Engineering Optimization

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

Using the knowledge gradient acquisition function in Bayesian optimization when searching for robust solutions

Hoai Phuong Le & Juergen Branke

To cite this article: Hoai Phuong Le & Juergen Branke (2022): Using the knowledge gradient acquisition function in Bayesian optimization when searching for robust solutions, Engineering Optimization, DOI: 10.1080/0305215X.2022.2145604

To link to this article: https://doi.org/10.1080/0305215X.2022.2145604

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2022.2145604

RESEARCH ARTICLE

Using the knowledge gradient acquisition function in Bayesian optimization when searching for robust solutions

Hoai Phuong Le[a] and Juergen Branke[b]

[a]Mathematics for Real-World Systems, University of Warwick, Coventry, UK; [b]Warwick Business School, University of Warwick, Coventry, UK

ABSTRACT
This article considers the use of Bayesian optimization to identify robust solutions, where robust means having a high expected performance given disturbances over the decision variables and independent noise in the output. A variant of the well-known knowledge gradient acquisition function is proposed specifically to search for robust solutions, with analytic expressions for uniformly and normally distributed disturbances. An empirical evaluation on a number of test problems demonstrates that the new acquisition function outperforms alternative approaches.

ARTICLE HISTORY
Received 4 January 2022
Accepted 3 September 2022

KEYWORDS
Reliability; Gaussian process; Bayesian optimization; robustness

1. Introduction

The problem of expensive black-box optimization is considered where the goal is to identify a robust solution, robust meaning here having a high expected performance despite disturbances over the decision variables and noise in the output.

Many engineering optimization problems can only be formulated as black-box functions, *e.g.* if the problem involves a simulation to evaluate solutions. Such problems are challenging to optimize, in particular if the evaluation of a solution is costly or computationally expensive, as is typical in engineering. In such cases, the budget for function evaluations is very limited. Bayesian optimization (Frazier 2018) has been shown to be a powerful technique of black-box optimization, especially for problems with a limited budget of function evaluations. It is an optimization method at the interface of machine learning that iteratively builds or updates a surrogate model and uses an acquisition function to determine the solution that would give the biggest additional information value when being added to the current data set.

When optimizing complicated systems, it is not always practicable to guarantee that the implemented solution follows exactly the design specification, and sometimes the environment where the solution is deployed is unpredictable. For instance, in engineering, manufacturing tolerances mean that the produced solution may deviate from the designed solution. In such situations, a solution is usually sought that is not merely good but is also robust.

This article is aimed at finding solutions having a good expected performance even with the presence of disturbances of the decision variables and output noise. Essentially, this means that not only the solution should be good, but its neighbourhood should also have high average quality (Branke 1998).

CONTACT H. P. Le Hoai-Phuong.Le@warwick.ac.uk

Supplemental data for this article can be accessed here. https://doi.org/10.1080/0305215X.2022.2145604

One of the simple ways to evaluate the expected performance of a solution is to estimate it directly by sampling from the distribution of disturbances and averaging. However this is of course very computationally expensive, particularly in higher dimensional search spaces and when the output is noisy, since the method would need many samples for an accurate estimation.

In Le and Branke (2020), the robust Knowledge Gradient (rKG) acquisition function has been suggested for Bayesian optimization, an adaptation of a method for simulation optimization with input uncertainty (Pearce and Branke 2017). This article summarizes the previous work and makes the following original contributions:

- The rKG is extended to normally distributed disturbances and an analytic expression for this case is derived.
- The consistency of the algorithm is proved.
- The proposed algorithm is empirically compared with a recently published benchmark, NES_EP, on a number of additional test functions and it is demonstrated that it outperforms alternative state-of-the-art algorithms.

The article is organized as follows. It begin with an overview of related literature on robust optimization in Section 2. The problem is formally defined in Section 3 and the method for simulation optimization for robust solutions is explained in Section 4, followed by empirical tests and comparison with some benchmarks in Section 5. Finally, Section 6 consists of conclusions and suggestions for future work.

2. Literature review

Bayesian optimization has recently become a popular method for solving many kinds of problem involving costly-to-evaluate functions. In every iteration, a response surface (often called a surrogate model, emulator or metamodel) is fitted to the data collected so far, with Gaussian Processes (GPs) being the most commonly used metamodel. Based on the information given by the metamodel, the next sample is chosen, maximizing a so-called infill criterion or acquisition function. The most popular acquisition function is Expected Improvement (EI), first proposed by Mockus, Tiesis, and Zilinskas (1978) and popularized in Efficient Global Optimization (EGO) (Jones, Schonlau, and Welch 1998). Other examples are Upper Confidence Bound (Srinivas *et al.* 2010), Knowledge Gradient (Scott, Frazier, and Powell 2011) and Entropy Search (Hennig and Schuler 2012). A variant of EI with an adaptive exploration component has recently been proposed by Xu, Guo, and Saleh (2021).

There are a large number of publications regarding robust global optimization, but mostly based on evolutionary algorithms (Beyer and Sendhoff 2007). So far, Bayesian optimization for solving that problem has not been applied widely. Articles can be split into articles looking for worst-case robustness, where disturbance of the decision variable is defined as a compact set, and those that look for good expected performance given a probability distribution of disturbances, as in this article. There is also the related area of reliability based design optimization where the goal is to identify the best solution that remains feasible despite disturbances to the decision variables, see *e.g.* Yang *et al.* (2016).

Among the articles searching for solutions robust with respect to the worst case, Marzat, Walter, and Piet-Lahanier (2013) propose a combination of Bayesian optimization and an iterative relaxation procedure. The basic idea is to maintain a discrete set of disturbances, and alternate between seeking the robust optimal solution given the disturbance set, and seeking a new worst-case disturbance to be added to the disturbance set. In each of the two alternating optimization steps, expected improvement-based algorithms are used. Note that, in this article, it is not the decision variables that are disturbed, but the simulation model's input parameters (Pearce and Branke 2017).

ur Rehman, Langelaar, and Keulen (2014) and ur Rehman and Langelaar (2017) modify the EI acquisition function to the case of searching for a robust solution in the worst-case context. First, the nominal optimum is computed on the constructed metamodel. It is the solution with the best worst-case prediction, not the best found solution. Then the next sampling location is determined by a modified EI. For any position x in the design variable space, according to the metamodel, the worst-case x_w in the disturbance region is identified, and then at that worst-case location x_w the predicted performance distribution is used to evaluate the acquisition function. The solution is the robust optimum of the metamodel in the last iteration. The proposed approach appeared to be much more sample-efficient than the method from Marzat, Walter, and Piet-Lahanier (2013) in an empirical comparison on some benchmark problems. ur Rehman and Langelaar (2017) deviate from ur Rehman, Langelaar, and Keulen (2014) by considering constrained problems instead of unconstrained ones.

Sanders *et al.* (2019) also identify the current best robust solution with respect to the worst case within the disturbance region, using the posterior mean of the constructed GP. A number of realizations (functions) are drawn from the fitted GP to determine the next disturbance region to sample in. The actual improvement over the current best solution for each realization is computed and the average of the improvements over all function realizations will be the overall expected improvement of a solution. The solution with the largest expected improvement is sought using an evolutionary algorithm. The next observation of the target function is made within the disturbance region of this solution, at the location with the largest variance as predicted by the Gaussian process. It is shown empirically on a number of benchmark problems that the method performs significantly better in comparison with the algorithm suggested by ur Rehman, Langelaar, and Keulen (2014). While the article focuses on worst-case performance, the authors mention that their algorithm, with a minor modification, could also be used for optimizing expected performance over a disturbance region.

Nogueira *et al.* (2016) propose to approximate the expectation of the EI acquisition function given the distribution of disturbances. This so-called unscented expected improvement is computed using the unscented transformation (Julier and Uhlmann 2004).

