Robustness kriging-based optimization

Mélina Ribaud, Christophette Blanchet-Scalliet, Frederic Gillot, Céline Helbert

Robustness kriging-based optimization

Mélina Ribaud*[1,2], Christophette Blanchet-Scalliet[1], Frédéric Gillot[2,3], and Céline Helbert[1]

[1]Univ Lyon, École centrale de Lyon, CNRS UMR 5208, Institut Camille Jordan, 36 avenue Guy de Collonge, F-69134 Ecully Cedex, France

[2]Univ Lyon, Université Ecole Centrale de Lyon, LTDS UMR CNRS 5513,36 avenue Guy de Collonge, F-69134 Ecully Cedex, France

[3]INRIA Rennes, I4S Team

July 4, 2018

Abstract

In the context of robust shape optimization, the estimation cost of some physical models is reduced by the use of a response surface. The multi objective methodology for robust optimization that requires the partitioning of the Pareto front (minimization of the function and the robustness criterion) has already been developed. However, the efficient estimation of the robustness criterion in the context of time-consuming simulation has not been much explored. We propose a robust optimization procedure based on the prediction of the function and its derivatives by a kriging. The usual moment 2 is replaced by an approximated version using Taylor theorem. A Pareto front of the robust solutions is generated by a genetic algorithm named NSGA-II. This algorithm gives a Pareto front in an reasonable time of calculation.

We detail seven relevant strategies and compare them for the same budget in two test functions (2D and 6D). In each case, we compare the results when the derivatives are observed and not.

Keywords. Gaussian process regression, robustness criterion, multi-objective optimization

1 Introduction

Complex physical phenomena are more and more studied through numerical simulations. These numerical models are able to mimic real experiments with a high accuracy. They predict the physical measures of interest (outputs) very precisely. Then, numerical simulations are used as a replacement for real experiments because they are less costly in primary materials. Sometimes, the solutions of the optimization problem could be sensitive to inputs' perturbations. For example, these perturbations are due to random fluctuations during production. A solution of a multi-objective optimization problem is then looked for where the first objective is the function itself and the second is a robustness criterion. These two objectives are assumed to be antagonistic. The issue of robust optimization (RO) is to find a Pareto

*melina.ribaud@gmail.com

front that makes a balance between the optimization of the function and the impact of input perturbations (uncertainties). As the simulations given by the numerical code are often time-consuming, only a few simulations are then affordable. So we cannot exploit intensively the computer code to provide the robust optimum.

In the context of costly simulations, the optimization procedure is often run on a kriging model that statistically approximates the computer code (kriging-based black-box optimization). Choosing where to sample the output in the input space to reach the optimum as fast as possible is a big issue. [Jones et al., 1998] developed the Efficient Global Optimization (EGO) algorithm that exploits the Expected Improvement (EI) criterion. However, the EGO algorithm is not an answer to the robust optimization problem because uncertainties are not taken into account.

Uncertainties can be of different kinds. They have to be well identified to make the robust optimization accurate. [Lelièvre et al., 2016] propose a classification of different approaches that deal with uncertainties in the context of reliability, optimization and robustness. In order to provide this survey, they sort uncertainties in two main groups: uncertainties that "are primitively linked to the environment and condition of use" and uncertainties that "are those connected with the production/manufacturing process".

In literature, we can find a sample of works that handle robust optimization with the first type of uncertainties. The aim is to find $\mathbf{x}$ such that $f(\mathbf{x}, \mathbf{u})$ is optimal with $\mathbf{u}$ the vector of the uncertain variables (cf [Janusevskis and Le Riche, 2013], [Marzat et al., 2013], [Apley et al., 2006] and [ur Rehman et al., 2014]). For example, [Janusevskis and Le Riche, 2013] propose a way to make a robust optimization based on a metamodel. They develop a Gaussian process (GP) model in the joint space $(\mathbf{x}, \mathbf{u})$ that takes into account the uncertainty of the $\mathbf{u}$-group of inputs. They define an adapted expected improvement and they maximize this criterion to enrich the design sequentially. [Marzat et al., 2013] propose an algorithm that make a Kriging Based Robust Optimization (KBRO) considering the worst-case. At each step i, they conduct two EGO. A first EGO is performed on the design space to found $\mathbf{x}^i$ that minimize $f(\mathbf{x}, \mathbf{u}^i)$. At the first iteration, they randomly choose $\mathbf{u}^i$ in the uncertain space. Then a second EGO is performed on the uncertain space to found $\mathbf{u}^i$ that minimizes $f(\mathbf{x}^i, \mathbf{u})$. They return $f(\mathbf{x}^i, \mathbf{u}^i)$ at the last iteration. In all these methods, the variables are clearly separated in two classes (design and uncertain) and the design is enriched sequentially.

In our context, we consider that the inputs we want to optimize deal with a little perturbation on which. The aim is to optimize the function $f(\mathbf{x} + \mathbf{H})$ where $\mathbf{x}$ are the design variables and $\mathbf{H}$ are the perturbations. This case is related to the second type of uncertainties described by [Lelièvre et al., 2016] (see section 4.3). [ur Rehman et al., 2014] propose an algorithm close to the EGO to answer to this problem. They add a previous optimization that localizes the worst-case on the response surface, $\min_{\mathbf{x}} \max_{\mathbf{H}} \widehat{y}(\mathbf{x}+\mathbf{H})$ where $\widehat{y}$ is the kriging prediction. Then, they maximize the EI calculated with the worst case instead of the local minimum for the reference value. This solution is a first step to the robust optimization issue because the only difference with the EGO is the reference value on the EI. In addition, it provides only one point that makes a balance between the function to be optimized and the inputs' perturbations to be minimized. The entire Pareto front is not explored.

In our work we propose a multi-objective strategy to detect the whole set of robust solutions. The first objective is the function itself while the second objective is the robustness criterion which needs to be described. The robustness quantification of a solution is challenging, [Göhler et al., 2016], [Gabrel et al., 2014] and [Coco et al., 2014] give some overviews of different robustness criteria. Our industrial partners quantify the variability of a solution by the local variance of the output in the neighborhood of a solution(see e.g. [Apley et al., 2006] and [Troian et al., 2016]). One aim of this paper is to propose an accurate estimation of this local variance.

That is why, the robustness criterion we look for is based on Taylor development as proposed by [Darlington et al., 1999]. But [Darlington et al., 1999] do not provide a RO in the context of time-consuming simulations. In our article, the RO is coupled with a kriging. Once the criterion defined, we perform a kriging-based multi-objectif optimization on both the function and the robustness criterion. We choose the function instead of the mean because the inputs perturbation have already been taken into account in the robustness criterion. In addition, we are interested in the optimum and not the optimum in mean.

[Pronzato and Éric Thierry, 2003] study the behavior of the mean and the variance of the function computed with the Taylor Theorem. They prove that the kriging variance has a huge influence on the two moments and it is highly recommended to take into account this variance to make an efficient KBRO. All the KBRO strategies we develop take into account the kriging variance.

Since the Taylor theorem needs the values of derivatives, co-kriging is well adapted (see e.g [Le Gratiet, 2013]). This model is an extension of the kriging model. More precisely kriging is an interpolation technique which aims at predicting the output using an adapted underlying correlation function (see e.g [Santner et al., 2003]). The co-kriging method consists in exploiting the natural covariance structure between the GP model of the function and all the derivatives. This structure is described in [Rasmussen and Williams, 2006]. The observation of the derivatives are not necessary to predict them, we only need observations of the function. However, all the observed derivatives are good to know to improve the prediction quality.

Then, the function and its robustness criterion are accessible through the co-kriging model. A muti-obtjective optimization is performed to provide solutions. [Wagner et al., 2010] make an overview of different multi-objective (MO) algorithms based on a kriging model: the aggregation methods (see [Knowles, 2006], [Liu et al., 2007] and [Zhang et al., 2010]), the Hypervolume methods (see [Ponweiser et al., 2008] and [Emmerich et al., 2011]), the maximin method (see [Svenson and Santner, 2016]), the uncertainty reduction method (see [Picheny, 2015]) and the MO method (see [Jeong and Obayashi, 2005]). [Henkenjohann and Kunert, 2007] shows that the aggregation methods are not efficient with a complex Pareto front. The hypervolume, maximin and uncertainty reduction algorithms need to make the multi-objective optimization on GP processes. As the robustness criterion we develop is not anymore Gaussian, it could be costly to adapt these methods in our case. We choose to develop some optimization procedures inspired by the MO EI introduced by [Jeong and Obayashi, 2005]. They propose to modify the reference value of the EI and they maximize the EI with a multi-objective algorithm. The MO EI is computed for each objective functions.

