in a single step; however, this is only possible when $h(x)$ is perfectly known. Instead, we will focus on analyzing their cumulative values as the algorithm progresses.

Definition 2 (Cumulative regret). *Given a total of T steps of CUQB, we define the cumulative regret and cumulative positive regret over T to be*

$$R_T = \sum_{t=0}^{T-1} r_{t+1}, \quad (25a)$$

$$R_T^+ = \sum_{t=0}^{T-1} r_{t+1}^+. \quad (25b)$$

Definition 3 (Cumulative constraint violation). *Given a total of T steps of CUQB, we define the cumulative constraint violation of constraint i over T to be*

$$V_{i,T} = \sum_{t=0}^T v_{i,t+1}, \quad \forall i \in \mathbb{N}_1^n. \quad (26)$$

4.2. Required assumptions

To derive bounds on R_T and $V_{i,T}$ based on the sequence of samples selected by CUQB, we must first make some assumptions on our problem setup.

Assumption 1. *The unknown function $h(x)$ is a sample of a GP with mean function $\mu_0(x)$ and covariance function $K_0(x, x')$ and its observations are corrupted by zero mean Gaussian noise. Without loss of generality, we will assume that the prior mean function is zero, i.e., $\mu_0(x) = 0$.*

Assumption 1 is required to ensure that the data collected on h can be used to construct a reasonable surrogate model. As noted previously, the choice of the kernel function is very flexible, which enables such multi-output GP models to capture a large class of functions, as long as they satisfy some degree of smoothness/continuity. We will establish convergence for the most popular kernel choices later.

Assumption 2. *The feasible set \mathcal{X} has finite cardinality $|\mathcal{X}| < \infty$.*

Assumption 2 is made to simplify the construction of high-probability bounds on the predictions of the objective and constraint functions. This is a common assumption in the BO literature, as discussed in [39]. There has been work on relaxing this assumption in the case of black-box functions, e.g., [51, 52, 56], though how to effectively extend these bounds to the case of composite functions remains an open question that we plan to address in our future work. Although this may seem like a very restrictive assumption, it is only needed to simplify the theoretical analysis and thus does not prevent CUQB from being practically applied to continuous sets. Furthermore, we will show that the cumulative regret and constraint violation bounds exhibit a logarithmic dependence on the cardinality $|\mathcal{X}|$ such that we do not expect substantial performance losses in the case of continuous \mathcal{X} .

Assumption 3. *The optimization problem (1) is feasible and has an optimal solution x^* with corresponding optimal objective value $f_0^* = f_0(x^*)$.*

We initially make Assumption 3 to focus on the feasible case first. The behavior of CUQB when this assumption is relaxed is studied in Section 4.5.

Assumption 4. *The functions $\{g_i(x, y)\}_{i \in \mathbb{N}_0^n}$ are Lipschitz continuous with respect to their second argument, i.e.,*

$$|g_i(x, y) - g_i(x, z)| \leq L_i \|y - z\|_1, \quad \forall x \in \mathbb{R}^d, \quad \forall y, z \in \mathbb{R}^m, \quad \forall i \in \mathbb{N}_0^n, \quad (27)$$

where $L_i < \infty$ is a finite Lipschitz constant.

Assumption 4 ensures that the outer transformations of the unknown function $h(x)$ are well-behaved in the sense that the distance between the outputs are bounded some constant times the distance between the inputs.

4.3. Cumulative regret and violation probability bounds

Our derived bounds depend on the *maximum information gain* (MIG), which is a fundamental quantity in Bayesian experimental design [57] that provides a measure of informativeness of any finite set of sampling points.

Definition 4 (Maximum information gain). *Let $\mathcal{A} \subset \mathcal{X}$ denote any potential subset of points sampled from \mathcal{X} . The maximum information gain for the j^{th} element of h given t noisy measurements is*

$$\gamma_{j,t} = \max_{\mathcal{A} \subset \mathcal{X}: |\mathcal{A}|=t} \frac{1}{2} \log \det (I + \sigma^{-2} \mathbf{K}_{j,\mathcal{A}}), \quad (28)$$

where $\mathbf{K}_{j,\mathcal{A}} = [k_{j,0}(x, x')]_{x, x' \in \mathcal{A}}$ where $k_{j,0}$ is the j^{th} diagonal element of K_0 .

The first key step is to show that the lower and upper quantile bounds can be guaranteed to bound the true function with high probability for proper selection of the quantile parameters $\{\alpha_{i,t}\}$.

Lemma 1. *Let Assumptions 1 and 2 hold. Then,*

$$\Pr \{g_i(x, h(x)) \in [l_{i,t}(x), u_{i,t}(x)], \quad \forall x \in \mathcal{X}, \forall t \geq 0, \forall i \in \mathbb{N}_0^n \mid \mathcal{D}_t\} \geq 1 - \delta, \quad (29)$$

where $l_{i,t}(x)$ and $u_{i,t}(x)$ are given by (12) with $\alpha_{i,t} = 6\delta/(\pi^2(t+1)^2(n+1)|\mathcal{X}|)$.

Next, we can use Lemma 1 to derive bounds on the instantaneous regret (22) and constraint violation (24) terms that are related to the width between the upper and lower quantiles for the given sample point x_{t+1} .

Lemma 2. *Let Assumptions 1, 2, and 3 hold. Then, with probability at least $1 - \delta$, feasibility is never declared in Line 3 of Algorithm 1 and for all $t \geq 0$*

$$r_{t+1} \leq r_{t+1}^+ \leq w_{0,t}(x_{t+1}), \quad (30a)$$

$$v_{i,t+1} \leq w_{i,t}(x_{t+1}), \quad \forall i \in \mathbb{N}_1^n, \quad (30b)$$

where $w_{i,t}(x_{t+1}) = u_{i,t}(x_{t+1}) - l_{i,t}(x_{t+1})$ is the width of the $\alpha_{i,t}$ -level quantile bound evaluated at the sample point x_{t+1} .

Lemma 2 shows that both the regret and violation must be less than the quantile width $w_{i,t}(x_{t+1})$ at every $t \geq 0$ of Algorithm 1. However, it is not obvious if $w_{i,t}(x_{t+1})$ is decaying at every iteration, which would imply that we are effectively “closing in” on the optimal point. Therefore, we next establish a relationship between $w_{i,t}(x_{t+1})$ and the standard deviations of the underlying GP predictions, which will be easier to show must decay as iterations increase.

Lemma 3. *Let $Y = g(X)$ be a nonlinear transformation of a normal random vector $X \sim \mathcal{N}(\mu_X, \Sigma_X)$ where $g : \mathbb{R}^m \rightarrow \mathbb{R}$ is Lipschitz continuous with constant L_g . Then,*

$$Q_Y(1 - \frac{\alpha}{2}) - Q_Y(\frac{\alpha}{2}) \leq 2L_g\beta^{1/2} \sum_{j=1}^m \sigma_{X,j}, \quad (31)$$

where $\sigma_{X,j}$ is the j^{th} diagonal element of Σ_X , $\alpha \in (0, 1)$, and $\beta = 2\log(2m/\alpha)$.

Now that we have a connection between quantiles of a transformed random variable and the standard deviation of the underlying Gaussian random vector in Lemma 3, it is useful to establish a bound on the cumulative sum of the posterior standard deviations in terms of the MIG.

Lemma 4 (Lemma 4, [51]). *Let x_1, \dots, x_T be a sequence of query points selected by an algorithm. Then, the following scaling laws hold for common choices of kernels*

$$\sum_{t=0}^{T-1} \sigma_{j,t}(x_{t+1}) \leq 2\sqrt{(T+2)\gamma_{j,T}}, \quad \forall j \in \mathbb{N}_1^n. \quad (32)$$

The relationship in Lemma 4 is important because there are known iteration-dependent bounds on the MIG that have been established for the commonly used kernels. These bounds were derived in [39], which we repeat below for completeness.

Lemma 5 (Theorem 5, [39]). *Let $\mathcal{X} \subset \Omega$ be a subset of a compact and convex set Ω , $d \in \mathbb{N}$, and assume $k_{j,0}(x, x') \leq 1$. Then,*

- *Linear:* $\gamma_{j,T} = \mathcal{O}(d \log T)$;
- *Squared exponential:* $\gamma_{j,T} = \mathcal{O}((\log T)^{d+1})$;
- *Matern ($\nu > 1$):* $\gamma_{j,T} = \mathcal{O}(T^{d(d+1)/(2\nu+d(d+1))}(\log T))$.

We are now in a position to state our main theorem, which establishes explicit bounds on the cumulative regret and constraint violation as a function of the total number of iterations T .

Theorem 1. *Under Assumptions 1, 2, 3, and 4, we have, with probability at least $1 - \delta$, that the sample points of CUQB (Algorithm 1) satisfy*

$$R_T \leq R_T^+ \leq 4L_0\beta_T^{1/2}\Psi_T = \tilde{\mathcal{O}}(\sqrt{\gamma_T^{\max}T}), \quad (33a)$$

$$V_{i,T} \leq 4L_i\beta_T^{1/2}\Psi_T = \tilde{\mathcal{O}}(\sqrt{\gamma_T^{\max}T}), \quad \forall i \in \mathbb{N}_1^n, \quad (33b)$$

where $\Psi_T = \sum_{j=1}^m \sqrt{(T+2)\gamma_{j,T}}$, $\beta_t = 2\log(m\pi^2t^2(n+1)|\mathcal{X}|/(3\delta))$, and $\gamma_T^{\max} = \max_{j \in \mathbb{N}_1^m} \gamma_{j,T}$ is the worst-case (slowest converging) MIG for unknown function h . Here, the notation $\tilde{\mathcal{O}}$ refers to a variation of the standard order of magnitude that suppresses logarithmic factors.

We can combine the results shown in Theorem 1 with Lemma 5 to derive kernel specific bounds on the growth of the cumulative regret and constraint violation as the number of iterations T increases. Intuitively, one would expect that a *sublinear* growth in T would allow for convergence since this implies the regret/violation gap must be shrinking, i.e., $R_T/T \rightarrow 0$ and $V_{i,T}/T \rightarrow 0$ as $T \rightarrow \infty$. We more formally study the impact of Theorem 1 on convergence rate with and without noise next.

4.4. Convergence rate bounds

With slight abuse of notation, we let $r(x_t) = f_0^* - f_0(x_t)$ and $v_i(x_t)$ denote the instantaneous regret and constraint violation as a function of x_t (as opposed to being indexed by t). We can then establish the following bounds on the rate of convergence of CUQB to the optimal solution.

Theorem 2. *Let Assumptions 1, 2, 3, and 4 hold and $\mathcal{L} = \sum_{i=0}^n L_i$. Then, with probability at least $1 - \delta$, there must exist a point $\tilde{x}_T \in \{x_1, \dots, x_T\}$ contained in the sequence generated by CUQB such that the following inequalities hold*

$$f_0^* - f_0(\tilde{x}_T) \leq \frac{4\mathcal{L}\beta_T^{1/2}\Psi_T}{T} = \tilde{\mathcal{O}}\left(\sqrt{\frac{\gamma_T^{\max}}{T}}\right), \quad (34a)$$

$$[-f_i(\tilde{x}_T)]^+ \leq \frac{4\mathcal{L}\beta_T^{1/2}\Psi_T}{T} = \tilde{\mathcal{O}}\left(\sqrt{\frac{\gamma_T^{\max}}{T}}\right), \quad \forall i \in \mathbb{N}_1^n. \quad (34b)$$

Theorem 2 provides explicit convergence rate bounds to the constrained optimal solution in terms of the MIG for general kernels. However, one important point is that the result holds for $\tilde{x}_T = x_{t^*}$ with $t^* = \operatorname{argmin}_{t \in \{1, \dots, T\}} (r_{t+1}^+ + \sum_{i=1}^n v_{i,t+1})$. In practice, we cannot identify t^* since it depends on f_0^* , which is unknown. To address this challenge, we proposed a practical recommendation procedure in (14). This procedure is derived from the fact, in the absence of noise, we can identify the index that minimizes a penalized version of this sequence, i.e.,

$$\operatorname{argmin}_{t \in \{1, \dots, T\}} \left(r_t + \rho \sum_{i=1}^n v_{i,t} \right) = \operatorname{argmax}_{t \in \{1, \dots, T\}} \left(f_0(x_t) - \rho \sum_{i=1}^n [-f_i(x_t)]^+ \right), \quad (35)$$

for any penalty factor $\rho > 0$ since $r_t = f_0^* - f_0(x_t)$, so f_0^* appears as a constant that will not change the location of the minimum. The result in (35) can only be used when observations of h are noise-free since, otherwise, we do not have direct access to $\{f_i(x_t)\}_{i \in \mathbb{N}_0^n}$. Therefore, we would like to understand the impact of this recommendation procedure on convergence, which we do in the following theorem.

Theorem 3. *Let Assumptions 1, 2, 3, and 4 hold and $\bar{\rho}$ be large enough such that $f_0(x) - \rho \sum_{i=1}^n [-f_i(x)]^+$ is an exact penalty function for (1) for any $\rho \geq \bar{\rho}$. Then, the recommended point (14), \tilde{x}_T^r , by CUBQ (Algorithm 1) must satisfy*

$$0 \leq f_0^* - f_0(\tilde{x}_T^r) + \rho \sum_{i=1}^n [-f_i(\tilde{x}_T^r)]^+ \leq \frac{4\mathcal{L}_r(\rho)\beta_T^{1/2}\Psi_T}{T}, \quad \forall \rho \in [\bar{\rho}, \infty), \quad (36)$$

with probability at least $1 - \delta$, where $\mathcal{L}_r(\rho) = L_0 + \rho \sum_{i=1}^n L_i$.

It is interesting to note that the established bounds in Theorems 2 and 3 have a similar structure, with the main difference being the constant out front (\mathcal{L} versus $\mathcal{L}_r(\rho)$). Theorem 2, however, provides a stronger result in the sense that both regret and constraint violation can be individually bounded by a smaller constant. The main value of Theorem 3 is that it allows us to establish convergence of our recommended point, i.e., $\tilde{x}_T^r \rightarrow x^*$ as long as the bound in (36) decays to 0 as $T \rightarrow \infty$. From Lemma 5, we see that this holds for the three most popular kernel types including linear, squared exponential, and Matern with $\nu > 1$. The convergence rates of CUQB for each of these cases are summarized in Table 1 assuming that the elements of h are modeled independently with the same choice of kernel.