Two very recent articles that optimize expected performance over the input disturbances have been published by Fröhlich *et al.* (2020) and Iwazaki, Inatsu, and Takeuchi (2020). The first article considers the same problem as in this article, but adapts the Entropy Search acquisition function (Hennig and Schuler 2012). It is shown to outperform the unscented expected improvement by Nogueira *et al.* (2016). The latter adapts the Upper Confidence Bound acquisition function (Srinivas *et al.* 2010) to search for a robust solution in a multi-objective context considering both mean and variance. Thus it is similar to this work when focusing on only the mean. Both articles assume normally distributed disturbances, whereas the rKG algorithm is based on the KG idea and tested with both uniformly and normally distributed disturbances.

The method is related to the method proposed for simulation optimization with input uncertainty (Pearce and Branke 2017) or optimizing expensive integrands (Toscano-Palmerin and Frazier 2018) but adapted to the case of searching for robust solutions.

3. Problem definition

Let black-box function f be defined over a compact input domain $X \subset \mathbb{R}^D$. There are two kinds of uncertainty, called *noise* and *disturbance*. Observing f at a solution x is noisy, *i.e.* $y = f(x) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. The white noise ϵ is assumed to have constant variance across the input domain. Additionally, the disturbance δ is distributed around each solution $x \in X$ with a probability distribution $\mathbb{P}[\delta]$. While in this article the approach is tested with uniformly and normally distributed disturbances, it also applies to other distributions of the disturbance, although this may require reverting to Monte Carlo estimation rather than analytical derivation of the acquisition function. The goal is to maximize the expected performance given noise and disturbances, *i.e.* to maximize the following *robustness function*:

$$\max_{x \in X} F(x) = \int_{\mathbb{R}^D} f(x+\delta)\mathbb{P}[\delta]\,\mathrm{d}\delta. \tag{1}$$

The Opportunity Cost (OC) is used as the measurement for the quality of the method. OC is the difference between the value of the robustness function at the true optimal robust solution and the solution that the algorithm recommends. Let x_r denote the final solution returned by the algorithm, the opportunity cost is then

$$\mathrm{OC}(x_r) = \max_{x'} F(x') - F(x_r). \tag{2}$$

It is assumed that the latent function f can be approximated by a Gaussian process reasonably well. There is a limited budget of N noisy observations of function f (N samples). The algorithm can choose the solution x^n at each iteration n to be sampled, dependent on the information collected so far. The goal is to sample solutions in such a way that minimizes the OC of the solution x_r that is returned at the end of optimization. The algorithm therefore needs to define where to sample next (*i.e.* what acquisition function to use) and what solution to return at the end.

4. Methodology

The method proposes to use Bayesian optimization, and in particular a variant of the KG acquisition function (Scott, Frazier, and Powell 2011), as an efficient way to search for robust solutions. The method uses a surrogate model and a novel acquisition function to define the next sample iteratively. The surrogate model and the acquisition function are described in the following subsections.

4.1. Gaussian process

A surrogate model is also called an emulator, metamodel or response surface. There are several choices, such as a Gaussian process or Tree Parzen estimator. In this approach, a Gaussian process is chosen following the majority of the literature. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams 2006).

It is characterized by its mean function $\mu_0(x)$ and kernel (covariance function) $\Sigma_0(x, x')$, where

$$\begin{aligned} \mu_0(x) &= \mathbb{E}[f(x)], \\ \Sigma_0(x, x') &= \mathbb{E}[(f(x) - \mu_0(x))(f(x') - \mu_0(x'))]. \end{aligned}$$

The widely used constant mean function and squared-exponential kernel are chosen, *i.e.*

$$\begin{aligned} \mu_0(x) &= \mu_0, \\ \Sigma_0(x, x') &= \alpha_0 \exp\left(-\frac{\|x - x'\|^2}{2l_x^2}\right). \end{aligned}$$

The choice of other mean functions and kernels is discussed by Rasmussen and Williams (2006).

Given the vector of observations $f(x^{1:n}) = (f(x^1), \dots, f(x^n))^{\mathrm{T}}$ at the vector of points $x^{1:n} = (x^1, \dots, x^n)^{\mathrm{T}}$, the posterior mean μ^n and posterior covariance Σ^n can be computed as follows:

$$\mu^n(x) = \Sigma_0(x, x^{1:n})(\Sigma_0(x^{1:n}, x^{1:n}) + \sigma_\epsilon^2 \mathbb{I}_n)^{-1}(f(x^{1:n}) - \mu_0(x^{1:n})) + \mu_0(x), \tag{3}$$

$$\Sigma^n(x', x) = \Sigma_0(x', x) - \Sigma_0(x', x^{1:n})(\Sigma_0(x^{1:n}, x^{1:n}) + \sigma_\epsilon^2 \mathbb{I}_n)^{-1}\Sigma_0(x^{1:n}, x), \tag{4}$$

where $\mathbb{I}_n$ is the identity matrix of size n and

$$\begin{aligned} \Sigma_0(x, x^{1:n}) &= (\Sigma_0(x, x^1), \dots, \Sigma_0(x, x^n)), \\ \Sigma_0(x^{1:n}, x) &= (\Sigma_0(x^1, x), \dots, \Sigma_0(x^n, x))^{\mathrm{T}}. \end{aligned}$$

The Gram matrix

$$\Sigma_0(x^{1:n}, x^{1:n}) = \begin{pmatrix} \Sigma_0(x^1, x^1) & \Sigma_0(x^1, x^2) & \dots & \Sigma_0(x^1, x^n) \\ \Sigma_0(x^2, x^1) & \Sigma_0(x^2, x^2) & \dots & \Sigma_0(x^2, x^n) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_0(x^n, x^1) & \Sigma_0(x^n, x^2) & \dots & \Sigma_0(x^n, x^n) \end{pmatrix}$$

is also called the covariance matrix and is positive semidefinite.

Maximum likelihood is usually used for tuning the hyperparameters of the model, which are signal variance α_0, lengthscale l_x and noise variance σ_ϵ.

4.2. Robust Bayesian optimization

4.2.1. Standard knowledge gradient

In Bayesian optimization, at each iteration, the point maximizing a so-called acquisition function is chosen as the next sampling point. One of the most widely used acquisition functions is the Knowledge Gradient (KG). Frazier, Powell, and Dayanik (2009) introduced KG for optimizing over a discrete and finite set which maximizes the expected improvement of the maximal value of the posterior mean conditioned on sampling once more at a specific point.

Let $\mathcal{F}^n$ denote the sigma-algebra generated by the data collected so far $\{(x^1, f(x^1)), \dots, (x^n, f(x^n))\}$. Denoting by $\mu^n(x)$ the posterior mean after n already taken samples $\{x^1, x^2, \dots, x^n\}$ and with the assumption that the next sample x^{n+1} will be at x, KG can be written as

$$\mathrm{KG}^n(x) := \mathbb{E}\left[\max_{x'} \mu^{n+1}(x') - \max_{x''} \mu^n(x'') | \mathcal{F}^n, x^{n+1} = x\right].$$

Scott, Frazier, and Powell (2011) present an extension of KG for correlated beliefs, KG for continuous parameters, which can be approximated by maximization over a finite subset of the input space. For instance, by discretizing X over a subset $X_{n_X} = \{x_1, \dots, x_{n_X}\} \subset X$, KG^n can be approximated as follows:

$$\begin{aligned} \mathrm{KG}^n(x) &= \mathbb{E}\left[\max_{i=1,\dots,n_X} \mu^{n+1}(x_i) - \max_{i=1,\dots,n_X} \mu^n(x_i) | \mathcal{F}^n, x^{n+1} = x\right] \\ &= \mathbb{E}[\max\{\mu_1 + Z\sigma_1, \dots, \mu_{n_X+1} + Z\sigma_{n_X+1}\} | \mathcal{F}^n, x^{n+1} = x], \end{aligned} \tag{5}$$

where $Z \sim \mathcal{N}(0, 1)$ and

$$\begin{aligned} \mu_i &= \mu^n(x_i) - x' \in X_{n_X} \cup \{x\} \mu^n(x'), \quad i = 1, \dots, n_X, \\ \sigma_i &= \tilde{\sigma}^n(x_i, x) = \frac{\Sigma^n(x_i, x)}{\sqrt{\sigma^n(x)^2 + \sigma_\epsilon^2}}, \quad i = 1, \dots, n_X, \\ \mu_{n_X+1} &= \mu^n(x) - \max_{x' \in X_{n_X} \cup \{x\}} \mu^n(x'), \\ \sigma_{n_X+1} &= \tilde{\sigma}^{n_X+1} = \frac{\Sigma^n(x, x)}{\sqrt{\sigma^n(x)^2 + \sigma_\epsilon^2}}. \end{aligned}$$

