arXiv:2007.05554v2 [stat.ML] 16 Jul 2020

Bayesian Optimization of Risk Measures

Sait Cakmak[a], Raul Astudillo[b], Peter Frazier[b], and Enlu Zhou[a]
[a]School of Industrial and Systems Engineering, Georgia Institute of Technology
[b]School of Operations Research and Information Engineering, Cornell University

July 17, 2020

Abstract

We consider Bayesian optimization of objective functions of the form $\rho[F(x, W)]$, where F is a black-box expensive-to-evaluate function and ρ denotes either the VaR or CVaR risk measure, computed with respect to the randomness induced by the environmental random variable W. Such problems arise in decision making under uncertainty, such as in portfolio optimization and robust systems design. We propose a family of novel Bayesian optimization algorithms that exploit the structure of the objective function to substantially improve sampling efficiency. Instead of modeling the objective function directly as is typical in Bayesian optimization, these algorithms model F as a Gaussian process, and use the implied posterior on the objective function to decide which points to evaluate. We demonstrate the effectiveness of our approach in a variety of numerical experiments.

1 Introduction

Traditional Bayesian optimization (BO) has focused on problems of the form $\min_x F(x)$, or more generally, $\min_x \mathbb{E}\left[F(x, W)\right]$ where $F(x, W)$ is a time-consuming black box function that does not provide derivatives, and W is a random variable. This has seen enormous impact, and has expanded from hyper-parameter tuning [1, 2] to real-world decisions [3, 4, 5, 6].

However, for many truly high-stakes decisions the expectation is inappropriate: we must be risk-averse. In such settings, risk measures have become a crucial tool for quantifying risk. For instance, by law, banks are regulated using the Value-at-Risk [7]. Other examples where risk measures have been used include cancer treatment planning [8, 9], healthcare operations [10], natural resource management [11], and disaster management [12].

In this work, we consider risk averse optimization of the form $\min_x \rho\left[F(x, W)\right]$, where ρ is a *risk measure* that maps the probability distribution of $F(x, W)$ (induced by a random W and fixed F) onto a real number describing its level of risk.

When F is inexpensive and has convenient analytic structure, optimizing against a risk measure is well understood [13, 14, 15]. However, to the best of our knowledge, when F is *expensive-to-evaluate*, *derivative-free*, or is a *black box*, there are *no* existing methods able to tackle the problem. While one can sample from W and obtain a high-accuracy estimate of $\rho[F(x, W)]$, this typically requires a large number of samples. For example, if each evaluation of F requires 1 hour, and we need 10^2 samples to obtain a high-accuracy estimate of $\rho[F(x, W)]$, this would require more than 4 days per evaluation. Moreover, if we do not obtain a high-accuracy estimate, estimators are typically biased [16] in a way that is unaccounted for by traditional Gaussian process regression.

In this paper, we develop a novel approach that overcomes these challenges. We provide a new one-step Bayes optimal algorithm ρKG, and a fast approximation ρKG^{apx} that works well

in numerical experiments. This significantly improves the sampling efficiency over the state art BO methods. We develop asymptotically unbiased and consistent gradient estimators for efficient optimization of ρKG. To further improve the numerical efficiency, we introduce a novel two time scale optimization approach that is broadly applicable in many BO methods that require nested optimization.

We focus on two risk measures commonly used in practice: Value at Risk (VaR) and Conditional Value at Risk (CVaR). VaR and CVaR have been applied in a wide range of settings, including robust optimization [14], simulation optimization under input uncertainty [17], insurance [18] and risk management [19]. We refer the reader to [20] for a broader discussion on VaR and CVaR.

The remainder of this paper is organized as follows. Section 2 gives a brief background on risk measures and BO. Section 3 formally introduces the problem setting. Section 4 introduces the statistical model on F, a Gaussian process (GP) prior, and explains how to estimate VaR and CVaR of a GP. We introduce ρKG, a value of information type of acquisition function for optimization of VaR and CVaR in Section 5. A cheaper approximation ρKG^{apx} and efficient optimization of both algorithms is also discussed. Finally, Section 6 lists a number of problems that we consider for testing and demonstrating the performance of the algorithms developed here.

2 Background

2.1 Bayesian optimization

The BO problem has been studied extensively over the past 20 years. Popular BO methods include expected improvement [21], knowledge gradient [22], (predictive) entropy search [23]; can work on discrete [22] or continuous [24] domains; with noisy or noise-free observations; has been extended to utilize parallel evaluations [25, 26], multi-fidelity observations [27], and gradient observations [28]. We refer the reader to [29] for a review of BO methods.

Within the BO literature, a closely related work to ours is [30], where (1) is studied when ρ is the expectation operator, Due to linearity of the expectation, the GP prior on $F(x, w)$ translates into a GP prior on $\mathbb{E}[F(x, W)]$, a fact that is leveraged to efficiently compute the one-step Bayes optimal policy, also known as the knowledge gradient (KG) policy. Although it is not the focus of this study, since CVaR at risk level $\alpha = 0$ is the expectation, our methods can be directly applied for solving the expectation problem, and the resulting algorithm is equivalent to the algorithm proposed in [30]. Other BO methods that consider the effect of a random environmental variable include [31, 32].

2.2 Risk measures

We recall that a *risk measure* (cf. [33, 34, 35, 36]) is a functional that maps probability distributions onto a real number. For a generic random variable Z, we often use the notation $\rho(Z)$ to indicate the risk value evaluated on the *distribution* of Z.

The most widely used risk measure is the Value-at-Risk (VaR) [37], which is $\rho(Z) = \text{VaR}_\alpha[Z] = \inf\{t : P_Z(Z \leq t) \geq \alpha\}$ where P_Z indicates the distribution of Z. Another widely used risk measure is the Conditional Value-at-Risk (CVaR), $\text{CVaR}_\alpha[Z] = \mathbb{E}_Z[Z \mid Z \geq \text{VaR}_\alpha(Z)]$.

Risk measures offer a middle ground between risk-neutral expectation operators and a worst-case performance measure, which is often more interpretable than expected utility [17]. They are widely used in finance, and value at risk is encoded in the Basel II accord [38].

3 Problem setup

We consider the optimization problem

$$\min_{x \in \mathcal{X}} \rho\left[F(x, W)\right], \tag{1}$$

where $\mathcal{X} \subset \mathbb{R}^{d_{\mathcal{X}}}$ is a simple compact set, e.g. a hyper-rectangle; W is a random variable with probability distribution $P(w)$ and compact support $\mathcal{W} \subset \mathbb{R}^{d_{\mathcal{W}}}$; and ρ is a known risk function, mapping the random variable $F(x, W)$ (induced by W) to a real number. We assume that $F : \mathcal{X} \times \mathcal{W} \to \mathbb{R}$ is a continuous black-box function whose evaluations are expensive, i.e., each evaluation takes hours or days, has significant monetary cost, or number of evaluations is limited for some other reason. We also assume that evaluations of F are either noise-free or observed with independent normally distributed noise with known variance.

We emphasize that the risk measure ρ is only over the randomness in W, and $\rho\left[F(x, W)\right]$ is calculated holding F fixed. For example, if ρ is the value at risk, and W is a continuous random variable, then this is explicitly written as:

$$\mathrm{VaR}_\alpha[F(x, W)] = \inf\left\{t : \int_{\mathcal{W}} \mathbb{1}\{F(x, w) \leq t\} p(w)\, dw \geq \alpha\right\} \tag{2}$$

Later, we will model F as being drawn from a GP, but we will continue to compute $\rho\left[F(x, W)\right]$ only over he randomness in W. Thus, this quantity is a function of F and x and will be random with a distribution induced by the distribution on F (and more precisely by the distribution on $F(x, \cdot)$).

4 Statistical model

We assume a level of familiarity with Gaussian processes and refer the reader to [39] for details. For notational purposes, we recall that, given a GP prior on F, the posterior distribution of F given the history after n evaluations of F, $\mathcal{F}_n$, is again a GP, $F \mid \mathcal{F}_n \sim \mathrm{GP}(\mu_n, \Sigma_n)$, where μ_n and Σ_n are the posterior mean and covariance functions.

We build on this standard analysis to build a Bayesian posterior over $\rho[F(\cdot, W)]$. The GP posterior distribution on F implies a posterior distribution on the mapping $x \mapsto \rho[F(x, W)]$. In contrast with the case where ρ is the expectation operator, this distribution is in general non-Gaussian, thus rendering computations more challenging.

The quantity of interest is the posterior mean of $\rho[F(x, W)]$, i.e.,

$$\mathbb{E}_n[\rho[F(x, W)]] \tag{3}$$

where $\mathbb{E}_n$ denotes the conditional expectation given the history up to time n, $\mathcal{F}_n$. We can estimate (3) following a simple Monte Carlo approach by first drawing multiple samples of the (possibly infinite dimensional) random vector $(F(x, w) : w \in \mathcal{W})$ according to the time n GP posterior distribution on F and then averaging over the value of ρ computed with respect to each of these samples. We note that if $\mathcal{W}$ is a finite set, sampling from $(F(x, w) : w \in \mathcal{W})$ reduces to sampling from a multivariate normal distribution with mean vector $(\mu_n(x, w) : w \in \mathcal{W})$ and covariance matrix $(\Sigma_n(x, w, x, w'))_{w, w' \in \mathcal{W}}$. We further discuss estimation of (3) and its derivative in Section 5.2.

5 The ρKG acquisition function

As is standard in the BO literature, our algorithm's search is guided by an acquisition function, whose maximization indicates the next point to evaluate. Our proposed acquisition function generalizes

the well known knowledge gradient acquisition function [24], which has been generalized to other settings such as parallel and multi-fidelity optimization [40, 41].

Before formally introducing our acquisition function, we note that, in our setting, choosing a decision x to be implemented after the evaluation stage is complete, is not a straightforward task. If the cardinality of $\mathcal{W}$ is large, then chances are $\rho[F(x, W)]$ will not be known exactly as this would require evaluating $F(x, w)$ for all $w \in \mathcal{W}$. A common approach in such scenarios is to choose the decision with best expected objective value according to the posterior distribution (c.f. [29]),

$$\min_{x \in \mathcal{X}} \mathbb{E}_N[\rho[F(x, W)]]. \tag{4}$$

Having defined the choice of the decision x to be implemented after the evaluation stage is complete, we are now in position to introduce our acquisition function, the knowledge gradient for risk measures. We motivate this acquisition function by noting that, if we had to choose a decision with the information available at time n, the expected objective value we would get according to equation (4) is simply $\min_{x \in \mathcal{X}} \mathbb{E}_n[\rho[F(x, W)]]$. On the other hand, if we were allowed to make on additional evaluation, the expected objective value we would get becomes $\min_{x \in \mathcal{X}} \mathbb{E}_{n+1}[\rho[F(x, W)]]$. Therefore,

$$\min_{x \in \mathcal{X}} \mathbb{E}_n[\rho[F(x, W)]] - \min_{x \in \mathcal{X}} \mathbb{E}_{n+1}[\rho[F(x, W)]]$$

can be interpreted as a measure of improvement due to making one additional evaluation.