Table 1: Convergence rate for specific covariance kernel choices for the proposed CUQB method.

| Kernel | Convergence rate |
|---------------------|---|
| Linear | $\tilde{\mathcal{O}}(T^{-1/2})$ |
| Squared exponential | $\tilde{\mathcal{O}}(T^{-1/2} \log(T)^{(d+1)/2})$ |
| Matern, $\nu > 1$ | $\tilde{\mathcal{O}}(T^{-\nu/(2\nu+d(d+1))} \log(T)^{1/2})$ |

Theorem 3 holds for any large enough $\rho \geq \bar{\rho}$ but does not discuss how to practically compute a ρ value satisfying this requirement. It turns out that we can use the quantile bounds established in Lemma 1 to compute such a ρ , which we summarize in the following corollary.

Corollary 1. *Let $\rho \geq 0$ be any scalar value that satisfies the following inequality*

$$\max_{x \in \mathcal{X}} \{u_{0,0}(x) - \rho \sum_{i=1}^n [-u_{i,0}(x)]^+\} \leq l_{0,0}^*, \quad (37)$$

where $l_{0,0}^* = \max_{x \in \mathcal{X}, l_{i,0}(x) \geq 0, \forall i \in \mathbb{N}_1^n} l_{0,0}(x)$ is a lower bound on f_0^* . Then, ρ must satisfy the assumption in Theorem 3, i.e., $\rho \geq \bar{\rho}$ with probability at least $1 - \delta$.

The inequality (37) is defined completely in terms of the initial quantile bounds (at iteration $t = 0$), which can be efficiently evaluated and optimized, as discussed in Section 3. As such, we can practically find ρ using (37) before selecting any new samples using Algorithm 1. In particular, one can start from an initial guess of ρ and keep iteratively increasing it until (37) is satisfied. Under Assumptions 1-4, a finite value for ρ must exist that satisfies (37). Note that the result holds at every iteration and so a new ρ can be computed as the algorithm proceeds at additional cost, though there is little benefit to updating this choice once a sufficiently large value has been found. Furthermore, we have found that a fixing ρ to be a large value (on the order of 10^5) is often sufficient in practice as long as the objective and constraint functions are reasonably scaled.

4.5. Infeasibility declaration

Theorems 2 and 3 assume the original problem is feasible; however, this may not always hold in practice and thus we have incorporated an infeasibility detection scheme in Algorithm 1, similar to [32]. Lemma 2 showed that, given the problem is feasible, infeasibility will not be declared with high probability. It turns out that we can further show that this step will be triggered with high probability if the original problem is infeasible as shown next.

Theorem 4. *Let Assumptions 1, 2, and 4 hold, assume the original problem (1) is infeasible, and that $\gamma_T^{\max} = o(\log(T)/T)$ (i.e., $\lim_{T \rightarrow \infty} \gamma_T^{\max} \log(T)/T = 0$). Then, given a desired confidence level $\delta \in (0, 1)$, the CUQB method (Algorithm 1) will*

declare infeasibility in Line 3 within a finite number of iterations

$$\bar{T} = \min_{T \geq 1} \left\{ T : \sqrt{\frac{\log(T)\gamma_T^{max}}{T}} < C\epsilon \right\}, \quad (38)$$

with probability at least $1 - \delta$, where $\epsilon = -\max_{x \in \mathcal{X}} f_i(x) > 0$ for some $i \in \mathbb{N}_1^n$ and C is some positive constant.

5. Computational Setup

5.1. Test problems

We consider a total of 22 test problems that have been aggregated from the optimization and control literature. The problems have different number of input dimensions (ranging from 2 to 10) with bound constraints, different number of internal black-box functions (ranging from 1 to 24), and different number of constraints (ranging from 0 to 3). The test problems are distributed into two major groups: (i) synthetic problems and (ii) realistic application problems. The synthetic problems can be further divided into a set of 10 unconstrained and 10 constrained problems that are summarized in Tables 2 and 3, respectively. Note that these synthetic problems have been modified from the original form to exhibit some degree of composite (grey-box) structure by splitting up the objective and/or constraint functions into known and unknown components. The full problem descriptions can be found in Appendices C and D. The realistic application problems include an environmental model calibration problem and a real-time reactor optimization problem. The first problem is meant to demonstrate that CUQB can scale to a problem with a large number of black-box functions. The second problem is meant to show how CUQB can be effective in the presence of measurement noise.

Table 2: Summary of the unconstrained set of synthetic test problems with input dimension d and number of black-box functions m . The detailed formulation and solution of each problem is provided in Appendix C.

| Name | d | m | Appendix | Reference |
|-----------------|-----|-----|----------|-----------|
| Booth | 2 | 1 | C.1 | [58] |
| Wolfe | 3 | 1 | C.2 | [59] |
| Rastrigin | 3 | 2 | C.3 | [60] |
| Colville | 4 | 1 | C.4 | [61] |
| Friedman | 5 | 1 | C.5 | [62] |
| Dolan | 5 | 2 | C.6 | [59] |
| Rosenbrock | 6 | 4 | C.7 | [63] |
| Zakharov | 7 | 1 | C.8 | [64] |
| Powell | 8 | 4 | C.9 | [65] |
| Styblinski-Tang | 9 | 4 | C.10 | [66] |

5.2. Baseline methods for comparison

We compare a collection of seven different solvers to the proposed CUQB method, which are summarized in detail below. These solvers represent a mixture of the top performing DFO solvers from the comprehensive study performed in [12] and more recently published state-of-the-art constrained BO methods.

Table 3: Summary of the constrained set of synthetic test problems with input dimension d , number of black-box functions m , and number of nonlinear constraints n . The detailed formulation and solution of each problem is provided in Appendix D.

| Name | d | m | n | Appendix | Reference |
|--------------|-----|-----|-----|----------|-----------|
| Bazaraa | 2 | 2 | 2 | D.1 | [67] |
| Spring | 3 | 2 | 4 | D.2 | [68] |
| Ex314 | 3 | 2 | 3 | D.3 | [69] |
| Rosen-Suzuki | 4 | 2 | 3 | D.4 | [70] |
| st_bpv1 | 4 | 3 | 4 | D.5 | [71] |
| Ex211 | 5 | 2 | 1 | D.6 | [69] |
| Ex212 | 6 | 2 | 2 | D.7 | [69] |
| g09 | 7 | 2 | 4 | D.8 | [72] |
| Ex724 | 8 | 3 | 4 | D.9 | [69] |
| Ex216 | 10 | 4 | 5 | D.10 | [69] |

1. **SNOBFIT** [73]: This is a deterministic global search DFO solver that uses a combination of search spacing partitioning and surrogate model building. It uses the global selection rule to split the search domain and builds a quadratic model around the current best iterate and linear models for all other evaluated points. We use the Python implementation of SNOBFIT that has been made available in the `scikit-quant` package [74].
2. **DIRECT** [18]: This is a deterministic global search DFO solver that systematically divides the search domain into smaller rectangles and eliminates regions that are unlikely to contain the global optimum. The elimination process is based on concept of Lipschitz continuity, which ensures that the function’s behavior within a rectangle is predictable. We use the Python implementation of DIRECT in the `scipy` package [75].
3. **BOBYQA** [76]: This is a deterministic local search DFO solver that belongs to the class of trust region algorithms. It iteratively constructs local quadratic models within the trust region to guide the search toward an optimum. We use the Python implementation of BOBYQA in the `Py-BOBYQA` package [77].
4. **CMA-ES** [14]: This is a stochastic global search DFO solver that belongs to the class of evolutionary algorithms. It maintains a population of candidate solutions and iteratively updates the mean and covariance matrix of a multivariate normal distribution to generate new candidate solutions. The algorithm adaptively adjusts the covariance matrix based on the success of previous iterations, allowing it to explore the search space efficiently. We use the Python implementation of CMA-ES in the `pycma` package [78].
5. **EIC** [27]: Expected improvement with constraints (EIC) is a deterministic global search constrained DFO solver that belongs to the family of black-box BO methods. It modifies the popular EI acquisition function to account for black-box constraints. In the absence of constraints, it simply reduces to the standard EI acquisition function [47]. One can interpret EIC as a special case of the COBALT acquisition function in (5) when $g_i(x, h(x)) = h_i(x)$ (i.e., fully black-box instead of grey-box) for all $i \in \mathbb{N}_0^n$. In this case, one can derive a closed-form expression for the expectation in (5), as shown in [27]. We implemented our own version of EIC in the `BoTorch` package [79].
6. **EPBO** [31]: Exact penalty Bayesian optimization (EPBO) is a deterministic global search constrained DFO solver that belongs to the family of black-box BO methods. It is an extension of the well-known upper confidence

bound (UCB) acquisition function [39] to handle black-box constraints in a rigorous manner. One can show that the proposed CUQB method (without the infeasibility detection scheme) is equivalent to EPBO when $g_i(x, h(x)) = h_i(x)$ (i.e., fully black-box instead of grey-box) for all $i \in \mathbb{N}_0^n$. We implemented EPBO using the `BoTorch` package [79].

7. **EIC-CF [38]:** EIC with composite functions (EIC-CF) is a deterministic global search constrained DFO solver that belongs to the family of grey-box BO methods. It is an extension of the black-box EIC method to account for the known (grey-box) composite structure present in (1). EIC-CF is closely related to the COBALT method developed by the authors; the main difference is how one goes about practically optimizing over (5). EIC-CF directly optimizes over (7) with the non-smooth indicator function replaced by a differentiable sigmoid approximation whereas COBALT employs several heuristics that we have found lead to significant practical improvements (see numerical experiments in [38, Section 4]). We opted to use the vanilla EIC-CF for a fairer comparison to CUQB that does not incorporate any additional heuristics to improve the optimization of the acquisition function. We implemented EIC-CF using the `BoTorch` package [79].

SNOBFIT, DIRECT, BOBYQA, and CMA-ES are not directly applicable to problems with unknown (black- or grey-box) constraints. To make them applicable to problems of the form (3), we use a standard penalty method (see, e.g., [80]). In this approach, the following penalty function is defined

$$F(x) = f_0(x) - \rho \sum_{i=1}^m [-f_i(x)]^+, \quad (39)$$

such that one can solve the box-constrained problem $\max_{x \in \mathcal{X}} F(x)$ in place of (3). We used a penalty weight factor of $\rho = 10^5$, which was found to provide a reasonable tradeoff between efficiency and constraint satisfaction. The other methods (including CUQB) handle constrained problems directly and so do not require further modifications. Unless otherwise stated, all solvers use their default settings. As discussed more in the next section, all constrained BO methods (EIC, EPBO, EIC-CF, and CUQB) were implemented by the authors in the same environment and therefore use the exact same GP model settings and same optimization strategy for tackling their respective acquisition subproblem.

5.3. Implementation details

All BO methods assume that the black-box functions are independent and have a GP prior with a zero mean function and a 3/2 Matern covariance function with automatic relevance determination (ARD) [41]. The hyperparameters of the covariance function were estimated using the standard maximum likelihood estimation (MLE) approach in [41, Chapter 2] using the `fit_gpytorch_mll` function in `BoTorch`.

The complete list of hyperparameters for the practical implementation of CUQB are summarized in Table 4 that are divided into 4 major groups including termination. Since the focus of this work is on expensive functions, we impose an overall budget of only 100 function evaluations in every run and all algorithms are compared based on the quality of the recommended solution at every iteration. We briefly discuss the hyperparameters in the other three groups below.

We note that these practical choices are made to achieve a good tradeoff between computational cost of solving (13) and high quality sampling performance. One could easily update these parameters to generate a more accurate solution to (13) at additional cost. The theory in Section 4 suggests that, the closer one gets to exactly solving (13), the better results one would observe in practice. This means

that the ideal CUQB method in Algorithm 1 should only further improve upon the performance results shown in Section 6.

A Python implementation of the proposed CUQB method, along with the other baseline solvers described previously, is available at [81].

Table 4: Default hyperparameter values for practical implementation of CUQB algorithm.

| | Parameter | Description | Value |
|-------------------|------------------|---|----------|
| Quantile approx. | L | Number of randomly drawn Monte Carlo samples | 50 |
| | ε | Regularization strength of soft sorting operator | 0.1 |
| | α | Probability level | 0.95 |
| Optimization | N_0 | Number of initial random samples used to select multi-start candidates | 8192 |
| | N_{opt} | Number of multi-start initialization candidates for L-BFGS-B | 3 |
| | ρ | Penalty factor used for recommendation procedure and approximately solving (13) | 10^5 |
| Budget allocation | T_0 | Budget for initial randomly drawn samples | $2d + 1$ |
| Termination | T | Overall evaluation budget | 100 |

5.3.1. Quantile approximation hyperparameters

As discussed in Section 3.3, the quantile functions $\{u_{i,t}(x)\}_{i \in \mathbb{N}_0^n}$ must be estimated using some number of MC samples L and sorting regularization parameter ε . Proposition 1 shows that the empirical estimate will converge to the true quantile function as $L \rightarrow \infty$ and $\varepsilon \rightarrow 0$; however, it is not clear what values will work well enough in practice. Through numerical tests, we found that the results are fairly insensitive to values of $\varepsilon < 1$, which matches the observations from [40]. This result was consistent across L and α values, which led us to select a default value of $\varepsilon = 0.1$ that provides a nice tradeoff between accuracy and useful gradient information for the optimization process. In Figure 3, we plot the empirical quantile function for a given test problem for a range of $L \in \{50, 100, 1000\}$ and $\alpha \in \{0.75, 0.85, 0.95\}$ with $\varepsilon = 0.1$. Due to the randomness of the MC samples, we plot the resulting empirical quantile function for ten independent replicates. As expected, the variance in the estimator reduces as L increases. Although we would like to select a large value for L , this increases the computational complexity of the optimization in (13) such that it is practically useful to select as small an L as possible without degrading the quality of the solution to (13). Even for $L = 50$, the variability is reasonably small and does not change the location of the maximizer for this test problem. We found similar results across other problems, which led us to select 50 as the default value.

Lemma 1 provides a rigorous way to select $\alpha_{i,t}$; however, it tends to be conservative as it was derived through use of the union bound. A simpler choice that often works well in practice is to set $\alpha_{i,t} = \alpha$ to a constant $\alpha \in (0, 1)$. The value α is a key factor impacting overall performance of CUQB. Empirically, if α is set to a small value, then there will be limited exploration and one is more likely to miss finding the globally optimal solution. On the other hand, if α is set too close to one,

then the algorithm will overemphasize exploration, which may cause a reduction in the rate of convergence. Through the results of our performance tests in Section 6.1, we found that setting $\alpha = 0.95$ works well across a variety of problems.