4.2.2. The Direct Robustness Approximation (DRA)

Perhaps the simplest way to apply Bayesian optimization to find the robust solution solving (1) is to estimate $F(x)$ by sampling over δ and to apply standard acquisition functions such as the standard KG. To reduce the impact of disturbances and noise, every observation can be averaged over k independent replications. This method is called Direct Robustness Approximation (DRA(k)) (Le and

Branke 2020). The idea is to approximate the robustness function $F(x)$ directly by the GP and each observation with random disturbance and output noise is taken as a sample of F.

The GP model for DRA(k) has always to allow for observation noise even if the underlying function f is deterministic, as observations are still stochastic owing to the random input disturbance. The method returns the solution with the best posterior mean of the approximated robustness function (Le and Branke 2020).

Because each observation is an average over multiple samples, the method is computationally expensive. The estimate of the solution quality can be improved if the k samples are drawn by Latin Hypercube Sampling (LHS) (McKay, Beckman, and Conover 1979) rather than random sampling (Le and Branke 2021).

4.3. *Robust knowledge gradient*

The robust Knowledge Gradient (rKG) is proposed as an acquisition function to identify robust solutions. The key insight is that disturbances can usually be controlled during optimization (*e.g.* when running a simulation model), and only the final solution is subject to disturbances. rKG thus uses a GP to approximate the latent f, not the robustness function F, and estimates values for F from evaluating f at locations without disturbance.

From the estimation of the underlying function f with $\mu^n(x)$, an approximation of the robustness function F by the robust mean is brought forward:

$$M^n(x) = \int_{\mathbb{R}^D} \mu^n(x+\delta)\mathbb{P}[\delta]\,\mathrm{d}\delta, \quad x \in X. \tag{6}$$

The rKG is then simply the expected value of the increase in maximal value of the posterior robust mean conditioned on sampling once more at a specific location. So the main difference between the standard KG and the rKG is that, instead of the posterior mean, the change in the posterior robustness mean from sampling a solution is looked at. Hence rKG after n samples for continuous parameters can be written as follows:

$$\mathrm{rKG}^n(x) := \mathbb{E}\left[\max_{x'} M^{n+1}(x') - \max_{x''} M^n(x'') | \mathcal{F}^n, x^{n+1} = x\right].$$

Similarly to the standard KG, the conditional mean can be rewritten as $\mu^{n+1}(x') = \mu^n(x') + \tilde{\sigma}^n(x', x)Z\,|\mathcal{F}^n,\ x^{n+1} = x$ with $Z \sim \mathcal{N}(0,1)$. Thus

$$M^{n+1}(x') = M^n(x') + \tilde{\Sigma}^n(x', x)Z,$$

where

$$\tilde{\Sigma}^n(x', x) = \int_{\mathbb{R}^D} \tilde{\sigma}^n(x'+\delta, x)\mathbb{P}[\delta]\,\mathrm{d}\delta, \quad x' \in X. \tag{7}$$

The robust KG, rKG(x), still has a formula similar to (5)

$$\begin{aligned}\mathrm{rKG}^n(x) &= \mathbb{E}\left[\max_{i=1,\dots,n_X} M^{n+1}(x_i) - \max_{i=1,\dots,n_X} M^n(x_i) | \mathcal{F}^n, x^{n+1} = x\right] \\ &= \mathbb{E}[\max\{M_1 + Z\tilde{\Sigma}_1, \dots, M_{n_X+1} + Z\tilde{\Sigma}_{n_X+1}\}],\end{aligned} \tag{8}$$

but with the adjusted components $M_i = M^n(x_i) - \max_{x' \in X_{n_X} \cup \{x\}} M^n(x')$, $M_{n_X+1} = M^n(x) - \max_{x' \in X_{n_X} \cup \{x\}} M^n(x')$ and $\tilde{\Sigma}_i = \tilde{\Sigma}^n(x_i, x)$, $\tilde{\Sigma}_{n_X+1} = \tilde{\Sigma}^n(x, x)$, $i = 1, \dots, n_X$.

With squared-exponential kernel and constant prior mean μ_0, the analytical formulae for each element M_i and $\tilde{\Sigma}_i$ can be derived for two of the most frequently observed distributions of disturbances:

uniform and normal. Each element in M_i and $\tilde{\Sigma}_i$ can be derived analytically using Equations (3) and (4).

For each $i = 1, \ldots, n_X$, from (3) and (6)

$$M^n(x_i) = \left(\int_{\mathbb{R}^D} \Sigma_0(x_i+\delta, x^1)\mathbb{P}[\delta]\,\mathrm{d}\delta, \ldots, \int_{\mathbb{R}^D} \Sigma_0(x_i+\delta, x^n)\mathbb{P}[\delta]\,\mathrm{d}\delta\right)(\Sigma_0(x^{1:n}, x^{1:n}) + \sigma_\epsilon^2 I_n)^{-1}(f(x^{1:n}) - \mu_0(x^{1:n})) + \int_{\mathbb{R}^D} \mu_0(x_i+\delta)\mathbb{P}[\delta]\,\mathrm{d}\delta \tag{9}$$

and from (4) and (7)

$$\tilde{\Sigma}^n(x_i, x) = \frac{1}{\sqrt{\sigma^n(x)^2 + \sigma_\epsilon^2}} \int_{\mathbb{R}^D} \Sigma_0(x_i+\delta, x)\mathbb{P}[\delta]\,\mathrm{d}\delta - \frac{1}{\sqrt{\sigma^n(x)^2 + \sigma_\epsilon^2}} \times \left(\int_{\mathbb{R}^D} \Sigma_0(x_i+\delta, x^{1:n})\mathbb{P}[\delta]\,\mathrm{d}\delta\right)(\Sigma_0(x^{1:n}, x^{1:n}) + \sigma_\epsilon^2 I_n)^{-1}\Sigma_0(x^{1:n}, x). \tag{10}$$

For a D-dimensional input, the k^{th} element of the vector in (9) can be computed as follows:

(a) for a uniformly distributed disturbance $\mathcal{U}(-\Delta, \Delta)$,

$$\int_{\mathbb{R}^D} \Sigma_0(x_i+\delta, x^k)\mathbb{P}[\delta]\,\mathrm{d}\delta = \alpha_0 \prod_{d=1}^{D} \frac{\sqrt{\pi} l_d}{\sqrt{2}\Delta_d}\left(\Phi\left(\frac{x_{i_d} - x_d^k + \Delta_d}{l_d}\right) - \Phi\left(\frac{x_{i_d} - x_d^k - \Delta_d}{l_d}\right)\right);$$

(b) for a normally distributed disturbance $\mathcal{N}(0, \Theta^2)$,

$$\int_{\mathbb{R}^D} \Sigma_0(x_i+\delta, x^k)\mathbb{P}[\delta]\,\mathrm{d}\delta = \alpha_0 \prod_{d=1}^{D} \frac{l_d}{\sqrt{l_d^2 + \Theta_d^2}} \exp\left(-\frac{(x_{i_d} - x_d^k)^2}{2(l_d^2 + \Theta_d^2)}\right).$$

Details of the derivation are described in Appendix A of the online supplemental data for this article, which can be accessed at https://doi.org/10.1080/0305215X.2022.2145604. The expectation in (8) can be computed using Algorithm 1 in Scott, Frazier, and Powell (2011).