We note that, given the information available at time n, $\min_{x \in \mathcal{X}} \mathbb{E}_{n+1}[\rho[F(x, W)]]$ is random due to its dependence on the yet unobserved $(n+1)^{th}$ evaluation, and thus the quantity above is itself random. The knowledge gradient for risk measures acquisition function is defined as the expected value of this quantity given the information available at time n and the candidate (x, w).

$$\rho\text{KG}(x, w) = \mathbb{E}_n \left[\min_{x \in \mathcal{X}} \mathbb{E}_n[\rho[F(x, W)]] - \min_{x \in \mathcal{X}} \mathbb{E}_{n+1}[\rho[F(x, W)] \mid (x_{n+1}, w_{n+1}) = (x, w)] \right] \tag{5}$$

Our algorithm sequentially chooses the next point to evaluate as the one that maximizes (5) and is thus, by construction, one-step Bayes optimal.

5.1 Evaluating ρKG

In this subsection, we explain how to estimate the value of ρKG for a given candidate (x, w) and the solutions to the inner optimization problems x_i^*. We note that the first term in (5) does not depend on the conditional expectation, and start by replacing the intractable expectation operators with their Monte-Carlo estimators and the inner optimization problem(s) with the given solutions.

$$\rho\text{KG}(x, w) \approx \frac{1}{M} \sum_{j=1}^{M} \rho[\widetilde{F}^j(x_0^*, W)] - \frac{1}{KM} \sum_{i=1}^{K} \sum_{j=1}^{M} \rho[\widehat{F}^{ij}(x_i^*, W)] \tag{6}$$

where $\widetilde{F}^j$ is the jth sample drawn from the current GP, and $\widehat{F}^{ij}$ is the jth sample drawn from i^{th} fantasy GP model. A fantasy GP model is the GP model obtained by conditioning the GP model on a fantasy observation simulated using the distribution implied by the GP model at the candidate point.

Let $w_{1:L}$ denote the vector $[w_1, \ldots, w_L]$. We approximate $\rho[\widehat{F}^{ij}(x, W)]$ (and $\widetilde{F}^j$) as follows. If $\mathcal{W}$ is a small finite set, let $\widetilde{\mathcal{W}} = \mathcal{W}$, otherwise let $\widetilde{\mathcal{W}} = \{w_{1:L}\} \overset{iid}{\sim} P(w)$. Draw the random vector $\{\widehat{F}^{ij}(x, w), w \in \widetilde{\mathcal{W}}\}$ from the ith fantasy GP, GP_f^i with mean and covariance functions μ_f^i, Σ_f^i,

$$[\widehat{F}(x_i, w_1), \ldots, \widehat{F}(x_i, w_L)] \sim \mathcal{N}(\mu_f^i(x_i, w_{1:L}), \Sigma_f^i(x_i, w_{1:L}, x_i, w_{1:L}));$$

order the samples so that $\widehat{F}(x_i, w_{(1)}) \leq \widehat{F}(x_i, w_{(2)}) \leq \ldots \leq \widehat{F}(x_i, w_{(L)})$, and set $\widehat{v}(x_i) := \widehat{F}(x_i, w_{(\lceil L\alpha \rceil)})$ and $\widehat{c}(x_i) := \frac{1}{\lceil L(1-\alpha) \rceil} \sum_{j=\lceil L\alpha \rceil}^{L} \widehat{F}(x_i, w_{(j)})$, where $\lceil \cdot \rceil$ is the ceiling operator, as estimates of VaR and CVaR respectively (cf. [20]). We note that the last step of the procedure outlined here assumes that W has either a continuous or a discrete uniform distribution. Otherwise, one should follow the definitions given in Section 2, and calculate the sample ρ while accounting for the probability mass of each $w \in \widetilde{\mathcal{W}}$.

5.2 Optimization of ρKG

Since the first term in (5) is independent of the candidate point, the optimization problem reduces to

$$\max_{(x,w)\in\mathcal{X}\times\mathcal{W}} \mathbb{E}_n \left[-\min_{x\in\mathcal{X}} \mathbb{E}_{n+1}[\rho[F(x,W)]] \mid (x_{n+1}, w_{n+1}) = (x,w) \right]. \tag{7}$$

The objective function involves two expectations that are not analytically available. Replacing these with their Monte-Carlo estimators, we obtain

$$\max_{(x,w)\in\mathcal{X}\times\mathcal{W}} \frac{1}{K} \sum_{i=1}^{K} -\min_{x_i\in\mathcal{X}} \frac{1}{M} \sum_{j=1}^{M} \rho \left[\widehat{F}^{ij}(x_i, W) \right], \tag{8}$$

where $\widehat{F}^{ij}$ is defined in Section 5.1. Let $\nabla_t \widehat{F}(x_i, w_l)$ be the gradient (w.r.t. the argument t) corresponding to the sample $\widehat{F}(x_i, w_l)$. Based on the ordering defined in Section 5.1, let $\nabla_t \widehat{v}(x_i) := \widehat{F}(x_i, w_{(\lceil L\alpha \rceil)})$ and $\nabla_t \widehat{c}(x_i) := \frac{1}{\lceil L(1-\alpha) \rceil} \sum_{j=\lceil L\alpha \rceil}^{L} \widehat{F}(x_i, w_{(j)})$ be the estimators of gradients of VaR and CVaR respectively. Under mild regularity conditions, [42] and [43] show that these are asymptotically unbiased and consistent estimators of $\nabla_t \rho[F(x_i, W)] \mid \mathrm{GP}_f^i$. Combining with results from the IPA literature (e.g. [44], Proposition 1), averaging over M i.i.d. samples of the estimators, we obtain asymptotically unbiased and consistent estimators of $\nabla_t \mathbb{E}[\rho[F(x_i, W)] \mid \mathrm{GP}_f^i]$. The estimators of $\nabla_{x_i} \mathbb{E}[\rho[F(x_i, W)] \mid \mathrm{GP}_f^i]$ can be used to solve the inner optimization problem.

Let x_i^* be the solution to the ith inner optimization problem. Using the Envelope theorem ([45], Corollary 4), we can propagate gradients one step further to obtain

$$\frac{1}{K} \sum_{i=1}^{K} -\frac{1}{M} \sum_{j=1}^{M} \nabla_u \widehat{c/v}^{\,j}(x_i^*) \tag{9}$$

where $\nabla_u \widehat{c/v}^{\,j}(x_i^*)$ is the estimator from j^{th} sample, as asymptotically unbiased and consistent gradient estimators for (7). We use these with the LBFGS algorithm [46] to optimize VaRKG.

Proposition 1. *Under mild regularity conditions, the estimators given by (9) are asymptotically unbiased and consistent gradient estimators of the* ρKG *optimization problem (7).*

A formal statement of the proposition and the proof is given in the supplement for the case of $d^{\mathcal{W}} = 1$. We show in the supplement that selecting the nth candidate to evaluate using the ρKG algorithm, including the training of the GP model, has a computational complexity of $\mathcal{O}(Q_1 n^3 + Q_2 Q_3 K L[n^2 + Ln + L^2 + ML])$, where Q_1, Q_2, Q_3 are the number of LBFGS [46] iterations performed for training the GP model, and the outer and inner optimization loops respectively.

5.3 The ρKG^{apx} approximation

The ρKG algorithm is computationally intensive, as it requires solving a nested non-convex optimization problem. While we are confident that, in a wide range of settings, the sampling efficiency it provides justifies its computational cost, we also reckon that in certain not-so-expensive settings, a faster algorithm is desirable.

Inspired by the EI [21] and KGCP [24] acquisition functions, we propose the following acquisition function, which replaces the inner optimization problem of ρKG with a much simpler one. In the following, the set $\widetilde{\mathcal{X}} = \{x_{1:n}\}$ is the x components of all the solutions evaluated so far, in addition the set $\widetilde{\mathcal{X}}^+ = \{x_{1:n}, x_{n+1}\}$ includes the x component of the current candidate in consideration.

$$\rho\text{KG}^{apx}(x, w) = \mathbb{E}_n \left[\min_{x \in \widetilde{\mathcal{X}}} \mathbb{E}_n[\rho[F(x, W)]] - \min_{x \in \widetilde{\mathcal{X}}^+} \mathbb{E}_{n+1}[\rho[F(x, W)]] \mid (x_{n+1}, w_{n+1}) = (x, w) \right]. \quad (10)$$

We note that, like ρKG, ρKG^{apx} has an appealing one-step Bayes optimal interpretation; the maximizer of ρKG^{apx} is the one-step optimal point to evaluate if we restrict the choice of the decision to be implemented, x, among those that have been evaluated for at least one environmental condition, w.

5.4 Two time scale optimization

In this paper, we introduce two acquisition functions, ρKG and ρKG^{apx}. These acquisition functions share a nested structure, making optimization computationally challenging. Here we describe a novel two-timescale optimization approach overcoming this challenge.

Because the posterior mean and kernel are continuous, if we fix the seed used to generate the fantasy model and the GP sample path, then continuity of the posterior mean and kernel causes a small perturbation to the candidate solutions to shift the sample path only slightly. Thus, optimal solutions obtained for the previous candidate should be in a small neighborhood of the current optimal solutions. We can thus use the inner solutions obtained in the previous iteration to obtain a good approximation of the acquisition function value and gradients for the current candidate. We utilize this observation by solving the inner optimization problem once every $T \approx 10$ iterations, and using this solution to evaluate ρKG for the remaining $T - 1$ iterations. We refer to this approach as *two-scale optimization* and present an algorithmic description with more detail in the supplement.

The two-time scale idea outlined here is not limited to ρKG, and can be applied to other acquisition functions that require nested optimization, such as [24, 25, 30]. In numerical testing, the two-time-scale optimization approach did not affect the performance of either of the algorithms while offering significant computational savings.

5.5 A visual analysis of the acquisition functions

In Figure 1, we provide a plot of a GP model based on 6 random samples taken from a 2D Branin test function, and the corresponding acquisition function values (VOI, short for value of information) of ρKG and ρKG^{apx} over the candidate space $\mathcal{X} \times \mathcal{W}$. In the example, we consider a CVaR objective with risk level $\alpha = 0.7$ and a uniform distribution over $\mathcal{W} = \{0/9, 1/9, \ldots, 9/9\}$.