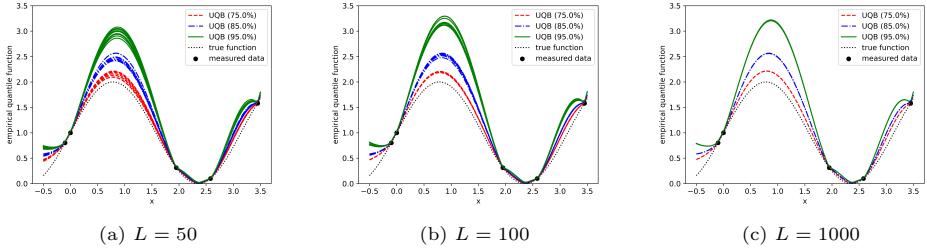


Figure 3: Empirical quantile function for a test problem estimated using $L = 50, 100, 1000$ Monte Carlo samples and $\alpha = 0.75, 0.85, 0.95$ probability levels for 10 independent replicates.

5.3.2. Optimization hyperparameters

Although we can solve (13) with a guaranteed global optimization method (such as branch and bound), it can be expensive in practice due to the generally non-convex and multi-modal structure of the acquisition function. There are a large number of alternatives that one can pursue to tackle (13), with the simplest being to use DFO solvers that do not require gradient information. These are feasible for low-dimensional problems but do not scale to higher dimensions. A more scalable approach, used in [79], is to rely on quasi-second order methods to provide strong convergence properties to a local solution. For simplicity, we focus on a penalty method for solving (13) in practice (though this could easily be replaced by interior point or sequential quadratic programming methods)

$$x_{t+1} \in \operatorname{argmax}_{x \in \mathcal{X}} a_t(x), \quad (40)$$

where $a_t(x) = u_{0,t}(x) - \rho \sum_{i=1}^n [-u_{i,t}(x)]^+$. As long as the upper quantile functions are monotonically non-increasing (i.e., $u_{i,t-1}(x) \geq u_{i,t}(x)$ for all $t \geq 0$), then (40) is a solution to the original problem (13) for the ρ derived in Corollary 1. We can practically compute gradients of $a_t(x)$ since it is differentiable everywhere except the kink points in the non-smooth max operator $[-u_{i,t}(x)]^+$. As such, (40) can be solved to local optimality using the unconstrained L-BFGS-B solver [82], which is supported in **BoTorch** (thus directly enabling GPU-accelerated computations).

Due to the structure of the acquisition function $a_t(x)$, the initial conditions supplied to L-BFGS-B are very important since one may get stuck in highly suboptimal local solutions. To reduce this risk, it is common to employ multi-start optimization (i.e., start the solver from multiple initial conditions and select the best of the final converged solutions). We use the following effective heuristic developed in [79] to select a set of N_{opt} initialization candidates:

1. Sample N_0 points $\{x_0^{(i)}\}_{i=1}^{N_0}$ from \mathcal{X} using quasi-random Sobol sequences;
2. Evaluate acquisition function at these candidate points $\{v_t^{(i)} = a_t(x_0^{(i)})\}_{i=1}^{N_0}$;
3. Let $\hat{\mu}$ and $\hat{\sigma}$ denote the empirical mean and standard deviation of the population $\{v_t^{(i)}\}_{i=1}^{N_0}$, respectively. Sample N_{opt} points from the set $\{x_0^{(i)}\}_{i=1}^{N_0}$ where each point is selected with probability $p_i \propto \exp(\eta(v_t^{(i)} - \hat{\mu})/\hat{\sigma})$ for some $\eta > 0$.

This sampling procedure ensures that the initial conditions are selected to tradeoff exploration/exploitation based on the magnitude of η . This procedure converges to standard Sobol sampling as $\eta \rightarrow 0$ and corresponds to greedy sampling as $\eta \rightarrow \infty$.

The latter approach is known to suffer from the clustering problem wherein all the high-valued points are near each other and thus converge to the same local solution. We therefore stick with the default value of $\eta = 1$. Interested readers are referred to [79, Appendix E] for detailed comparison of this practical optimization approach to various alternative methods.

5.3.3. Budget allocation hyperparameters

CUQB can be easily applied in the absence of initial data when an accurate GP prior is known. Since this is not the case in many real-world applications, we instead rely on the MLE training procedure discussed previously, which can be inaccurate when there is little-to-no initial data. Therefore, it is common in the BO literature to select a set of T_0 points uniformly at random to act as purely explorative samples in the early phase of the algorithm. There have been limited studies on the impact of T_0 on performance, though it is commonly suggested to use $d + 1$ random points. We study the impact of T_0 , along with the choice of α , in Section 6.1 and find that a value of $2d + 1$ provides a better overall balance between quick performance gains without getting stuck in local solutions.

5.3.4. Estimating average performance

The performance of most solvers used in this study are affected by the selected set of starting points (e.g., N_0 initial samples for the BO-related methods). Therefore, each problem is solved for ten different randomly generated starting points. The same set of points are used for all considered solvers. We then compare the average behavior of each algorithm by the median recommended feasible objective value from the ten runs for each problem.

5.4. Performance metric

To evaluate the performance of the different algorithms, we modify the so-called “performance profiles” suggested in [83] to work for constrained problems. Specifically, for a given tolerance $\tau \in [0, 1]$ and starting point x_0 , the following test is used to measure the increase in a penalized version of the objective function value:

$$F_\omega - F(x_0) \geq (1 - \tau)(F_U - F(x_0)), \quad (41)$$

where $F(x) = f_0(x) - \rho \sum_{i=1}^n [-f_i(x)]^+$ is the penalized objective function, $F(x_0)$ is the value of the penalized objective function at the starting point x_0 , F_ω is the value of the penalized objective function at the solution reported by solver ω , and F_U is the largest penalized objective function obtained by any of the solvers in the set of interest within the budget of T function evaluations. This condition implies that a problem is considered “solved” by ω if the median solution improved the starting point by at least a fraction of $(1 - \tau)$ of the largest attained reduction.

For the set of test problems \mathcal{K} , the performance profile of any solver ω is defined as the fraction of problems that can be solved for a given tolerance τ within a given number of function evaluations t , i.e.,

$$\varrho_\omega(t) = \frac{\text{card}\{\kappa \in \mathcal{K} : \pi_{\kappa, \omega} \leq t\}}{|\mathcal{K}|}, \quad (42)$$

where $\pi_{\kappa, \omega}$ denotes the minimum number of function evaluations required by solver ω to satisfy (41) for problem κ .

6. Results and Discussion

6.1. Choice of initial sampling and probability level parameter

The initial sample budget T_0 and the probability level α are key hyperparameters in CUQB. To illustrate their influence, we conduct numerical experiments by varying them over $T_0 \in \{d + 1, 2d + 1, 3d + 1\}$ and $\alpha \in \{0.75, 0.85, 0.95\}$. Figure 4 shows the performance profiles for CUQB under these different settings for the set of 10 unconstrained problems (Table 2) with $\tau = 0.01$. The first major increase in performance happens around iteration 15, with $T_0 = d + 1$ performing better than the alternatives that devote more of the initial budget to random samples. However, we see $T_0 = d + 1$ has slightly worse performance in the later iterations, which is likely due to inaccurate estimates of the GP hyperparameters causing it to under-explore. Similarly, the lower α values of 0.75 and 0.85 show slightly worse performance after iteration 40 than 0.95 as a result of under-exploration (due to the quantile function not providing an accurate upper bound the true function). As such, the combination of $T_0 = 2d + 1$ and $\alpha = 0.95$ appears to be the best at providing balanced data for learning accurate covariance hyperparameters while not under- or over-exploring the design space. We emphasize that, under these settings, the median CUQB result was within 1% of the best improvement for 9 out of 10 problems of varying size and difficulty within only 40 function evaluations.

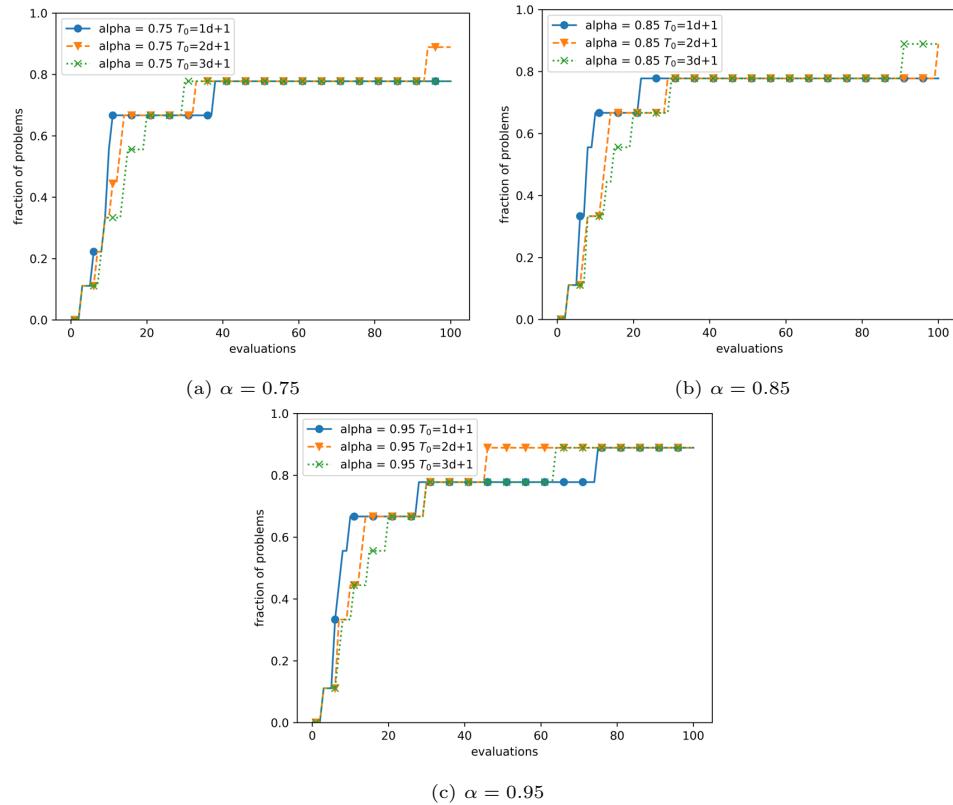


Figure 4: Performance profiles comparing CUQB under different initial sampling and probability level settings over the set of unconstrained test problems for $\tau = 0.01$.

6.2. Computational results for unconstrained problems

Figure 5 compares the performance of CUQB to the solvers described in Section 5.2 for the set of unconstrained test problems for $\tau = 0.01$. We see that CUQB

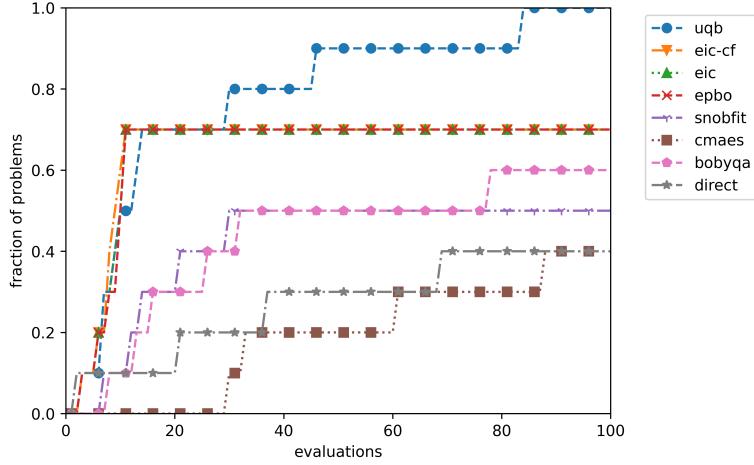


Figure 5: Performance profiles comparing CUQB and the selected set of comparison solvers in Section 5.2 over the set of unconstrained test problems for $\tau = 0.01$.

substantially outperforms all tested solvers, as it is the only solver capable of solving $> 70\%$ of the test problems (according to (41)). All BO-related methods (CUQB, EIC-CF, EIC, EPBO) perform similarly for the first 25 iterations and clearly outperform the non-BO solvers, solving 70% of the test problems within around 15 iterations. SNOBFIT and BOBYQA, which are both model-based DFO solvers, end up solving 50% and 60% of test problems, respectively, compared to only 40% for DIRECT and CMA-ES. This is not unexpected since model-based DFO solvers are known to be quite efficient for small- to medium-sized problems. It is interesting to note that the BO methods show faster convergence than deterministic model-based methods likely due to a better balance between exploration and exploitation. However, EIC-CF, EIC, and EPBO are unable to solve 30% of the test problems whereas CUQB ends up solving 100% of them. EIC and EPBO do not take advantage of prior knowledge and so their performance is fundamentally limited especially on higher-dimensional problems. Although EIC-CF does account for prior knowledge, the acquisition surface is much more challenging to optimize, which leads to poor performance in the later iterations. CUQB is able to overcome both of these limitations, making it the best solver among the tested solvers.

6.3. Computational results for constrained problems

Next, we analyze the performance of CUQB on problems with black- and grey-box constraints. Figure 6 shows the performance profiles for CUQB and the selected baseline solvers for the set of constrained test problems with $\tau = 0.01$. CUQB remains the best solver in terms of both speed of finding a solution and number of problems solved, being the only solver to satisfy (41) for $> 80\%$ of the test problems. Similarly to the unconstrained case, we see that the BO-related solvers continue to outperform the non-BO solvers by a significant margin. Furthermore, due to the lumped penalty approach employed by SNOBFIT, DIRECT, BOBYQA, and CMA-ES, all of these methods struggle to select feasible solutions during the search process, resulting in much worse performance. In particular, DIRECT and CMA-ES are unable to solve any constrained test problem well enough to satisfy (41) while BOBYQA and SNOBFIT can only solve 10% and 20%, respectively. All BO-related methods, on the other hand, are able to solve $\geq 50\%$ of the constrained test problems since they explicitly model the constraints to better shape the acquisition function to identify feasible points. An important difference between the results for

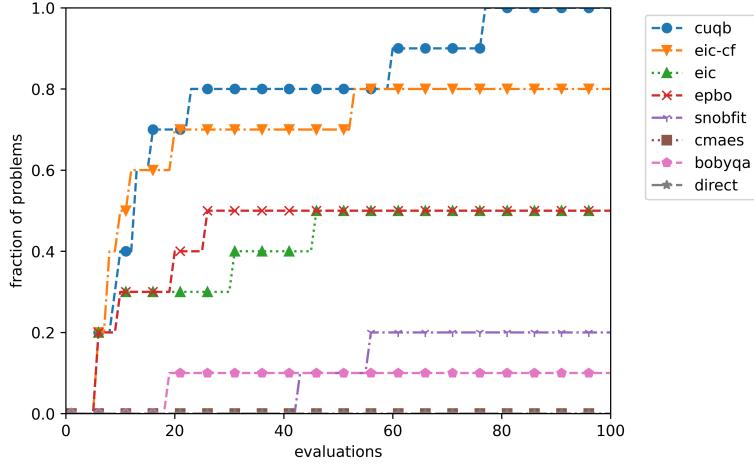


Figure 6: Performance profiles comparing CUQB and the selected set of comparison solvers in Section 5.2 over the set of constrained test problems for $\tau = 0.01$.

the constrained and unconstrained test problems is that there is a much bigger gap between the black-box methods (EIC and EPBO) and the grey-box methods (EIC-CF and CUQB) on the constrained problems. This suggests that prior knowledge is even more helpful when it comes to finding feasible points that satisfy unknown constraint functions, which is a key advantage of the grey-box perspective adopted in this work. EIC-CF continues to underperform relative to CUQB, likely due to the complexity of the acquisition surface such that CUQB excels over all other tested solvers. CUQB solves 80% of the constrained test problems within the first 25 iterations and 100% of the constrained problems in the 100 evaluation budget.

