Two-step lookahead Bayesian optimization with inequality constraints

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent advances in computationally efficient non-myopic Bayesian optimization offer improved query efficiency over traditional myopic methods like expected improvement, with only a modest increase in computation cost. These advances have been largely limited to the problems without constraints, however, with the one existing multi-step lookahead constrained BO method [1] relying on computationally expensive unreliable brute-force derivative-free optimization of a Monte Carlo rollout acquisition function. Methods that use the reparameterization trick for more efficient derivative-based optimization of non-myopic acquisition functions in the unconstrained setting, like sample average approximation and infinitesimal perturbation analysis, do not extend: constraints introduce discontinuities in the sampled acquisition function surface. Moreover, we argue here that being non-myopic is even more important in constrained problems because fear of violating constraints pushes myopic methods away from sampling the boundary between feasible and infeasible regions, slowing the discovery of optimal solutions with tight constraints. In this paper, we propose a computationally efficient two-step lookahead constrained Bayesian optimization acquisition function (2-OPT-C) supporting both sequential and batch settings. To enable fast acquisition function optimization, we develop a novel likelihood-ratio based unbiased estimator of the gradient of the two-step optimal acquisition function that does not use the reparameterization trick. In numerical experiments, 2-OPT-C typically improves query efficiency by 2x or more over previous methods, and in some cases by 10x or more.

1 Introduction

We consider constrained optimization of a continuous black-box function f under continuous black-box constraints g_i, $\min_{x \in \mathbb{A}} f(x)$ subject to $g_i(x) \leq 0$, $i = 1, \ldots, I$, within the compact design space $\mathbb{A} \subseteq \mathbb{R}^d$. We suppose both $f(x)$ and $g_i(x)$ are derivative-free, time-consuming-to-evaluate and also noise-free. Such problems arise, for example, in tuning hyperparameters of machine learning models subject to runtime or fairness constraints and policy optimization in RL with safety constraints.

Bayesian optimization (BO) has proven successful at solving black-box optimization problems with expensive objectives [2, 3], including constrained problems of the form above. BO methods for constrained problems include constrained expected improvement (EIC as in [4], rediscovered by [5]), constrained BO with stepwise uncertainty reduction [6], predictive entropy search with unknown constraints [7], Alternating Direction Method of Multipliers Bayesian optimization [8], constrained BO with max-value entropy search [9], and augmented Lagrangian techniques that convert the constrained problems into a sequence of unconstrained ones [10]. It also includes methods designed specifically for equality and mixed constraints [11], for batch observations [12] and for problems with high dimensions [13].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

All of these existing methods, however, for constrained Bayesian optimization (CBO), are myopic. (Notable exceptions are [1, 14], discussed below.). They only consider the immediate improvement in solution quality resulting from a function evaluation and ignore how what we will do in the future will affect this value. This greedy behavior may hinder an algorithm's ability to find good solutions efficiently. While a recent flurry of activity is addressing this issue for unconstrained problems [15, 16, 17, 18, 19], and performance improvements from non-myopic algorithms in unconstrained settings suggest that non-myopic BO is promising for constraints as well, substantial non-myopic development has not reached the constrained setting. Moreover, we argue below that being non-myopic provides even more value in constrained settings than it does in unconstrained ones. This is born out in our numerical experiments, which show several-fold improvement over the state-of-the-art in some problems.

One existing non-myopic constrained BO method [1] first formulates constrained BO within a dynamic programming (DP) framework. However, this original DP problem is intractable. To circumvent the issue, rollout, an approximate DP technique, is applied to solving the problem. This approach requires an extremely large amount of computation to approximate the multi-step lookahead policy well, especially in problems with more than a few dimensions. This limits its applicability.

The constrained multi-information source BO method recently proposed by [14] is also non-myopic. As their focus is on the multi-information source setting, their method assumes the ability to evaluate the objective and constraint in a decoupled fashion. Indeed, it uses a multi-information source version of the two-step lookahead acquisition function proposed by [14] to select the point at which to query the objective function and (separately) a one-step lookahead information gain acquisition function to select the point at which to query constraints. In contrast, our applications of interest typically compute the objective and constraints simultaneously with a single code. For example, when tuning ML hyperparameters to maximize accuracy subject to a model execution time constraint, the marginal cost of evaluating accuracy is negligible given that we are already planning to evaluate model execution time and must incur the cost of training. Thus, a method that evaluates the objective and constraints in a decoupled way essentially throws information away in such settings.

Our Contributions. We provide a novel non-myopic computationally efficient method for batch CBO. It substantially outperforms myopic CBO methods. In relation to [1] it requires substantially less computation to decide where to sample and its query efficiency is substantially better in high-dimensional problems, and is at least as good in low-dimensional ones.

The key to our approach is a new method for optimizing stochastic acquisition functions, leveraging the likelihood ratio method [20]. Standard efficient approaches to optimizing stochastic non-myopic acquisition functions, such as IPA [21] or the one-shot method [22] rely on a sampled acquisition function surface created using the reparameterization trick. In CBO, however, this surface is discontinuous, preventing the efficient use of these methods. Our novel approach is potentially generalizable to other settings where such discontinuities prevent the use of IPA and one-shot optimization.

Our work function builds on the unconstrained two-step optimal method [17], overcoming substantial computational difficulties requiring created by constraints' inherent discontinuities. These difficulties require abandoning the IPA approach used in [17] and instead developing a new likelihood-ratio-based approach.