The article is structured as follows. Our robustness kriging-based criterion is introduced in section 2. In section 3, we introduce the estimation of our criterion in a context of a Gaussian process metamodeling. We present the general multi-objective scheme in section 4. The multi-objective optimization procedure is described in section 5. And finally, in section 6, we study the behavior of our methodology on two test cases.

2 Robustness criterion

The global aim of this article is to conduct a robust optimization of a two times differentiable function

$$\begin{array}{rccc} f: & D \subset \mathbb{R}^p & \longrightarrow & [a;b] \subset \mathbb{R} \\ & \mathbf{x} & \longmapsto & f(\mathbf{x}) \end{array} \tag{1}$$

To catch the robustness of f around a design point we consider a local variance, that's to say the variance of f in the neighborhood of the given point. However, we cannot compute the variance on the real function because it is too expensive. This section gives an approximation of this local variance that can easily be predicted in the context of a Gaussian process model of f.

Let $\mathbf{x} \in D$, an observation point. The variance of the function f around $\mathbf{x}$ is written $v_f(\mathbf{x}) = Var\left(f(\mathbf{x}+\mathbf{H})\right)$ where $\mathbf{H}$ represents fluctuations that can appear during fabrication. We consider that the production error $\mathbf{H}$ follows a Gaussian law. Then $\mathbf{H} \sim \mathcal{N}(0_{\mathbb{R}^d}, \Delta^2)$ where Δ^2 is defined by:

$$\Delta^2 = \begin{pmatrix} \delta_1^2 & 0 & \dots & 0 \\ 0 & \delta_2^2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \delta_p^2 \end{pmatrix}$$

Variances $\delta_1, \dots, \delta_p$ associated to each input are not necessary the same and are given by the experts.

A point $\mathbf{x}^1 \in D$ is considered less robust than a point $\mathbf{x}^2 \in D$ if $v_f(\mathbf{x}^1) > v_f(\mathbf{x}^2)$. In Figure 1, the minimum on the right (circles) is less robust than the one on the left (triangles). Let $\mathbf{h}^1, \dots, \mathbf{h}^N$, $\mathbf{h}^j \in \mathbb{R}^p$, $j = 1, \dots, N$ be N realizations of $\mathbf{H}$. The empirical estimation of the variance $v_f(\mathbf{x})$ is:

$$\widehat{v}_f(\mathbf{x}) = \frac{1}{N-1} \sum_{j=1}^{N} \left(f(\mathbf{x}+\mathbf{h}^j) - \bar{f}(\mathbf{x})\right)^2 \tag{2}$$

where $\bar{f}(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N}\left(f(\mathbf{x}+\mathbf{h}^j)\right)$ is the empirical mean (first moment). The estimation of the variance around only one point needs N calls to f.
In order to have a good estimation, N should satisfy the following inequality :

$$N \geq \left(z_{1-\alpha/2} \frac{\sqrt{\mu_4 - S_f^2}}{|e_n|} \right)^2 \tag{3}$$

where $S_f^2 = \frac{1}{N}\sum_{j=1}^{N}\left(f(\mathbf{x}+\mathbf{h}^j) - \bar{f}(\mathbf{x})\right)^2$, $\mu = \mathbb{E}[f(\mathbf{x}+\mathbf{H})]$, $\mu_4 = \mathbb{E}[(f(\mathbf{x}+\mathbf{H}) - \mu)^4]$, $z_{1-\alpha/2}$ is the quantile of risk α of the standard normal distribution and $|e_n|$ is the precision chosen by the user.

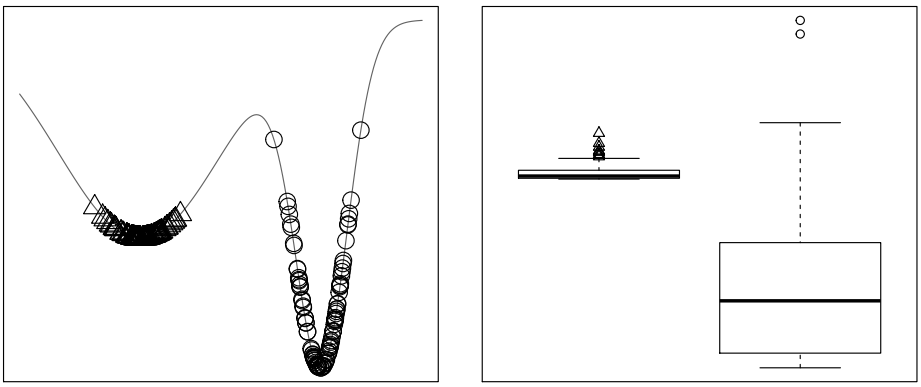

Figure 1: Illustration of the robustness. The figure on the left shows a function in one dimension with two optima, the right one (circles) is the less robust. The figure on the right shows the variability of the points simulated by the same Gaussian law around the two optima.

The demonstration is in appendix A. N is often too large. To overcome this difficulty, we propose to use the Taylor approximation introduced by [Darlington et al., 1999] to quantify the robustness.

For all $\mathbf{h} \in \mathbb{R}^p$, one has:

$$f(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + \nabla_f(\mathbf{x}) \cdot \mathbf{h} + \frac{1}{2}\mathbf{h}'\mathbb{H}_f(\mathbf{x})\mathbf{h} + o(\|\mathbf{h}\|^2)$$

where ∇_f is the gradient of f and $\mathbb{H}_f$ the Hessian matrix of f. We introduce:

$$\widetilde{f}(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + \nabla_f(\mathbf{x}) \cdot \mathbf{h} + \frac{1}{2}\mathbf{h}'\mathbb{H}_f(\mathbf{x})\mathbf{h}$$

Then, we define the robustness criterion by the following approximation of the local variance :

$$RC_f(\mathbf{x}) = Var\left(\widetilde{f}(\mathbf{x}+\mathbf{H})\right)$$

An analytical form of this expression can be calculated (see [Beyer and Sendhoff, 2007]) and is given by the following expression:

$$RC_f(\mathbf{x}) = tr\left(\nabla_f \nabla_f' \Delta^2\right) + \frac{1}{2} tr\left(\mathbb{H}_f^2(\delta_1^2, \ldots, \delta_p^2)(\delta_1^2, \ldots, \delta_p^2)'\right) \tag{4}$$

where tr is the matrix trace. If the output of a simulation provides the results of the function and the first derivatives, RC_f criterion can be computed with only one call to the computer code. However in the context of costly simulations RO cannot be directly done on f and RC_f.
In the next section we present how with a kriging approach these quantities can be predicted.

3 Kriging prediction of the robustness criterion

As it can be seen in Equation (4), the robustness criterion depends on the first and second derivatives. A Gaussian process metamodel is well suited to this context in the sense that all derivatives can easily be predicted from that model. In this section, we present the co-kriging model and the predictions of the two objectives used in the robust optimization.

3.1 Co-kriging Model

This subsection is divided in two main parts. First, we present the general model for the function and its derivatives. Secondly, we introduce the kriging equations of prediction.

3.1.1 General model

We introduce a Gaussian Process (GP) to model the function, its first and second derivatives.

Let $(Y(\mathbf{x}))_{\mathbf{x}\in D}$ be a process with covariance function $k(\mathbf{x},\tilde{\mathbf{x}})$, $\forall(\mathbf{x},\tilde{\mathbf{x}}) \in D\times D$. This process is differentiable in mean square at point $(\mathbf{x},\tilde{\mathbf{x}})$ if and only if $\frac{\partial^2 \mathbf{k}}{\partial x_i \partial \tilde{x}_j}(\mathbf{x},\tilde{\mathbf{x}})$ exists $\forall i,j \in \{1,\dots,p\}$ and finite at point $(\mathbf{x},\tilde{\mathbf{x}}) = (\mathbf{t},\mathbf{t})$. In addition we have:

$$cov\left(Y(\mathbf{x}), \frac{\partial Y(\tilde{\mathbf{x}})}{\partial \tilde{x}_j}\right) = \frac{\partial k(\mathbf{x},\tilde{\mathbf{x}})}{\partial \tilde{x}_j}$$
$$cov\left(\frac{\partial Y(\mathbf{x})}{\partial x_i}, \frac{\partial Y(\tilde{\mathbf{x}})}{\partial \tilde{x}_j}\right) = \frac{\partial^2 k(\mathbf{x},\tilde{\mathbf{x}})}{\partial x_i \partial \tilde{x}_j}$$

We denote by $(Y_{x_i}(\mathbf{x}))_{\mathbf{x}\in D} = \left(\frac{\partial Y}{\partial x_i}(\mathbf{x})\right)_{\mathbf{x}\in D}$ the process $(Y(\mathbf{x}))_{\mathbf{x}\in D}$ differentiated in direction i and by $\left(Y_{x_i,x_j}(\mathbf{x})\right)_{\mathbf{x}\in D} = \left(\frac{\partial^2 Y}{\partial x_i \partial x_j}(\mathbf{x})\right)_{\mathbf{x}\in D}$ two times differentiated process in direction i,j.