6.4. Large-scale environmental model function

We further compare CUQB and the selected solvers on a more realistic case related to Bayesian calibration of expensive computer models, originally proposed in [84]. The main purpose of this study is to show that CUQB can effectively scale to problems with a large number of black-box functions.

Consider the spill of a chemical pollutant at two locations along a long narrow channel, which leads to the following concentration profile

$$c(s, t; M, D, \Lambda, \tau) = \frac{M}{\sqrt{4\pi Dt}} \exp\left(\frac{-s^2}{4Dt}\right) + \frac{M\mathbb{I}(t > \tau)}{\sqrt{4\pi D(t - \tau)}} \exp\left(\frac{-(s - \Lambda)^2}{4D(t - \tau)}\right), \quad (43)$$

where s and t are, respectively, the space and time points, M the mass of the pollutant spilled at each location, D the diffusion coefficient, Λ is the location of the second spill, and τ the time of the second spill. The true underlying parameters $M_0 = 10$, $D_0 = 0.07$, $\Lambda_0 = 1.505$, and $\tau_0 = 30.1525$ are unknown; however, we can estimate these parameters based on observations obtained from various monitoring stations. Specifically, we observe $c(s, t; M_0, D_0, \Lambda_0, \tau_0)$ on a 4×6 grid of values at $s \in \mathcal{S} = \{1, 1.5, 2.5, 3\}$ and $t \in \mathcal{T} = \{10, 20, 30, 40, 50, 60\}$. Therefore, our goal is to find the parameter values $x = (M, D, \Lambda, \tau)$ that minimize the mean squared error between our model predictions and available observations

$$g_0(x, h(x)) = - \sum_{(s,t) \in \mathcal{S} \times \mathcal{T}} (c(s, t; M_0, D_0, \Lambda_0, \tau_0) - c(s, t; M, D, \Lambda, \tau))^2, \quad (44)$$

where $h(x) = \{c(s, t; M, D, \Lambda, \tau) : (s, t) \in \mathcal{S} \times \mathcal{T}\} \in \mathbb{R}^{24}$ is our black-box function.

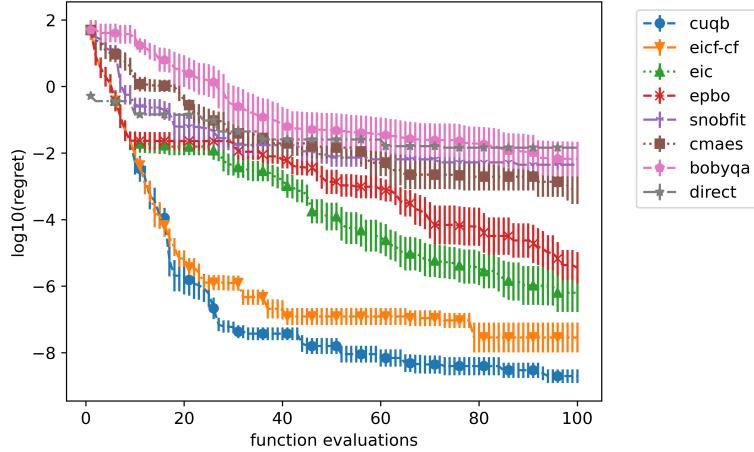


Figure 7: Expected simple regret (on a logarithmic scale) for the environmental model problem with approximate confidence bounds estimated from 10 independent realizations. Results are shown for all comparison solvers discussed in Section 5.2.

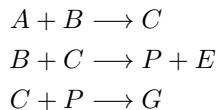
In summary, the environmental calibration problem has the following dimensions: $d = 4$, $m = 24$, and $n = 0$.

The average simple regret values $\min_{t'=1,\dots,t} \{r_{t'} = f_0^* - f(x_{t'})\}$ versus the number of iterations t for all solvers are shown in Figure 7. CUQB remains the best solver converging to a regret value less than 10^{-6} within the first 20 simulations and a value of less than 10^{-8} by the end of 100 simulation budget (implying it quickly and accurately identified the true global solution). EIC-CF, which also exploits the composite structure of the objective function, performs similarly to CUQB, with only slightly worse performance after 20 iterations. The same trend remains for the other solvers, with EIC and EPBO coming in third and fourth, respectively, with a substantially slower rate of convergence than CUQB and EIC-CF due to neglected problem structure. However, these model-based solvers outperform the alternatives (DIRECT, CMA-ES, SNOBFIT, and BOBYQA), which end up with around 6 orders of magnitude worse regret values than CUQB after 100 simulations. These results demonstrate CUQB’s ability to effectively exploit problem structure to reduce simulation cost in an important and common application problem.

6.5. Application to real-time reactor optimization

As a final case study, we consider the application of CUQB to the real-time optimization (RTO) of a chemical reactor system. The main purpose of this study is to demonstrate CUQB’s ability to effectively handle measurement noise, which commonly occurs in RTO and other process control applications. Noise affects both the sample selection (13) and recommendation (14) processes, so we analyze both impacts in the results below, after providing a short description of the problem.

We consider the Williams-Otto system from [4], which is a benchmark problem in the RTO literature, in which a continuously stirred tank reactor (CSTR) is fed with two streams of pure reactants A and B with inlet flowrates F_A and F_B . The reactor operates at a fixed temperature T_r , with the following three reactions occurring



where C is an intermediate, G is an undesirable byproduct, and P and E are two

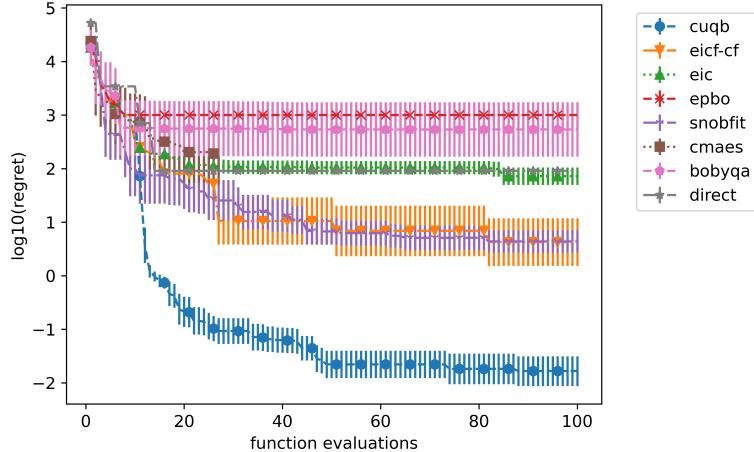


Figure 8: Expected simple regret (on a logarithmic scale) for the Williams-Otto RTO problem without measurement noise ($\sigma = 0$) with approximate confidence bounds estimated from 10 independent realizations. Results are shown for all comparison solvers discussed in Section 5.2.

main products. The complete set of mass balance and kinetic rate equations can be found in [85]. Let X_A , X_G , X_P , and X_E denote the mass fraction of species A , G , P , and E , respectively, which depend on the two input variables $x = (F_B, T_r)$. The unknown black-box function is $h(x) = (X_A(x), X_G(x), X_P(x), X_E(x)) \in \mathbb{R}^4$. We assume that we only have access to noisy measurements of these mass fractions

$$y = h(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I_4), \quad (45)$$

where σ denotes the standard deviation of the noise. We are interested in maximizing the economic profit of this system by manipulating F_B and T_r subject to operating constraints on the mass fraction of unreacted A and byproduct G . We can formulate these as composite functions as follows

$$g_0(x, y) = (1043.38X_P + 2092X_E)(F_A + F_B) - 79.23F_A - 118.34F_B, \quad (46a)$$

$$g_1(x, y) = 0.12 - X_A, \quad (46b)$$

$$g_2(x, y) = 0.08 - X_G. \quad (46c)$$

In summary, the Williams-Otto RTO problem has the following dimensions: $d = 2$, $m = 4$, and $n = 2$.

For comparison purposes, we first present results for the ideal perfect measurement case, i.e., $\sigma = 0$. As discussed in Section 4.4, an advantage of the noise-free case is that we can easily identify the point in the sample sequence $\{x_1, \dots, x_t\}$ that has minimum penalty-based regret according to (35). Figure 8 shows the average simple penalty-based regret (minimum over the sequence) versus the number of iterations for all solvers. CUQB continues to perform the best, achieving penalty-based regret values of $\sim 10^{-2}$ within 50 iterations, which is more than a factor of 100 better than any other tested solver. EIC-CF performs significantly worse than CUQB in this case, which is likely due to a combination of how it handles constraints and the complexity of optimizing the acquisition function. Another interesting observation is that the black-box BO methods (EIC and EPBO) are outperformed by SNOBFIT, which is likely due to the problem being low-dimensional ($d = 2$) and the sensitivity of the constraint functions with respect to the inputs that results in over-exploration of points with relatively small amounts of constraint violation.

Next, we look at how the performance changes when a small ($\sigma = 0.01$) and large

$(\sigma = 0.05)$ amount of noise is included in the observations. Since we can no longer identify the best sample point in the sequence using (35), we need an updated recommendation process. We proposed a quantile bound recommender (14) for which we establish convergence rates in Theorem 2. To demonstrate the practical importance of the proposed recommender, we also compare it to a “naive recommender” that replaces the noise-free values in (35) with their noisy counterparts. The true penalty-based regret values (evaluated at the recommended point) versus number of iterations for both recommenders and both noise levels are shown in Figure 9. The CUQB search process shows the best performance (lowest true penalty-based regret values) in all cases. We see a considerable increase in the regret values (more than a factor of 10) when using the naive recommender (Figure 9ac) compared to the proposed quantile bound recommender (Figure 9bd) – this highlights the quantile bound recommender’s ability to adequately filter the measurement noise when making a final recommendation, which is critical in the presence of constraints. As expected, we do see a drop in performance of all algorithms due to the presence of noise; however, CUQB appears to be the least impacted by noise since the mean and standard deviation remain the smallest out of all tested solvers for both noise levels. These results demonstrate CUQB’s unique capability of simultaneously exploiting problem structure and filtering noise in an industrially-relevant application.

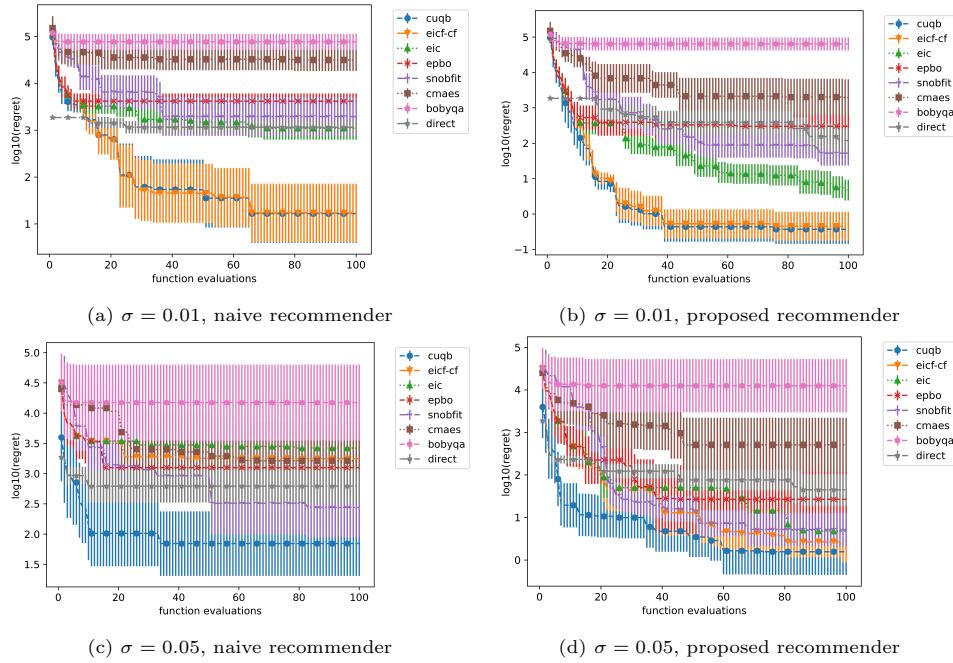


Figure 9: Expected simple penalty-based regret (on a logarithmic scale) for the Williams-Otto RTO problem with measurement noise with approximate confidence bounds estimated from 10 independent realizations. Results are shown for two measurement noise levels $\sigma = 0.01$ (top) and $\sigma = 0.05$ (bottom), the naive (left) and proposed quantile bound recommender (right), and for all comparison solvers discussed in Section 5.2.

7. Conclusions

This paper proposes a new algorithm, CUQB, as a simple and effective approach for efficient constrained global optimization of composite functions whose input is an expensive vector-valued black-box function. These types of problems commonly

arise when modeling real-world science, engineering, and control applications and allows for easy incorporation of so-called hybrid (grey-box) model structures. CUQB takes advantage of the principle that one should be optimistic in the face of uncertainty, which has been successfully exploited in its unconstrained and black-box counterparts. In particular, our algorithm solves a constrained auxiliary problem defined in terms of quantile bound functions that can be used to identify a high probability range of prediction outcomes for the objective and constraints given all available data and prior knowledge. In addition to developing an efficient optimization procedure for CUQB, we also perform extensive theoretical analysis on its performance. Specifically, we develop bounds on the cumulative regret and constraint violation that can be translated into convergence rate bounds to the global solution under certain regularity assumptions. We also derive an infeasibility detection scheme as an extension of recent work for the black-box case. Several detailed numerical experiments are performed that demonstrate the effectiveness of CUQB, which we find to be highly competitive with state-of-the-art alternative methods.