Through the rest of this paper, §2 reviews myopic CBO methods, Gaussian processes and the standard constrained expected improvement (EIC). §3 defines notations and proposes our new lookahead acquisition function 2-OPT-C for batch CBO. §4 discusses the discontinuity issue introduced by reparameterization trick in the constrained setting. §5 shows how to efficiently optimize 2-OPT-C via a gradient estimator obtained from the likelihood ratio method. §6 compares the empirical performance of 2-OPT-C against various CBO algorithms on a collection of problems wildly used in CBO literature. §7 concludes the paper.

2 Background

We briefly review the literature on myopic CBO. We then briefly state well-known results and notation regarding Gaussian processes and CBO, including the widely used myopic method EIC and its batch version, which will be needed later.

2.1 Myopic Constrained Bayesian Optimization

This work builds on the larger literature on myopic CBO. [5] proposes constrained expected improvement, a constraint-weighted expected improvement acquisition function which multiplies the expected improvement with the probability of feasibility associated with each constraint, rediscovering an approach due to [4]. [23] proposes an approach in which the point to sample is found by maximizing EIC and then a decision is made whether to evaluate the objective or a constraint based on the information gain. [6] proposes a stepwise uncertainty reduction method in which the acquisition function aims to maximally decrease our uncertainty on the location of the optimizer with a single evaluation. Predictive entropy search with constraints (PESC) [7], an extension of predictive entropy search [24] is another information-gain based approach. It chooses the point to evaluate by approximating the expected information gain on the value of the constrained minimizer. [10] proposes a hybrid approach combining the expected improvement with an augmented Lagrangian framework. [11] extends this technique by introducing an alternative slack variable formulation that handles equality and mixed constraints. Integrated expected conditional improvement (IECI) [25] proposes a new acquisition function that integrates a conditional improvement with respect to a reference point over the design space. [12] develops a quasi-Monte Carlo approximation of expected improvement under batch optimization with noisy observations and noisy constraints. [9] modifies the mutual information criterion of max-value entropy search [26] and extends to the constrained setting with the ability to handle both continuous and binary constraints. [8] leverages the ADMM framework to convert constrained problems into multiple unconstrained subproblems by introducing auxiliary variables for each constraint, then solves the subproblems using standard BO. Similar to TuRBO [27], [13] proposes an acquisition function for constrained optimization that scales to high dimensions by maintaining and adjusting trust regions.

2.2 Constrained Expected Improvement

We briefly review constrained expected improvement (EIC) [5, 4], introducing notation used later. Suppose we have independent Gaussian process (GP) priors on f and g and observations $f(D) := \{f(x) : x \in D\}$ and $g(D) := \{g(x) : x \in D\}$ at a collection of datapoints $D = \{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(n)}\}$. Let f^* be the best point observed so far subject to our constraints, $f^* = \min_{x \in D, g(x) \leq 0} f(x)$. The constrained expected improvement of x is:

$$\text{EIC}(x) = \mathbb{E}\left[[f^* - f(x)]^+ \cdot \mathbb{1}\{g(x) \leq 0\}\right], \tag{1}$$

where $\mathbb{E}[\cdot]$ is the expectation taken with respect to the posterior given D, $f(D)$, and $g(D)$. If the prior on f and g are independent, as assumed in [5], then

$$\text{EIC}(x) = \mathbb{E}\left[f^* - f(x)\right]^+ \mathbb{E}\left[\mathbb{1}\{g(x) \leq 0\}\right] = \text{EI}(x) \cdot \text{PF}(x),$$

where $\text{EI}(x)$ and $\text{PF}(x)$ are the expected improvement and the probability of feasibility at x respectively, and both have analytic forms. We refer readers to [28] and [5] for more details.

We now define EIC in the batch setting, building on the first discussion of which we are aware [12]. Let $\mathbf{X} = \{x^{(n+1)}, x^{(n+2)}, \dots, x^{(n+q)}\}$ be the batch of q candidate points that we consider evaluating next. Then the *constrained multi-points expected improvement* $\text{EIC}(\mathbf{X})$ is,

$$\text{EIC}(\mathbf{X}) = \mathbb{E}\left[\max_{x \in \mathbf{X}} \ (f^* - f(x))^+ \cdot \mathbb{1}\{g(x) \leq 0\}\right] \tag{2}$$

3 Constrained Two-step Acquisition Function

We define our two-step optimal acquisition function for CBO, called 2-OPT-C, for both sequential and batch settings, and then discuss computational optimization in §5. In this acquisition function, we find the point to measure now that is optimal under the assumption that we will be able to measure one additional point before being awarded the value of the best point found over two steps. In other words, it looks two steps ahead instead of the one-step ahead considered by myopic methods. For clarity, we assume a single constraint and treat multiple constraints in the supplement.

We first define notation. We use subscripts $\{0, 1, 2\}$ to denote stages. Stage 0 is the current stage when we will evaluate a batch of points. In stage 1, we will evaluate one additional point based on the

results of stage 0 and then in stage 2 will be judged based on the best of these points evaluated. We put independent GP priors on f and g. Let D be the collection of data points that we have observed so far. Then given the information in D and their corresponding objective $f(D)$ and constraint $g(D)$ values, the posterior distribution on the objective f is a Gaussian process with a mean function μ_0 and kernel K_0. Similarly, the posterior distribution on the constraint g is also a Gaussian process with a mean function μ_0^c and kernel K_0^c. $\mathbb{E}_0$ denotes taking the expectation with respect to the posterior distributions given D.