Let p be the number of input variables. Then each observation $\mathbf{x}$ is a vector with p coordinates, such that $\mathbf{x} = (x_1,\dots,x_p)$, $\mathbf{x} \in D$. The outputs (function and derivatives) at point $\mathbf{x}^k \in D$ are denoted by $y^k \in \mathbb{R}$, $y^k_{x_i} \in \mathbb{R}$ and $y^k_{x_i,x_j} \in \mathbb{R}$, where $i \in \{1,\dots,p\}$, $j \in \{i,\dots,p\}$ and $k \in \{1,\dots,n\}$. We note the collection of outputs $\mathbf{y}$, $\mathbf{y}_{x_i}$ and $\mathbf{y}_{x_i,x_j}$ such that :

$$\mathbf{y} = (y^1,\dots,y^n)'$$
$$\mathbf{y}_{x_i} = (y^1_{x_i},\dots,y^n_{x_i})'$$
$$\mathbf{y}_{x_i,x_j} = (y^1_{x_i,x_j},\dots,y^n_{x_i,x_j})'$$

Let $d = 1+\frac{3p}{2}+\frac{p^2}{2}$. In kriging context, $(y^k, y^k_{x_1},\dots,y^k_{x_p}, y^k_{x_1,x_1},\dots,y^k_{x_i,x_j},\dots,y^k_{x_p,x_p})$, $k \in \{1,\dots,n\}$ is assumed to be a realization of the following d dimensional GP:

$$Z(\mathbf{x}) = (Y(\mathbf{x}), Y_{x_1}(\mathbf{x}),\dots,Y_{x_p}(\mathbf{x}), Y_{x_1,x_1}(\mathbf{x}),\dots,Y_{x_i,x_j}(\mathbf{x}),\dots,Y_{x_p,x_p}(\mathbf{x})),\ 1\le i\le p,\ i\le j\le p$$

at points $\mathbf{x}^1,\dots,\mathbf{x}^n$ where $\mathbf{x}^{\mathbf{k}} \in D$, $k \in \{1,\dots,n\}$, such that :

$$Y(\mathbf{x}) = \mu + \eta(\mathbf{x})$$
$$Y_{x_i}(\mathbf{x}) = \eta_{x_i}(\mathbf{x})$$
$$Y_{x_i,x_j}(\mathbf{x}) = \eta_{x_i,x_j}(\mathbf{x})$$

where $\mu \in \mathbb{R}$ is the trend, the process $(\eta(\mathbf{x}))_{\mathbf{x}\in D}$ is a centered GP with a stationary covariance function that depends on a vector of range parameters $\boldsymbol{\theta} \in \mathbb{R}_+^p$ such that $Cov(\eta(\mathbf{x}), \eta(\mathbf{x})) = k_{\boldsymbol{\theta}}(\mathbf{x} - \tilde{\mathbf{x}}) = \sigma^2 r_{\boldsymbol{\theta}}(\mathbf{x} - \tilde{\mathbf{x}})$, $\forall(\mathbf{x}, \tilde{\mathbf{x}}) \in D \times D$. In this paper the trend μ and the variance σ^2 are assumed to be constants.

The process vector is then modeled as follow:

$$Z(\mathbf{x}) = m + \epsilon(\mathbf{x}) \tag{5}$$

where $\mathbf{m} = (\mu, 0, \ldots, 0)' \in \mathbb{R}^d$ is the trend vector, the process $(\epsilon(\mathbf{x}))_{\mathbf{x}\in D}$ is the vector of d centered Gaussian processes i.e.

$$\epsilon(\mathbf{x}) = (\eta(\mathbf{x}), \eta(\mathbf{x})_{x_1}, \ldots, \eta(\mathbf{x})_{x_p}, \eta(\mathbf{x})_{x_1,x_1}, \ldots, \eta(\mathbf{x})_{x_i,x_j}, \ldots, \eta(\mathbf{x})_{x_p,x_p}),\ 1 \leq i \leq p,\ i \leq j \leq p$$

3.1.2 Kriging predictions

The co-kriging model presented by [Le Gratiet, 2013] is used to surrogate the function itself and its derivatives. The problem is to predict Z considering observations of z at points $x^1, ..., x^n$. But, the entire vector z is not always observable. Let $u_{obs} \subset \{1, \ldots, d\}$ be the components that are observable. For exemple only the fonction and its first derivatives can be affordable. In the same way it is not always necessary to predict the whole vector z. Let $u_{pred} \subset \{1, \ldots, d\}$ be the components that are to be predicted.

The kriging mean is then given by the following equation :

$$\widehat{\mathbf{z}}_{u_{pred}}(\mathbf{x}) = \mathbf{m} + \mathbf{c}_{\boldsymbol{\theta}}(\mathbf{x})'\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}(\mathbf{z}_{u_{obs}} - \mathbf{m}'\mathbf{F}), \widehat{\mathbf{z}}_{u_{pred}}(\mathbf{x}) \in \mathbb{R}^{d_{pred}} \tag{6}$$

where $\mathbf{z}_{u_{obs}} = (z^1_{u_{obs}}, \ldots, z^n_{u_{obs}})$ the observation vector, $1 \in u_{obs}$ and $d_{obs} = \#u_{obs}$. $\widehat{\mathbf{z}}_{u_{pred}}(\mathbf{x})$ is the prediction vector and $d_{pred} = \#u_{pred}$. The mean square error (MSE) at point $\mathbf{x} \in D$ is given by :

$$\widehat{\mathbf{s}}^2_{u_{pred}}(\mathbf{x}) = \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) - \begin{pmatrix} f' & \boldsymbol{c}_{\boldsymbol{\theta}}(\mathbf{x})' \end{pmatrix} \begin{pmatrix} 0 & \mathbf{F}' \\ \mathbf{F} & \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \end{pmatrix}^{-1} \begin{pmatrix} f \\ \boldsymbol{c}_{\boldsymbol{\theta}}(\mathbf{x}) \end{pmatrix} \tag{7}$$

where $\widehat{\mathbf{s}}^2_{u_{pred}}(\mathbf{x}) \in \mathcal{M}_{d_{pred}\times d_{pred}}$, $\mathbf{F} = (E_{1,1}^{d_{obs}\times d_{pred}}, \ldots, E_{1,1}^{d_{obs}\times d_{pred}})' \in \mathcal{M}_{nd_{obs}\times d_{pred}}$ and $f = E_{1,1}^{d_{pred}\times d_{pred}}$ with $E_{1,1}^{n_1\times n_2}$ the canonical matrix of size $n_1 \times n_2$.

$\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is the covariance matrix of size $nd_{obs} \times nd_{obs}$ given by :

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}} = \begin{pmatrix} \Sigma_{\mathbf{x}_1,\mathbf{x}_1}(u_{obs}, u_{obs}) & \ldots & \Sigma_{\mathbf{x}_1,\mathbf{x}_n}(u_{obs}, u_{obs}) \\ \vdots & \ddots & \vdots \\ \Sigma_{\mathbf{x}_n,\mathbf{x}_1}(u_{obs}, u_{obs}) & \ldots & \Sigma_{\mathbf{x}_n,\mathbf{x}_n}(u_{obs}, u_{obs}) \end{pmatrix}$$

where

$$\Sigma_{\mathbf{x},\tilde{\mathbf{x}}} = \begin{pmatrix} \Sigma_{Y,Y} & \Sigma_{Y,Y_{\tilde{x}_j}} & \Sigma_{Y,Y_{\tilde{x}_j\tilde{x}_k}} & \Sigma_{Y,Y_{\tilde{x}_j^2}} \\ \Sigma_{Y_{x_i},Y} & \Sigma_{Y_{x_i},Y_{\tilde{x}_j}} & \Sigma_{Y_{x_i},Y_{\tilde{x}_j\tilde{x}_k}} & \Sigma_{Y_{x_i},Y_{\tilde{x}_j^2}} \\ \Sigma_{Y_{x_ix_l},Y} & \Sigma_{Y_{x_ix_l},Y_{\tilde{x}_j}} & \Sigma_{Y_{x_ix_l},Y_{\tilde{x}_j\tilde{x}_k}} & \Sigma_{Y_{x_ix_l},Y_{\tilde{x}_j^2}} \\ \Sigma_{Y_{x_i^2},Y} & \Sigma_{Y_{x_i^2},Y_{\tilde{x}_j}} & \Sigma_{Y_{x_i^2},Y_{\tilde{x}_j\tilde{x}_k}} & \Sigma_{Y_{x_i^2},Y_{\tilde{x}_j^2}} \end{pmatrix}$$

$i, j, k, l \in \{1, \dots p\}$ with $l > i$ and $k > j$. For instance $\Sigma_{Y_{x_i}, Y_{\tilde{x}_j}} = cov(Y_{x_i}, Y_{\tilde{x}_j}) = cov(\eta_{x_i}, \eta_{\tilde{x}_j}) = \frac{\partial^2 k(\mathbf{x}-\tilde{\mathbf{x}})}{\partial x_i \partial \tilde{x}_j}$. $\mathbf{c}_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathcal{M}_{nd_{obs} \times d_{pred}}$ is the covariance matrix between $Z_{u_{pred}}(\mathbf{x})$ and the observations. $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) \in \mathcal{M}_{d_{pred} \times d_{pred}}$ is the covariance of $Z_{u_{pred}}(\mathbf{x})$.