Three future directions for extending upon the version of CUQB presented in this paper are briefly discussed next. First, the acquisition optimization subproblem is generally nonconvex and thus is difficult to solve to global optimality. This is a known challenge due to the structure of the GP model since the number of nonlinear terms in GPs equals the product of the number of interpolated points and the dimensionality of the input space [86]. Several works have considered global optimization with embedded GP surrogate models, e.g., [87, 88, 89]; however, there has been limited work on extending these ideas to acquisition functions that incorporate the predictive variance term, along with other nonlinear transformations. In [90], a spatial branch-and-bound technique is proposed that uses generalized McCormick relaxations to efficiently compute lower and upper bounds. Although finite-time convergence to a global solution can be guaranteed under relatively mild assumption, this implementation is currently too slow to apply in practice and so additional work is needed to make global optimization viable in realistic settings. Second, the theoretical results technically hold for design sets with finite cardinality. Although we show that the impact on the number of design points on the rate of convergence is quite weak (logarithmic), it remains an open question exactly how CUQB will theoretically perform in the worst-case in continuous design spaces. We conjecture that similar results can be derived for CUQB in continuous spaces as long as the quantile bound is appropriately related to GP bounds in continuous spaces that have been recently established in [51]. Third, while technically possible to apply CUQB to equality-constrained problems by transforming $c(x) = 0$ into two inequality constraints $c(x) \geq 0$ and $-c(x) \geq 0$, it is unclear how effective such a reformulation will work in practice. Noisy observations of the black-box functions will further complicate this task, so that additional modifications may be needed to overcome the challenges of a limited feasible region.

A. Appendix: Proofs for Theoretical Results in Section 3

A.1. Proof of Proposition 1

We start by proving the following lemma.

Lemma 6. *Let $y_{(\lceil \alpha L \rceil)}$ denote the $[pL]$ -th order statistic of the dataset y_1, \dots, y_L for $p \in (0, 1)$. Let the samples $\{y_i\}_{i=1}^L$ be i.i.d. with cumulative distribution function $F(y) = \Pr\{Y \leq y\}$ and $F_-(y) = \Pr\{Y < y\}$. Let $Q(p) = \inf\{y : F(y) \geq p\}$ be the corresponding p -quantile. Then, $y_{(\lceil pL \rceil)} \rightarrow Q(p)$ in probability as $L \rightarrow \infty$.*

Proof. By definition, the k -th order statistic must satisfy

$$y_{(k)} \leq y \Leftrightarrow \sum_{l=1}^L \mathbb{I}(y_l \leq y) \geq k, \quad (\text{A.1})$$

$$y_{(k)} \geq y \Leftrightarrow \sum_{l=1}^L \mathbb{I}(y_l \geq y) \geq L - k + 1. \quad (\text{A.2})$$

For $L \geq 1$ and $p \in (0, 1)$, $\lceil pL \rceil > 0$ so that $y_{(\lceil pL \rceil)}$ is well-defined. Recall the formal definition of convergence in probability, i.e., we must show that

$$\lim_{L \rightarrow \infty} \Pr \{ |y_{(\lceil pL \rceil)} - Q(p)| \geq \epsilon \} = 0, \quad \forall \epsilon > 0. \quad (\text{A.3})$$

We can rewrite the probability expression as follows

$$\begin{aligned} & \Pr \{ |y_{(\lceil pL \rceil)} - Q(p)| \geq \epsilon \} \\ &= \Pr \{ y_{(\lceil pL \rceil)} \leq Q(p) - \epsilon \} + \Pr \{ y_{(\lceil pL \rceil)} \geq Q(p) + \epsilon \}. \end{aligned} \quad (\text{A.4})$$

We will analyze these two terms separately. Using the equivalence in (A.1), we can establish the following

$$\Pr \{ y_{(\lceil pL \rceil)} \leq Q(p) - \epsilon \} = \Pr \left\{ \frac{1}{L} \sum_{l=1}^L \mathbb{I}(y_l \leq Q(p) - \epsilon) \geq \frac{\lceil pL \rceil}{L} \right\}. \quad (\text{A.5})$$

By the strong law of large numbers, $\frac{1}{L} \sum_{l=1}^L \mathbb{I}(y_l \leq Q(p) - \epsilon) \rightarrow F(Q(p) - \epsilon)$ in probability and $\frac{\lceil pL \rceil}{L} \rightarrow p$ as $L \rightarrow \infty$. Therefore, taking the limit of the above expression leads to

$$\lim_{L \rightarrow \infty} \Pr \{ y_{(\lceil pL \rceil)} \leq Q(p) - \epsilon \} = \Pr \{ F(Q(p) - \epsilon) \geq p \} = 0, \quad (\text{A.6})$$

which follows from the definition of $Q(p)$ (smallest y value such that $F(y) \geq p$) and the fact that $F(y)$ is non-decreasing such that $F(Q(p) - \epsilon) < p$ for any $\epsilon > 0$. We can repeat a similar analysis for the second term after algebraic manipulations

$$\begin{aligned} & \Pr \{ y_{(\lceil pL \rceil)} \geq Q(p) + \epsilon \} \\ &= \Pr \left\{ \frac{1}{L} \sum_{l=1}^L \mathbb{I}(y_l \geq Q(p) + \epsilon) \geq \frac{L - \lceil pL \rceil + 1}{L} \right\}, \\ &= \Pr \left\{ \frac{1}{L} \sum_{l=1}^L \mathbb{I}(y_l < Q(p) + \epsilon) \leq \frac{\lceil pL \rceil}{L} - \frac{1}{L} \right\}, \end{aligned} \quad (\text{A.7})$$

where we have used $\mathbb{I}(y_l \geq y) = 1 - \mathbb{I}(y_l < y)$. Again, applying the strong law of large numbers, we have $\frac{1}{L} \sum_{l=1}^L \mathbb{I}(y_l < Q(p) + \epsilon) \rightarrow F_-(Q(p) + \epsilon)$ and $\frac{\lceil pL \rceil}{L} - \frac{1}{L} \rightarrow p$ as $L \rightarrow \infty$. Taking the limit of then yields

$$\lim_{L \rightarrow \infty} \Pr \{ y_{(\lceil pL \rceil)} \geq Q(p) + \epsilon \} = \Pr \{ F_-(Q(p) + \epsilon) < p \} = 0, \quad (\text{A.8})$$

which follows from the fact that the limit approaches from below p and F is increasing with $F(Q(p)) \geq p$. It follows that $y_{(\lceil pL \rceil)} \in (Q(p) - \epsilon, Q(p) + \epsilon)$ with probability approaching 1 for any $\epsilon > 0$. \square

From [40, Proposition 2], we have that $s_\varepsilon(\boldsymbol{\theta}) \rightarrow s(\boldsymbol{\theta})$ as $\varepsilon \rightarrow 0$ for all $\boldsymbol{\theta} \in \mathbb{R}^L$. This means that the soft version of the empirical quantile will converge to the true empirical quantile value in (18) as $\varepsilon \rightarrow 0$ for any $p \in (0, 1)$, any $x \in \mathcal{X}$, and any $L \in \mathbb{N}$. Then, according to Lemma 6, we must have that $\hat{u}_{i,t}^{(L)}(x) \rightarrow u_{i,t}(x)$ in

probability for all $x \in \mathcal{X}$. \square

A.2. Proof of Proposition 2

This result follows from the fact that the posterior distribution of $h(x)$ is an m -dimensional normal with mean vector $\mu_t(x)$ and covariance matrix $\Sigma_t(x)$. By the closure of Gaussian distributions under linear transformations, the posterior distribution of $a_i^\top(x)h(x) + b_i(x)$ must be a univariate normal with mean $a_i^\top(x)\mu_t(x) + b_i(x)$ and variance $a_i^\top(x)\Sigma_t(x)a_i(x)$. The stated result then follows from the analytic expression for the quantile function of normal distribution and the fact that $\Phi^{-1}(\alpha) = -\Phi^{-1}(1 - \alpha)$ for any $\alpha \in (0, 1)$. \square

B. Appendix: Proofs for Theoretical Results in Section 4

B.1. Proof of Lemma 1

The quantile bounds $l_{i,t}(x)$ and $u_{i,t}(x)$ are random variables since they depend on the noisy observations y_t . Therefore, we can define the following event that the true function is contained in these bounds for all $x \in \mathcal{X}$ and $t \geq 0$ as follows

$$\mathcal{E}_i = \cap_{x \in \mathcal{X}} \cap_{t \geq 0} \{l_{i,t}(x) \leq g_i(x, h(x)) \leq u_{i,t}(x)\}, \quad \forall i \in \mathbb{N}_0^n. \quad (\text{B.1})$$

We can then establish the following sequence of inequalities on the joint probability of these events holding for all functions

$$\Pr \{\cap_{i=0}^n \mathcal{E}_i\} = 1 - \Pr \{\cup_{i=0}^n \overline{\mathcal{E}_i}\}, \quad (\text{B.2a})$$

$$\geq 1 - \sum_{i=0}^n \Pr \{\overline{\mathcal{E}_i}\}, \quad (\text{B.2b})$$

$$\geq 1 - \sum_{i=0}^n \sum_{x \in \mathcal{X}} \sum_{t \geq 0} (F_{Y_{i,t}(x)}(l_{i,t}(x)) + 1 - F_{Y_{i,t}(x)}(u_{i,t}(x))), \quad (\text{B.2c})$$

$$\geq 1 - \sum_{i=0}^n \sum_{x \in \mathcal{X}} \sum_{t \geq 0} \left(\frac{\alpha_{i,t}}{2} + 1 - \left(1 - \frac{\alpha_{i,t}}{2}\right) \right), \quad (\text{B.2d})$$

$$= 1 - \delta, \quad (\text{B.2e})$$

where (B.2a) follows from De Morgan's complement law; (B.2b) follows from Boole's inequality (union bound) being applied across the different functions; (B.2c) follows from Boole's inequality being applied across $x \in \mathcal{X}$, $t \geq 0$ and the definition of the CDF; (B.2d) follows from the identity $F_Y(Q_Y(\alpha)) \geq \alpha$ for any random variable Y and probability value $\alpha \in (0, 1)$; and (B.2e) follows from the fact that the infinite series $\sum_{t \geq 0} \frac{6}{\pi^2(t+1)^2} = 1$ is convergent such that $\sum_{t \geq 0} \alpha_{i,t} = \frac{\delta}{(n+1)|\mathcal{X}|}$. \square

B.2. Proof of Lemma 2

According to Lemma 1, the set of functions $\{g_i(x, h(x))\}_{i \in \mathbb{N}_0^n}$ are bounded by their lower $l_{i,t}(x)$ and upper quantile bounds $u_{i,t}(x)$ for all $x \in \mathcal{X}$ and iterations $t \geq 0$ with probability at least $1 - \delta$. The statements below are conditioned on this joint event being true, so must also hold with probability at least $1 - \delta$.

Under Assumption 3, x^* is feasible and therefore must satisfy $g_i(x^*, h(x^*)) \geq 0$ for all $i \in \mathbb{N}_1^n$. Since $u_{i,t}(x^*) \geq g_i(x^*, h(x^*)) \geq 0$, there must exist at least one feasible solution to (13), meaning infeasibility is not declared. We then establish

the following sequence of inequalities on the instantaneous regret

$$r_t \leq r_t^+, \quad (\text{B.3a})$$

$$= [f_0^* - u_{0,t}(x_{t+1}) + u_{0,t}(x_{t+1}) - f_0(x_{t+1})]^+, \quad (\text{B.3b})$$

$$\leq [f_0^* - u_{0,t}(x_{t+1})]^+ + [u_{0,t}(x_{t+1}) - f_0(x_{t+1})]^+, \quad (\text{B.3c})$$

$$\leq [u_{0,t}(x^*) - u_{0,t}(x_{t+1})]^+ + [u_{0,t}(x_{t+1}) - l_{0,t}(x_{t+1})]^+, \quad (\text{B.3d})$$

$$= u_{0,t}(x_{t+1}) - l_{0,t}(x_{t+1}) = w_{0,t}(x_{t+1}), \quad (\text{B.3e})$$

where (B.3b) follows from the definition of positive instantaneous regret; (B.3c) follows from $[a+b]^+ \leq [a]^+ + [b]^+$, $\forall a, b \in \mathbb{R}$; (B.3d) follows from the assumed upper and lower bounds on the objective function; and (B.3e) follows from (13), which implies $u_{0,t}(x^*) \leq u_{0,t}(x_{t+1})$ (since we are maximizing over the feasible region). We can perform a similar analysis on the constraint violation

$$v_{i,t} = [-f_i(x_{t+1})]^+, \quad (\text{B.4a})$$

$$= [-f_i(x_{t+1}) + u_{i,t}(x_{t+1}) - u_{i,t}(x_{t+1})]^+, \quad (\text{B.4b})$$

$$\leq [-f_i(x_{t+1}) + u_{i,t}(x_{t+1})]^+ + [-u_{i,t}(x_{t+1})]^+, \quad (\text{B.4c})$$

$$= [-f_i(x_{t+1}) + u_{i,t}(x_{t+1})]^+, \quad (\text{B.4d})$$

$$\leq [u_{i,t}(x_{t+1}) - l_{i,t}(x_{t+1})]^+, \quad (\text{B.4e})$$

$$= w_{i,t}(x_{t+1}), \quad (\text{B.4f})$$

where (B.4d) follows from feasibility of x_{t+1} for the CUQB subproblem (B.2d). \square

B.3. Proof of Lemma 3

From [39, Lemma 5.1], the following concentration inequality must hold for each element of the random vector X

$$\Pr\{|X_j - \mu_{X,j}| \leq \beta^{1/2} \sigma_{X,j}\} \geq 1 - e^{-\beta/2} = 1 - \frac{\alpha}{2m}. \quad (\text{B.5})$$

where the latter equality follows by substituting $\beta = 2 \log(2m/\alpha)$. Applying Boole's inequality across $j \in \mathbb{N}_1^m$, we have

$$\Pr\{|X_j - \mu_{X,j}| \leq \beta^{1/2} \sigma_{X,j}, \forall j \in \mathbb{N}_1^d\} \geq 1 - \frac{\alpha}{2}. \quad (\text{B.6})$$

From the Lipschitz continuity of g , we have that $g(x) - g(\mu) \leq L_g \sum_{j=1}^m |x_j - \mu_{X,j}|$ holds for any x , which can be combined with (B.6) to yield

$$\Pr\{g(X) - g(\mu) \leq L_g \beta^{1/2} \sum_{j=1}^m \sigma_{X,j}\} \geq 1 - \frac{\alpha}{2}. \quad (\text{B.7})$$