Let f_0^* be the best evaluated point satisfying the constraint so far, i.e. $f_0^* = \min_{x \in D, g(x) \leq 0} f(x)$. Let X_1 be the set of q candidate points that we consider evaluating at the first stage and let $Y_f = \{f(x) : x \in X_1\}$ and $Y_g = \{g(x) : x \in X_1\}$ be the sets of corresponding function values and constraint values respectively. We let μ_1, μ_1^c and K_1, K_1^c denote the mean function and kernel for the posterior distributions of f and g respectively given D and X_1. Let $\sigma_1(x) = \sqrt{K_1(x,x)}$ and $\sigma_1^c(x) = \sqrt{K_1^c(x,x)}$. Let $\mathbb{E}_1$ indicate the expectation with respect to the corresponding Gaussian processes given both D and X_1. Finally, we use x_2 to denote a single point to be evaluated in the second stage, based on the results of the first.

Let f_1^* and f_2^* be the best evaluated feasible point by the end of the first and second stage, respectively.

Following the principle of dynamic programming, our goal is to choose X_1 to minimize the overall expected objective $\mathbb{E}_1(f_2^*)$. This is done under the assumption that x_2 will be chosen optimally leveraging observations of X_1. Equivalently, our goal is to maximize

$$\mathbb{E}_0\left[\max_{x_2}[f_0^* - f_2^*]\right] = \mathbb{E}_0[f_0^* - f_1^* + \max_{x_2}\mathbb{E}_1[f_1^* - f_2^*]],$$

over X_1 chosen in the first stage. $\mathbb{E}_1[f_1^* - f_2^*]$ depends implicitly on the information obtained from X_1, which is Y_f and Y_g, and is included in the posterior over which $\mathbb{E}_1$ is taken.

Thus, we define the constrained two-step acquisition function:

$$\text{2-OPT-C}(X_1) := \mathbb{E}_0\left[f_0^* - f_1^* + \max_{x_2 \in A(\delta)} \mathbb{E}_1\left[f_1^* - f_2^*\right]\right]$$

where $A(\delta)$ is a compact subset of A consisting of points at least δ away from sampled points in $D \cup X_1$. With $\delta = 0$, $A = A(\delta)$. We introduce the parameter $\delta \geq 0$ purely to overcome a technical hurdle in our theoretical analysis: that the standard deviation of the posterior distribution is not smooth at sampled points. We believe δ can be set to 0 in practice. Indeed, the theoretical analysis (Theorem 1) allows setting δ at any arbitrary small positive value.

For use in §5, we derive a more directly computable expression for $\text{2-OPT-C}(X_1)$. We rewrite

$$\text{2-OPT-C}(X_1) = \mathbb{E}_0\left[\max_{x_2 \in A(\delta)}\left[f_0^* - f_1^* + \mathbb{E}_1\left[f_1^* - f_2^*\right]\right]\right] := \mathbb{E}_0\left[\max_{x_2 \in A(\delta)} \alpha(X_1, x_2, Y)\right]$$

where $Y = (Y_f, Y_g) \sim p(y; X_1)$ and $p(y; X_1)$ is the distribution of Y given $f(D)$ and $g(D)$, specified explicitly as

$$p(y; X_1) = \mathcal{N}\left(\begin{bmatrix}\mu_0(X_1)\\ \mu_0^c(X_1)\end{bmatrix}, \begin{bmatrix}\sigma_0^2(X_1) & 0\\ 0 & (\sigma_0^c(X_1))^2\end{bmatrix}\right). \tag{3}$$

Then, $\alpha(X_1, x_2, Y)$ can be written in closed form:

$$\alpha(X_1, x_2, Y) = f_0^* - f_1^* + \text{EI}(f_1^* - \mu_1(x_2), \sigma_1(x_2)^2) \cdot \text{PF}(\mu_1(x_2)^c, (\sigma_1(x_2)^c)^2),$$

where $\text{EI}(m, v) := m\Phi(m/\sqrt{v}) + \sqrt{v}\varphi(m/\sqrt{v})$ and $\text{PF}(m^c, v^c) = \Phi\left(-m^c/\sqrt{v^c}\right)$.

The Importance of Being Non-Myopic in CBO Here we give intuition why the acquisition function just described is substantially better than the myopic approach. We also argue that being non-myopic is even more important in constrained problems than it is in unconstrained ones.