Our first observation is that the VOI plots closely resemble each other, supporting our claim that ρKG^{apx} is a good approximation to ρKG. When we look at the implied posterior on the objective, we see a large uncertainty for large x values, and expect a large VOI to encourage exploration. When we look at the VOI plots, we see that this is indeed the case, but more so with certain w than others. Looking at the μ_n plot, we see that the subset of $\mathcal{W}$ corresponding to CVaR objective is where w is

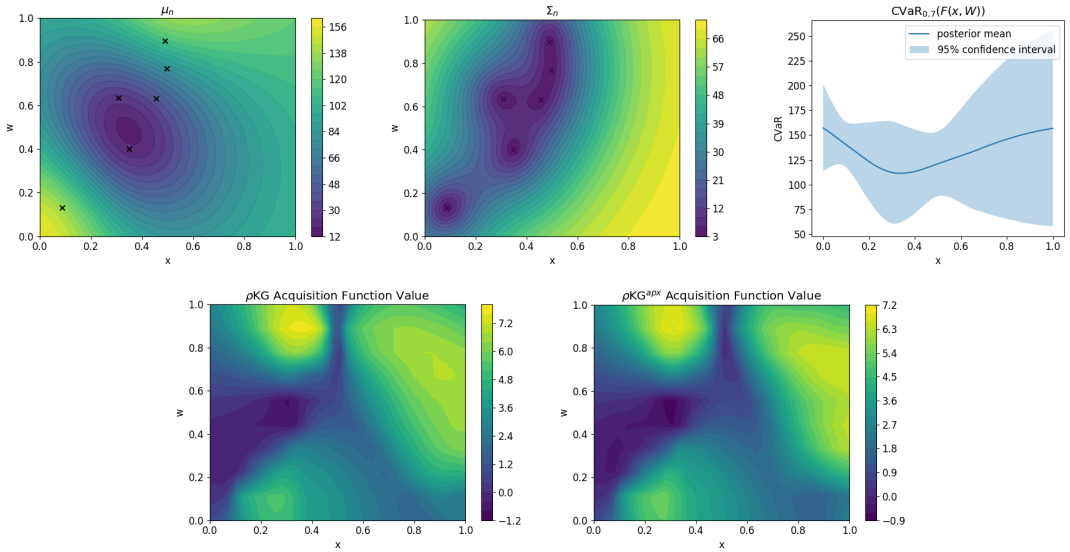

Figure 1: Top: the underlying GP model, its mean function, standard deviation, and implied posterior on the objective. Left: ρKG, Right: $\rho\mathrm{KG}^{apx}$ acquisition function values.

large, and see that VOI is larger in this region. Thus, for a given x, ρKG prefers a sample from w which directly affects the value of the objective, and can efficiently search the candidate space.

We observe a similar behavior in the promising region $x \in [0.2, 0.5]$. Certain w in this are both have a small uncertainty and do not directly affect the value of the objective, thus have small VOI despite being in the promising region of the GP model. However, w at the two ends directly affect the value of CVaR objective in the promising region, thus have the largest VOI in the whole candidate space. We hope that these plots and accompanying discussion help the reader appreciate the value of considering the context variables in designing the acquisition function.

6 Numerical experiments

In this section, we present several numerical examples that demonstrate the sampling efficiency of our algorithms. We compare our algorithms with Expected Improvement (EI, [21]), Knowledge Gradient (KG, [25]), Upper Confidence Bound (UCB), and Max Value Entropy Search (MES, [47]) algorithms. We use the readily available implementations from the BoTorch package [48], and the default parameter values given there. For benchmark algorithms, we fit a GP on observations of $\rho[F(x, W)]$, which are obtained by evaluating a given $x \in \mathcal{X}$ for all $w \in \mathcal{W}$ (or a subset $\widetilde{\mathcal{W}}$) and then calculating the value of the risk measure on these samples. Thus, the benchmark algorithms require $|\mathcal{W}|$ (or $|\widetilde{\mathcal{W}}|$) samples per iteration whereas ρKG and $\rho\mathrm{KG}^{apx}$ require only one.

We optimize each acquisition function using the LBFGS [46] algorithm with $10 \times (d^{\mathcal{X}} + d^{\mathcal{W}})$ restart points. The restart points are selected from $500 \times (d^{\mathcal{X}} + d^{\mathcal{W}})$ raw samples using a heuristic. For the inner optimization problem of ρKG, we use $5 \times d^{\mathcal{X}}$ random restarts with $25 \times d^{\mathcal{X}}$ raw samples. For both ρKG and $\rho\mathrm{KG}^{apx}$, we use the two time scale optimization where we solve the inner optimization problem once every 10 optimization iterations. ρKG and $\rho\mathrm{KG}^{apx}$ are both estimated using $K = 10$ fantasy GP models, and $M = 40$ sample paths for each fantasy model.

We initialize each run of the test problems with $2d^{\mathcal{X}} + 2$ starting points from the $\mathcal{X}$ space. We evaluate $F(x, w)$ for each $2d^{\mathcal{X}} + 2$ points for each $w \in \mathcal{W}$ (or $\widetilde{\mathcal{W}}$, to be specified for each problem).

ρKG and ρKG^{apx} then use each of these evaluations to initialize their respective GP models, and the benchmark algorithms use $\rho[F(x, W)]$ computed using these evaluations to initialize the GP models. Further details on experiment settings is given in the supplement. The code for our implementation of the algorithms and experiments can be found at `https://github.com/saitcakmak/BoRisk`.

6.1 Synthetic test problems

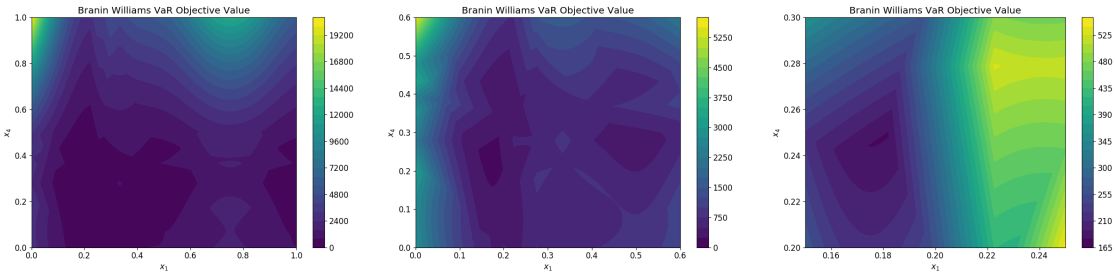

Figure 2: The VaR objective of Branin Williams function. Left to right, full domain and close-ups of the optimal region. It is notable that the objective function is quite flat across most of its domain, making it challenging to identify the optimal solution using a small number of evaluations.

The first two problems we consider are synthetic test functions from the BO literature. The first problem is the 4 dimensional Branin-Williams problem formulated in [49]. We consider the minimization of VaR at risk level $\alpha = 0.7$ w.r.t. the distribution of context variables $w = (x_2, x_3)$. A plot of the VaR objective of the Branin Willams function is given in Figure 2.

The second problem we consider is the 7 dimensional $f_6(x_c, x_e)$ function from [50]. We formulate this problem for minimization of CVaR at risk level α w.r.t. the distribution of the 3 dimensional environmental variable x_e. More details can be found in the supplement.

6.2 Portfolio optimization problem

In this test problem, our goal is to tune the hyper-parameters of a trading strategy so as to maximize return under risk-aversion to random environmental conditions. We use CVXPortfolio [51] to simulate and optimize the evolution of a portfolio over a period of four years using open-source market data. Each evaluation of this simulator returns the average daily return over this period of time under the given combination of hyper-parameters and environmental conditions. The details of this simulator can be found in Sections 7.1-7.3 of [51].

The hyper-parameters to be optimized are the risk and trade aversion parameters, and the holding cost multiplier over the ranges $[0.1, 1000]$, $[5.5, 8.]$, and $[0.1, 100]$, respectively. The environmental variables are the bid-ask spread and the borrow cost, which we assume are uniform over $[10^{-4}, 10^{-2}]$ and $[10^{-4}, 10^{-3}]$, respectively. For this problem, we use the VaR risk measure at risk level $\alpha = 0.8$ We use a random subset $\widetilde{\mathcal{W}}$ of size 40 for the inner computations of our algorithms, and a random subset of size 10 for the benchmark evaluations of $\text{VaR}_{0.8}[F(x, W)]$.

Since this simulator is indeed expensive-to-evaluate, with each evaluation taking around 3 minutes, evaluating the performance of the various algorithms becomes prohibitively expensive. Therefore, in our experiments we do not use the simulator directly. Instead, we build a surrogate function obtained as the mean function of a GP trained using evaluations of the actual simulator across $3,000$ points chosen according to a Sobol sampling design [52].

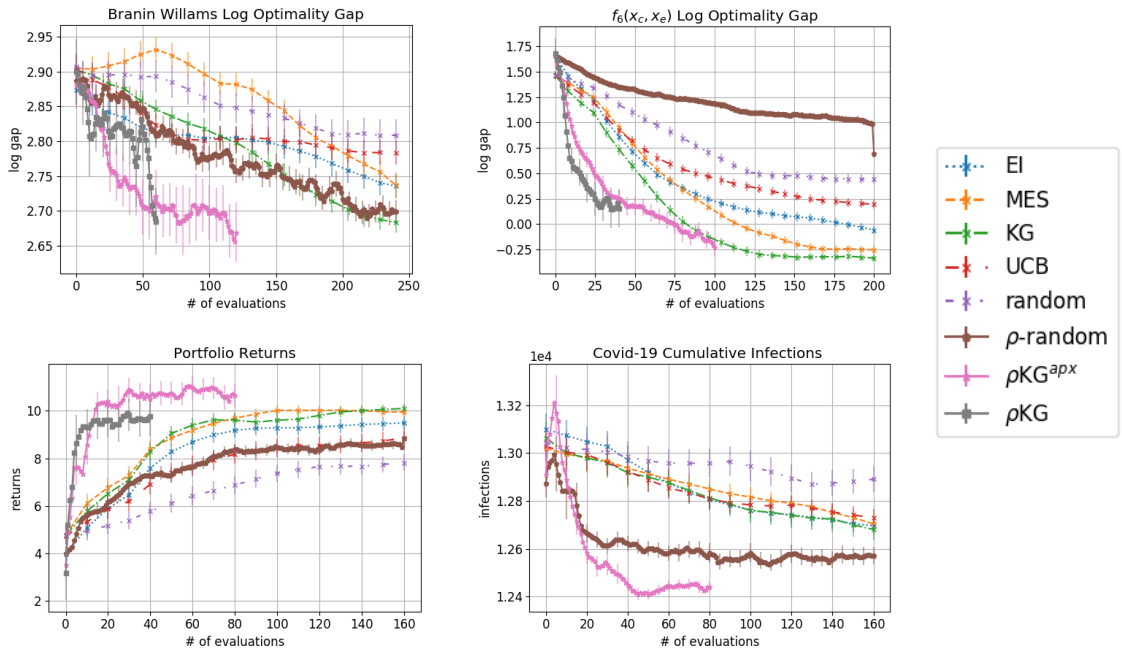

Figure 3: Top: The log optimality gap in Branin Williams (left) and $f_6(x_c, x_e)$ (right). Bottom left: The returns in Portfolio problem. Bottom right: Cumulative number of infections in the COVID-19 problem. The plots are plotted against the number of $F(x, w)$ evaluations, and are smoothed using a moving average of 3 iterations. Non-smoothed plots are given in the supplement.

6.3 Allocating COVID-19 testing capacity

In this example, we study allocation of a joint COVID-19 testing capacity between three neighboring populations (e.g. cities, counties). The objective is to allocate the testing capacity between the populations to minimize the total number of people infected with COVID-19 in a span of two weeks. We use the COVID-19 simulator provided in [53] and discussed in [54, 55]. The simulator models the interactions between individuals within the population, the disease spread and progression, testing and quarantining of positive cases. Contact tracing is performed for people who test positive, and the contacts that are identified are placed in quarantine.