At the end of optimization, a GP model over all sampled points is fitted and the point that maximizes the robustness performance of that model is the solution to return, *i.e.*

$$x_r^N = \arg\max_x M^N(x).$$

Algorithm 1 summarizes the details of rKG method of determining sampling points.

Algorithm 1: Pseudo-code for robustness optimization

```
Place a Gaussian process prior on $f$
Update distribution on $f$ at $n_0$ initially sampled points.
Set the number of sampled points $n$ to $n_0$.
while $n \leq N$ do
    Fit a GP on $n$ samples.
    Identify the point with the largest value of rKG.
    Add it to the set of sampled points and increase $n$ by 1.
Result: $x_r^N = \arg\max_x M^N(x)$
```

4.4. Consistency of the algorithm

It is worth noting that the algorithm is myopically optimal by construction. It can be proved that the rKG algorithm finds the true robust optimum x^{OPT} on a compact domain X, given an infinite budget.

Theorem 4.1: *If the input space is compact, the noise variance of the output is strictly positive across the input domain, f can be modelled by a GP, and with the assumption that*

$$\forall\, x' \neq x \in X, \quad \lim_{n\to\infty} \sup |\text{Corr}^n[f(x'), f(x)]| \leq \text{Const.} < 1, \tag{11}$$

the solution returned by the rKG algorithm converges to the true robust optimum, given an infinite budget of evaluations.

The theorem is based on the fact that rKG at any location converges to zero given an infinite budget of evaluations. This fact is proved combining the techniques of proving the convergence in Pearce, Poloczek, and Branke (2019) and for the Knowledge Gradient for Continuous Parameters (KGCP) in Scott, Frazier, and Powell (2011). Details of the proof are given in Appendix B of the online supplemental data.

5. Experiments

Robust KG is tested on several one- and two-dimensional benchmark problems with both uniformly and normally distributed disturbances, and compared with several alternative approaches.

5.1. Experimental setup

Initial points are sampled by LHS, the number of points is five times the number of dimensions, except for the rocket simulation, which is difficult to model by a GP and thus requires a larger number of initial points. For uniformly distributed cases, the total budget is 75, for normally distributed cases it is 50 samples for one- and two-dimensional functions and 100 for three- and four-dimensional functions. Unless stated otherwise, at each step, the HyperParameters (HP) of the GP are tuned by maximizing the marginal likelihood. Functions from the TensorFlow (TF) library (Abadi *et al.* 2016) are used to compute the log marginal likelihood, including the Cholesky decomposition and solving triangular systems. The maximization of the marginal likelihood, as well as of the acquisition function, starts by evaluating some random points, followed by gradient ascent from the best points using L-BFGS-B (Zhu *et al.* 1997) to find the maximum.

All experimental results are averaged over 100 independent runs for one-dimensional problems and 25 independent runs for two-dimensional problems. OC figures show the mean plus and minus one standard error.

5.2. Benchmark functions

The following nine test functions are used, summarized in Table 1 and visualized in Figures 1 and 2.

(1) A simple function from Le and Branke (2020) $\max f_1(x) = -0.5(x+1)\sin(\pi x^2)$.
(2) A function from Paenke, Branke, and Jin (2006) $\min f_2(x) = 2\sin(10e^{-0.2x}x)e^{-0.25x}$.
(3) The first function is also considered in a two-dimensional version by simply adding up over the two dimensions: $\max f_3(x_1, x_2) = -0.5(x_1+1)\sin(\pi x_1^2) - 0.5(x_2+1)\sin(\pi x_2^2)$.
(4) A simple one-dimensional function from Fröhlich *et al.* (2020) $\max f_4(x) = \sin(5\pi x^2) + 0.5x$.
(5) The one-dimensional Reproducing Kernel Hilbert Space (RKHS) function from Nogueira *et al.* (2016), which is to be minimized.
(6) A two-dimensional Gaussian Mixture Model (GMM 2D) from Nogueira *et al.* (2016).

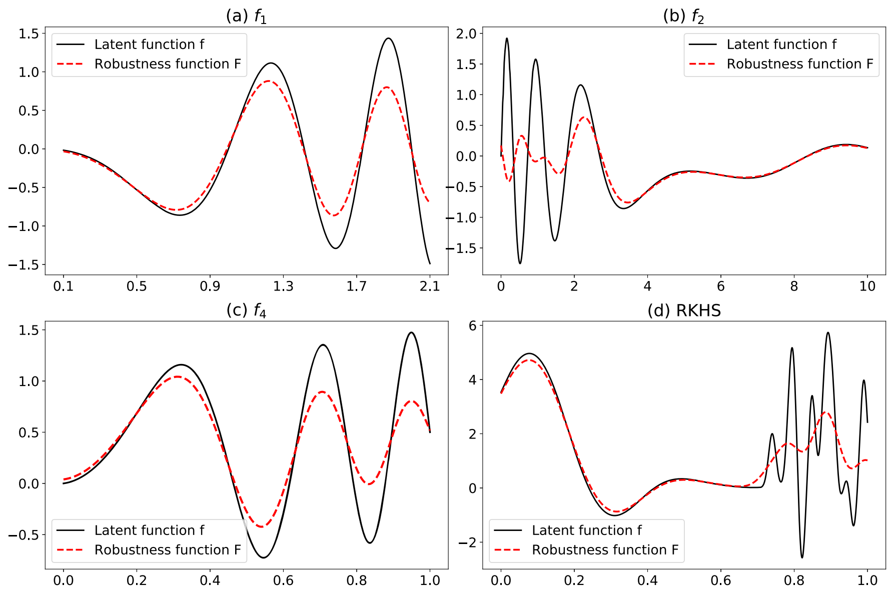


Figure 1. One-dimensional benchmark functions.

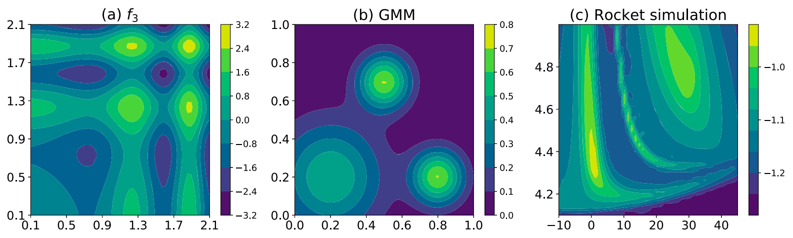


Figure 2. Two-dimensional benchmark functions.

Table 1. Benchmark functions for experiments.

Name	Optimum	Domain X	Distribution	n_0	Figure
f_1	Max	$[0.1, 2.1]$	$\mathcal{U}(-\Delta, \Delta)$, Δ=0.15	5	1(a)
f_2	Min	$[0.0, 10.0]$	$\mathcal{U}(-\Delta, \Delta)$, Δ=0.5	5	1(b)
f_3	Max	$[0.1, 2.1]^2$	$\mathcal{U}(-\Delta, \Delta)$, Δ=(0.15, 0.15)	10	2(a)
f_4	Max	$[0.0, 1.0]$	$\mathcal{N}(0, 0.05^2)$	5	1(c)
RKHS	Min	$[0.0, 1.0]$	$\mathcal{N}(0, 0.03^2)$	5	1(d)
GMM	Max	$[0, 1]^2$	$\mathcal{N}((0, 0), (0.1, 0.1)^2)$	10	2(b)
Rocket	Max	$[-10, 45] \times [4.1, 5]$	$\mathcal{N}((0, 0), (3, 0.05)^2)$	20	2(c)
Hartmann	Max	$[0, 1]^3$	$\mathcal{N}(0, 0.1)$	15	
Piston	Max	$[0, 1]^4$	$\mathcal{N}(0.0, 0.05^2)$	20	

(7) A two-dimensional gravity assisted rocket trajectory simulation model from Fröhlich *et al.* (2020), where the goal is to identify an angle α_0 and launch speed ν_0 that minimize a cost term proportional to the launch speed and the distance of the resulting trajectory from the target planet. The two terms are combined in a single objective function $J(\alpha_0, \nu_0) = \log_{10}(d_{\text{target}} + \beta\nu_0)$, with β being a parameter that trades-off the two terms. The disturbance for ν_0 and α_0 is $\Delta_\nu = 0.05$ and $\Delta_\alpha = 3^o$, respectively.