Suppose the (unobserved) best feasible point is at or near where the constraint is tight. To find this point with as few function evaluations as possible, the algorithms need to efficiently learn the feasible

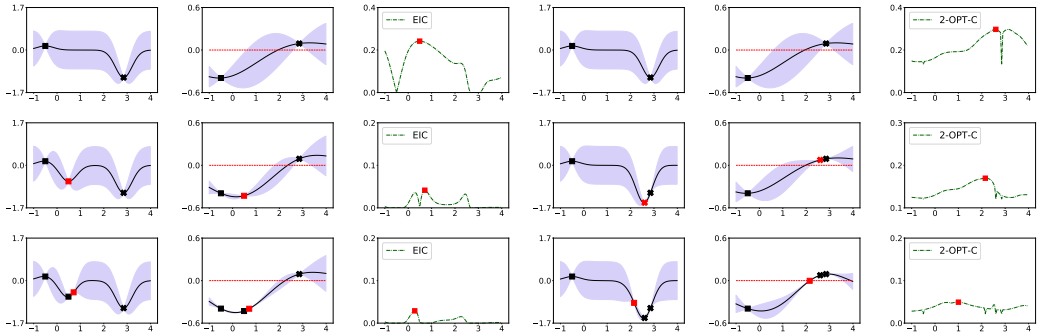

Figure 1: EIC and 2-OPT-C optimizing a 1-d synthetic function. The first three graphs of each row are the posterior on f, the posterior on g, and the acquisition function under EIC, and the last three graphs are the corresponding quantities under 2-OPT-C. Each row shows one iteration, proceeding from the first iteration at the top to the last iteration at the bottom. In the plots of the posterior on f and g, "x" denotes infeasible points and a square denotes feasible points. The constraint threshold (0) is plotted as a red line. The red dot is the point newly sampled in the current iteration. 2-OPT-C explores potentially infeasible points more aggressively and discovers a better feasible point.

region's boundaries. This happens most quickly when evaluating points close to this boundary. Since the boundary is uncertain, this requires an algorithm to be willing to evaluate points that may be infeasible. However, the value of such points to myopic CBO algorithms like EIC is significantly reduced because the probability of infeasibility is high — typically near 50%. Thus, myopic CBO algorithms may insufficiently explore the boundary, instead sticking well inside the feasible region.

Figure 1 illustrates this with an example. On the first iteration, EIC evaluates a point in a region around an existing feasible point that is likely feasible, getting a small one-step improvement but little information on f and g for future evaluations. However, 2-OPT-C explores more aggressively. The point it evaluates is not feasible but provides a significant amount of information about the boundary between the feasible and infeasible region. As a result, after progressing two steps, 2-OPT-C locates a new feasible point ($g(x) = -0.02$) with a much lower objective function value while EIC remains stuck near the previous feasible point.

4 Discontinuities Hinder Optimization via the Reparameterization Trick

The reparameterization trick [29] has been widely used in BO to construct gradient estimators to support optimization of multi-step lookahead acquisition functions, both together with infinitesimal perturbation analysis (IPA) [30, 17] and sample average approximation (SAA) [22]. It has also been used to obtain gradient estimators for one-step lookahead acquisition functions in [31]. However, it encounters a key difficulty in *constrained* BO. We describe this difficulty here.

To attempt to apply the reparameterization trick to optimize 2-OPT-C, first note that Y_f and Y_g can be written as deterministic functions of normal random variables: $Y_f = \mu_0(X_1) + R_0(X_1)Z_f$, $Y_g = \mu_0^c(X_1) + R_0^c(X_1)Z_g$, where Z_f, Z_g are two q-dimensional independent standard normal random variables and $R_0(X_1)$, $R_0^c(X_1)$ are the Cholesky decompositions of $K_0(X_1, X_1)$ and $K_0^c(X_1, X_1)$.

To optimize 2-OPT-C, we would replace the values of Y_f and Y_g in the definition of 2-OPT-C(X_1) with these expressions. We focus on the term $\mathbb{E}_0[f_0^* - f_1^*]$, since this is where the difficulty lies, and focus on the non-batch case for simplicity. We have:

$$\begin{aligned}\mathbb{E}_0[f_0^* - f_1^*] &= \mathbb{E}_0[(f_0^* - f(X_1))^+ \cdot \mathbb{1}\{g(X_1) \le 0\}] \\ &= \mathbb{E}_0[(f_0^* - \mu_0(X_1) - R_0(X_1)Z_f)^+ \cdot \mathbb{1}\{\mu_0^c(X_1) + R_0^c(X_1)Z_g \le 0\}].\end{aligned}$$

The key difficulty for both IPA and SAA arises because the term $\mathbb{1}\left\{\mu_0^c(X_1) + R_0^c(X_1)Z_g \le 0\right\}$ has a discontinuity as we vary X_1, at points where $\mu_0^c(X_1) + R_0^c(X_1)Z_g$ is 0.

We describe this difficulty, beginning with IPA. To use IPA to estimate the gradient of 2-OPT-C in support of a gradient-based optimization method, one would first sample Z_f and Z_g and then use the

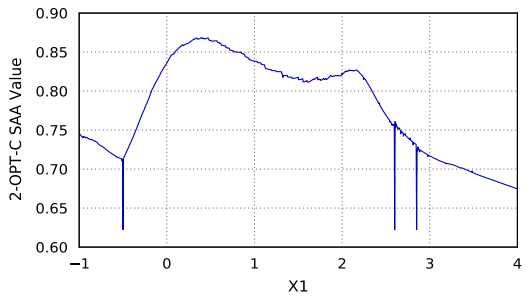

Figure 2: SAA approximation to 2-OPT-C(X_1) using 256 base samples, plotting the approximation vs. X_1 in a 1d problem with one constraint. Computation grows as dimensions, constraints, or samples are added. As sample size increases, jumps shrink in size but grow in number. The surface's roughness (zoom to see it better) prevents efficient optimization with more than a few dimensions or constraints.

gradient with respect to X_1 of the expression inside the expectation, $(f_0^* - \mu_0(X_1) - R_0(X_1)Z_f)^+ \cdot \mathbb{1}\{\mu_0^c(X_1) + R_0^c(X_1)Z_g \leq 0\}$, as an estimator of $\nabla_{X_1}\mathbb{E}_0[f_0^* - f_1^*]$.