We can rearrange this expression to be in terms of Y and compare it to the definition of the quantile function

$$\Pr\{Y \leq Q_Y(1 - \frac{\alpha}{2})\} = 1 - \frac{\alpha}{2} \leq \Pr\{Y \leq g(\mu) + L_g \beta^{1/2} \sum_{j=1}^m \sigma_{X,j}\}. \quad (\text{B.8})$$

Since these events are one-sided, this can be true if and only if

$$Q_Y(1 - \frac{\alpha}{2}) \leq g(\mu) + L_g \beta^{1/2} \sum_{j=1}^m \sigma_{X,j}. \quad (\text{B.9})$$

We can repeat a similar analysis for the lower bound to find

$$Q_Y(\frac{\alpha}{2}) \geq g(\mu) - L_g \beta^{1/2} \sum_{j=1}^m \sigma_{X,j}. \quad (\text{B.10})$$

Subtracting (B.10) from (B.9) then yields the stated result. \square

B.4. Proof of Theorem 1

Using previous results, we can establish the following sequence of inequalities for the cumulative regret

$$R_T = \sum_{t=0}^{T-1} r_t, \quad (\text{B.11a})$$

$$\leq \sum_{t=0}^{T-1} r_t^+ = R_T^+, \quad (\text{B.11b})$$

$$\leq \sum_{t=0}^{T-1} w_{0,t}(x_{t+1}), \quad (\text{B.11c})$$

$$\leq \sum_{t=0}^{T-1} 2L_0 \beta_{t+1}^{1/2} \sum_{j=1}^m \sigma_{j,t}(x_{t+1}), \quad (\text{B.11d})$$

$$\leq 2L_0 \beta_T^{1/2} \sum_{j=1}^m \sum_{t=0}^{T-1} \sigma_{j,t}(x_{t+1}), \quad (\text{B.11e})$$

$$\leq 4L_0 \beta_T^{1/2} \sum_{j=1}^m \sqrt{(T+2)\gamma_{j,T}} = 4L_0 \beta_T^{1/2} \Psi_T. \quad (\text{B.11f})$$

The third and fourth line follow from Lemmas 2 and 3, respectively. The fifth line follows from monotonicity of the sequence $\{\beta_t^{1/2}\}_{t \geq 1}$ and the final line follows from Lemma 4 and the definition of Ψ_T . We can repeat a similar procedure for the cumulative violation of constraint i to show

$$V_{i,T} = \sum_{t=0}^{T-1} v_{i,t+1}, \quad (\text{B.12a})$$

$$\leq \sum_{t=0}^{T-1} w_{i,t}(x_{t+1}), \quad (\text{B.12b})$$

$$\leq \sum_{t=0}^{T-1} 2L_i \beta_{t+1}^{1/2} \sum_{j=1}^m \sigma_{j,t}(x_{t+1}), \quad (\text{B.12c})$$

$$\leq 2L_i \beta_T^{1/2} \sum_{j=1}^m \sum_{t=0}^{T-1} \sigma_{j,t}(x_{t+1}), \quad (\text{B.12d})$$

$$\leq 4L_i \beta_T^{1/2} \sum_{j=1}^m \sqrt{(T+2)\gamma_{j,T}} = 4L_i \beta_T^{1/2} \Psi_T. \quad (\text{B.12e})$$

Lastly, note $\beta_T^{1/2}$ grows logarithmically with T and Ψ_T is a finite summation over terms $\mathcal{O}(\sqrt{\gamma_{j,T} T})$ such that the claimed scaling laws must hold true, which completes the proof. \square

B.5. Proof of Theorem 2

Consider the following sum

$$\sum_{t=0}^{T-1} \left(r^+(x_{t+1}) + \sum_{i=1}^n v_i(x_{t+1}) \right) = R_T^+ + \sum_{i=1}^n V_{i,T} \leq 4\mathcal{L} \beta_T^{1/2} \Psi_T, \quad (\text{B.13})$$

The minimum of a sequence of points must be less than or equal to their average

$$\begin{aligned} \min_{t \in \{1, \dots, T\}} \left(r^+(x_{t+1}) + \sum_{i=1}^n v_i(x_{t+1}) \right) &\leq \frac{1}{T} \sum_{t=0}^{T-1} \left(r^+(x_{t+1}) + \sum_{i=1}^n v_i(x_{t+1}) \right), \quad (\text{B.14}) \\ &\leq \frac{4\mathcal{L} \beta_T^{1/2} \Psi_T}{T}. \end{aligned}$$

This implies that there must exist a point $\tilde{x}_T \in \{x_1, \dots, x_T\}$ that satisfies

$$r^+(\tilde{x}_T) + \sum_{i=1}^n v_i(\tilde{x}_T) \leq \frac{4\mathcal{L} \beta_T^{1/2} \Psi_T}{T}. \quad (\text{B.15})$$

Since each of these terms is non-negative, the bound must hold for all terms, i.e.,

$$r(\tilde{x}_T) \leq r^+(\tilde{x}_T) \leq \frac{4\mathcal{L}\beta_T^{1/2}\Psi_T}{T}, \quad (\text{B.16a})$$

$$v_i(\tilde{x}_T) \leq \frac{4\mathcal{L}\beta_T^{1/2}\Psi_T}{T}, \quad \forall i \in \mathbb{N}_1^n. \quad (\text{B.16b})$$

The stated result then immediately follows. \square

B.6. Proof of Theorem 3

The lower bound holds since $f_0(x) - \rho \sum_{i=1}^n [-f_i(x)]^+ \leq f_0^*$ for all $x \in \mathcal{X}$ by the exact penalty property for any $\rho \geq \bar{\rho}$. To derive the upper bound, we first define a quantity that we refer to as instantaneous penalty-based regret

$$r_{EP}(x_t; \rho) = r_t + \rho \sum_{i=1}^n v_{i,t} = f_0^* - f_0(x_t) + \rho \sum_{i=1}^n [-f_i(x_t)]^+. \quad (\text{B.17})$$

Next, we define a conservative (pessimistic) estimate of $r_{EP}(x_t; \rho)$ by replacing the true function with their lower quantile bounds

$$\bar{r}_{EP}(x_t; \rho) = f_0^* - l_{0,t-1}(x_t) + \rho \sum_{i=1}^n [-l_{i,t-1}(x_t)]^+, \quad (\text{B.18})$$

such that $r_{EP}(x_t; \rho) \leq \bar{r}_{EP}(x_t; \rho)$ for any choice of x_t with probability $\geq 1 - \delta$ by Lemma 1. Following the same logic used in Theorem 1, we can derive

$$\sum_{t=1}^T \bar{r}_{EP}(x_t; \rho) \leq 4\mathcal{L}_r(\rho)\beta_T^{1/2}\Psi_T, \quad (\text{B.19})$$

since the bounds established on r_{t+1} and $v_{i,t+1}$ in Lemma 2 also hold for their pessimistic counterparts. The final step is to recognize that

$$r_{EP}(\tilde{x}_T^r; \rho) \leq \bar{r}_{EP}(\tilde{x}_T^r; \rho) = \min_{t \in \{1, \dots, T\}} \bar{r}_{EP}(x_t; \rho) \leq \frac{1}{T} \sum_{t=1}^T \bar{r}_{EP}(x_t; \rho), \quad (\text{B.20})$$

where $\tilde{x}_T^r = x_{t^*}$ with index t^* given by (14). The upper bound then follows by combining (B.19) and (B.20) and substituting the definition of $r_{EP}(\tilde{x}_T^r; \rho)$. \square

B.7. Proof of Corollary 1

The following bounds must hold with probability $\geq 1 - \delta$

$$l_{i,t}(x) \leq f_i(x) \leq u_{i,t}(x), \quad \forall i \in \mathbb{N}_0^n, \forall t \geq 0, \forall x \in \mathcal{X}, \quad (\text{B.21})$$

based on Lemma 1. By definition, an exact penalty function for (1) must satisfy

$$f_0(x) - \rho \sum_{i=1}^n [-f_i(x)]^+ - f_0^* \leq 0, \quad \forall x \in \mathcal{X}, \quad (\text{B.22})$$

for some $\rho \geq 0$. We can construct an upper bound for the left-hand side of this inequality by individually bounding each term in (B.22) using (B.21) as follows

$$\begin{aligned} f_0(x) &\leq u_{0,0}(x), & \forall x \in \mathcal{X}, \\ [-f_i(x)]^+ &\geq [-u_{i,0}(x)]^+, & \forall x \in \mathcal{X}, \forall i \in \mathbb{N}_1^n, \\ f_0^* &= \max_{x \in \mathcal{X}, f_i(x) \geq 0, \forall i \in \mathbb{N}_1^n} \geq l_{0,0}^* = \max_{x \in \mathcal{X}, l_{i,0}(x) \geq 0, \forall i \in \mathbb{N}_1^n} l_{0,0}(x), \end{aligned}$$

which also hold with probability $\geq 1 - \delta$. Using these inequalities, we can establish

$$(f_0(x) - \rho \sum_{i=1}^n [-f_i(x)]^+ - f_0^*) \leq (u_{0,0}(x) - \rho \sum_{i=1}^n [-u_{i,0}(x)]^+ - l_{0,0}^*), \quad (\text{B.23})$$

for all $x \in \mathcal{X}$, meaning (37) is a sufficient condition for the exact penalty property (B.22) to hold, which completes the proof. \square

B.8. Proof of Theorem 4

The proof follows the same line of reasoning as [32, Theorem 5.1]. Suppose that infeasibility has not been declared up until step T and let i be in the index of the infeasible constraint. By Lemma 1, we have $f_i(x_{t+1}) \in [l_{i,t}(x_{t+1}), u_{i,t}(x_{t+1})]$ for all $t \in \mathbb{N}_0^{T-1}$ with probability at least $1 - \delta$. Then, since infeasibility has yet to be declared, we must have

$$u_{i,t}(x_{t+1}) \geq 0, \quad \forall t \in \mathbb{N}_0^{T-1}, \quad (\text{B.24})$$

and

$$l_{i,t}(x_{t+1}) \leq f_i(x_{t+1}) \leq \max_{x \in \mathcal{X}} f_i(x) = -\epsilon, \quad (\text{B.25})$$

with probability $\geq 1 - \delta$. We can use these to lower bound the sum

$$\sum_{t=0}^{T-1} (u_{i,t}(x_{t+1}) - l_{i,t}(x_{t+1})) \geq \sum_{t=0}^{T-1} \epsilon = T\epsilon. \quad (\text{B.26})$$

Combining Lemmas 3 and 4, we can upper bound this sum as follows

$$\sum_{t=0}^{T-1} (u_{i,t}(x_{t+1}) - l_{i,t}(x_{t+1})) \leq \sum_{t=0}^{T-1} 2L_i \beta_{t+1}^{1/2} \sum_{j=1}^m \sigma_{j,t}(x_{t+1}), \quad (\text{B.27a})$$

$$\leq 2L_i \beta_T^{1/2} \sum_{j=1}^m \sum_{t=0}^{T-1} \sigma_{j,t}(x_{t+1}), \quad (\text{B.27b})$$

$$\leq 4L_i \beta_T^{1/2} \sum_{j=1}^m \sqrt{(T+2)\gamma_{j,T}}, \quad (\text{B.27c})$$

where the expression for $\beta_T^{1/2}$ is given in Theorem 1. Therefore, we have

$$T\epsilon \leq 4L_i \beta_T^{1/2} \sum_{j=1}^m \sqrt{(T+2)\gamma_{j,T}}, \quad (\text{B.28})$$

which implies

$$\epsilon \leq \frac{4L_i \beta_T^{1/2} \sum_{j=1}^m \sqrt{(T+2)\gamma_{j,T}}}{T} = \mathcal{O}\left(\sqrt{\frac{\log(T)\gamma_T^{\max}}{T}}\right), \quad (\text{B.29})$$

since $\beta_T = \mathcal{O}(\log(T))$ by definition. The inequality (B.29) implies there exists a constant $\tilde{C} > 0$ such that

$$\epsilon \leq \tilde{C} \sqrt{\frac{\log(T)\gamma_T^{\max}}{T}}, \quad (\text{B.30})$$

which can be rearranged, with $C = 1/\tilde{C}$, as follows

$$C\epsilon \leq \sqrt{\frac{\log(T)\gamma_T^{\max}}{T}}. \quad (\text{B.31})$$

However, the worst-case MIG satisfies $\lim_{T \rightarrow \infty} \frac{\log(T)\gamma_T^{\max}}{T} = 0$ by assumption such that (B.31) will be violated when T is large enough. Thus, infeasibility will be declared on or before \bar{T} , which is the first iteration that (B.31) is violated. \square

C. Appendix: Unconstrained Synthetic Test Problems

C.1. Booth

The Booth problem is defined by the following functions

$$g_0(x, y) = -(y_1 + (2x_1 + x_2 - 5)^2).$$

$$h(x) = (x_1 + 2x_2 - 7)^2.$$

$$-10 \leq x_i \leq 10, \quad \forall i = 1, 2.$$

The global maximum is 0 with $x^* = [1, 3]^\top$.

C.2. Wolfe

The Wolfe problem is defined by the following functions

$$g_0(x, y) = -\left(\frac{4}{3}y_1 + x_3\right).$$

$$h(x) = (x_1^2 + x_2^2 - x_1 x_2)^{0.75}.$$

$$0 \leq x_i \leq 2, \quad \forall i = 1, \dots, 3.$$

The global maximum is 0 with $x^* = [1, 1, 1]^\top$.

C.3. Rastrigin

The Rastrigin problem is defined by the following functions

$$g_0(x, y) = -(y_1 + y_2 + 30 + x_3^2 - 10 \cos(2\pi x_3)).$$

$$h(x) = \begin{bmatrix} x_1^2 - 10 \cos(2\pi x_1) \\ x_2^2 - 10 \cos(2\pi x_2) \end{bmatrix}.$$

$$-5 \leq x_i \leq 5, \quad \forall i = 1, \dots, 3.$$

The global maximum is 0 with $x^* = [0, 0, 0]^\top$.

C.4. Colville

The Colville problem is defined by the following functions

$$\begin{aligned} g_0(x, y) = & -\left(y_1 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) \right. \\ & \left. + 19.8(x_2 - 1)(x_4 - 1) \right). \end{aligned}$$

$$h(x) = 100(x_1^2 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^2.$$

$$-10 \leq x_i \leq 10, \quad \forall i = 1, \dots, 4.$$

The global maximum is 0 with $x^* = [1, 1, 1, 1]^\top$.