(8) The three-dimensional Hartmann function

$$f(x) = -\sum_{i=1}^{4} \alpha_i \exp\left(\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2\right),$$

where

$$\alpha = (1.0, 1.2, 3.0, 3.2)^{\mathrm{T}}$$

$$A = \begin{pmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}; \quad P = 10^{-4}\begin{pmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{pmatrix}.$$

(9) The piston design example, adapted from Arendt, Apley, and Chen (2013), with more details in Section 5.7.

All functions have multiple local optima with the optimum of the undisturbed function being different from the robust optimum given disturbances.

In case the disturbance distribution is uniform, it is capped at the boundary of the search space, *i.e.*

$$\mathbb{P}[\delta] = \begin{cases} \dfrac{1}{\min\{ub, x+\Delta\} - \max\{lb, x-\Delta\}} & \delta \in [\max\{lb, x-\Delta\}, \min\{ub, x+\Delta\}] \\ 0 & \text{otherwise,} \end{cases}$$

where lb and ub are the lower and upper bounds of the search space, respectively.

Unless stated otherwise, the output noise is set to have standard deviation 0.1, *i.e.* $\epsilon \sim \mathcal{N}(0, 0.1^2)$, although some experiments are also run without output noise. In the case of normally distributed disturbances, unless stated otherwise, the standard deviation of the output noise is set, following Fröhlich *et al.* (2020), to 0.001, *i.e.* $\epsilon \sim \mathcal{N}(0, 0.001^2)$.

5.3. Benchmark methods

The rKG algorithm is compared with the following three alternative approaches.

- The Direct Robustness Approximation (DRA) at each sample location as described in Section 4.2.2.
- Uniform allocation (UA), which allocates the samples by LHS (McKay, Beckman, and Conover 1979) rather than sequentially by an acquisition function. Apart from the acquisition function, the algorithm is identical to the algorithm used with rKG, *i.e.* it builds a GP from the data collected on the undisturbed function f and returns the solution with best robust posterior mean $x_r^N = \arg\max_x M^N(x)$.
- Noisy Entropy Search–Expectation Propagation (NES-EP): an approach proposed by Fröhlich *et al.* (2020) that uses ES (Hennig and Schuler 2012) as acquisition function. As NES-EP has only been derived for normally distributed disturbances, it is only used as a benchmark on such test problems.

Note that NES-EP as proposed by Fröhlich *et al.* (2020) uses a different HP tuning algorithm based on GPy (2012). Because of primary interest in the relative performances of the acquisition functions, the original HP tuning in NES-EP has been replaced by the same method used for rKG, *i.e.* acquisition functions compared using the same HP tuning method. However, to explore the impact of the HP

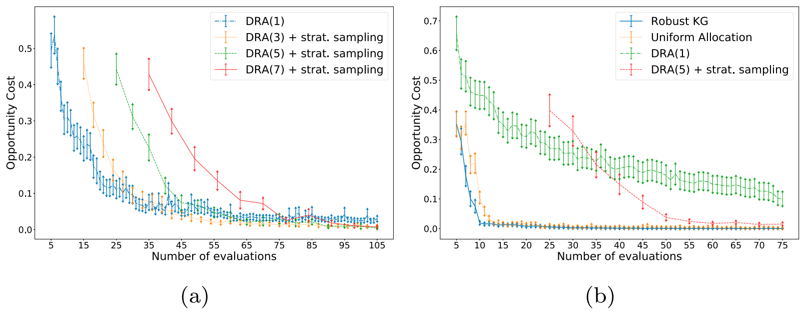


Figure 3. OC over number of evaluations for f_1 .

tuning method, some experiments where rKG and NES-EP use the HP tuning of the original NES-EP algorithm based on GPy are also run. These algorithms are then denoted as rKG_GPy and NES-EP_GPy.

5.4. Results with uniform disturbance

5.4.1. One-dimensional test function f_1

For test function $f_1(x)$, the robust optimum is approximately at $x = 1.22$. However, optimization of f using standard KG would return a solution with lower robustness performance near $x = 1.873$.

The experiments are started by examining parameter k of DRA(k), the number of replications to average over for a single observation. The bigger k, the smaller observation noise would be, but the more computational time per iteration. The result is shown in Figure 3(a), which plots OC over the total number of function observations so far.

With five initial solutions, DRA with $k = 1$ only needs five evaluations to initialize, whereas for example the run with $k = 7$ requires $5 \times 7 = 35$ evaluations to initialize. Thus, larger k means a delayed use of the acquisition function, but towards the end, the runs with $k \in \{3, 5, 7\}$ find a slightly better solution compared to $k = 1$. It can be concluded that, to evaluate a single solution, it is better to use small values of k and the noise handling capability of the GP rather than many replications (bigger k), at least for this test problem. While this would have been expected for problems with just output noise, this was not so clear for problems with disturbance of decision variables, as the disturbance leads to heteroscedastic observation noise.

Figure 3(b) compares the results of rKG, UA, DRA(1) and DRA(5). Visibly, the rKG method performs better than all other methods. UA is not far behind on this simple function whereas throughout the run DRA is significantly worse. It is interesting to notice that, after the initial five samples, DRA(1) already has a worse opportunity cost, using the same information as rKG. It seems that making the relationship between f and F explicit by learning f and deriving F through integration leads to a better model than approximating F directly.

Finally, the algorithms are tested on a deterministic version of f_1. Similarity is observed in the results of this case and the case of noisy observation, thus an additional figure was not included. It seems the implicit averaging of the kernel function makes all approaches very robust to additional output noise.

5.4.2. More complicated one-dimensional test function f_2

This is a minimization problem taken from Paenke, Branke, and Jin (2006). The test function has several local minima, concentrated on the left side of the input domain. Results are similar to those of test function f_1, although UA converges more slowly compared to rKG.

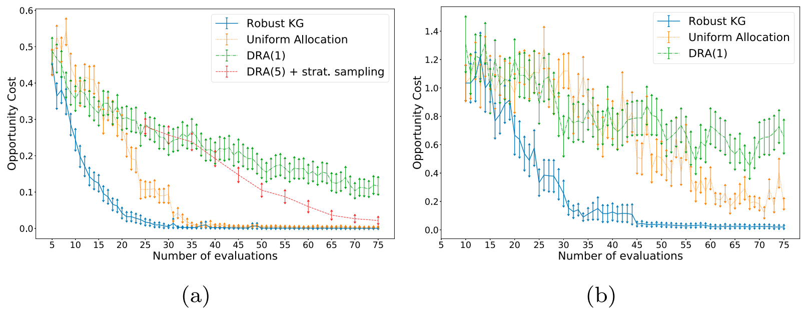


Figure 4. OC over number of evaluations for (a) f_2, (b) f_3.

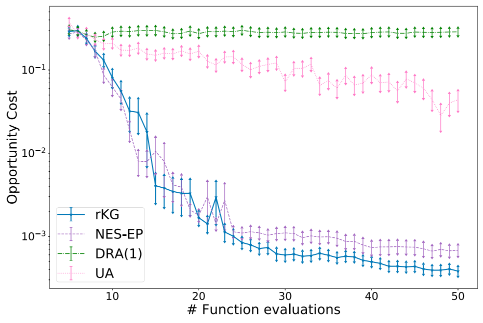


Figure 5. OC over number evaluations of f_4.