3.2 Prediction of f

The prediction of the real function f is given by the co-kriging model and corresponds to the first coordinate of the vector $\widehat{z}_{u_{pred}}(\mathbf{x})$ in Equation (6) when $1 \in u_{pred}$ written:

$$\widehat{z}_{u_{pred}}(\mathbf{x}) = (\widehat{y}(\mathbf{x}), \dots)$$

where $\widehat{y}$ is the prediction of the function f.

3.3 Prediction of RC_f

We propose to predict our robustness criterion by the co-kriging metamodel. The prediction $\widehat{z}_{u_{pred}}$ is used instead of the function to compute the criterion. The prediction of $RC_f(\mathbf{x})$ is given by:

$$RC_{\widehat{y}}(\mathbf{x}) = tr\left(\nabla_{\widehat{y}} \nabla'_{\widehat{y}} \Delta^2\right) + \frac{1}{2} tr\left(\mathbb{H}^2_{\widehat{y}} (\delta_1^2, \dots, \delta_p^2)' (\delta_1^2, \dots, \delta_p^2)\right) \tag{8}$$

where $\nabla_{\widehat{y}}$ is the vector $\begin{pmatrix} \widehat{y}_{x_1} \\ \vdots \\ \widehat{y}_{x_p} \end{pmatrix}$ and is the prediction of the gradient. $\mathbb{H}_{\widehat{y}}$ is the matrix $\begin{pmatrix} \widehat{y}_{x_1,x_1} & \dots & \widehat{y}_{x_1,x_p} \\ \vdots & \ddots & \vdots \\ \widehat{y}_{x_p,x_1} & \dots & \widehat{y}_{x_p,x_p} \end{pmatrix}$ and corresponds to the prediction of the hessian matrix. $\nabla_{\widehat{y}}$ and $\mathbb{H}_{\widehat{y}}$ are obtained from different components of $\widehat{z}_{upred}$.

3.4 Illustration with the six-hump Camel function

The studied function is the six-Hump Camel function, defined by:

$$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1 x_2 + \left(-4 + 4x_2^2\right) x_2^2, \mathbf{x} \in [-2; 2] \times [-1; 1]$$

We consider for the kriging model a kernel which is *anisotropic*:

$$cov\left(Y(\mathbf{x}), Y(\tilde{\mathbf{x}})\right) = k(\mathbf{x} - \tilde{\mathbf{x}}) = \sigma^2 \prod_{j=1}^{p} \rho_{\theta_j}\left(|x_j - x'_j|\right), \boldsymbol{\theta} = (\theta_1, \dots, \theta_p) \in \mathbb{R}_+^p \tag{9}$$

where ρ_{θ_j} is a correlation function which only depends on the one dimensional range parameter θ_j, see e.g [Santner et al., 2003] and [Stein, 1999]. The *anisotropic* kernel contains as many parameters as the number of variables p. We use a Matern 5/2 kernel because the output is supposed to be two times continuously differentiable:

$$\forall \theta \in \mathbb{R}^+, \forall h \in \mathbb{R}^+, \rho_\theta(h) = \left(1 + \frac{\sqrt{5}|h|}{\theta} + \frac{5h^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}|h|}{\theta}\right).$$

We choose a maximin latin hypercube learning set of 10 points. The test set is a space filling of 1500 points. In the first kriging, without the derivatives, the observations are $y_1, \dots, y_{10}$ where $y_i = f(\mathbf{x}_i)$.

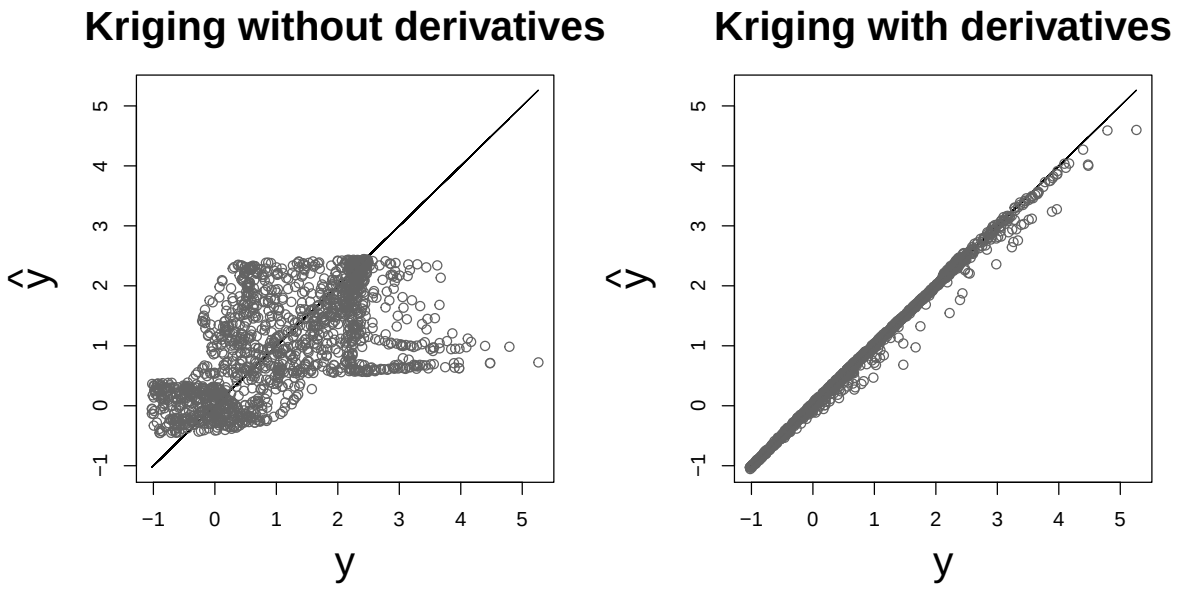

Figure 2: Prediction plots for the Six-Hump Camel function: 10 points without observation of the derivatives (on the left), 10 with 5 derivatives (on the middle) and 60 points without observation of the derivatives (on the right).

In the second kriging, with the derivatives, the observations are $z_1, \ldots, z_{10}$ where

$$z_i = \left(f(\mathbf{x}_i), \frac{\partial f(\mathbf{x}_i)}{\partial x_1}, \frac{\partial f(\mathbf{x}_i)}{\partial x_2}, \frac{\partial^2 f(\mathbf{x}_i)}{\partial x_1 \partial x_2}, \frac{\partial^2 f(\mathbf{x}_i)}{\partial x_1^2}, \frac{\partial^2 f(\mathbf{x}_i)}{\partial x_2^2} \right).$$

In the third kriging, without the derivatives but with more observation points written $y_1, \ldots, y_{60}$ where $y_i = f(\mathbf{x}_i)$.

As expected, Figure 2 shows that kriging with derivatives does much better than without in the case of 10 points. If we consider that the computational cost of one derivative is the same as computing a new point, kriging without derivatives is better. In industrial application, computing all derivatives is cheaper than computing a new point.