Unfortunately, however, the term $\mathbb{1}\{\mu_0^c(X_1)+R_0^c(X_1)Z_g \leq 0\}$ is discontinuous, taking two different constant values for different values of X_1. While modifying X_1 affects the probability of feasibility, the derivative of this term with respect to X_1 will be 0 (as long as X_1 avoids the boundary between feasible and infeasible regions). This causes the gradient estimator to be biased, preventing its use for optimizing the acquisition function.

The discontinuity in $\mathbb{1}\{\mu_0^c(X_1) + R_0^c(X_1)Z_g \leq 0\}$ also creates a problem for SAA. SAA optimizes Monte Carlo acquisition functions by first approximating them using randomly sampled "base samples" and then optimizing this approximation while holding these base samples fixed. In our context, to optimize our acquisition function $\mathbb{E}_0[f_0^* - f_1^*]$, SAA would sample standard normal random vectors $Z_{f,m}$ and $Z_{g,m}$ of appropriate dimension and then optimize the following over X_1:

$$\frac{1}{M}\sum_{m=1}^{M}\left[(f_0^* - \mu_0(X_1) - R_0(X_1)Z_{f,m})^+ \cdot \mathbb{1}\{\mu_0^c(X_1) + R_0^c(X_1)Z_{g,m} \leq 0\}\right]. \tag{4}$$

Unfortunately, the discontinuity in $\mathbb{1}\{\mu_0^c(X_1) + R_0^c(X_1)Z_g \leq 0\}$ causes (4) to be discontinuous, as shown in Fig 2, making it difficult to optimize. To combat these discontinuities, one approach is to average over a larger number of samples M, paying a higher computational cost. This shrinks the jumps but increases their number. Many jumps (even small ones) make gradient-based methods fail. Instead, SAA for constrained problems requires derivative-free methods. Derivative-free methods are much slower for acquisition-function optimization than derivative-based ones ([22] Appendix F, [31] Figs 3 & 4), especially for the problems with higher dimensions or more constraints. In numerical experiments §6.1, we compare with an SAA-based implementation of two-step lookahead and find that it requires more computation and offers degraded query efficiency compared to our method.

5 Optimizing 2-OPT-C Using Likelihood Ratios

We develop a novel approach for optimizing stochastic acquisition functions like 2-OPT-C, overcoming the challenges created by discontinuities. This approach uses the likelihood ratio method to derive an unbiased estimator of the gradient for 2-OPT-C. To the best of our knowledge, we are the first to adopt the likelihood ratio method to design a gradient estimator for BO algorithms.

We begin by rewriting the expression to be differentiated using importance sampling. In importance sampling, rather than sampling a random variable from its original distribution, we sample it from an importance sampling distribution and then use a likelihood ratio to compensate.

To construct an importance sampling distribution for Y, we choose a point $\widetilde{X}_1$ (or, if X_1 is a batch of points, $\widetilde{X}_1$ will be a batch of the same number of points). $\widetilde{X}_1$ is arbitrary for now but will be specified below. Our importance sampling distribution will be $Y \sim p(y; \widetilde{X}_1)$, where $p(y; \widetilde{X}_1)$ is defined by (3), substituting $\widetilde{X}_1$ for X_1. To compensate for the change in distribution from $p(y; X_1)$ to $p(y; \widetilde{X}_1)$, we use the likelihood ratio [20] defined by

$$L(y; X_1, \widetilde{X}_1) = \frac{p(y; X_1)}{p(y; \widetilde{X}_1)}.$$

Since $L(y; X_1, \widetilde{X}_1)p(y; \widetilde{X}_1) = p(y; X_1)$, we can rewrite 2-OPT-C using the likelihood ratio as:

$$\text{2-OPT-C}(X_1^\epsilon) = \int \max_{x_2 \in A(\delta)} \alpha(X_1^\epsilon, x_2, y) L(y; X_1^\epsilon, \widetilde{X}_1) p(y; \widetilde{X}_1)\, dy$$

where $X_1^\epsilon = X_1 + \epsilon$, ϵ is an arbitrary vector (later, we will differentiate with respect to ϵ at $\epsilon = 0$). This integral can be understood as the expectation of $\max_{x_2 \in A(\delta)} \alpha(X_1^\epsilon, x_2, Y) L(Y; X_1^\epsilon, \widetilde{X}_1)$, where Y is distributed according to $p(y; \widetilde{X}_1)$ rather than $p(y; X_1)$.

Critically, since $\widetilde{X}_1$ stays fixed as we vary ϵ, the distribution of Y also stays the same in this new alternate expression for 2-OPT-C. Moreover, for any given Y, the expression $\max_{x_2 \in A(\delta)} \alpha(X_1^\epsilon, x_2, Y) L(Y; X_1^\epsilon, \widetilde{X}_1)$ is continuous in ϵ. This is the key property that overcomes the previously described challenge created by discontinuities.