C.5. Friedman

The Friedman problem is defined by the following functions

$$g_0(x, y) = - (10y_1 + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5).$$

$$h(x) = \sin(\pi x_1 x_2).$$

$$0 \leq x_i \leq 1, \quad \forall i = 1, \dots, 5.$$

The global maximum is 27.5 with $x^* = [x_1^*, x_2^*, 0.5, -1.5, -1.5]^\top$ for any x_1^* and x_2^* satisfying $\sin(\pi x_1^* x_2^*) = -1$.

C.6. Dolan

The Dolan problem is defined by the following functions

$$g_0(x, y) = - (y_1 - y_2 + 0.2x_5^2 - x_2 - 1).$$

$$h(x) = \begin{bmatrix} (x_1 + 1.7x_2) \sin(x_1) \\ 1.5x_3 - 0.1x_4 \cos(x_5 + x_4 - x_1) \end{bmatrix}.$$

$$-100 \leq x_i \leq 100, \quad \forall i = 1, \dots, 5.$$

The global maximum is 529.87 with $x^* = [98.964, 100, 100, 99.224, -0.25]^\top$.

C.7. Rosenbrock

The Rosenbrock problem is defined by the following functions

$$g_0(x, y) = - \left(\sum_{i=1}^3 (100y_i^2 + (1 - x_i)^2) + 100(x_5 - x_4^2) + y_4 + 100(x_6 - x_5^2) + (1 - x_5)^2 \right).$$

$$h(x) = \begin{bmatrix} x_2^2 - x_1^2 \\ x_3^2 - x_2^2 \\ x_4^2 - x_3^2 \\ (1 - x_4)^2 \end{bmatrix}.$$

$$-2 \leq x_i \leq 2, \quad \forall i = 1, \dots, 6.$$

The global maximum is 0 with $x^* = [0, 0, 0, 0, 0, 0]^\top$.

C.8. Zakharov

The Zakharov problem is defined by the following functions

$$g_0(x, y) = - \left(\sum_{i=1}^7 x_i^2 + \sum_{i=1}^7 (0.5ix_i)^2 + y_1 \sum_{i=1}^7 (0.5ix_i)^2 \right).$$

$$h(x) = \sum_{i=1}^7 (0.5ix_i)^2.$$

$$-5 \leq x_i \leq 10, \quad \forall i = 1, \dots, 7.$$

The global maximum is 0 with $x^* = [0, 0, 0, 0, 0, 0, 0]^\top$.

C.9. Powell

The Powell problem is defined by the following functions

$$g_0(x, y) = - \left(y_1 + (x_5 + 10x_6)^2 + y_2 + 5(x_7 - x_8)^2 + (x_2 - 2x_3)^4 + y_3 + 10(x_1 - x_4)^4 + y_4 \right).$$

$$h(x) = \begin{bmatrix} (x_1 + 10x_2)^2 \\ 5(x_3 - x_4)^2 \\ (x_6 - 2x_7)^4 \\ 10(x_5 - x_8)^4 \end{bmatrix}.$$

$$-4 \leq x_i \leq 5, \quad \forall i = 1, \dots, 8.$$

The global maximum is 0 with $x^* = [0, 0, 0, 0, 0, 0, 0, 0]^\top$.

C.10. Styblinski-Tang

The Styblinski-Tang problem is defined by the following functions

$$g_0(x, y) = - \left(\sum_{i=1}^4 y_i + \sum_{i=5}^9 (0.5x_i^4 - 16x_i^2 + 5x_i) \right).$$

$$h(x) = \begin{bmatrix} 0.5(x_1^4 - 16x_1^2 + 5x_1) \\ 0.5(x_2^4 - 16x_2^2 + 5x_2) \\ 0.5(x_3^4 - 16x_3^2 + 5x_3) \\ 0.5(x_4^4 - 16x_4^2 + 5x_4) \end{bmatrix}.$$

$$-5 \leq x_i \leq 5, \quad \forall i = 1, \dots, 9.$$

The global maximum is 352.49 with $x^* = -2.904[1, 1, 1, 1, 1, 1, 1, 1, 1]^\top$.

D. Appendix: Constrained Synthetic Test Problems

D.1. Bazaraa

The Bazaraa problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= -(2x_1^2 + 2x_2^2 - y_2), \\ g_1(x, y) &= -(5x_1 + x_2 - 5), \\ g_2(x, y) &= -(y_1 - x_1). \end{aligned}$$

$$h(x) = \begin{bmatrix} 2x_2^2 \\ 2x_1x_2 + 6x_1 + 4x_2 \end{bmatrix}.$$

$$0.01 \leq x_i \leq 1, \quad \forall i = 1, 2.$$

The global maximum is 6.613 with $x^* = [0.868, 0.659]^\top$.

D.2. Spring

The Spring problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= -(2y_1 + 2x_3), \\ g_1(x, y) &= -(2x_2^2 - x_1), \\ g_2(x, y) &= -\left(\frac{4x_2^2 - x_1x_2}{12566(x_1^3x_2 - x_1^4)} + \frac{1}{5108x_1x_2 - 1}\right), \\ g_3(x, y) &= -\left(1 - 140.45\frac{x_1}{y_2}\right), \\ g_4(x, y) &= -\left(\frac{2}{3}(x_1 + x_2) - 1\right). \end{aligned}$$

$$h(x) = \begin{bmatrix} x_1^2x_2 \\ x_2^3x_3 \end{bmatrix}.$$

$$\begin{aligned} 0.05 \leq x_1 &\leq 2, \\ 0.25 \leq x_2 &\leq 1.3 \\ 2 \leq x_3 &\leq 15. \end{aligned}$$

The global maximum is -0.0127 with $x^* = [0.052, 0.357, 11.289]^\top$.

D.3. Ex314

The Ex314 problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= y_2, \\ g_1(x, y) &= -x_1y_1 - 2x_2^2 + 2x_1x_2 + 2x_2x_3 - 2x_1x_3 - 2x_3^2 + 20x_1 - 9x_2 + 13x_3 - 24, \\ g_2(x, y) &= x_1 + x_3 + x_3 - 4, \\ g_3(x, y) &= 3x_2 + x_6 - 6. \end{aligned}$$

$$h(x) = \begin{bmatrix} 4x_1 - 2x_2 + 2x_3 \\ x_2 - x_3 - 2x_1 \end{bmatrix}.$$

$$\begin{aligned} -2 \leq x_1 &\leq 2, \\ 0 \leq x_2 &\leq 6 \\ -3 \leq x_3 &\leq 3. \end{aligned}$$

The global maximum is 4 with $x^* = [0.5, 0.0, 3.0]^\top$.

D.4. Rosen-Suzuki

The Rosen-Suzuki problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= -(x_1^2 + x_2^2 + x_4^2 - 5x_1 - 5x_2 + y_1), \\ g_1(x, y) &= 8 - x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4, \\ g_2(x, y) &= 10 - x_1^2 - 2x_2^2 - y_2 + x_1 + x_4, \\ g_3(x, y) &= 5 - 2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4. \end{aligned}$$

$$h(x) = \begin{bmatrix} 2x_3^2 - 21x_3 + 7x_4 \\ x_3^2 + 2x_4^2 \end{bmatrix}.$$

$$-2 \leq x_i \leq 2, \quad \forall i = 1, \dots, 4.$$

The global maximum is 44 with $x^* = [0, 1, 2, -1]^\top$.

D.5. st_bpv1

The st_bpv1 problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= -(y_1 + x_2 x_4), \\ g_1(x, y) &= -(30 - y_2), \\ g_2(x, y) &= -(20 - y_3), \\ g_3(x, y) &= -(x_3 + x_4 - 15). \end{aligned}$$

$$h(x) = \begin{bmatrix} x_1 x_3 \\ x_1 + 3x_2 \\ 2x_1 + x_2 \end{bmatrix}.$$

$$\begin{aligned} 0 \leq x_1 &\leq 27, \\ 0 \leq x_2 &\leq 16, \\ 0 \leq x_3 &\leq 10, \\ 0 \leq x_4 &\leq 10. \end{aligned}$$

The global maximum is -10 with $x^* = [27, 1, 0, 10]^\top$.

D.6. Ex211

The Ex211 problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= -(42x_1 - 50y_1 + 44x_2 + 45x_3 + 47x_4 + 47.5x_5), \\ g_1(x, y) &= -(20x_1 + y_2 + 4x_5 - 39). \end{aligned}$$

$$h(x) = \begin{bmatrix} x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \\ 12x_2 + 11x_3 + 7x_4 \end{bmatrix}.$$

$$0 \leq x_i \leq 1, \quad \forall i = 1, \dots, 5.$$

The global maximum is 17 with $x^* = [1, 1, 0, 1, 0]^\top$.

D.7. Ex212

The Ex212 problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= 10x_6 + y_1 + 0.5(x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2), \\ g_1(x, y) &= -(6x_1 + 3x_2 + 3x_3 + 2x_4 + x_5 - 6.5), \\ g_2(x, y) &= -(y_2 - 20). \end{aligned}$$

$$h(x) = \begin{bmatrix} 10.5x_1 + 7.5x_2 + 3.5x_3 + 2.5x_4 + 1.5x_5 \\ 10x_1 + 10x_3 + x_6 \end{bmatrix}.$$

$$0 \leq x_i \leq 30, \quad \forall i = 1, \dots, 6.$$

The global maximum is 213 with $x^* = [0, 1, 0, 1, 1, 20]^\top$.

D.8. g09

The g09 problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= -y_1 - x_3^4 - 3(x_4 - 11)^2 - 10x_5^6 - 7x_6^2 - x_7^4 + 4x_6x_7 + 10x_6 + 8x_7, \\ g_1(x, y) &= 127 - 2x_1x_2 - y_2 - 5x_5, \\ g_2(x, y) &= 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5, \\ g_3(x, y) &= 196 - 23x_1 + x_2^2 - 6x_6^2 + 8x_7, \\ g_4(x, y) &= -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7. \end{aligned}$$

$$h(x) = \begin{bmatrix} (x_1 - 10)^2 + 5(x_2 - 12)^2 \\ 3x_2^4 + x_3 + 4x_4^2 \end{bmatrix}.$$

$$-10 \leq x_i \leq 10, \quad \forall i = 1, \dots, 7.$$

The global maximum is -680.63 with $x^* = [2.33, 1.95, -0.48, 4.37, -0.62, 1.04, 1.59]^\top$.

D.9. Ex724

The Ex724 problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= -(y_3 + 0.4(x_2/x_8)^{0.67} - x_1 + 10), \\ g_1(x, y) &= -(0.0588x_5x_7 + 0.1x_1 - 1), \\ g_2(x, y) &= -(0.0588x_6x_8 + 0.1x_1 + 0.1x_2 - 1), \\ g_3(x, y) &= -(4(x_3/x_5) + 2/y_1 + 0.0588(x_7/x_3)^{1.3} - 1), \\ g_4(x, y) &= -(y_2 + 0.0588x_4^{1.3}x_8 - 1). \end{aligned}$$

$$h(x) = \begin{bmatrix} x_3^{0.71}x_5 \\ 4(x_4/x_6) + 2/(x_4^{0.71}x_6) \\ 0.4(x_1/x_7)^{0.67} - x_2 \end{bmatrix}.$$

$$0.1 \leq x_i \leq 10, \quad \forall i = 1, \dots, 8.$$

The global maximum is -3.92 with $x^* = [6.35, 2.34, 0.67, 0.53, 5.95, 5.32, 1.04, 0.42]^\top$.

D.10. Ex216

The Ex216 problem is defined by the following functions

$$\begin{aligned} g_0(x, y) &= 48x_1 - 0.5y_1 - 50x_5^2 - 50x_6^2 - 50x_7^2 - 50x_8^2 - 50x_9^2 - 50x_{10}^2 \\ &\quad + 42x_2 + y_3 + 47x_7 + 42x_8 + 45x_9 + 46x_{10}, \\ g_1(x, y) &= y_2 - 2x_7 - 6x_8 - 2x_9 - 2x_{10} + 4, \\ g_2(x, y) &= 6x_1 - 5x_2 + 8x_3 - 3x_4 + x_6 + 3x_7 + 8x_8 + 9x_9 - 3x_{10} - 22, \\ g_3(x, y) &= -5x_1 + 6x_2 + 5x_3 + 3x_4 + 8x_5 - 8x_6 + 9x_7 + 2x_8 - 9x_{10} + 6, \\ g_4(x, y) &= y_4 + 3x_7 - 9x_8 - 9x_9 - 3x_{10} + 23, \\ g_5(x, y) &= -8x_1 + 7x_2 - 4x_3 - 5x_4 - 9x_5 + x_6 - 7x_7 - x_8 + 3x_9 - 2x_{10} + 12. \end{aligned}$$

$$h(x) = \begin{bmatrix} 100x_1^2 + 100x_2^2 + 100x_3^2 + 100x_4^2 \\ -2x_1x_2 - x_3 - 3x_5 - 3x_6 \\ 48x_3 + 45x_4 + 44x_5 + 41x_6 \\ 9x_1 + 5x_2 - 9x_4 + x_5 - 8x_6 \end{bmatrix}.$$

$$0 \leq x_i \leq 1, \quad \forall i = 1, \dots, 10.$$

The global maximum is 39 with $x^* = [1, 0, 0, 1, 1, 0, 1, 1, 1]^\top$.

References

- [1] D. Piga, M. Forgione, S. Formentin, A. Bemporad, Performance-oriented model learning for data-driven MPC design, *IEEE control systems letters* 3 (3) (2019) 577–582.
- [2] J. A. Paulson, A. Mesbah, Data-driven scenario optimization for automated controller tuning with probabilistic performance guarantees, *IEEE Control Systems Letters* 5 (4) (2020) 1477–1482.
- [3] F. Sorourifar, G. Makrygirgos, A. Mesbah, J. A. Paulson, A data-driven automatic tuning method for MPC under uncertainty using constrained Bayesian optimization, *IFAC-PapersOnLine* 54 (3) (2021) 243–250.
- [4] E. A. del Rio Chanona, P. Petsagkourakis, E. Bradford, J. A. Graciano, B. Chachuat, Real-time optimization meets Bayesian optimization and derivative-free optimization: A tale of modifier adaptation, *Computers & Chemical Engineering* 147 (2021) 107249.
- [5] J. Wu, S. Toscano-Palmerin, P. I. Frazier, A. G. Wilson, Practical multi-fidelity bayesian optimization for hyperparameter tuning, in: *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, Vol. 115 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 788–798.
- [6] I. Kapetanovic, Computer-aided drug discovery and development (CADD): in silico-chemico-biological approach, *Chemico-biological Interactions* 171 (2) (2008) 165–176.
- [7] S. Ju, T. Shiga, L. Feng, Z. Hou, K. Tsuda, J. Shiomi, Designing nanostructures for phonon transport via Bayesian optimization, *Physical Review X* 7 (2) (2017) 021024.
- [8] A. M. Schweidtmann, A. D. Clayton, N. Holmes, E. Bradford, R. A. Bourne, A. A. Lapkin, Machine learning meets continuous flow chemistry: Automated optimization towards the Pareto front of multiple objectives, *Chemical Engineering Journal* 352 (2018) 277–282.
- [9] J. Vrugt, J. W. Hopmans, J. Šimunek, Calibration of a two-dimensional root water uptake model, *Soil Science Society of America Journal* 65 (4) (2001) 1027–1037.
- [10] L. Schultz, V. Sokolov, Bayesian optimization for transportation simulators, *Procedia Computer Science* 130 (2018) 973–978.
- [11] J. Paulson, M. Martin-Casas, A. Mesbah, Fast uncertainty quantification in dynamic flux balance analysis models using sparse multi-element polynomial chaos, in: *2019 AIChE Annual Meeting*, AIChE, 2019.