4 Robust optimization procedure

In this section, we present the robust optimization procedure that uses our robustness criterion (cf Equation (8)). The approach to solve this optimization problem in the context of time consuming simulations is based on a classical black-box optimization scheme (see [Jones et al., 1998]). The robust optimization problem is written as:

$$\begin{aligned} &\text{Find vectors } \mathbf{x}_0 \text{ in a Pareto optimal sense such that} \\ &\mathbf{x}_0 = \underset{\mathbf{x} \in \mathbb{R}^p}{\operatorname{argmin}} \{ f(\mathbf{x}), RC_f(\mathbf{x}) \} \end{aligned} \tag{10}$$

The size of the initial sample set and the size of the batches are given by the user. The procedure is divided in two main parts: step 1, 2, 3 and step 4, 5, 6. The first is the initial part for the design and the Gaussian process. The second part solves the optimization problem using the metamodel and is

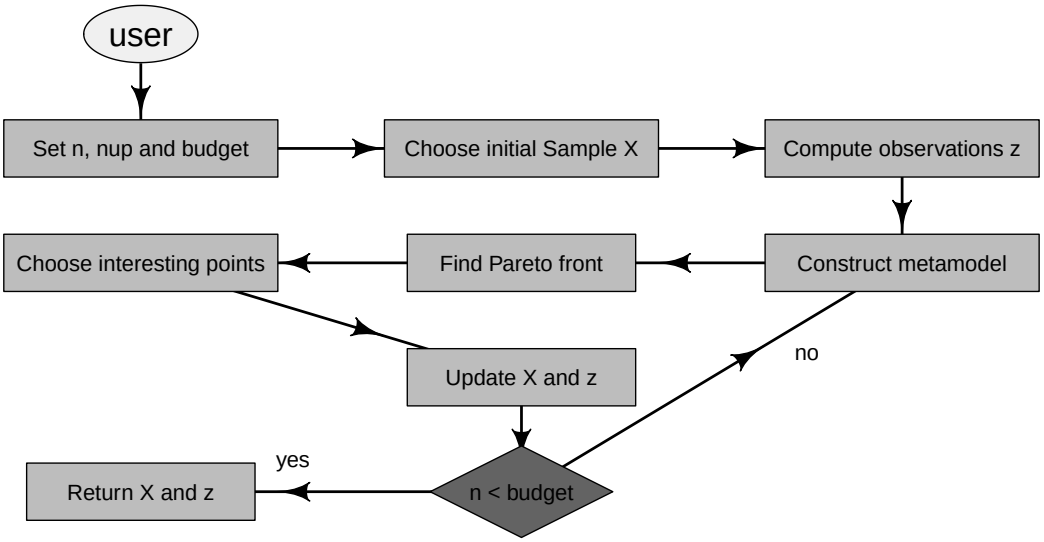

Figure 3: The robust optimization procedure

iteratively repeated until the budget is reached. **Step 1** generates the initial sample $\mathbb{X}$ of n points spread in the p design variable space. An Optimized Latin Hypercube (OLH) is chosen. The observations of the function and its derivatives, when there are available, are computed in **Step 2**. In **Step 3** the co-kriging metamodel is estimated based on the observations. **Step 4** searches for non-dominated solutions of a kriging-based multi-objective problem using the optimizer NSGA II see e.g. [Deb et al., 2002]. **Step 5** aims at choosing the best batch of points among the Pareto front. These points are added to the design $\mathbb{X}$ in **Step 6** and the simulation is run on these new points. The response surface f is then updated with the new observations. Figure 3 shows the general scheme.

Several choices are possible for **Step 4** and **Step 5**. The next section describes them.

5 Sequential procedure for the acquisition of new points

We have developed seven enrichment strategies based on three main MO optimization problems using the previous criterion. For the first problem, a NSGA II algorithm is applied to the prediction of the function and to the prediction of the robustness criterion (cf Equation (6) and (8)). Once the Pareto front is found, points are chosen using the kriging variance. That's the enrichment step. A problem could appear during the Pareto front construction if kriging predictions turn out to be of poor quality. Some interesting areas can be missed. That is why, we introduce a second approach. The multiobjective optimization is conducted on the EI of the function and the robustness criterion. In this case the kriging variance is already taken into account during the optimization. Then several points are chosen among the Pareto front through different criteria. Selecting the good points from the Pareto front is not easy, so we introduce a last approach using qEI. This criterion measures the improvement of a batch of points. This strategy takes into account the kriging variance in the MO and add points by batch. The best point in the qEI space is selected in the Pareto front and corresponds to a set of q points in the design space. Before doing this, we recall some results on EI.

5.1 Background

In the EGO algorithm, the expected improvement (EI) criterion measures the improvement of a point $\mathbf{x}$ in the minimization of function f and is used to add new points to the learning set. The expression of the EI (cf [Jones et al., 1998]) at point $\mathbf{x}$ is:

$$EI(\mathbf{x}) = \mathbb{E}\left[(\min(y(\mathbb{X})) - Y(\mathbf{x}))^+ \,|Y(\mathbb{X}) = \mathbf{y}\right]$$

where $\min(y(\mathbb{X})) = \min(y^1, \ldots, y^n)$.
The analytical expression of the EI for a Gaussian process is given by:

$$EI(\mathbf{x}) = (\min(y(\mathbb{X})) - \widehat{y}(\mathbf{x})\Phi\left(\frac{\min(y(\mathbb{X})) - \widehat{y}(\mathbf{x})}{\widehat{s}}\right) + \widehat{s}\phi\left(\frac{\min(y(\mathbb{X})) - \widehat{y}(\mathbf{x})}{\widehat{s}}\right)$$

where $\widehat{y}(\mathbf{x})$ is the kriging mean, $\widehat{s}(\mathbf{x})$ is the kriging standard deviation, Φ and ϕ are the cdf and pdf of the standard normal law.

In our case, we perform a multi-objective optimization on f and RC_f. Then, to evaluate an EI on RC_f we need to define the process $(RC_Y(\mathbf{x}))_{\mathbf{x}\in D}$. From Equation 4 the process is:

$$\begin{aligned} RC_Y(\mathbf{x}) &= tr\left(\begin{pmatrix} Y_{x_1}(\mathbf{x}) \\ \vdots \\ Y_{x_p}(\mathbf{x}) \end{pmatrix}\begin{pmatrix} Y_{x_1}(\mathbf{x}) & \ldots & Y_{x_p}(\mathbf{x}) \end{pmatrix}\begin{pmatrix} \delta_1^2 & & \\ & \ddots & \\ & & \delta_p^2 \end{pmatrix}\right) \\ &+ \frac{1}{2}tr\left(\begin{pmatrix} Y^2_{x_1,x_1}(\mathbf{x}) & \ldots & Y^2_{x_1,x_p}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ Y^2_{x_p,x_1}(\mathbf{x}) & \ldots & Y^2_{x_p,x_p}(\mathbf{x}) \end{pmatrix}\begin{pmatrix} \delta_1^2 \\ \vdots \\ \delta_p^2 \end{pmatrix}\begin{pmatrix} \delta_1^2 & \ldots & \delta_p^2 \end{pmatrix}\right) \\ &= \sum_{i=1}^{p} Y_{x_i}(\mathbf{x})^2\delta_i^2 + \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} Y^2_{x_i,x_j}(\mathbf{x})\delta_i^2\delta_j^2 \end{aligned} \tag{11}$$

We can not conduct the multi-objective optimization on two independent EI. That is why,in [Jeong and Obayashi, 2005] the authors change the reference value of the EI to adapt it to the multi-objective case. We recall that $\mathbf{y} = (y^1 = f(\mathbf{x}^1), \ldots, y^n = f(\mathbf{x}^n))$. Let $RC_{\mathbf{y}} = \left(RC_{y^1} = RC_f(\mathbf{x}^1), \ldots, RC_{y^n} = RC_f(\mathbf{x}^n)\right)$ be the evaluation of the robustness criterion on the design points. The multi-objective EI are:

$$\begin{aligned} EI_y(\mathbf{x}) &= \mathbb{E}\left[(\max(y(\mathbb{X}^*)) - Y(\mathbf{x}))^+ \,|\mathbf{z}_{u_{obs}}\right] \\ EI_{RC_y}(\mathbf{x}) &= \mathbb{E}\left[(\max(RC_y(\mathbb{X}^*)) - RC_Y(\mathbf{x}))^+ \,|\mathbf{z}_{u_{obs}}\right] \end{aligned}$$

where $\mathbb{X}^*$ is the set of non-dominated points for the objectives $\{y, RC_y\}$ of the learning set $\mathbb{X}$.

Remark:

- *A solution $\mathbf{x}^1$ dominates another solution $\mathbf{x}^2$ for the m objectives $g_1, \ldots, g_m$ if and only if $\forall i \in \{1, \ldots, m\}$ $g_i(\mathbf{x}^1) \leq g_i(\mathbf{x}^2)$ and $\exists i \in \{1, \ldots, m\}$ $g_i(\mathbf{x}^1) < g_i(\mathbf{x}^2)$. Among a set of solution $\mathbb{X}$, the non-dominated set $\mathbb{X}^*$ (Pareto front) are those that are not dominated by any member of the set $\mathbb{X}$.*

- *When the derivatives used to compute the robustness criterion are not observed we replace them*

by the kriging prediction in $\max(RC_y(\mathbb{X}^*))$.