We study three populations of size $5 \times 10^4, 7.5 \times 10^4, 10^5$ that share a combined testing capacity of 10^4 tests per day. The initial disease prevalence within each population is estimated to be in the range of $0.1 - 0.4\%, 0.2 - 0.6\%$ and $0.2 - 0.8\%$ respectively. We assign a probability of 0.5 to the middle of the range and 0.25 to the two extremes, independently for each population. Thus, the initial prevalence within each population defines the context variables. We pick the fraction of testing capacity allocated to the first two populations as the decision variable (remaining capacity is allocated to the third), with the corresponding decision space $\mathcal{X} = \{x \in \mathbb{R}^2_+ : x_1 + x_2 \leq 1\}$. For the inner computations of ρKG^{apx}, we use the full $\mathcal{W}$ set, however, for the evaluations of benchmark algorithms we randomly sample a subset $\widetilde{\mathcal{W}}$ of size 10 to avoid using 27 evaluations at once.

6.4 Results

The results of the experiments are plotted in Figure 3. Evaluations reported exclude initialization and error bars denote one standard error. In each experiment, ρKG and ρKG^{apx} match the performance of all benchmarks using fewer evaluations, thus demonstrating superior sampling efficiency. In the

experiments $\widetilde{W}$ is intentionally kept small to avoid giving ρKG an outsize advantage. If $\widetilde{W}$ were larger, the benchmarks would use up more evaluations per iteration and our algorithms would provide an even larger benefit.

To demonstrate the benefit of using our novel statistical model on $\rho[F(x, w)]$, we compare two random sampling strategies. The one labeled "random" evaluates $\rho[F(x, w)]$ and uses the corresponding GP model over $\mathcal{X}$. "ρ-random" on the other hand, evaluates $F(x, w)$ at a randomly selected (x, w), uses our statistical model and reports $\arg\min_x \mathbb{E}_n[\rho[F(x, w)]]$ as its solution. The ability to survey $\mathcal{X} \times \mathcal{W}$ more freely gives "ρ-random" an edge over "random" in all experiments except for $f_6(x_c, x_e)$ which we attribute to the size of $\mathcal{W}$. Moreover, in the COVID-19 problem, we see that the "ρ-random" is performing better than all the benchmarks considered. This demonstrates the added value of our statistical model and suggests an additional cheap-to-implement algorithm that is useful whenever $F(x, w)$ is cheap enough to render typical BO algorithms too expensive.

Broader Impact

Our work is of interest whenever one needs to make a decision guided by an expensive simulator, and subject to environmental uncertainties. Such scenarios arise in simulation assisted medical decision making [4], in financial risk management [7, 56, 57], in public policy, and disaster management [58].

The impact of our algorithms could be summarized as facilitating risk averse decision making. In many scenarios, risk averse approaches result in decisions that are more robust to environmental uncertainties compared to decisions resulting from common risk neutral alternatives. For example, in a financial crisis, an earlier risk-averse decision of holding more cash and other low-risk securities might prevent large losses or even the default of a financial institution. As another example, a risk averse decision of stockpiling of excess medical supplies in non-crisis times would alleviate the shortages faces during crisis times, such as the COVID-19 pandemic we are facing today.

On the negative side of things, the risk averse decisions we facilitate may not always benefit all stakeholders. For a commercial bank, a risk averse approach may suggest a higher credit score threshold for making loans, which might end up preventing women and minorities, who systematically have lower credit scores, from access to much needed credit.

In our case, the failure of the system would mean a poor optimization of the objective, and recommendation of a bad solution. Implementation of a bad decision can have harmful consequences in many settings, however, we imagine that any solution recommended by our algorithms would then be evaluated using the simulator, thus preventing the implementation of said decision.

Our methods do not rely on any training data, and only require noisy evaluations of the function value. Thus, it can be said that our method does not leverage any bias in any training data. However, the solutions recommended by our algorithms are only good up to the supplied function evaluations, thus are directly affected by any biases built into the simulator used for these evaluations.

Acknowledgements

The authors gratefully acknowledge the support by the National Science Foundation under Grants CAREER CMMI-1453934 and CCF-1740822; and the Air Force Office of Scientific Research under Grants FA9550-19-1-0283 and FA9550-15-1-0038.

References

[1] J. Snoek, H. Larochelle, and R. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems 25*, Advances in Neural Information Processing Systems, pp. 2951–2959, 12 2012.

[2] M.-A. Zöller and M. F. Huber, *Benchmark and Survey of Automated Machine Learning Frameworks*, 2019.

[3] D. Negoescu, P. Frazier, and W. Powell, "The knowledge-gradient algorithm for sequencing experiments in drug discovery," *ORSA journal on computing*, vol. 23, pp. 346–363, 6 2011.

[4] J. Xie, P. Frazier, S. Sankaran, A. Marsden, and S. Elmohamed, "Optimization of computationally expensive simulations with gaussian processes and parameter uncertainty: Application to cardiovascular surgery," pp. 406–413, 10 2012.

[5] B. Letham, B. Karrer, G. Ottoni, and E. Bakshy, *Efficient tuning of online systems using Bayesian optimization*, 2018. *Available at* `https://research.fb.com/blog/2018/09/efficient-tuning-of-online-systems-using-bayesian-optimization/`.

[6] A. Rai, R. Antonova, F. Meier, and C. G. Atkeson, "Using simulation to improve sample-efficiency of bayesian optimization for bipedal robots.," *Journal of machine learning research*, vol. 20, no. 49, pp. 1–24, 2019.

[7] J. A. Lopez, *Regulatory Evaluation of Value-at-Risk Models*, Dec 1997. Staff Report, Federal Reserve Bank of New York.

[8] Y. An, J. Liang, S. E. Schild, M. Bues, and W. Liu, "Robust treatment planning with conditional value at risk chance constraints in intensity-modulated proton therapy," *Medical physics*, vol. 44, no. 1, pp. 28–36, 2017.

[9] G. J. Lim, L. Kardar, S. Ebrahimi, and W. Cao, "A risk-based modeling approach for radiation therapy treatment planning under tumor shrinkage uncertainty," *European Journal of Operational Research*, vol. 280, no. 1, pp. 266–278, 2020.

[10] P. C. Austin and M. J. Schull, "Quantile regression: A statistical tool for out-of-hospital research," *Academic Emergency Medicine*, vol. 10, no. 7, pp. 789–797, 2003.

[11] S. A. Sethi and M. Dalton, "Risk measures for natural resource management: Description, simulation testing, and r code with fisheries examples," *Journal of Fish and Wildlife Management*, 2012.

[12] M. Meraklı and S. Küçükyavuz, "Risk aversion to parameter uncertainty in markov decision processes with an application to slow-onset disaster relief," *IISE Transactions*, vol. 52, no. 8, pp. 811–831, 2020.

[13] A. Ruszczyński and A. Shapiro, "Optimization of risk measures," in *Probabilistic and Randomized Methods for Design under Uncertainty* (G. Calafiore and F. Dabbene, eds.), pp. 119–157, London: Springer London, 2006.

[14] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton University Press, 2009.

[15] D. Bertsimas, D. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.

[16] A. A. Trindade, S. Uryasev, A. Shapiro, and G. Zrazhevsky, "Financial prediction with constrained tail risk," *Journal of Banking & Finance*, vol. 31, no. 11, pp. 3524 – 3538, 2007. Risk Management and Quantitative Approaches in Finance.

[17] D. Wu, H. Zhu, and E. Zhou, "A bayesian risk approach to data-driven stochastic optimization: Formulations and asymptotics," *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 1588–1612, 2018.

[18] S. Zymler, B. Rustem, and D. Kuhn, "Robust portfolio optimization with derivative insurance guarantees," *European Journal of Operational Research*, vol. 210, pp. 410–424, April 2011.

[19] P. Jorion, *Financial Risk Manager Handbook*. Wiley, New York, 2010.

[20] L. J. Hong, Z. Hu, and G. Liu, "Monte carlo methods for value-at-risk and conditional value-at-risk: A review," *ACM Trans. Model. Comput. Simul.*, vol. 24, pp. 22:1–22:37, Nov. 2014.

[21] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. of Global Optimization*, vol. 13, p. 455–492, Dec. 1998.

[22] P. Frazier, W. Powell, and S. Dayanik, "The knowledge-gradient policy for correlated normal beliefs," *INFORMS Journal on Computing*, vol. 21, no. 4, pp. 599–613, 2009.

[23] J. M. Hernandez-Lobato, M. Gelbart, M. Hoffman, R. Adams, and Z. Ghahramani, "Predictive entropy search for bayesian optimization with unknown constraints," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1699–1707, PMLR, 07–09 Jul 2015.

[24] W. Scott, P. Frazier, and W. Powell, "The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 996–1026, 2011.

[25] J. Wu and P. 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, (USA), pp. 3134–3142, Curran Associates Inc., 2016.

[26] J. Wang, S. C. Clark, E. Liu, and P. I. Frazier, *Parallel Bayesian Global Optimization of Expensive Functions*, 2016.

[27] K. Swersky, J. Snoek, and R. P. Adams, *Freeze-Thaw Bayesian Optimization*, 2014.

[28] J. Wu, M. Poloczek, A. G. Wilson, and P. I. Frazier, "Bayesian optimization with gradients," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 5273–5284, Curran Associates Inc., 2017.

[29] P. I. Frazier, "A tutorial on bayesian optimization," in *Recent Advances in Optimization and Modeling of Contemporary Problems*, ch. 11, pp. 255–278, 2018.

[30] S. Toscano-Palmerin and P. I. Frazier, "Bayesian optimization with expensive integrands," *arXiv*, vol. 1803.08661, 2018.

[31] J. Kirschner, I. Bogunovic, S. Jegelka, and A. Krause, "Distributionally robust bayesian optimization," 2020.

[32] I. Bogunovic, J. Scarlett, S. Jegelka, and V. Cevher, "Adversarially robust optimization with gaussian processes," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 5760–5770, Curran Associates, Inc., 2018.

[33] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[34] R. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443 – 1471, 2002.

[35] F. Delbaen, "Coherent risk measures on general probability spaces," in *Advances in Finance and Stochastics: Essays in Honour of Dieter Sondermann* (K. Sandmann and P. J. Schönbucher, eds.), pp. 1–37, Springer Berlin Heidelberg, 2002.

[36] R. Rockafellar and S. Uryasev, "The fundamental risk quadrangle in risk management, optimization and statistical estimation," *Surveys in Operations Research and Management Science*, vol. 18, p. 33–53, 10 2013.

[37] P. Jorion, *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill, 2007.

[38] Basel Committee on Banking Supervision, "Fundamental review of the trading book," *Consultative Document*, 2012.

[39] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

[40] J. Wu and P. 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, (Red Hook, NY, USA), p. 3134–3142, Curran Associates Inc., 2016.