- [12] L. M. Rios, N. V. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, *Journal of Global Optimization* 56 (3) (2013) 1247–1293.
- [13] R. Eberhart, J. Kennedy, Particle swarm optimization, in: *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, Citeseer, 1995, pp. 1942–1948.
- [14] N. Hansen, S. D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evolutionary Computation* 11 (1) (2003) 1–18.
- [15] D. M. Mukhopadhyay, M. O. Balitanas, A. Farkhod, S.-H. Jeon, D. Bhattacharyya, Genetic algorithm: A tutorial review, *International Journal of Grid and Distributed Computing* 2 (3) (2009) 25–32.
- [16] J. A. Nelder, R. Mead, A simplex method for function minimization, *The Computer Journal* 7 (4) (1965) 308–313.
- [17] T. G. Kolda, R. M. Lewis, V. Torczon, Optimization by direct search: New perspectives on some classical and modern methods, *SIAM Review* 45 (3) (2003) 385–482.
- [18] D. R. Jones, C. D. Perttunen, B. E. Stuckman, Lipschitzian optimization without the lipschitz constant, *Journal of Optimization Theory and Applications* 79 (1) (1993) 157–181.
- [19] R. Chen, M. Menickelly, K. Scheinberg, Stochastic optimization using a trust-region method and random models, *Mathematical Programming* 169 (2018) 447–487.
- [20] F. E. Curtis, V. Kungurtsev, D. P. Robinson, Q. Wang, A stochastic-gradient-based interior-point algorithm for solving smooth bound-constrained optimization problems, *arXiv preprint arXiv:2304.14907* (2023).
- [21] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. De Freitas, Taking the human out of the loop: A review of Bayesian optimization, *Proceedings of the IEEE* 104 (1) (2015) 148–175.
- [22] P. I. Frazier, A tutorial on Bayesian optimization, *arXiv preprint arXiv:1807.02811* (2018).
- [23] Y. Sui, A. Gotovos, J. Burdick, A. Krause, Safe exploration for optimization with Gaussian processes, in: *International Conference on Machine Learning*, PMLR, 2015, pp. 997–1005.
- [24] D. Bergmann, K. Graichen, Safe Bayesian optimization under unknown constraints, in: *In Proceedings of the Conference on Decision and Control*, IEEE, 2020, pp. 3592–3597.
- [25] F. Berkenkamp, A. Krause, A. P. Schoellig, Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics, *Machine Learning* (2021) 1–35.
- [26] D. Krishnamoorthy, F. J. Doyle, Safe Bayesian optimization using interior-point methods—applied to personalized insulin dose guidance, *IEEE Control Systems Letters* 6 (2022) 2834–2839.

[27] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, J. P. Cunningham, Bayesian optimization with inequality constraints., in: International Conference on Machine Learning, Vol. 2014, 2014, pp. 937–945.

[28] V. Picheny, R. B. Gramacy, S. Wild, S. L. Digabel, Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 1443–1451.

[29] M. J. Sasena, P. Papalambros, P. Goovaerts, Exploration of metamodeling sampling criteria for constrained global optimization, *Engineering Optimization* 34 (3) (2002) 263–278.

[30] R. Priem, N. Bartoli, Y. Diouane, A. Sgueglia, Upper trust bound feasibility criterion for mixed constrained bayesian optimization with application to aircraft design, *Aerospace Science and Technology* 105 (2020) 105980.

[31] C. Lu, J. A. Paulson, No-regret Bayesian optimization with unknown equality and inequality constraints using exact penalty functions, *IFAC-PapersOnLine* 55 (7) (2022) 895–902.

[32] W. Xu, Y. Jiang, C. N. Jones, Constrained efficient global optimization of expensive black-box functions, *arXiv preprint arXiv:2211.00162* (2022).

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

[34] J. P. Eason, L. T. Biegler, A trust region filter method for glass box/black box optimization, *AICHE Journal* 62 (9) (2016) 3124–3136.

[35] B. Beykal, F. Boukouvala, C. A. Floudas, E. N. Pistikopoulos, Optimal design of energy systems using constrained grey-box multi-objective optimization, *Computers & Chemical Engineering* 116 (2018) 488–502.

[36] I. Bajaj, S. S. Iyer, M. F. Hasan, A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point, *Computers & Chemical Engineering* 116 (2018) 306–321.

[37] S. H. Kim, F. Boukouvala, Surrogate-based optimization for mixed-integer non-linear problems, *Computers & Chemical Engineering* 140 (2020) 106847.

[38] J. A. Paulson, C. Lu, COBALT: COnstrained Bayesian optimizAtion of computationally expensive grey-box models exploiting derivative information, *Computers & Chemical Engineering* 160 (2022) 107700.

[39] N. Srinivas, A. Krause, S. M. Kakade, M. Seeger, Gaussian process optimization in the bandit setting: No regret and experimental design, in: International Conference on Machine Learning, 2015, pp. 2171–2180.

[40] M. Blondel, O. Teboul, Q. Berthet, J. Djolonga, Fast differentiable sorting and ranking, in: International Conference on Machine Learning, PMLR, 2020, pp. 950–959.

[41] C. K. Williams, C. E. Rasmussen, Gaussian Processes for Machine Learning, Vol. 2, MIT Press Cambridge, MA, 2006.

[42] H. Liu, J. Cai, Y.-S. Ong, Remarks on multi-output Gaussian process regression, *Knowledge-Based Systems* 144 (2018) 102–121.

- [43] D. Eriksson, M. Poloczek, Scalable constrained Bayesian optimization, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 730–738.
- [44] W. J. Maddox, M. Balandat, A. G. Wilson, E. Bakshy, Bayesian optimization with high-dimensional outputs, *Advances in Neural Information Processing Systems* 34 (2021) 19274–19287.
- [45] R. Astudillo, P. I. Frazier, Thinking inside the box: A tutorial on grey-box Bayesian optimization, in: 2021 Winter Simulation Conference, IEEE, 2021, pp. 1–15.
- [46] J. Moćkus, On Bayesian methods for seeking the extremum, in: Optimization Techniques IFIP Technical Conference, Springer, 1975, pp. 400–404.
- [47] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, *Journal of Global Optimization* 13 (4) (1998) 455.
- [48] J. T. Wilson, R. Moriconi, F. Hutter, M. P. Deisenroth, The reparameterization trick for acquisition functions, *arXiv preprint arXiv:1712.00424* (2017).
- [49] Y. Zhang, X. Zhang, P. Frazier, Constrained two-step look-ahead Bayesian optimization, *Advances in Neural Information Processing Systems* 34 (2021) 12563–12575.
- [50] R. Astudillo, P. Frazier, Bayesian optimization of composite functions, in: International Conference on Machine Learning, PMLR, 2019, pp. 354–363.
- [51] S. R. Chowdhury, A. Gopalan, On kernelized multi-armed bandits, in: International Conference on Machine Learning, PMLR, 2017, pp. 844–853.
- [52] S. Vakili, K. Khezeli, V. Picheny, On information gain and regret bounds in Gaussian process bandits, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 82–90.
- [53] T. G. Epperly, E. N. Pistikopoulos, A reduced space branch and bound algorithm for global optimization, *Journal of Global Optimization* 11 (3) (1997) 287.
- [54] V. J. Bowman, Permutation polyhedra, *SIAM Journal on Applied Mathematics* 22 (4) (1972) 580–589.
- [55] K. Wang, B. Wilder, S.-c. Suen, B. Dilkina, M. Tambe, Improving GP-UCB algorithm by harnessing decomposed feedback, in: *Machine Learning and Knowledge Discovery in Databases*, Springer, 2020, pp. 555–569.
- [56] M. Wüthrich, B. Schölkopf, A. Krause, Regret bounds for Gaussian-process optimization in large domains, *Advances in Neural Information Processing Systems* 34 (2021) 7385–7396.
- [57] K. Chaloner, I. Verdinelli, Bayesian experimental design: A review, *Statistical Science* (1995) 273–304.
- [58] M. Subotic, M. Tuba, N. Stanarevic, Different approaches in parallelization of the artificial bee colony algorithm, *International Journal of Mathematical Models and Methods in Applied Sciences* 5 (4) (2011) 755–762.
- [59] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimisation problems, *International Journal of Mathematical Modelling and Numerical Optimisation* 4 (2) (2013) 150–194.

[60] L. A. Rastrigin, Systems of extremal control, Nauka (1974).

[61] S. Rahnamayan, H. R. Tizhoosh, M. M. Salama, A novel population initialization method for accelerating evolutionary algorithms, *Computers & Mathematics with Applications* 53 (10) (2007) 1605–1614.

[62] R. B. Gramacy, H. K. Lee, Cases for the nugget in modeling computer experiments, *Statistics and Computing* 22 (2012) 713–722.

[63] H. Rosenbrock, An automatic method for finding the greatest or least value of a function, *The Computer Journal* 3 (3) (1960) 175–184.

[64] R. Chelouah, P. Siarry, Tabu search applied to global optimization, *European Journal of Operational Research* 123 (2) (2000) 256–270.

[65] M. Laguna, R. Marti, Experimental testing of advanced scatter search designs for global optimization of multimodal functions, *Journal of Global Optimization* 33 (2005) 235–255.

[66] M. Styblinski, T.-S. Tang, Experiments in nonconvex optimization: stochastic approximation with function smoothing and simulated annealing, *Neural Networks* 3 (4) (1990) 467–483.

[67] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, 2013.

[68] S. R. Nekoo, J. Á. Acosta, A. Ollero, A search algorithm for constrained engineering optimization and tuning the gains of controllers, *Expert Systems with Applications* 206 (2022) 117866.

[69] C. A. Floudas, P. M. Pardalos, C. Adjiman, W. R. Esposito, Z. H. Gümüs, S. T. Harding, J. L. Klepeis, C. A. Meyer, C. A. Schweiger, *Handbook of test problems in local and global optimization*, Vol. 33, Springer Science & Business Media, 2013.

[70] W. Hock, K. Schittkowski, Test examples for nonlinear programming codes, *Journal of Optimization Theory and Applications* 30 (1) (1980) 127–129.

[71] M. Tawarmalani, N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, Vol. 65, Springer Science & Business Media, 2013.

[72] D. Karaboga, B. Akay, A modified artificial bee colony (ABC) algorithm for constrained optimization problems, *Applied Soft Computing* 11 (3) (2011) 3021–3031.

[73] W. Huyer, A. Neumaier, SNOBFIT–stable noisy optimization by branch and fit, *ACM Transactions on Mathematical Software* 35 (2) (2008) 1–25.

[74] scikit-quant (2023).
URL <https://github.com/scikit-quant/scikit-quant>

[75] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, *Nature Methods* 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.

[76] M. J. Powell, The BOBYQA algorithm for bound constrained optimization without derivatives, Cambridge NA Report NA2009/06, University of Cambridge, Cambridge 26 (2009).

[77] C. Cartis, J. Fiala, B. Marteau, L. Roberts, Improving the flexibility and robustness of model-based derivative-free optimization solvers, *ACM Transactions on Mathematical Software (TOMS)* 45 (3) (2019) 1–41.

[78] N. Hansen, Y. Akimoto, P. Baudis, CMA-ES/pycma on Github, Zenodo, DOI:10.5281/zenodo.2559634 (Feb. 2019). doi:10.5281/zenodo.2559634.
URL <https://doi.org/10.5281/zenodo.2559634>

[79] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, E. Bakshy, BoTorch: A framework for efficient Monte-Carlo Bayesian optimization, *Advances in Neural Information Processing Systems* 33 (2020) 21524–21538.

[80] J. Larson, M. Menickelly, S. M. Wild, Derivative-free optimization methods, *Acta Numerica* 28 (2019) 287–404.

[81] C. Lu, J. A. Paulson, Constrained Upper Quantile Bound (2023).
URL https://github.com/PaulsonLab/Constrained_Upper_Quantile_Bound

[82] C. Zhu, R. H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization, *ACM Transactions on Mathematical Software* 23 (4) (1997) 550–560.

[83] J. J. Moré, S. M. Wild, Benchmarking derivative-free optimization algorithms, *SIAM Journal on Optimization* 20 (1) (2009) 172–191.

[84] N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, P. Mugunthan, Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation, *Journal of Computational and Graphical Statistics* 17 (2) (2008) 270–294.

[85] D. F. Mendoza, J. E. A. Graciano, F. dos Santos Laporace, G. A. C. Le Roux, Assessing the reliability of different real-time optimization methodologies, *The Canadian Journal of Chemical Engineering* 94 (3) (2016) 485–497.

[86] F. Boukouvala, C. A. Floudas, ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems, *Optimization Letters* 11 (2017) 895–913.

[87] M. Tawarmalani, N. V. Sahinidis, A polyhedral branch-and-cut approach to global optimization, *Mathematical programming* 103 (2) (2005) 225–249.

[88] R. Misener, C. A. Floudas, ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations, *Journal of Global Optimization* 59 (2-3) (2014) 503–526.

[89] A. Cozad, N. V. Sahinidis, D. C. Miller, Learning surrogate models for simulation-based optimization, *AIChE Journal* 60 (6) (2014) 2211–2227.

[90] A. M. Schweidtmann, D. Bongartz, D. Grothe, T. Kerkenhoff, X. Lin, J. Nadjman, A. Mitsos, Deterministic global optimization with Gaussian processes embedded, *Mathematical Programming Computation* 13 (3) (2021) 553–581.