To create an unbiased estimator of the gradient of 2-OPT-C with this understanding and alternate expression, we first observe that Theorem 1 below allows us to interchange the integral and gradient:

$$\begin{aligned} \nabla_\epsilon \text{2-OPT-C}(X_1^\epsilon)\Big|_{\epsilon=0} &= \int \nabla_{\epsilon=0} \left[\max_{x_2 \in A(\delta)} \alpha(X_1^\epsilon, x_2, y) L(y; X_1^\epsilon, \widetilde{X}_1) p(y; \widetilde{X}_1) \right] dy \\ &= \int \Gamma(X_1, \widetilde{X}_1, y) p(y; \widetilde{X}_1) dy \end{aligned} \tag{5}$$

Here,

$$\begin{aligned} \Gamma(X_1, \widetilde{X}_1, y) &:= \nabla_{X_1} \left[\max_{x_2 \in A(\delta)} \alpha(X_1, x_2, y) L(y; X_1, \widetilde{X}_1) \right] \\ &= \left[\nabla_{X_1} \alpha(X_1, x_2^*, Y) \right] L(y; X_1, \widetilde{X}_1) + \alpha(X_1, x_2^*, Y) \left[\nabla_{X_1} p(y; X_1) \right] / p(y; \widetilde{X}_1) \end{aligned}$$

with $x_2^* \in \arg\max_{x_2 \in A(\delta)} \alpha(X_1, x_2, Y)$. The last equality is by the envelope theorem [32], i.e., $\max_{x_2 \in A(\delta)} \alpha(X_1, x_2, Y)$ can be differentiated with respect to X_1 by first optimizing over x_2 given X_1, then holding x_2^* fixed while differentiating with respect to X_1. From now on, we will drop the subscript of the differential operator ∇ for simplicity.

By (5), $\Gamma(X_1, \widetilde{X}_1, Y)$ is an unbiased estimator of ∇ 2-OPT-C(X_1) when Y is drawn according to $p(y; \widetilde{X}_1)$. With this stochastic gradient estimator, we then can use stochastic gradient ascent [33] with multiple restarts to find a collection of stationary values for X_1. Then we use simulation to evaluate 2-OPT-C(X_1) at these values and select the one with the highest estimated 2-OPT-C. This then provides a computationally efficient algorithm for optimizing 2-OPT-C. Pseudocode for using 2-OPT-C is provided in the supplement.

As discussed above, the key step to developing our unbiased estimator is to interchange the integral and gradient operator, which is justified by Theorem 1 below. Both a detailed set of regularity conditions (Assumption 1) and the proof are provided in the supplement. Although Theorem 1 assumes $\delta > 0$, in practice, we choose $\delta = 0$ in our stochastic gradient estimator.

Theorem 1. *Under Assumption 1 (in the supplement),* 2-OPT-C(X_1)*'s partial derivatives exist almost everywhere for any* $\delta > 0$*. When* 2-OPT-C(X_1) *is differentiable,*

$$\nabla \text{2-OPT-C}(X_1) = \int \Gamma(X_1, \widetilde{X}_1, y) p(y; \widetilde{X}_1) dy$$

Choice of Importance Sampling Distribution We have constructed an estimator of the gradient ∇ 2-OPT-C at X_1. This estimator was constructed using an importance sampling distribution parameterized by $\widetilde{X}_1$. We are free to choose this $\widetilde{X}_1$ as we wish. We recommend setting $\widetilde{X}_1$ equal to X_1, as our gradient estimator takes a particularly simple form and offers robust performance. In particular, when $\widetilde{X}_1 = X_1$, $L(y; X_1, X_1) = 1$, so our gradient estimator is

$$\Gamma(X_1, X_1, y) := \alpha(X_1, x_2^*, y) \left[\nabla p(y; X_1) \right] / p(y; X_1) + \left[\nabla \alpha(X_1, x_2^*, y) \right],$$

where $x_2^* \in \arg\max_{x_2} \alpha(X_1, x_2, y)$.

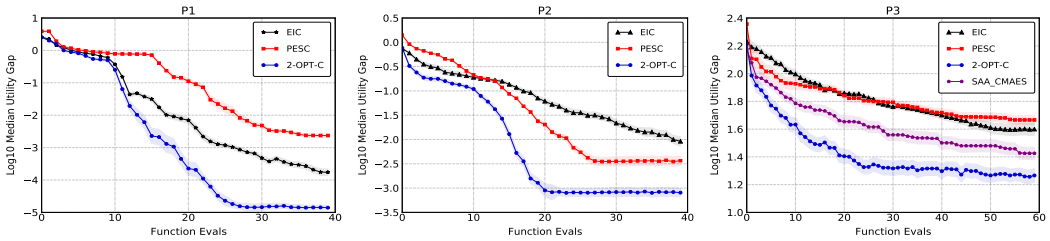

Figure 3: Log10 median utility gap of EIC, PESC, and 2-OPT-C with 95% confidence intervals for problems P1, P2, and P3.

6 Numerical Experiments

This section numerically investigates our algorithms on problems widely used as benchmarks in the constrained BO literature. Benchmarks demonstrate that 2-OPT-C provides significant improvements in query efficiency over state-of-the-art methods, often by a factor of 3 or more. In the main paper, we focus on query efficiency for the sequential setting. The supplement includes results incorporating the computational overhead of acquisition function optimization and batch evaluations.