[41] M. Poloczek, J. Wang, and P. Frazier, "Multi-information source optimization," in *Advances in Neural Information Processing Systems*, pp. 4288–4298, 2017.

[42] G. Jiang and M. C. Fu, "Technical note—on estimating quantile sensitivities via infinitesimal perturbation analysis," *Operations Research*, vol. 63, no. 2, pp. 435–441, 2015.

[43] L. J. Hong and G. Liu, "Simulating sensitivities of conditional value at risk," *Management Science*, vol. 55, no. 2, pp. 281–293, 2009.

[44] M. Broadie and P. Glasserman, "Estimating security price derivatives using simulation," *Management Science*, vol. 42, no. 2, pp. 269–285, 1996.

[45] P. Milgrom and I. Segal, "Envelope theorems for arbitrary choice sets," *Econometrica*, vol. 70, no. 2, pp. 583–601, 2002.

[46] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Softw.*, vol. 23, p. 550–560, Dec. 1997.

[47] Z. Wang and S. Jegelka, "Max-value entropy search for efficient Bayesian optimization," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 3627–3635, PMLR, 06–11 Aug 2017.

[48] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "BoTorch: Programmable Bayesian Optimization in PyTorch," *arxiv e-prints*, 2019.

[49] B. Williams, T. Santner, and W. Notz, "Sequential design of computer experiments to minimize integrated response functions," *Statistica Sinica*, vol. 10, pp. 1133–1152, 10 2000.

[50] J. Marzat, E. Walter, and H. Piet-Lahanier, "Worst-case global optimization of black-box functions through kriging and relaxation," *Journal of Global Optimization*, vol. 55, pp. 707–727, 04 2013.

[51] S. Boyd, "Multi-period trading via convex optimization," *Foundations and Trends® in Optimization*, vol. 3, no. 1, pp. 1–76, 2017.

[52] A. B. Owen, "Scrambling sobol' and niederreiter–xing points," *Journal of Complexity*, vol. 14, no. 4, pp. 466 – 489, 1998.

[53] M. Cashore, P. Frazier, Y. Zhang, and J. Wan, *Group Testing*, 2020. *Available at* `https://github.com/peter-i-frazier/group-testing/`.

[54] P. Frazier, Y. Zhang, and M. Cashore, *Feasibility of COVID-19 Screening for the U.S. Population with Group Testing*, 2020. *Available at* `https://docs.google.com/document/d/1hw5K5V7XOug_r6CQOUYt25szQxXFPmZmFhK15ZpH5UO/`.

[55] L. Kotlikoff, *Drs. Fauci & Birx: Here's A Way To Contain Covid-19 And Reopen The Economy In As Little As One Month*, May 2020. *Available at* `https://www.forbes.com/sites/kotlikoff/2020/05/03/dr-fauci-heres-a-way-to-contain-covid-19-and-reopen-the-economy-in-as-little-as-one-month/`.

[56] A. J. McNeil, R. Frey, and P. Embrechts, *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton: Princeton University Press, 2005.

[57] M. B. Gordy and S. Juneja, "Nested simulation in portfolio risk measurement," *Management Science*, vol. 56, no. 10, pp. 1833–1848, 2010.

[58] D. Steward and T. T. H. Wan, "The role of simulation and modeling in disaster management," *Journal of Medical Systems*, 2007.

arXiv:2007.05554v2 [stat.ML] 16 Jul 2020

Supplementary Material to "Bayesian Optimization of Risk Measures"

Sait Cakmak[a], Raul Astudillo[b], Peter Frazier[b], and Enlu Zhou[a]
[a]School of Industrial and Systems Engineering, Georgia Institute of Technology
[b]School of Operations Research and Information Engineering, Cornell University

July 17, 2020

Contents

1 Computational Complexity **1**

2 Two time scale optimization **2**

3 Experiments **3**
3.1 Algorithm Parameters . . . 3
3.2 GP Model and Fitting . . . 4
3.3 Synthetic Test Problems . . . 4
3.4 Experiment Details . . . 5
3.5 Results . . . 6

4 Proofs **6**

1 Computational Complexity

In this section, we analyze the computational complexity of using ρKG to pick the nth candidate to evaluate. For simplicity, we ignore the multiple restart points used for optimization, and present the analysis for a single restart point. The resulting complexity involves terms for number of L-BFGS-B iterations performed. These terms can be adjusted if needed to account for the cost of multiple restart points.

In this analysis, we assume that evaluating the GP prior mean and kernel functions (and the corresponding derivatives) takes $\mathcal{O}(1)$ time. We break down the computations into several chunks and analyze them one by one.

An iteration starts by fitting the GP hyper-parameters using MAP estimation with Q_1 iterations of L-BFGS-B [**?**] algorithm. Each iteration requires calculating the likelihood of the data, which requires computing

$$A_n = \Sigma_0(x_{1:n}, w_{1:n}, x_{1:n}, w_{1:n}) + \mathrm{diag}(\sigma^2(x_1, w_1), \ldots, \sigma^2(x_n, w_n))$$

($\mathcal{O}(n^2)$ kernel evaluations), its Cholesky factor and inverse ($\mathcal{O}(n^3)$), and solving a triangular set of equations ($\mathcal{O}(n^2)$), putting the total cost of fitting the GP at $\mathcal{O}(Q_1 n^3)$.

We can use the Cholesky factor of A_n and the inverse A_n^{-1} computed in the previous step to calculate the posterior mean and variance at the candidate point at a cost of $\mathcal{O}(n^2)$, which are then used to draw K fantasy observations at a cost of $\mathcal{O}(K)$.

Let A_{n+1} be the matrix obtained by adding A_n one row and column corresponding to the candidate (x_{n+1}, w_{n+1}) at a cost of $\mathcal{O}(n)$. Using the pre-computed Cholesky factor of A_n, we can compute the Cholesky factor of A_{n+1} at a cost of $\mathcal{O}(n^2)$. The inverse A_{n+1}^{-1} can also be obtained from A_n^{-1} at a cost of $\mathcal{O}(n^2)$. Note that A_{n+1} is identical for each fantasy.

For each fantasy model, we need to compute the posterior mean and covariance matrix for the L points on which we draw the sample paths. Given A_{n+1}^{-1}, we first compute $\Sigma_0(x_i, w_{1:L}, x_{1:n+1}, w_{1:n+1})A_{n+1}^{-1}$ at a cost of $\mathcal{O}(Ln^2)$, then use it to compute the mean vector and the covariance matrix at a total cost of $\mathcal{O}(L^2n)$. To sample from the multivariate normal distribution, we also need the Cholesky factor of the covariance matrix, which is obtained at a cost of $\mathcal{O}(L^3)$. Repeating this for each fantasy, we obtain all K mean vectors, covariance matrices and the Cholesky factors at a total cost of $\mathcal{O}(KLn^2 + KL^2n + KL^3)$.

Given the mean vector and the Cholesky factor of the covariance matrix, we can generate a sample $\widehat{F}^{ij}(x_i, w_{1:L})$ at a cost of $\mathcal{O}(L^2)$. This results in a total cost of $\mathcal{O}(KML^2)$ to generate all samples. The sorting and other arithmetic operations needed for computing ρ given the sample paths are then all $\mathcal{O}(L^2)$ per sample-path, thus introducing no additional terms.

Note that each iteration of L-BFGS-B performs a bounded number of line search and a bounded number of function evaluations. Thus, if we use Q_3 iterations of L-BFGS-B to solve the inner optimization problem, this translates to a total computational cost of $\mathcal{O}(Q_3KL[n^2 + Ln + L^2 + ML])$ for the inner optimization of all K fantasies.

If we then use Q_2 iterations of L-BFGS-B to optimize the acquisition function, we end up with a total cost of $\mathcal{O}(Q_2Q_3KL[n^2+Ln+L^2+ML])$ for optimization, and a total cost of $\mathcal{O}(Q_1n^3+Q_2Q_3KL[n^2+Ln+L^2+ML])$ to pick the nth candidate to evaluate, including the GP fitting.

2 Two time scale optimization

Among the state of the art Bayesian optimization algorithms, knowledge gradient (KG) algorithms are on the more expensive side due to the inherent nested optimization structure. Although the added computational cost is typically justified by the superior sampling efficiency they offer (see [1, 2] for numerical comparisons), reducing the computational cost would make KG algorithms applicable in much broader settings.

To reduce the computational cost of KG optimization, [2] proposes a one-shot approach for optimizing the KG acquisition function, which converts the $d^{\mathcal{X}}$ dimensional nested optimization problem into a (non-nested) $(K+1)d^{\mathcal{X}}$ dimensional optimization problem. The objective function of the one-shot problem is substantially cheaper to evaluate, as it does not require solving an optimization problem. Moreover, this new problem problem is deterministic and smooth, allowing the use of efficient gradient-based algorithms such as L-BFGS-B. However, the one-shot problem is again non-convex, and its large dimension makes it challenging to find a global optimal solution.

Using a gradient based approach, the one-shot problem is optimized locally from a given set of restart points, resulting in local solutions to the inner problems, rather than a global solution one would get with a nested approach. This does not cause any significant issues if one has large enough K and the majority of them are at their global optimum, however, this requires starting with a large enough number of raw samples.

When the evaluations of the objective are cheap, as in the classical setting, one can afford to evaluate the one-shot acquisition function at a large number of (carefully crafted) raw samples, and select a good set of restart points for optimization. As K increases, both the number of raw samples and the restart points need to increase as well. When the acquisition function evaluations (even without the inner optimization) are significantly more expensive than the classical setting, one is restricted to a relatively small number of raw samples, which requires using a small K (< 10) to achieve a good performance within a reasonable time.

If one uses an insufficient number of raw samples to generate the restart points, it is possible to end up with a bad candidate that has all the globally optimal inner solutions, thus has a very large acquisition function value, and at the same time to have the optimal candidate, however with bad local optimal solutions, thus with a relatively small acquisition function value. In such a scenario, the one-shot approach will recommend the bad candidate with the large acquisition function value, leading to a poor sampling decision. This so called *instability* is what became an issue with the one-shot implementation of ρKG in our preliminary testing, and led to a search for a *stable* alternative.

In this paper, we propose *two time scale optimization* (TTS), a novel approach for optimizing KG type of acquisition functions that addresses the issues raised here about the one-shot method while still providing significant computational savings. The main idea behind the TTS optimization is that the objective of the inner optimization problem is a continuous function of the candidate, and the optimal inner solution changes only a small amount between iterations when we update the candidate using a gradient based algorithm.

Algorithm 1 Two time scale optimization of ρKG

Input: $Q_2, Q_3, M, K, L, \alpha$, starting candidate (x^0, w^0), and TTS-frequency T.
for $t = 0$ **to** $Q_2 - 1$ **do**
  Generate the fantasy models $\mathrm{GP}_f^i, i = 1, \ldots, K$ using the candidate (x^t, w^t).
  for $i = 1$ **to** K **do**
    if $t \bmod T = 0$ **then**
      Generate a set of random restart points for optimization of ith inner problem. Include the previous solution x_i^T if available.
      Use Q_3 iterations of the multi-start L-BFGS-B algorithm to optimize i^{th} inner problem, and set x_i^T as the optimizer.
    end if
    Return x_i^T as the solution to i^{th} inner problem
  end for
  Set $\frac{1}{K}\sum_{i=1}^{K} -\frac{1}{M}\sum_{j=1}^{M} \nabla_{(x,w)}\widehat{c/v}^{\,j}(x_i^T)$ as the gradient at the current candidate.
  Do an iteration of L-BFGS-B using this gradient to obtain new candidate (x^{t+1}, w^{t+1}).
end for
Return: (x^{Q_2}, w^{Q_2}) as the next candidate to evaluate.