- *The link between* $RC_Y(\mathbf{x})$ *and* $Z(\mathbf{x})$ *being not linear, the process* $(RC_Y(\mathbf{x}))_{\mathbf{x}\in D}$ *is not Gaussian anymore.* EI_{RC_y} *is then estimated by a Monte Carlo method.*

The EI makes a good balance between exploration and minimization but it computes the improvement of a single point. The multi-point EI (q-EI) developed by [Ginsbourger et al., 2010] is used to measure the improvement of q points $\mathbf{X} = (\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+q})'$.

$$\begin{aligned} qEI(\mathbf{X}) &= EI(\mathbf{x}^{n+1}, \ldots, \mathbf{x}^{n+q}) \\ &= \mathbb{E}\left[\left(\min(y(\mathbb{X})) - \min(Y(\mathbf{x}^{n+1}), \ldots, Y(\mathbf{x}^{n+q}))\right)^+ |\mathbf{z}_{u_{obs}}\right] \end{aligned}$$

In a multi-objective problem the q-EI are:

$$\begin{aligned} qEI_y(\mathbf{X}) &= \mathbb{E}\left[\left(\max(y(\mathbb{X}^*)) - \min(Y(\mathbf{x}^{n+1}), \ldots, Y(\mathbf{x}^{n+q}))\right)^+ |\mathbf{z}_{u_{obs}}\right] \\ qEI_{RC_y}(\mathbf{X}) &= \mathbb{E}\left[\left(\max(RC_y(\mathbb{X}^*)) - \min(RC_Y(\mathbf{x}^{n+1}), \ldots, RC_Y(\mathbf{x}^{n+q}))\right)^+ |\mathbf{z}_{u_{obs}}\right] \end{aligned}$$

5.2 Multi-objective optimization on the kriging predictor

The kriging prediction $\hat{y}$ and the robustness criterion $RC_{\hat{y}}$ are used instead of the real function f and the robustness criterion RC_f in the first group of strategies. The optimization problem of **Step 4** is written as:

$$\begin{aligned} &\text{Find vectors } \mathbf{x}_0 \text{ in a Pareto optimal sense such that} \\ &\mathbf{x}_0 = \underset{\mathbf{x}\in\mathbb{R}^p}{\operatorname{argmin}}\{\hat{y}, RC_{\hat{y}}\} \end{aligned}$$

The NSGA II is used to compute the Pareto front. Five enrichment approaches to select q points from the Pareto front have been benchmarked and are described below (**Step 5**):

1. MyAlea: $\lfloor \frac{q}{2} \rfloor$ [1] points are selected randomly on the Pareto front and $q - \lfloor \frac{q}{2} \rfloor$ points are randomly chosen in the parameter space.

2. MyEI: $-EI_y$ as well as $-EI_{RC_y}$ are computed for each point of the Pareto front. A k-means clustering using the method of [Hartigan and Wong, 1979] is applied to the non-dominated points of $\{-EI_y, -EI_{RC_y}\}$ to provide q clusters. Then the q clusters' medoids are added to the design.

3. MyqEI: a simulated annealing algorithm gives the set of q points among the Pareto front that minimizes the function $-qEI_y - qEI_{RC_y}$.

Two sequential approaches presented in [Ginsbourger et al., 2010] can be used as the replacement of the q-EI to measure the improvement of q points: the Kriging Believer and the Constant Liar.

4. MyKB: q points are sequentially selected from the Pareto front based on the Kriging Beliver strategy. The $-EI_y$ and $-EI_{RC_y}$ are computed on the Pareto front, then a point $\mathbf{x}_0^1$ is randomly

[1] $\lfloor . \rfloor$ is the floor function

chosen from the EI Pareto front and added. $\widehat{y}(\mathbf{x}_0^1)$ is then considered known and is assumed to be equal to $\widehat{y}(\mathbf{x}_0^1)$. Another computation of $-EI_y$ and $-EI_{RC_y}$ provides one more point based on the same strategy up to the q requested points.

5. MyCL: q points are sequentially selected based on the Constant Liar strategy. The $-EI_y$ and $-EI_{RC_y}$ are computed on the Pareto front, then a point $\mathbf{x}_0^1$ is randomly chosen from the EI Pareto front and added. $y(\mathbf{x}_0^1)$ is then considered known and is assumed to be equal to $\min y(\mathbb{X}^*)$. Another computation of $-EI_y$ and $-EI_{RC_y}$ provides one more point based on the same strategy up to the q requested points.

The problem with this group of strategies is that the kriging variance is not taken into account during the multi-objective optimization. Some interesting areas can be missed because the methods will always add points in the same place except for the MyAlea strategy. The second approach solves this issue by conducting the MO optimization directly on the EI.

5.3 Multi-objective optimization on the expected improvement criterion

In the second group of strategies, the multi-objective optimization is performed on the EI of the output and of the robustness criterion. This approach takes into account the kriging variance from the beginning of the procedure. The multi-objective problem that is computed in **Step 4** is the following:

$$\begin{aligned}&\text{Find vectors } \mathbf{x}_0 \text{ in a Pareto optimal sense such that}\\ &\mathbf{x}_0 = \underset{\mathbf{x}\in D}{\operatorname{argmin}}\{-EI_y, -EI_{RC_y}\}\end{aligned}$$

The Pareto front is found by the NSGA II algorithm. For this approach, one enrichment strategy is proposed to add one point in **Step 5** and is the following:

6. MEIyAlea: a point is randomly chosen and sequentially added until the total budget is reached.

This strategy add point sequentially ($q = 1$) not anymore by batch ($q > 1$).

The last group is introduced to overcome this remark. The qEI is used instead of the EI to measure the improvement of a batch of points instead of one point.

5.4 Multi-objective optimization on the multi-point expected improvement criterion

In order to take into account the kriging variance in the optimization and to add points by batch, we modify the objectives. The multi-objective optimization is performed on the qEI of the outputs kriging prediction and of the robustness criterion. **Step 4** searches for non-dominated points of

Method	Minimization	Interesting points	Updates
MyAlea	y, RC_y	Random points on the Pareto front and the parameters space	Batch
MyEIClust	y, RC_y	Cluster on EIy and EIRCy	Batch
MyqEI	y, RC_y	annealing algorithm on qEIy and qEIRCy	Batch
MyKB	y, RC_y	Kriging believer	Batch
MyCL	y, RC_y	Constant liar	Batch
MEIyAlea	EI_y, EI_{RC_y}	Random point on the Pareto front	Seq
MqEIyAlea	qEI_y, qEI_{RC_y}	Random point on the Pareto front	Batch

Table 1: Minimization problems and methods to choose the interesting points.

the following problem:

$$\text{Find vectors } \mathbf{x}_0 \text{ in a Pareto optimal sense such that}$$
$$\mathbf{x}_0 = \underset{\mathbf{x} \in D \subset \mathbb{R}^{p \times q}}{\operatorname{argmin}} \{-qEI_y, -qEI_{RC_y}\}$$

This optimization scheme is of size $p \times q$, which limits its use. One enrichment approach has been benchmarked and is described below to add q points in **Step 5**.

7. MqEIyAlea : for each Pareto front from **Step 4**, one point is randomly extracted from the qEI space, this point will provide q points in the parameter space for the next optimization step.

The seven methods to choose interesting points in **Step 5** are summarized in the Table 1.

6 Applications

In order to compare the seven strategies, several measures exist to quantify the quality of a Pareto front (cf [Van Veldhuizen, 1999], [Schott, 1995], [Deb et al., 2002] and [Zitzler and Thiele, 1999]). We decide to focus on two of them that are the most popular. Let $\mathbf{f} = (f_1, \ldots, f_m)$ be the objective functions, $\mathcal{P}$ the theoretical Pareto front and $\mathbb{X}^*$ the empirical Pareto front where $N = \#\mathcal{P}$. The chosen performance metrics are:

- Inverted Generational distance (IGD) see [Van Veldhuizen, 1999]:

$$IGD(\mathbb{X}^*) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i}$$

where $d_i = \min_{\mathbf{x} \in \mathbf{X}^*}(\|\mathbf{f}(\mathbf{x}^i) - \mathbf{f}(\mathbf{x})\|_2)$, $\mathbf{f}(\mathbf{x}^i) \in \mathcal{P}$. This metric evaluates the distance between the empirical and the theoretical Pareto front. A small value is better.

- Hypervolume (HV) see [Zitzler and Thiele, 1999]. The Figure 4 shows the Hypervolume (HV) of a Pareto front. [Fonseca et al., 2006] introduce an algorithm to compute this volume. We compare the empirical HV to the theoretical.

This section compares the strategies in two test functions. The first one is the six-hump Camel in two dimensions. We apply the seven strategies in two cases: the observations of the function and the derivatives are available and only the observations of the function are available. The second test function is the

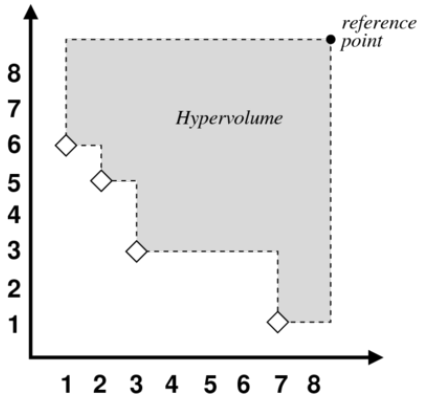

Figure 4: Hypervolume: the diamond represent the individuals of the empirical Pareto front $\mathbb{X}^*$. The black circle is the Nadir point of the set $\mathbb{X}^*$.