5.4.3. Simple two-dimensional test function f_3

For this two-dimensional benchmark problem, 10 initial sample points are used, which means DRA(5) would need 50 evaluations just to initialize, which renders this method unsuitable. Therefore only rKG, UA and DRA(1) are compared.

The opportunity cost over the number of evaluations is shown in Figure 4(b). While the rKG acquisition function once more quickly converges to the correct robust optimum (opportunity cost of zero), UA and DRA(1) are significantly slower.

5.5. Results on test problems with normally distributed disturbance

5.5.1. Noisy simple one-dimensional test function f_4

For this test problem, the output noise standard deviation is set to 0.1, *i.e.* $\epsilon \sim \mathcal{N}(0, 0.1^2)$. As shown in Figure 5, rKG is slightly better than NES-EP and both converge to a good solution, whereas DRA(1) does not even converge to good solutions and the opportunity cost is almost a flat line. It seems DRA(1) doesn't work well in the case of normally distributed disturbance, thus this approach is not tested with other test problems. UA is a bit better than DRA(1) but still does not return a good solution.

5.5.2. RKHS function

The RKHS function is tested with small output noise ($\sigma_\epsilon = 0.001$) and large output noise ($\sigma_\epsilon = 0.1$). As shown in Figure 6, the rKG approach is much better than the benchmarks for both levels of output noise. Similarly to the test with function f_4, UA cannot return a good solution. Since UA allocates the samples uniformly, it is not able to focus on the region of the robust optimum. For higher dimensional

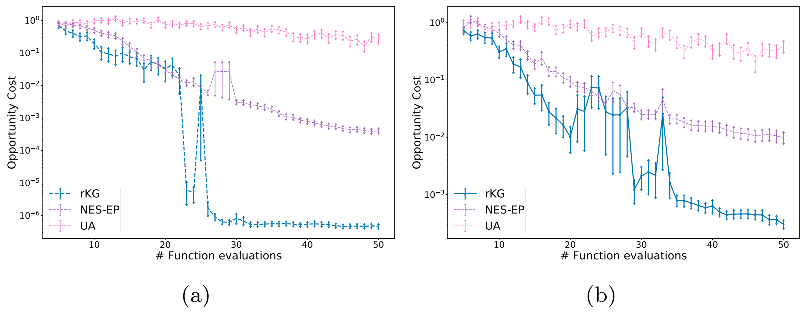


Figure 6. OC over number of evaluations of the RKHS function in (a) small noise case and (b) large noise case.

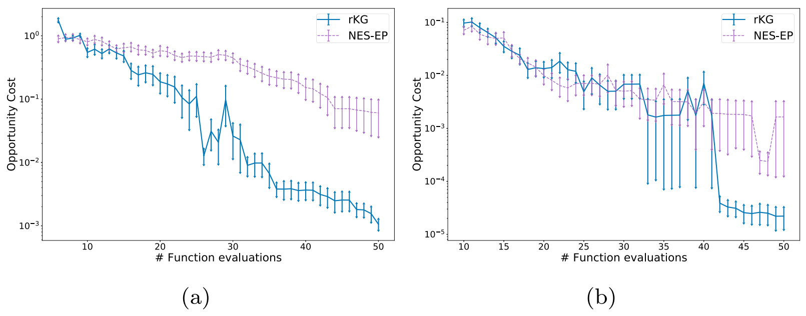


Figure 7. OC over number of evaluations for (a) f_3 with normally distributed disturbance, (b) GMM.

cases, UA requires many more samples (exponential in the number of dimensions) to have a good GP fit. Therefore this method is not tested any further.

5.5.3. Two-dimensional test function f_3 for normally distributed disturbance

In order to be able to compare with NES-EP, normally distributed noise is also used with test function f_3. As can be seen in Figure 7(a), rKG works better than NES-EP for this problem.

5.5.4. Gaussian mixture model

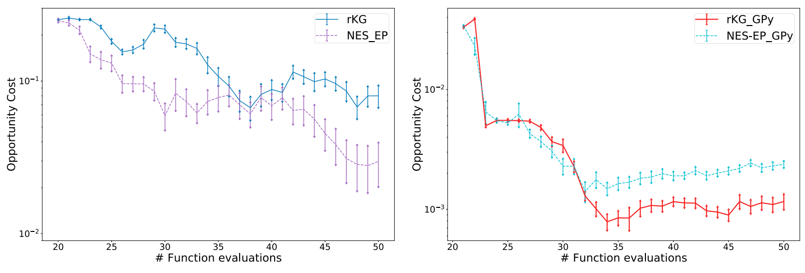
As depicted in Figure 7(b), for this problem, rKG is comparable in the first 30 evaluations but then better than NES-EP towards the end.

5.5.5. Gravity assist manoeuvre

The proposed method is applied to the problem of the gravity assist manoeuvre as described in Section 4.3 of Fröhlich *et al.* (2020). As can be seen in Figure 8(a), rKG does not outperform NES-EP although both solutions are relatively poor and the error bars are pretty big. This may be because this problem has some very sharp peaks that are difficult to model with a squared exponential kernel. This issue and the dependence on HP tuning is explored more deeply in Section 5.6.

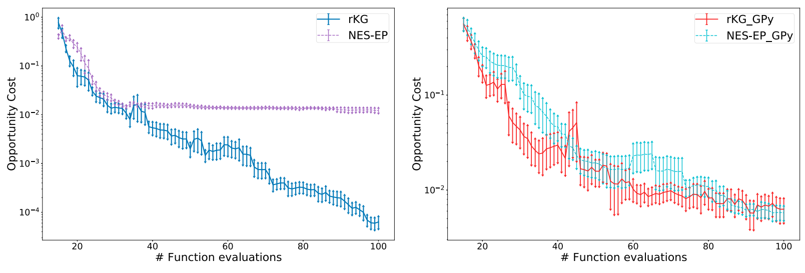
5.5.6. Hartmann 3D

As depicted in Figure 9(a), rKG is similar to NES-EP in the first 35 evaluations but then NES-EP converges prematurely to an inferior solution.



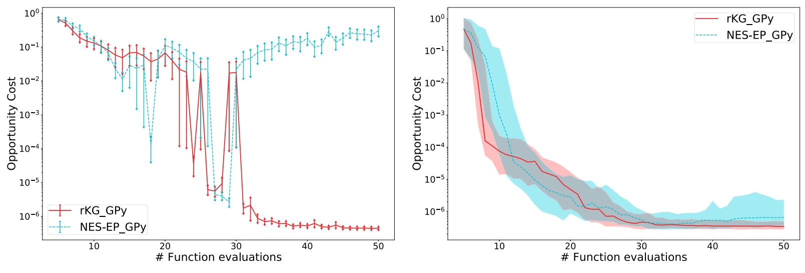
(a) (b)

Figure 8. OC for the gravity assist problem when (a) using the TF approach for HP tuning, (b) using GPy for HP tuning.



(a) (b)

Figure 9. OC for Hartmann 3D when (a) using the TF approach for HP tuning, (b) using GPy for HP tuning.



(a) Mean and error bars (b) Median and percentiles

Figure 10. OC over number of evaluations of RKHS function with small noise and GPy used for HP tuning. (a) Mean and error bars and (b) Median and percentiles.

5.6. *Results with GPy HP tuning*

So far, the HP tuning method with the TF library (Abadi *et al.* 2016) has been used. As NES-EP uses GPy HP tuning as default, the impact of the HP tuning procedure is also tested on the relative comparison. Since GPy requires prior values, the same set of prior values given by Fröhlich *et al.* (2020) is used.

5.6.1. *RKHS function with small noise and HP tuning using GPy*

As shown in Figure 10(a), rKG and NES-EP perform similarly, but NES-EP loses at the end owing to few outlier runs with very poor results. This is still consistent with the results reported in Fröhlich *et al.* (2020) if the OC is plotted with median and percentile as shown in Figure 10(b).