Thus, the solution to the inner problem obtained with the previous candidate provides a good approximation of the acquisition function value and derivatives. TTS optimization leverages this observation to provide computational savings by solving the inner optimization problem once every T iterations. We provide a detailed description of TTS in Algorithm 1. We note that L-BFGS-B performs multiple line searches within an iteration and evaluates the solution found at the end of each line search. The description of Algorithm 1 treats each line search as an iteration of L-BFGS-B, and the candidate (x^t, w^t) is updated at the end of each line search. This essentially means that we optimize the inner solution at every T^{th} evaluation of the ρKG and not necessarily at every T^{th} iteration of L-BFGS-B.

In the algorithm description the parameter T is the *TTS-frequency* that determines how often the inner problem is solved. If $T = 1$, the TTS optimization reduces to the standard nested optimization. In numerical testing, we observed that TTS optimization reduces the cost of optimizing ρKG roughly by a factor of T. In our experiments, we used $T = 10$ without any noticeable change in algorithm performance. Notice that the only ρKG specific part of the TTS optimization is the inner optimizer and the resulting derivative. By modifying these parts, the TTS optimization can be used with many BO algorithms that involve a nested optimization, including all the knowledge gradient algorithms.

We mentioned earlier that the TTS optimization would address the *instability* we faced with the one-shot optimization. Going back to our example with one good and one bad candidate, TTS, by periodically solving the inner problem globally for each candidate, ensures that each candidates acquisition function value is calculated using a global solution, reducing the chances of misidentifying the bad candidate as the better one.

We conclude by noting that the TTS implementation presented here is a primitive one, and has lots of room for improvement. We would also like to point out that our discussion of one-shot optimization is mostly an observational one, based on certain shortcomings we faced. An in-depth comparison and further development of the two optimization methods is left as a subject for future research.

3 Experiments

3.1 Algorithm Parameters

- Common Parameters: Each algorithm is optimized using L-BFGS-B algorithm [3] (except for Covid-19 where we use SLSQP [?] due to the linear constraint on the decision variables), with $10 \times (d^{\mathcal{X}} + d^{\mathcal{W}})$ random restart points which are selected from $500 \times (d^{\mathcal{X}} + d^{\mathcal{W}})$ raw samples. We use low-discrepancy Sobol sequences [4] to generate the raw samples. The restart points are selected using a heuristic implemented in the BoTorch package [2], which assigns each raw sample a weight that is an exponential factor of its acquisition function value, then selects the restart points with probability proportional to

this weight.

- EI: Does not have any specific parameters.
- MES: A set of $500 \times (d^{\mathcal{X}} + d^{\mathcal{W}})$ randomly drawn points is used to discretize the design space, and the max values are sampled over this discretization using the Gumbel approximation. We draw 10 max value samples over the discretization, and use 128 fantasies to compute the approximate information gain from using the candidate.
- KG: We use the one-shot KG implementation with 64 fantasies.
- UCB: We use parameter $\beta = 0.2$, i.e. the acquisition function is given by $\text{UCB}(x) = \mu_n(x) + \sqrt{0.2}\Sigma_n(x, x)$.
- ρKG and ρKG^{apx}: We use $K = 10$ fantasy models, and $M = 40$ sample paths per fantasy. For the inner optimization problem of ρKG, we use $5 \times d^{\mathcal{X}}$ random restart points selected from $50 \times d^{\mathcal{X}}$ raw samples. We use TTS optimization with $T = 10$.

All fantasy models and sample paths are generated using low-discrepancy Sobol sequences [4]. For our algorithms, we observed a small improvement when increasing the number of fantasy models, however, this comes with a linear increase in run time and memory usage. The number of sample paths were chosen rather arbitrarily since the majority of computational effort is spent in calculating the Cholesky factor, which is then used for generating all sample paths. We observed very similar performance using $M = 4$ and $M = 40$, and went with the larger value as since it only increased the run time by about 50%.

3.2 GP Model and Fitting

We use the "SingleTaskGP" model from the BoTorch package [2] with the given priors on hyper-parameters, which, quoting from the code base, "... work best when covariates are normalized to the unit cube and outcomes are standardized (zero mean, unit variance).". We use the Matérn 5/2 kernel and fit the hyper-parameters using MAP estimation.

We project the function domain $\mathcal{X} \times \mathcal{W}$ to the unit hypercube $[0, 1]^{d^{\mathcal{X}}+d^{\mathcal{W}}}$, and all the calculations are done on the unit hypercube. For for calculations involving the GP, we use the "Standardize" transformation available in BoTorch which transforms the outcomes (observations) into zero mean, unit variance observations for better compatibility with the hyper-parameter priors.

3.3 Synthetic Test Problems

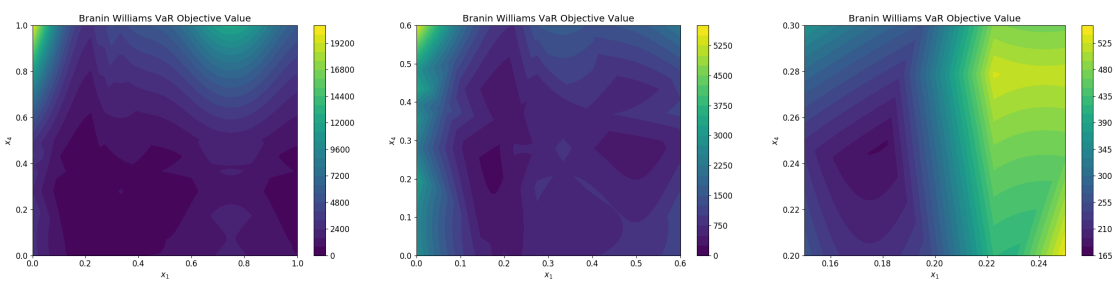

Figure 1: The VaR objective of Branin Williams function. Left to right, full domain and close-ups of the optimal region. It is notable that the objective function is quite flat across most of its domain, making it challenging to identify the optimal solution using a small number of evaluations.

The first two problems we use are synthetic test functions from the BO literature. The first problem is the 4 dimensional Branin-Williams problem formulated in [5]. For $x \in [0, 1]^4$, the function is defined as

Table 1: Probability distribution of x_2 and x_3 in Branin Williams problem.

		x_3			
		0.2	0.4	0.6	0.8
	0.25	0.0375	0.0875	0.0875	0.0375
x_2	0.5	0.0750	0.1750	0.1750	0.0750
	0.75	0.0375	0.0875	0.0875	0.0375

$y(x) = y_b(15x_1 - 5, 15x_2) \times y_b(15x_3 - 5, 15x_4)$, where

$$y_b(u, v) = \left(v - \frac{5.1}{4\pi^2}u^2 + \frac{5}{\pi}u - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(u) + 10.$$

The function is observed with additive Gaussian noise with standard deviation 10. The decision variables are taken to be x_1 and x_4, and the context variables are $w = (x_2, x_3)$. The set $\mathcal{W}$ of size 12 and the corresponding probability distribution is provided in the supplementary material.

The problem is originally formulated for the expectation objective, however, here we consider the minimization of VaR at risk level $\alpha = 0.7$. A plot of the VaR objective of the Branin Willams function is given in Figure 1.

The second problem we consider is $f_6(x_c, x_e)$ function from [6] given by

$$\begin{aligned} f_6(x_c, x_e) &= x_{e1}\left(x_{c1}^2 - x_{c2} + x_{c3} - x_{c4} + 2\right) + x_{e2}\left(-x_{c1} + 2x_{c2}^2 - x_{c3}^2 + 2x_{c4} + 1\right) \\ &\quad + x_{e3}\left(2x_{c1} - x_{c2} + 2x_{c3} - x_{c4}^2 + 5\right) + 5x_{c1}^2 + 4x_{c2}^2 + 3x_{c3}^2 + 2x_{c4}^2 - \sum_{i=1}^{2} x_{ei}^2, \end{aligned}$$

where $x_c \in [-5, 5]^4$ are the decision variables and $x_e \in [-2, 2]^3$ are the context variables. For the experiments, the function evaluations are with additive Gaussian noise with standard deviation of 1. We formulate this problem for minimization of CVaR at risk level α where $w = x_e$ has a continuous uniform distribution over its domain. For the computation of ρKG and ρKG^{apx}, we use a randomly drawn $\widetilde{\mathcal{W}}$ of size 40 at each iteration, and use $\widetilde{\mathcal{W}}$ of size 8 for evaluating $\text{CVaR}[F(x, W)]$ in benchmarks.

3.4 Experiment Details

To initialize each experiment, we draw $2d^{\mathcal{X}} + 2$ random samples from $x_i \in \mathcal{X}$, and for each x_i we evaluate $F(x_i, w), w \in \widetilde{\mathcal{W}}$ where $\widetilde{\mathcal{W}}$ is the set corresponding to the benchmark algorithms as described below. For ρKG and ρKG^{apx}, we use these $(2d^{\mathcal{X}} + 2) \times |\widetilde{W}|$ evaluations for initialization, and for the benchmarks we use $2^{\mathcal{X}} + 2$ $\rho[F(x_i, W)]$ calculated using these observations for initialization. These random samples are redrawn for each replication of the experiments, and are synchronized across different algorithms, i.e. k^{th} replication of each experiment uses the same set of samples.

In what follows, we describe the use of $\mathcal{W}$ in each experiment. Note that all the random sets $\widetilde{\mathcal{W}}$ are independently redrawn at each iteration.

1. Branin-Williams: The set $\mathcal{W}$ and the corresponding probability distribution is given in Table 1. Our algorithms use the full set $\mathcal{W}$ in the inner computations, and all the benchmarks evaluate $\text{VaR}_{0.7}[F(x, W)]$ using the full $\mathcal{W}$, thus using 12 evaluations per iteration.

2. $f_6(x_c, x_e)$: The set $\mathcal{W}$ is the hyper-rectangle $[-2, 2]^3$ with a continuous uniform distribution. At each iteration of our algorithms, we sample a subset $\widetilde{\mathcal{W}}$ of size 40, and use this for the inner computations. The w component of the candidate is not restricted to this subset $\widetilde{\mathcal{W}}$, and can take any value in $\mathcal{W}$. For each evaluation of the benchmarks, we sample a subset $\widetilde{\mathcal{W}}$ of size 10, and calculate $\text{CVaR}_{0.75}[F(x, W)]$ on this set, thus using 10 evaluations per iteration.