We select as our benchmark problems three problems from [1], named P1, P2, and P3. We use these problems in a set of comparisons with myopic methods in §6.1 and with the non-myopic method from [1] in §6.2. Additional details of the experimental setup are in the supplement.

6.1 Comparison with Myopic Methods

We compare with the myopic methods EIC [5, 4] and PESC [7]. Results are shown in Figure 3.

2-OPT-C offers significantly lower median utility gaps than the myopic methods across all three problems. For example, in P1, 2-OPT-C quickly converges to a utility gap 10^{-5} (in 27 evaluations), while the best of the other benchmarks (EIC) has a utility gap of 10^{-3} at this many evaluations. Moreover, while EIC continues to improve its utility gap (down to 10^{-4}) with more iterations, PESC seems to reach a plateau after roughly 35 function evaluations. Comparisons between 2-OPT-C are qualitatively similar in P2 and P3, with 2-OPT-C outperforming the best of the other benchmarks in all three problems, offering a utility gap that is roughly $10^{0.5} = 2$ times lower than the better of PESC and EIC in these given problems.

Also, in P3, we additionally compare with an SAA implementation of 2-OPT-C that uses CMA-ES [34] to optimize the SAA to the acquisition function. We see that it outperforms EIC and PESC, supporting the value of two-step lookahead for CBO over myopic approaches, but underperforms our method due to the difficulty of optimizing the two-step lookahead acquisition function using SAA. It also requires substantially more computation. This is because, as we note in §4, discontinuities in the SAA to the acquisition function make it extremely difficult to optimize.

6.2 Comparison with the Non-myopic Method

We compare 2-OPT-C with the non-myopic rollout algorithm of [1] on the aforementioned three problems. We use the terms Rollout-1, Rollout-2, and Rollout-3 to denote this algorithm with horizons of 1,2, and 3 respectively.

As code was unavailable for this rollout algorithm, we extract experimental data from [1] and replicate their experimental setup, adding 2-OPT-C's results to their Figures 2 and 3. The setup in that paper, and thus the one we use here in §6.2 is slightly different from the more standard one used in §6.1. In particular, [1] penalizes infeasibility when the recommended point x_n is infeasible, while §6.1 uses the best feasible value observed so far.

Figure 4 reports the results of these comparisons. It shows that 2-OPT-C provides query efficiency that is at least as good as the non-myopic rollout methods with different horizons on all three problems and is sometimes substantially better. In detail, for P1, 2-OPT-C increases solution quality by more than a factor of 2 ($10^{-4.59}/10^{-4.92}$) compared to the best rollout strategies, i.e., Rollout-1. For P3,

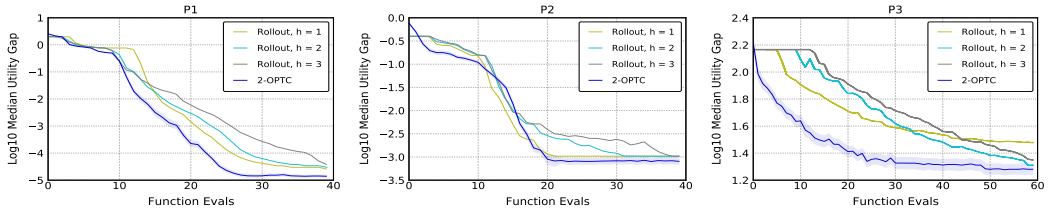

Figure 4: Log10 median utility gap of 2-OPT-C and the existing non-myopic constrained BO algorithm with different rollout horizons on P1, P2, and P3. The three lines of the rollout algorithm are extracted from Figure 3 in [1]. Additional information is provided in Table 1 in the supplement.

2-OPT-C is able to improve solution quality over the best competing rollout method by roughly a factor 2 when the evaluation budget is tight.

In addition, 2-OPT-C uses less computation than the rollout strategies in [1]. Since the code from [1] is not publicly available, we refer to the discussion of the computational time of the unconstrained version of the rollout strategies from [17], which we understand from private communication is similar to the one used in [1]. This states that the unconstrained rollout strategies require between 10 minutes and 1 hour to evaluate a single point even on a low-dimensional problem. This may be even larger in constrained settings, since we need a separate GP to model the constraint function. 2-OPT-C is more computationally efficient. On P3, for instance, 2-OPT-C optimizes a batch of 5 points in 25-30 minutes.

Rollout-1 is actually attempting to optimize the same theoretical acquisition function used by 2-OPT-C. Indeed, the rollout strategy used by [1] is EIC, which is optimal when it is used over a horizon of $h = 1$. Rollout-1 is then trying to identify the point to evaluate now that would be optimal if one additional point were evaluated afterward using EIC. If this point could be found exactly, it would be 2-step optimal.

Despite attempting to optimize an acquisition function that is conceptually the same as 2-OPT-C, the performance of Rollout-1 is substantially worse (the utility gap is roughly 2x larger over all budgets), despite requiring more than double the amount of computation. This is because [1] optimizes this two-step acquisition function without access to the type of derivative estimates provided by our likelihood ratio methodology.

7 Conclusion

This paper presents a non-myopic CBO algorithm, supporting both batch and sequential settings, that substantially improves both query efficiency and computation time over the one previous method focused on this class of problems. for both sequential and batch settings. This method overcomes a challenge in using the reparameterization trick, arising from discontinuities created by constraints. In spite of this challenge, it efficiently optimizes a two-step optimal acquisition function using a novel likelihood-ratio-based unbiased estimator of its gradient. Through numerical experiments, we demonstrate that the proposed algorithm substantially improves query efficiency compared with existing myopic and non-myopic methods.