Hartmann in six dimensions. We divided the case in same way than the previous function: observations of the functions and the derivatives then only observations of the function. In the same way as previously we consider that derivatives are affordable on one hand; on the other hand that only the function is available. For the sake of efficiency only three of the best strategies are applied on Hartmann function.

6.1 Six-hump Camel function: 2D

In this application, we consider the six-hump Camel function. The two input variables are supposed to suffer uncertainties modeled with a Gaussian law with a standard deviation of $\delta_j = \frac{0.05}{4}(\max(x_j) - \min(x_j))$, $j = \{1, 2\}$. Then:

$$(\mathbf{x} + H) \sim \mathcal{N}\left(\mathbf{x}, \begin{pmatrix} \delta_1^2 & 0 \\ 0 & \delta_2^2 \end{pmatrix}\right)$$

The Figure 5 shows that the algorithms have to found four areas.

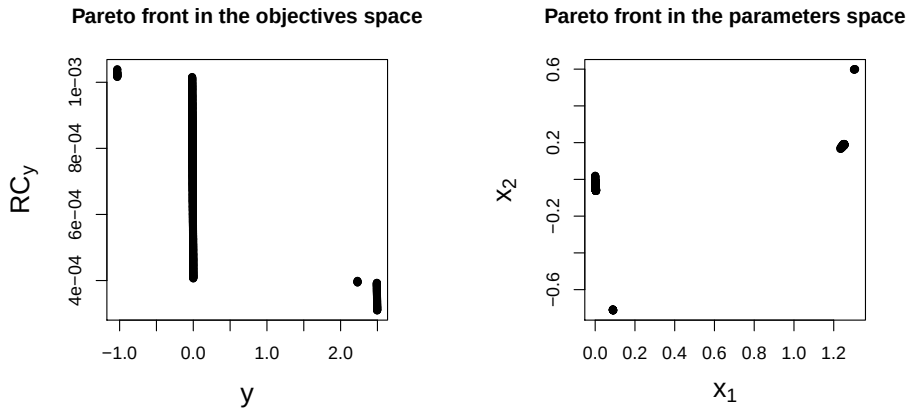

Figure 5: Pareto front of the six-hump Camel function in the objectives space (left) and in the parameters space (right)

As we aim at performing a robust optimization, the function and all the first and second derivatives are to be predicted. The set of predicted indexes is $u_{pred} = \{1, \ldots, 6\}$ that corresponds to the processes

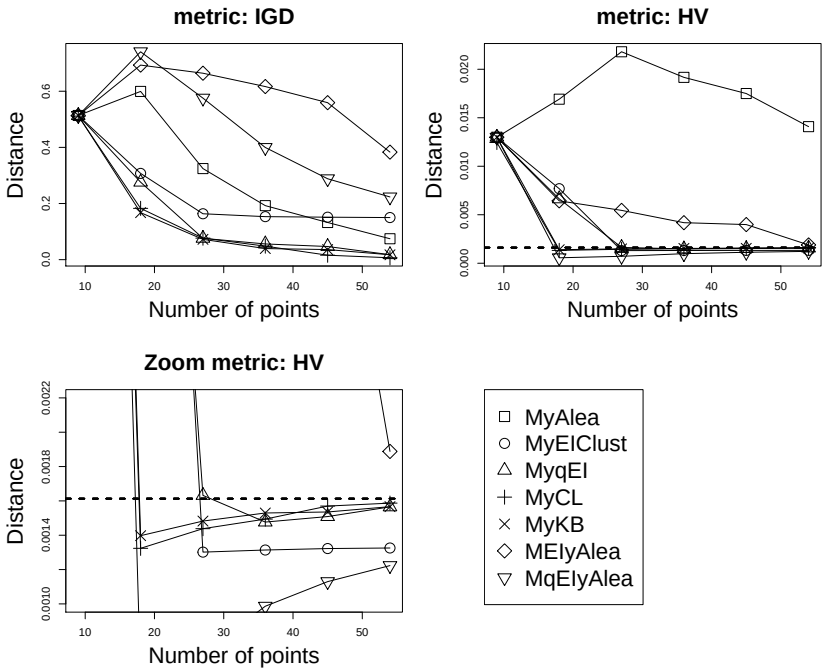

Figure 6: Six-hump Camel function with derivatives observations. Evolution of the Pareto metrics with the number of points compute for all the methods over 100 different runs of the algorithm. The HV value of the theoretical front is represented by the doted line.

vector:

$$Z_{u_{pred}} = (Y, Y_{x_1}, Y_{x_2}, Y_{x_1,x_2}, Y_{x_1,x_1}, Y_{x_2,x_2})$$

6.1.1 Derivatives' observations

In this first part of the study we consider that the function and all derivatives are available at each evaluated point. $u_{obs} = \{1, \ldots, 6\}$ that corresponds to the processes vector:

$$Z_{u_{obs}} = (Y, Y_{x_1}, Y_{x_2}, Y_{x_1,x_2}, Y_{x_1,x_1}, Y_{x_2,x_2})$$

The initial sample set is composed of 5 points. Nine updates of 5 points are added for a total budget of 54 points. The optimization scheme is performed 100 times with different initial learning sets to compare the seven strategies.

Results are provided in Figure 6 and Table 2. In the table, we compare the methods with the computation time and the number of areas found after 54 evaluations. In the figure the methods are compared through two Pareto front performances metrics.

Our analysis is as follow: the MyKB and MyCL are the two most efficient strategies in terms of metrics, areas found and computation times. Then MyqEI, MEIClust and MqEIyAlea gives good results for the metrics and the areas. Even if the MyqEI is quite better in metrics and MqEIyAlea in areas. Finally MyAlea and MEIyAlea are the worst efficient metrics in areas and metrics. In addition, MEIyAlea and

Method	Updates	Computation time	Nb areas
MyAlea	Batch	2 min	1.83
MyEIClust	Batch	2 min	2.73
MyqEI	Batch	6 min 30 sec	2.85
MyKB	Batch	3 min	3.77
MyCL	Batch	3 min	3.68
MEIyAlea	Seq	1 h	1.61
MqEIyAlea	Batch	3 h 30 min	3.06

Table 2: Summarize of the results obtained with the seven strategies on 100 simulation on the six-hump Camel function with derivatives observation. The theoretical number of areas is 4.

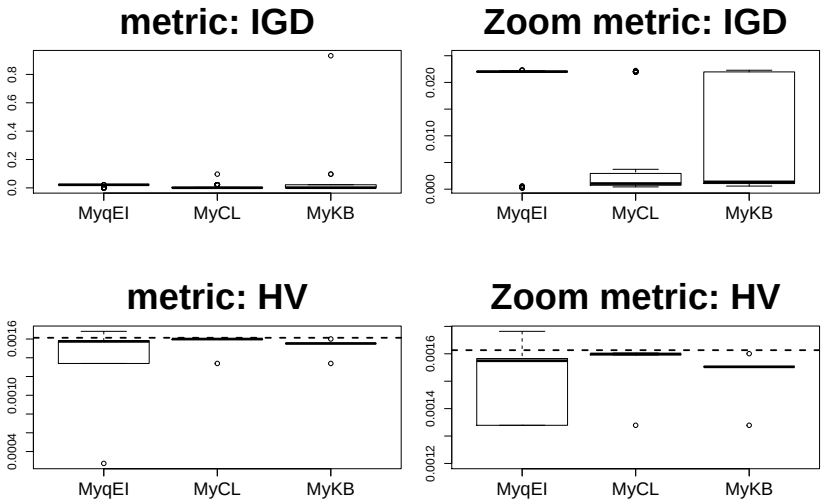

Figure 7: Boxplots of the metrics computed for the three best methods over 100 simulations for the six-hump Camel function with derivative observations.

MqEIyAlea are really time consuming. Then, the best methods selected to be used for robust optimization of limited budget application are MyqEI, MyCL and MyKB, which fully exploit batch computation of EI without excessive computational cost. Figure 7 shows the boxplots of these three methods for each distance. We can see on this figure that the MyqEI method gives in mean the worst results. It comes from the annealing simulation of the strategy that is difficult to tune.