3. Portfolio Optimization: The set

$$\mathcal{W} = \{w \in \mathbb{R}^2 : w_1 \in [10^{-4}, 10^{-2}], w_2 \in [10^{-4}, 10^{-3}]\}$$

gives the range of bid-ask spread (w_1) and the borrowing cost (w_2), and W follows a uniform distribution on $\mathcal{W}$. For the inner computations of our algorithms, we use a randomly drawn subset $\widetilde{\mathcal{W}}$ of size 40, and use a subset $\widetilde{\mathcal{W}}$ of size 10 for benchmark evaluations of $\mathrm{VaR}_{0.8}[F(x,w)]$, thus using 10 evaluations per iteration.

4. Covid-19: The set $\mathcal{W}$ of initial disease propensity is given by

$$\mathcal{W} = \left\{ w \in \mathbb{R}^3 : \begin{array}{c} w_1 \in \{0.001, 0.0025, 0.004\}, \\ w_2 \in \{0.002, 0.004, 0.006\}, \\ w_3 \in \{0.002, 0.005, 0.008\} \end{array} \right\}$$

with the probability distribution given by

$$P(w) = 2^{-8+\mathbb{1}(w_1=0.0025)+\mathbb{1}(w_2=0.004)+\mathbb{1}(w_3=0.005)}.$$

For the inner computation of $\rho\mathrm{KG}^{apx}$, we use the full set $\mathcal{W}$. For evaluations of the benchmarks, we draw a random sample $\widetilde{\mathcal{W}}$ of size 10 and use this to calculate $\mathrm{CVaR}[F(x,W)]$, thus using 10 evaluations per iteration.

The output of the Covid-19 simulator is stochastic with a large variation. To enable fair comparison between the algorithms, we fix a set of 10 seeds, and each function evaluation is done using a randomly selected seed from this set. The reported performance is computed as the average performance of using all 10 seeds.

We ran the benchmark algorithms EI, MES, KG, UCB and random for 100 replications in each experiment. For $\rho-$random, we ran 100 replications in all experiments except for Covid-19 where we used 50 replications. We ran 10 replications of ρKG in all experiments except Covid-19; and ran 20 replications of $\rho\mathrm{KG}^{apx}$ in all experiments except Covid-19, where we ran 30 replications.

3.5 Results

In the main body of the paper, we present the results using a moving average window of 3 iterations, to smooth out the oscillations and make the plots more readable. In Figure 2, we present the non-smoothed plots to address any questions this may raise. The comparatively large variance of our algorithms is due to: (i) having roughly 10 times as many data points; (ii) having fewer replications than the benchmark algorithms.

4 Proofs

We start with a simple result showing that the value of information defined by ρKG is non-negative.

$$\begin{aligned} \mathbb{E}_n[\min_x \mathbb{E}_n[\rho[F(x,W)]]] &\geq \mathbb{E}_n[\mathbb{E}_n[\min_x \mathbb{E}_{n+1}[\rho[F(x,W)]] \mid (x_{n+1}, w_{n+1})]] \\ &= \mathbb{E}_n[\min_x \mathbb{E}_{n+1}[\rho[F(x,W)]] \mid (x_{n+1}, w_{n+1})] \end{aligned}$$

The rest of the section is dedicated to showing that the derivative estimators we propose in this paper are asymptotically unbiased and consistent. Here, we only present the proofs for the case of $d^{\mathcal{W}} = 1$.

The derivative of a GP is again a GP (cf. [7] and section 9.4 of [8]). If we model $F(x)$ as a GP with mean and covariance functions μ, Σ, then F and ∇F follow a multi output GP with mean and covariance functions given by [7]

$$\widetilde{\mu}(x) = (\mu(x), \nabla\mu(x))^\top \text{ and } \widetilde{\Sigma}(x,x') = \begin{pmatrix} \Sigma(x,x') & \nabla\Sigma(x,x') \\ \nabla\Sigma(x',x)^\top & \nabla^2\Sigma(x,x') \end{pmatrix};$$

where $\nabla\Sigma$ and $\nabla^2\Sigma$ are the Jacobian and Hessian of the covariance function. We note that, although this result is found in the literature without any assumptions on the GP, it in fact requires the GP to possess differentiable sample paths. For example, the Brownian motion is a well known example of a GP that has nowhere differentiable sample paths, thus violates this framework. To this end, we refer the reader to [9] for precise conditions on the differentiability of the GP sample paths, and *assume* that these conditions are satisfied.

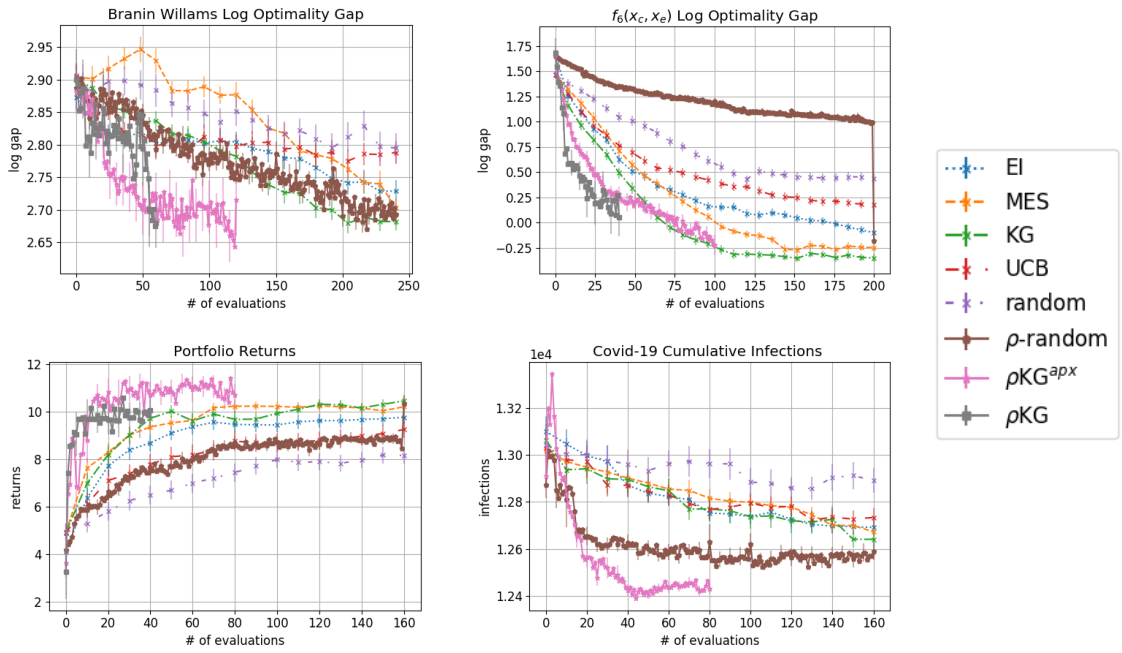

Figure 2: Top: The log optimality gap in Branin Williams (left) and $f_6(x_c, x_e)$ (right). Bottom left: The returns in Portfolio problem. Bottom right: Cumulative number of infections in the COVID-19 problem. The plots are plotted against the number of $F(x, w)$ evaluations.

Remark 1. *Before discussing the assumptions below, we note that the assumptions listed here are more restrictive than needed. The main results we build on require certain conditions on the distribution of the function in a neighborhood of its VaR. However, when the function is draw from a GP, it is impossible to know where this neighborhood, which would be different for each draw, might be, and thus it is impossible to impose assumptions on such an unknown and ever changing neighborhood. The main purpose of the results presented here is to show that there indeed exists a set of conditions under which our claims hold. However, we are confident that our algorithms are applicable in much broader settings where no such assumption can be verified.*

We *assume* that the kernel $\widetilde{\Sigma}$ of the joint GP of the GP and its derivative is strictly positive definite; and the random context variable W admits a strictly positive and continuous density $p(w)$ over its domain. Observe that when the kernel is strictly positive definite, the sample paths of the GP are non-zero w.p.1., i.e. the equation $F(x, w) = 0$ has at most countably many solutions, where $F(\cdot, \cdot)$ denotes a sample drawn from the GP.

Lemma 1. *Under conditions outlined above, for any t such that $P_W(F(x, W) \leq t) \in (0, 1)$, the CDF $P_W(F(x, W) \leq t)$ is continuously differentiable w.r.t. both x and t; and the density $\partial_t P_W(F(x, W) \leq t)$ is strictly positive for each x.*

Proof. Since there are at most countably many points where the derivative is zero, we can construct distinct intervals $\mathcal{W}_i, i \in I$ where $F(x, w)$ is monotone (in w) in each $\mathcal{W}_i$ and $\mathcal{W}_i$ are such that $P_W(\cup_{i \in I} \mathcal{W}_i) = 1$ and

$P_W(\mathcal{W}_i) > 0, i \in I$. Then,

$$\begin{aligned}
\partial_x P_W(F(x,W) \le t) &= \partial_x \int_{\{w:F(x,w)\le t\}} p(w)dw \\
&= \partial_x \sum_{i\in I} P_W(\mathcal{W}_i) \int_{\{w\in\mathcal{W}_i:F(x,w)\le t\}} \frac{p(w)}{P_W(\mathcal{W}_i)} dw \\
&= \sum_{i\in I} \partial_x P_W(\mathcal{W}_i) \int_{\{w\in\mathcal{W}_i:F(x,w)\le t\}} p(w)dw \qquad = (*)
\end{aligned}$$

where the interchange of derivative and summation is justified as the sum is bounded. From here on, we ignore the $\mathcal{W}_i$ that do not produce any root to the equation $F(x,w) = t$, as the corresponding derivatives are zero. Since $F(x,w)$ is continuously differentiable and monotone in w in each $\mathcal{W}_i$, by the inverse function theorem [10], there exists a continuously differentiable inverse function $F_i^{-1}(t';x)$ mapping t' to $w \in \mathcal{W}_i$ for a given x. For the remaining $\mathcal{W}_i$, define the sets I^+ and I^- such that, I^+ is the set of $\mathcal{W}_i$ that has $F_i^{-1}(t;x)$ as the upper bound of the integral and I^- as the ones with the lower bound. For $i \in I^+$, let w_i^- be the lower bound of the integral, and similarly w_i^+ as the upper bound for I^-. Then,

$$\begin{aligned}
(*) &= \sum_{i\in I^+} \partial_x \int_{w_i^-}^{F_i^{-1}(t;x)} p(w)dw + \sum_{i\in I^-} \int_{F^{-1}(t;x)}^{w_i^+} p(w)dw \\
&= \sum_{i\in I^+} p(w)\partial_x F_i^{-1}(t;x) - \sum_{i\in I^-} -p(w)\partial_x F_i^{-1}(t;x) \\
&= \sum_{i\in I^+\cup I^-} p(w)\partial_x F_i^{-1}(t;x)
\end{aligned}$$

where the result follows by the Leibniz rule. Applying the same steps with ∂_t, we get

$$\partial_t P_W(F(x,W) \le t) = \sum_{i\in I^+\cup I^-} p(w)\partial_t F_i^{-1}(t;x).$$

Since $p(w)$ is continuous and $F_i^{-1}(t;x)$ is continuously differentiable, $P_W(F(x,W) \le t)$ is continuously differentiable w.r.t. both x and t. Since $p(w)$ is strictly positive and the derivative is non-zero, it follows that $\partial_t P_W(F(x,W) \le t)$ is strictly positive, thus completing the proof. □

Proposition 1. *Under the conditions outlined above, for $\alpha \in (0,1)$,*

$$\frac{1}{M}\sum_{j=1}^{M} \nabla_x \widehat{v}^j(x) \xrightarrow{p} \nabla_x \mathbb{E}_{n+1}[\,VaR_\alpha[F(x,W)]]$$

as $M, L \to \infty$ where $\xrightarrow{p}$ denotes convergence in probability. Moreover, the gradient $\nabla_x \mathbb{E}_{n+1}[\,VaR_\alpha[F(x,w)]]$ is continuous.