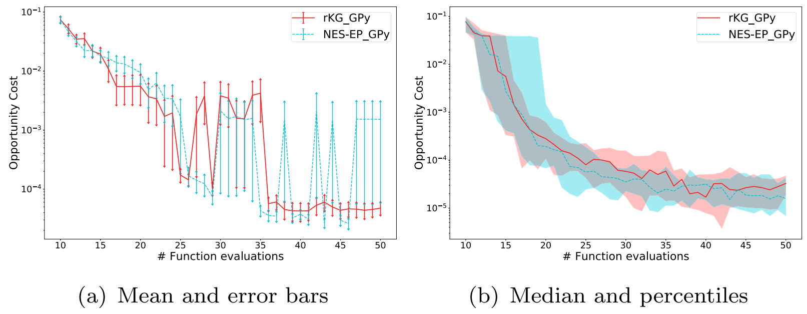


Figure 11. Plot of OC over number of evaluations of the GMM function. (a) Mean and error bars and (b) Median and percentiles.

5.6.2. GMM function with HP tuning using GPy

As can be seen in Figure 11(a), once again rKG and NES-EP are similar, but NES-EP loses at the end owing to few outlier runs with very poor results. Again, results are consistent with the article by Fröhlich *et al.* (2020) if the OC is plotted with median and percentile as shown in Figure 11(b).

5.6.3. Hartmann 3D

As depicted in Figure 9(b), rKG is slower midway but catches up and converges to a similarly good solution as NES-EP.

5.6.4. Gravity assist

Gravity assist was the only test problem where NES-EP performed better than rKG. However, as noted above, this problem has some very sharp peaks in an otherwise smooth area, and thus is not suitable to be modelled by a GP with squared exponential kernel. A poor HP choice will substantially disrupt GP-based algorithms. This may be somewhat alleviated by choosing a good prior and optimizing HPs using GPy. Thus the pairing of rKG_GPy and NES-EP_GPy is also compared.

As can be seen in Figure 8(b), performance of rKG_GPy and NES-EP_GPy is similar for the first 10 new evaluations. After that, rKG_GPy performs better, albeit the opportunity cost went back up a little bit towards the end. This supports the conjecture that the relatively poor performance of rKG on the gravity assist problem observed in Section 5.5.5 is due to a model mismatch.

5.6.5. Function f_3 for dependency on the choice of hyperparamenter prior

As the results of NES-EP_GPy are noticed to be quite sensitive to the choice of prior values of HPs, the simple two-dimensional function f_3 is tested with varying values of prior lengthscale and fixed signal variance or varying prior signal variance and fixed true lengthscale to see how the rKG method and the benchmark method depend on the choice of HPs. The fixed values of lengthscale and signal variance have been learned by maximum likelihood with a meshgrid of 6400 points (lengthscale = 0.316,639,81, signal variance = 342.074,631,937,6998). The mean of opportunity cost after 25 independent runs is tested, using the acquisition function rKG and NES-EP_GPy with the same method of HP tuning using GPy. Figure 12 shows that the rKG method is notably less sensitive to the setting of the prior.

5.7. Real-world application

Normally distributed disturbances of solution variables have an infinite support (they are not bounded). That means the integration limits in the calculation of the “expected performance” robustness are the whole input space, irrespective of the constraints on the solution space. However, in practice it is necessary to consider the reliability as it does not make sense if the disturbed solution (solution with disturbance) lies outside the region of feasibility. One way to deal with this is to assign

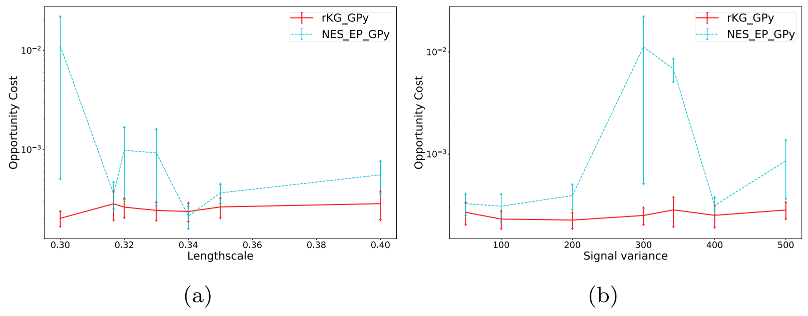


Figure 12. (a) Varying lengthscale prior; (b) varying signal variance prior.

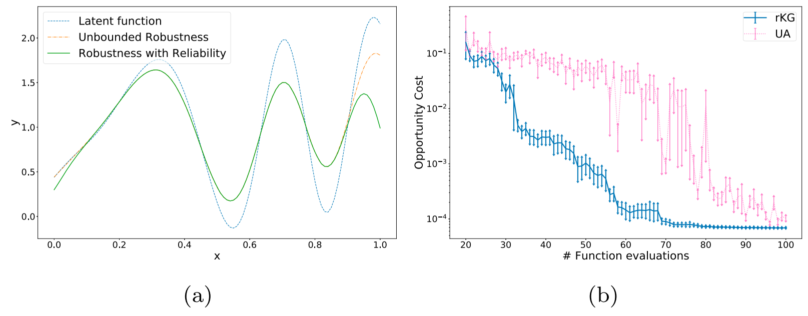


Figure 13. (a) Robustness with and without reliability compared on test function f_4; (b) opportunity cost over number of evaluations of piston design.

all solutions outside the region of feasibility a value of zero. The area where disturbed samples take non-zero values depends on the location of the solution in the feasibility region. Therefore, the limits are adjusted accordingly and the robustness with the adjusted limits is called 'Robustness with Reliability'. Note that, with the adjustment, the robust optimum solution is less likely to be at the edge of the feasibility region. An example of robustness with and without reliability is depicted in Figure 13(a).

The application of the rKG and 'Robustness with Reliability' are demonstrated by the piston design example that was analysed in Arendt, Apley, and Chen (2013). Owing to the expensive computational cost of the simulation used for the automotive piston design, whose true response surface is unknown, a GP built over the data set of 200 equally spaced points is used as a replacement (Arendt, Apley, and Chen 2013). The posterior mean of this GP is a function of four-dimensional design variable d and two-dimensional noise variable w. The data used in this experiment are normalized. More details can be found in Arendt, Apley, and Chen (2013). In this work, the problem is adapted, replacing the additional noise variables by the disturbance of the decision variables. It aims to search for the robust maximum of the function of four-dimensional design variables d, which is the posterior mean when the noise variable is fixed to the mean value ($w = [0.5, 0.5]$). The disturbance is distributed normally with zero mean and variance $\Theta^2 = 0.05^2$ around each design variable. Robustness without reliability would yield a robust optimum solution on the edge at $d = [0.807{,}302{,}15,\ 1.0,\ 0.0,\ 0.0]$, whereas robustness with reliability returns a solution within the region of feasibility $d^* = [0.791{,}864{,}73,\ 0.856{,}594{,}09,\ 0.162{,}340{,}84,\ 0.150{,}678{,}62]$.

Since NES-EP seeks for a robust solution without reliability, it cannot be used in this case. Therefore rKG is compared with UA. It starts with 20 initial samples, chosen by LHS, and finishes after 80 iterations and averages the results over 8 runs. Figure 13(b) shows that rKG returns a solution close to the true optimum much faster and more consistently (fewer fluctuations) compared to UA.

6. Conclusion

In many real-world optimization problems, disturbances to the decision variables, for instance manufacturing tolerances, may affect the solutions. It becomes crucial to search for a robust solution in order to reduce the sensitivity to the disturbances and achieve a high expected performance.