While our method offers significant improvements in query efficiency compared to the state-of-the-art that more than offset its computational expense, there are likely more opportunities for reducing its computational cost. This, as well as a further theoretical investigation regarding our method providing such significant improvements in query efficiency compared to other methods, presents valuable directions for future work.

Faster constrained optimization of time-consuming-to-evaluate functions with our methodology enables creation of new engineering systems, better supervised learning methods, and lower-cost business operations. While we believe technological innovation is on balance good, new technology created using our methods also has the potential for negative effects. Understanding the societal effects of improved optimization capabilities is an important area for future research.

References

[1] Remi Lam and Karen Willcox. Lookahead bayesian optimization with inequality constraints. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1890–1900. Curran Associates, Inc., 2017.

[2] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 2951–2959. Curran Associates, Inc., 2012.

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

[4] Matthias Schonlau, William J. Welch, and Donald R. Jones. Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, 34:11–25, 1998.

[5] Jacob R. Gardner, Matt J. Kusner, Zhixiang Xu, Kilian Q. Weinberger, and John P. Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–937–II–945. JMLR.org, 2014.

[6] Victor Picheny. A Stepwise uncertainty reduction approach to constrained global optimization. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 787–795, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

[7] Jose Miguel Hernandez-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1699–1707, Lille, France, 07–09 Jul 2015. PMLR.

[8] Setareh Ariafar, Jaume Coll-Font, Dana Brooks, and Jennifer Dy. Admmbo: Bayesian optimization with unknown constraints using admm. *Journal of Machine Learning Research*, 20(123):1–26, 2019.

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

[10] Robert B Gramacy, Genetha A Gray, Sébastien Le Digabel, Herbert KH Lee, Pritam Ranjan, Garth Wells, and Stefan M Wild. Modeling an augmented lagrangian for blackbox constrained optimization. *Technometrics*, 58(1):1–11, 2016.

[11] Victor Picheny, Robert B Gramacy, Stefan Wild, and Sebastien Le Digabel. Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 1435–1443. Curran Associates, Inc., 2016.

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

[13] David Eriksson and Matthias Poloczek. Scalable constrained bayesian optimization. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 730–738. PMLR, 13–15 Apr 2021.

[14] Seyede Fatemeh Ghoreishi and Douglas Allaire. Multi-information source constrained bayesian optimization. *Structural and Multidisciplinary Optimization*, 59(3):977–991, 2019.

[15] Remi Lam, Karen Willcox, and David H. Wolpert. Bayesian optimization with a finite budget: An approximate dynamic programming approach. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 883–891. Curran Associates, Inc., 2016.

[16] Javier Gonzalez, Michael Osborne, and Neil Lawrence. Glasses: Relieving the myopia of bayesian optimisation. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 790–799, Cadiz, Spain, 09–11 May 2016. PMLR.

[17] Jian Wu and Peter Frazier. Practical two-step lookahead bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9810–9820. Curran Associates, Inc., 2019.

[18] Shali Jiang, Henry Chai, Javier Gonzalez, and Roman Garnett. BINOCULARS for efficient, nonmyopic sequential experimental design. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4794–4803. PMLR, 13–18 Jul 2020.

[19] Shali Jiang, Daniel Jiang, Maximilian Balandat, Brian Karrer, Jacob Gardner, and Roman Garnett. Efficient nonmyopic bayesian optimization via one-shot multi-step trees. *Advances in Neural Information Processing Systems*, 33, 2020.

[20] Søren Asmussen and Peter W. Glynn. Stochastic simulation - algorithms and analysis. In *Stochastic modeling and applied probability*, 2007.

[21] Jian Wu and Peter I. Frazier. The parallel knowledge gradient method for batch bayesian optimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 3134–3142, USA, 2016. Curran Associates Inc.

[22] Maximilian Balandat, Brian Karrer, Daniel R Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. *arXiv preprint arXiv:1910.06403*, 2019.

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

[24] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 918–926. Curran Associates, Inc., 2014.

[25] J Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West. Optimization under unknown constraints. *Bayesian Statistics*, 9(9):229, 2011.

[26] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3627–3635. PMLR, 06–11 Aug 2017.

[27] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[28] J. Mockus. On bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.

[29] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 9884–9895, 2018.

[30] Philip Heidelberger, Xi-Ren Cao, Michael Zazanis, and Rajan Suri. Convergence properties of infinitesimal perturbation analysis estimates. *Management Science*, 34:1281–1302, 11 1988.

[31] Jialei Wang, Scott C Clark, Eric Liu, and Peter I Frazier. Parallel bayesian global optimization of expensive functions. *arXiv preprint arXiv:1602.05149*, 2016.

[32] Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

[33] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[34] Nikolaus Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation*, pages 75–102, 2006.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See §7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See §7.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See the Supplement.
   (b) Did you include complete proofs of all theoretical results? [Yes] See the Supplement.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the Supplement.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the Supplement.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report error bars for all experiments that we ran. We also include data from [1] that do not include error bars.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the Supplement.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See the supplement.
   (b) Did you mention the license of the assets? [Yes] This is discussed in the supplement.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See the supplement for our code.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]