Proof. As a sample path of the GP, $F(x,w)(\omega)$ is continuously differentiable w.r.t. w w.p.1. By Lemma 1, the CDF $P_W(F(x,W) \le t)$ is continuously differentiable w.r.t. both x and t, and the density $\partial_t P_W(F(x,W) \le t)$ is strictly positive. Then, by Lemma 2 of [11],

$$\lim_{L\to\infty} \mathbb{E}[\nabla_x \widehat{v}(x)] = \nabla_x \mathrm{VaR}_\alpha[F(x,W)]. \tag{1}$$

Since the gradient is also a GP defined in a compact space, $\nabla_x F(x,w)(\omega)$ is bounded w.p.1., so is $\nabla_x \mathrm{VaR}_\alpha[F(x,W)]$. Then, by proposition 1 of [12],

$$\nabla_x \mathbb{E}_{n+1}[\mathrm{VaR}_\alpha[F(x,W)]] = \mathbb{E}_{n+1}[\nabla_x \mathrm{VaR}_\alpha[F(x,W)]]. \tag{2}$$

$F(x,w)$ is bounded by the same argument as $\nabla_x F(x,w)$, thus has finite second moment. Therefore, the main result follows by an application of Chebyshev's inequality. The continuity of $\nabla_x \mathbb{E}_{n+1}[\mathrm{VaR}_\alpha[F(x,W)]]$ follows by Theorem 1 of [11] and the fact that both gradients are continuous by Lemma 1. □

Before going into the main result, we discuss the mapping from the candidate point (x, w) to the sample $F^{ij}(\cdot,\cdot)$. Let's start with the mapping from the candidate to the mean and covariance functions of GP_f^i. Let Z_i be the (a fixed realization of) standard normal random variable used to generate the fantasy. The mean and covariance functions of GP_f^i are given by

$$\mu_f^i(x', w') = \mu_n(x', w') + \frac{\Sigma_n(x', w', x, w)}{\sqrt{\Sigma_n(x, w, x, w) + \sigma^2(x, w)}} Z_i,$$
$$\Sigma_f^i(x', w', x'', w'') = \Sigma_n(x', w', x'', w'') - \frac{\Sigma_n(x', w', x, w)\Sigma_n(x, w, x'', w'')}{\Sigma_n(x, w, x, w) + \sigma^2(x, w)},$$

where $\sigma^2(x, w)$ is the known observation noise level (see [13] for more details). *Assuming* that the prior mean and covariance functions are continuously differentiable, μ_f^i and Σ_f^i are continuously differentiable w.r.t. the candidate (x, w).

For a given x_i and finite set $\widetilde{W} = \{w_{1:L}\}$, and a (realization of) L - dimensional standard normal random vector Z_L^{ij}, the jth sample path of ith fantasy model is generated as

$$\widehat{F}^{ij}(x_i, w_{1:L}) = \mu_f^i(x_i, w_{1:L}) + \mathbf{C}(x_i, w_{1:L}) Z_L^{ij}, \tag{3}$$

where $\mathbf{C}(x_i, w_{1:L})$ is the Cholesky factor of $L \times L$ covariance matrix $\Sigma_f^i(x_i, w_{1:L}, x_i, w_{1:L})$. Since Σ_f^i is continuously differentiable w.r.t. the candidate, so is $\mathbb{C}(x_i, w_{1:L})$, making the sample $\widehat{F}^{ij}(\cdot,\cdot)$ continuously differentiable w.r.t. the candidate (x, w).

The discussion above establishes continuous differentiability of the sample $\widehat{F}^{ij}(\cdot,\cdot)$ w.r.t. the candidate (x, w) only for a finite collection of points $(x_i, w_{1:L})$. However, to be able to use the analysis of Proposition 1, we need continuous differentiability of $\widehat{F}^{ij}(x, w), w \in W$ which, with W being a continuous random variable, is an infinite dimensional random variable. The concept of an infinite dimensional standard Gaussian random variable, or a standard Gaussian random function, is not well defined, and we can no longer use equation (3). In the proposition below, we *assume* that $F(\cdot,\cdot)$ is indeed differentiable w.r.t. the candidate (x, w) and hope that this discussion justifies the assumption. Below, we use $u := (x, w)$ as the candidate to simplify the notation.

Proposition 2. *Under the conditions outlined above, and assuming that the solution to the inner problem x^* is unique w.p.1.,*

$$\frac{1}{K}\sum_{i=1}^{K} -\frac{1}{M}\sum_{j=1}^{M} \nabla_u \widehat{v}^j(x_i^*) \xrightarrow{p} \nabla_u \mathbb{E}_n[-\min_x \mathbb{E}_{n+1}[\,VaR_\alpha[F(x, w)]] \mid u] \tag{4}$$

as $K, L, M \to \infty$.

Proof. Following the proof of proposition 1 with ∇_u replacing ∇_x, we get

$$\lim_{L\to\infty} \mathbb{E}_{n+1}[\mathbb{E}[\nabla_u \widehat{v}(x)]] = \nabla_u \mathbb{E}_{n+1}[\mathrm{VaR}_\alpha[F(x, W)]].$$

By the assumption that x^* is unique w.p.1. and the continuity of $\nabla_u E_{n+1}[\mathrm{VaR}_\alpha[F(x, W)]]$ by proposition 1, we can apply the envelope theorem (corollary 4, [14]) to get

$$\lim_{L\to\infty} \mathbb{E}_{n+1}[\mathbb{E}[\nabla_u \widehat{v}(x^*)]] = \nabla_u \mathbb{E}_{n+1}[\mathrm{VaR}_\alpha[F(x^*, W)]].$$

Again by the arguments in the proof of proposition 1, the gradient is continuous and bounded w.p.1., thus applying proposition 1 of [12], we get

$$\lim_{L\to\infty} \mathbb{E}_n[\mathbb{E}_{n+1}[\mathbb{E}[\nabla_u \widehat{v}(x^*)]] \mid u] = \nabla_u \mathbb{E}_n[\mathbb{E}_{n+1}[\mathrm{VaR}_\alpha[F(x^*, W)]] \mid u]. \tag{5}$$

Since the second moment of the estimators are bounded (see the proof of proposition 1), the result follows by Chebyshev's inequality. □

Proposition 3. *The results presented in Propositions 1 & 2 extend to the risk measure CVaR, i.e.*

$$\frac{1}{M}\sum_{j=1}^{M}\nabla_x \widehat{c}^j(x) \xrightarrow{p} \nabla_x \mathbb{E}_{n+1}[CVaR_\alpha[F(x,W)]];$$

and

$$\frac{1}{K}\sum_{i=1}^{K} -\frac{1}{M}\sum_{j=1}^{M}\nabla_u \widehat{c}^j(x_i^*) \xrightarrow{p} \nabla_u \mathbb{E}_n[-\min_x \mathbb{E}_{n+1}[CVaR_\alpha[F(x,w)]] \mid u] \tag{6}$$

as $K, L, M \to \infty$.

Proof. We have established in the proof of Proposition 1 that $\nabla F(x,w)$ is bounded, and that $\mathrm{VaR}_\alpha[F(x,W)]$ is continuously differentiable. Combining these with the assumption that W is a continuous random variable, we can apply Theorem 3.1 of [15] to get

$$\nabla \mathrm{CVaR}_\alpha[F(x,W)] = \mathbb{E}[\nabla F(x,W) \mid F(x,W) \geq \mathrm{VaR}_\alpha[F(x,W)]].$$

Since $\nabla F(x,W)$ is continuous and bounded, it follows by bounded convergence theorem [**?**] that $\mathrm{CVaR}_\alpha[F(x,W)]$ is continuously differentiable. The remaining arguments are identical to those of Propositions 1 & 2. □

We conclude the proofs by noting that while, the proposition in the main paper claims both asymptotic unbiasedness and consistency, the propositions proved here only establish the consistency of the estimators, and the asymptotic unbiasedness result is hidden within the proof argument. In the proof of Proposition 1, combining the equations (1) and (2) gives us the asymptotic unbiasedness of the gradient estimators for the inner problem, where the interchange of limit and expectation is justified by the dominated convergence theorem [**?**]. Similarly, equation (5) shows that the estimators for the candidate optimization of ρKG are asymptotically unbiased.

References

[1] J. Wu and P. 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, (USA), pp. 3134–3142, Curran Associates Inc., 2016.

[2] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "BoTorch: Programmable Bayesian Optimization in PyTorch," *arxiv e-prints*, 2019.

[3] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Softw.*, vol. 23, p. 550–560, Dec. 1997.

[4] A. B. Owen, "Scrambling sobol' and niederreiter–xing points," *Journal of Complexity*, vol. 14, no. 4, pp. 466 – 489, 1998.

[5] B. Williams, T. Santner, and W. Notz, "Sequential design of computer experiments to minimize integrated response functions," *Statistica Sinica*, vol. 10, pp. 1133–1152, 10 2000.

[6] J. Marzat, E. Walter, and H. Piet-Lahanier, "Worst-case global optimization of black-box functions through kriging and relaxation," *Journal of Global Optimization*, vol. 55, pp. 707–727, 04 2013.

[7] J. Wu, M. Poloczek, A. G. Wilson, and P. I. Frazier, "Bayesian optimization with gradients," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 5273–5284, Curran Associates Inc., 2017.

[8] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

[9] R. J. Adler and J. E. Taylor, *Random Fields and Geometry*. Springer, 2007.

[10] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill Inc, 1976.

[11] G. Jiang and M. C. Fu, "Technical note—on estimating quantile sensitivities via infinitesimal perturbation analysis," *Operations Research*, vol. 63, no. 2, pp. 435–441, 2015.

[12] M. Broadie and P. Glasserman, "Estimating security price derivatives using simulation," *Management Science*, vol. 42, no. 2, pp. 269–285, 1996.

[13] J. Wu and P. 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, (Red Hook, NY, USA), p. 3134–3142, Curran Associates Inc., 2016.

[14] P. Milgrom and I. Segal, "Envelope theorems for arbitrary choice sets," *Econometrica*, vol. 70, no. 2, pp. 583–601, 2002.

[15] L. J. Hong and G. Liu, "Simulating sensitivities of conditional value at risk," *Management Science*, vol. 55, no. 2, pp. 281–293, 2009.