This article introduced an algorithm based on the knowledge gradient idea for efficiently searching for the robust optimum of expensive-to-evaluate functions, adapting the technique used in Pearce and Branke (2017) and Toscano-Palmerin and Frazier (2018) for Bayesian optimization with input uncertainty. The article suggests a novel acquisition function, called the robust Knowledge Gradient (rKG), used for iteratively identifying the next point to sample. It has been demonstrated that, in the case of uniformly distributed disturbance, with a much lower number of required function evaluations, rKG can obtain the same solution quality as alternative approaches much more efficiently. And for the case of normally distributed disturbance, the rKG method has been shown to be at least comparable to benchmark methods. Moreover, it can deal with output noise better and is less sensitive to the hyperparameter prior.

Future work entails several avenues. The approach should be tested with more problems, especially higher dimensional and real-world problems. Also, it should be tested with other measures of robustness where a quantile or the worst case is of interest rather than the expected performance.

Acknowledgments

The authors would like to thank Dr Michael Pearce for his technical assistance in completing the experiments. The authors also acknowledge Dr Paul Arendt, Professor Wei Chen and Professor Daniel Apley for providing the data for the piston design example. Authors are also grateful to Lukas Fröhlich, one of the authors of Fröhlich *et al.* (2020), for making their code public, help in running the code and for providing the data for the gravity assist experiment.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The authors acknowledge support from the EPSRC [grant EP/L015374/1]; and GE as part of the Colibri project initiative.

Data availability statement

The data that support the findings of this study are available from the corresponding author, Hoai Phuong Le, upon request.

References

Abadi, Martin, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, *et al.* 2016. "TensorFlow: A System for Large-Scale Machine Learning." In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*, 265–283. Berkeley, CA: USENIX Association.

Arendt, Paul D., Daniel W. Apley, and Wei Chen. 2013. "Objective-Oriented Sequential Sampling for Simulation Based Robust Design Considering Multiple Sources of Uncertainty." *Journal of Mechanical Design* 135 (5): 051005.

Beyer, H.-G., and Bernhard Sendhoff. 2007. "Robust Optimization—A Comprehensive Survey." *Computer Methods in Applied Mechanics and Engineering* 196 (33): 3190–3218.

Branke, Juergen. 1998. "Creating Robust Solutions by Means of Evolutionary Algorithms." In *Proceedings of Parallel Problem Solving from Nature (PPSN V)*, edited by Agoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, 119–128. Berlin: Springer.

Frazier, Peter. 2018. "A Tutorial on Bayesian Optimization." https://arxiv.org/abs/1807.0281.

Frazier, Peter, Warren Powell, and Savas Dayanik. 2009. "The Knowledge-Gradient Policy for Correlated Normal Beliefs." *INFORMS Journal on Computing* 21 (4): 517–656.

Fröhlich, Lukas P., Edgar D. Klenske, Julia Vinogradska, Christian Daniel, and Melanie Nicole Zeilinger. 2020. "Noisy-Input Entropy Search for Efficient Robust Bayesian Optimization." In *Proceedings of the Twenty Third International*

Conference on Artificial Intelligence and Statistics, edited by Silvia Chiappa and Roberto Calandra, *PMLR Proceedings of Machine Learning Research* 108: 2262–2272.

GPy. 2012. *A Gaussian Process Framework in Python*. http://github.com/SheffieldML/GPy.

Hennig, Philipp, and Christian J. Schuler. 2012. "Entropy Search for Information-Efficient Global Optimization." *Journal of Machine Learning Research* 13 (1): 1809–1837.

Iwazaki, Shogo, Yu Inatsu, and Ichiro Takeuchi. 2020. "Mean-Variance Analysis in Bayesian Optimization Under Uncertainty." arXiv:2009.08166.

Jones, Donald R., Matthias Schonlau, and William J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions." *Journal of Global Optimization* 13 (4): 45–492.

Julier, S. J., and J. K. Uhlmann. 2004. "Unscented Filtering and Nonlinear Estimation." *Proceedings of the IEEE* 92 (3): 401–422.

Le, Hoai Phuong, and Juergen Branke. 2020. "Bayesian Optimization Searching for Robust Solutions." In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. Bae, Ben Feng, S. Lazarova-Molnar S. Kim, Z. Zheng, T. Roeder, and R. Thiesing, 2268–2278. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Le, Hoai Phuong, and Juergen Branke. 2021. "Bayesian Optimization for Robust Solutions Under Uncertain Input." In *Advances in Uncertainty Quantification and Optimization Under Uncertainty with Aerospace Applications*, edited by Massimiliano Vasile and Domenico Quagliarella, 245–259. Cham, Switzerland: Springer International.

Marzat, Julien, Eric Walter, and Hélène Piet-Lahanier. 2013. "Worst-Case Global Optimization of Black-Box Functions Through Kriging and Relaxation." *Journal of Global Optimization* 55 (4): 707–727.

McKay, M. D., Richard J. Beckman, and William J. Conover. 1979. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code." *Technometrics* 21 (2): 239–245.

Mockus, Jonas, Vytautas Tiesis, and Antanas Zilinskas. 1978. "The Application of Bayesian Methods for Seeking the Extremum." *Towards Global Optimization*, Vol. 2, 117–129. https://www.researchgate.net/publication/248818761_The_application_of_Bayesian_methods_for_seeking_the_extremum.

Nogueira, J., R. Martinez-Cantin, A. Bernardino, and L. Jamone. 2016. "Unscented Bayesian Optimization for Safe Robot Grasping." In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1967–1972.

Paenke, Ingo, Juergen Branke, and Yaochu Jin. 2006. "Efficient Search for Robust Solutions by Means of Evolutionary Algorithms and Fitness Approximation." *IEEE Transactions on Evolutionary Computation* 10 (4): 405–420.

Pearce, Michael, and Juergen Branke. 2017. "Bayesian Simulation Optimisation with Input Uncertainty." In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. Chan, A. D'Ambrogiom, and G. Zacharewicz, 2268–2278. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Pearce, Michael, Matthias Poloczek, and Juergen Branke. 2019. "Bayesian Optimization Allowing for Common Random Numbers." arXiv:1910.09259.

Rasmussen, Carl E., and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.

Sanders, Nicholas D., Richard M. Everson, Jonathan E. Fieldsend, and Alma A. M. Rahat. 2019. "A Bayesian Approach for the Robust Optimisation of Expensive-to-Evaluate Functions." arXiv:1904.11416v2.

Scott, Warren, Peter Frazier, and Warren Powell. 2011. "The Correlated Knowledge Gradient for Simulation Optimization of Continuous Parameters Using Gaussian Process Regression." *SIAM Journal on Optimization* 21 (3): 996–1026.

Srinivas, Niranjan, Andreas Krause, Sham Kakade, and Matthias Seeger. 2010. "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design." In *Proceedings of the 27th International Conference on Machine Learning*, 1015–1022. Madison, WI: Omnipress.

Toscano-Palmerin, Saul, and Peter Frazier. 2018. "Bayesian Optimization with Expensive Integrands." arXiv:1803.08661.

ur Rehman, Samee, and Matthijs Langelaar. 2017. "Expected Improvement Based Infill Sampling for Global Robust Optimization of Constrained Problems." *Optimization and Engineering* 18 (3): 723–753.

ur Rehman, Samee, Matthijs Langelaar, and Fredvan Keulen. 2014. "Efficient Kriging-Based Robust Optimization of Unconstrained Problems." *Journal of Computational Science* 5 (6): 872–881.

Xu, Z., Y. Guo, and J. H. Saleh. 2021. "Efficient Hybrid Bayesian Optimization Algorithm with Adaptive Expected Improvement Acquisition Function." *Engineering Optimization* 53 (10): 1786–1804. doi:10.1080/0305215X.2020.1826467.

Yang, J., Z. Zhan, K. Zheng, J. Hu, and L. Zheng. 2016. "Enhanced Similarity-Based Metamodel Updating Strategy for Reliability-Based Design Optimization." *Engineering Optimization* 48 (12): 2026–2045. doi:10.1080/0305215X.2016.1150469.

Zhu, Ciyou, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. 1997. "Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization." *ACM Transactions on Mathematical Software* 23 (4): 550–560. doi:10.1145/279232.279236.