6.1.2 No derivatives' observations

The aim of this section is to analyze the behavior of the seven strategies when the derivatives observations are not available.
The indexes set is $u_{obs} = \{1\}$ and $u_{pred} = \{1, \dots, 6\}$ that corresponds to the processes vectors:

$$Z_{u_{obs}} = Y$$
$$Z_{u_{pred}} = (Y, Y_{x_1}, Y_{x_2}, Y_{x_1,x_2}, Y_{x_1,x_1}, Y_{x_2,x_2})$$

The initial sample set is still a space filling of 5 points. Because available information is poorer than in the previous section more points need to be added to allow the detection of the front. That's why

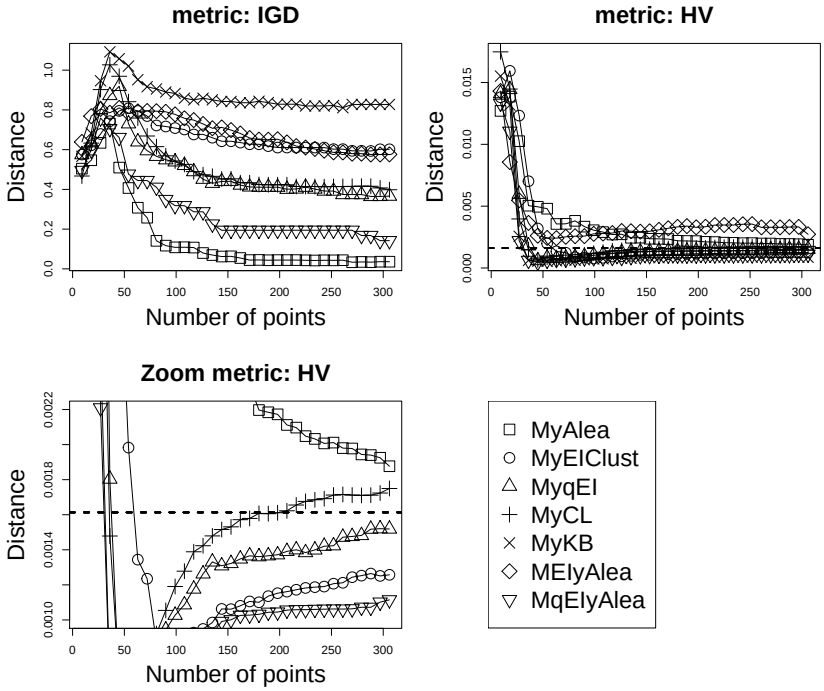

Figure 8: Six-hump Camel function without derivatives observations. Evolution of the Pareto metrics with the number of points compute for all the methods over 100 different runs of the algorithm. The HV value of the theoretical front is represented by the doted line.

35 updates of 5 points are performed until a total budget of 324 points. The optimization scheme is performed 100 times with different initial learning sets to compare the seven strategies.

Results are provided in Figure 8 and Table 3. Our analysis is as follow : the six hump Camel function is

Method	Updates	Computation time	Nb areas
MyAlea	Batch	18 min	2.98
MyEIClust	Batch	11 min	1.94
MyqEI	Batch	58 min	2.53
MyCL	Batch	15 min	2.58
MyKB	Batch	15 min	1.91
MEIyAlea	Seq	5 h 47 min	1.15
MqEIyAlea	Batch	15h17 min	3.57

Table 3: Summarize of the results obtained with the seven strategies on 100 simulation on the six-hump Camel function without derivatives observation. The theoretical number of areas is 4.

difficult to approximate without the information on derivatives. The MyAlea strategy that does not used too much kriging informations to enrich the set gives the best results. In this context, it is a good thing to not entirely trust kriging. The MyqEI and MqEIyAlea strategies provide quite good results because they use the qEI criterion that takes into account the improvement provided by a batch of points of the front. However, MqEIyAlea is too time consuming. The MyCL strategy that does not trust the response surface gives quite good results too, contrary to the MyKB. Finally, the MyEIClust and MEIyAlea stategies that use the EI criterion provide poor results. Even, if the MyEIClust strategy is quite better thanks to the clustering used to enrich the set. The best strategy is MyAlea but we also retain MyqEI and MyCL in

order to test them in higher dimension.

6.2 Hartmann function: 6D

In this section, we benchmark the three best strategies identified in Section 6.1.1 in higher dimension (6D). The kriging model uses the anisotropic kernel with the Matern5_2 covariance function. The function studied is the Hartmann six-dimensional defined by:

$$f(\mathbf{x}) = -\sum_{i=1}^{4} \alpha_i exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right) x_1^2, \mathbf{x} \in [0;1]^2$$

with $\alpha = (1, 1.2, 3, 3.2)'$,

$$A = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 18 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}$$

and

$$P = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}$$

We consider that the random variables are $\mathbf{x}_4$ and $\mathbf{x}_5$ and follow a Gaussian law with a standard deviation of $\delta_j = \frac{0.05}{4}(\max(x_j) - \min(x_j))$, $j = \{4, 5\}$.

We consider two cases : in the first one we have access to the observations of the function and the derivatives associated to the perturbations variables and in the second one we have only access to the observations of the function.

6.2.1 Derivatives' observations

The indexes set is $u_{obs} = u_{pred} = \{1, 5, 6, 20, 26, 27\}$ that corresponds to the processes vector

$$Z_{u_{obs}} = Z_{u_{pred}} = (Y, Y_{x_4}, Y_{x_5}, Yx_4, x_5, Y_{x_4,x_4}, Y_{x_5,x_5})$$

The initial sample set is composed of 18 points. Five updates are made and 18 points are added by update for a total budget of 108 points. We apply the best methods found in the previous test case with derivatives informations: MyqEI, MyCL and MyKB strategies.

The left part of Figure 9 shows that the three methods converge to the real front. At step 2, MyqEI gives the more advanced front. At final step the three methods give the same good results (see the right part of Figure 9). MyKB and MyCL take 10 min for the five steps when MyqEI takes 12 minutes.

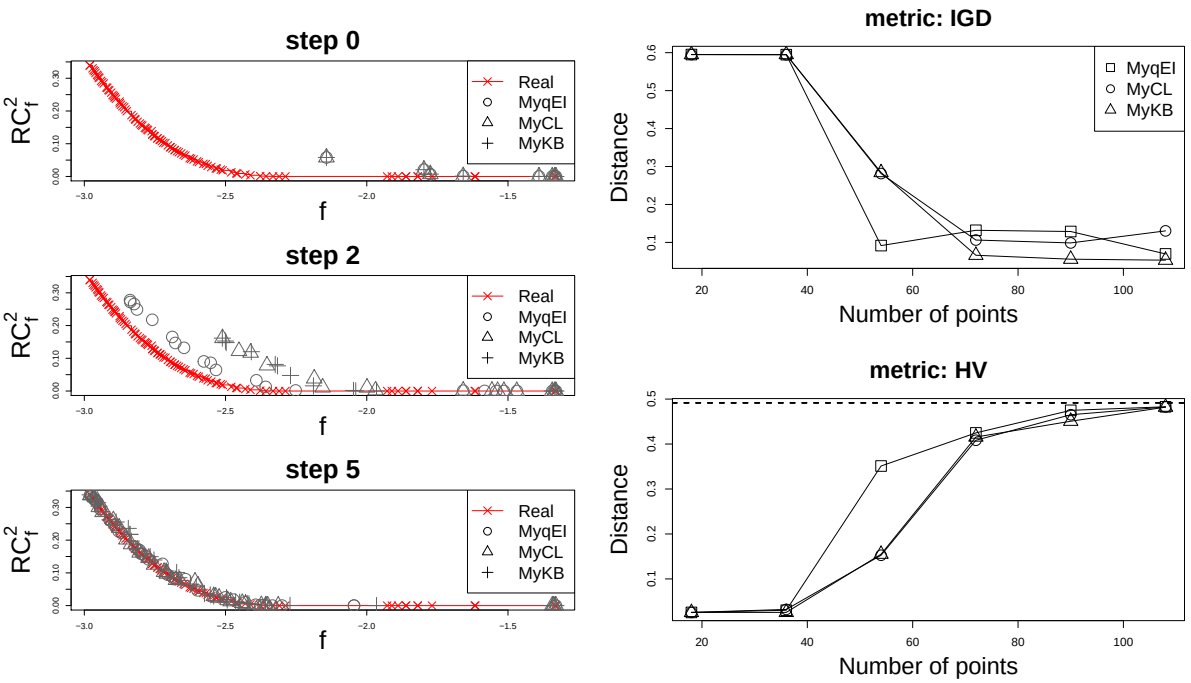

Figure 9: On the left: Pareto fronts obtained during the optimization procedure of the three strategies at initial step (step 0), middle step (step 2) and final step (step 5). On the right: Evolution of the metrics computed during the algorithm for all the methods over 100 simulations for the Hartmann function with derivatives observations. The HV value of the theoretical front is represented by the doted line.

6.2.2 No derivatives' observations

The indexes set is $u_{obs} = \{1\}$ and $u_{pred} = \{1, 5, 6, 20, 26, 27\}$ that corresponds to the processes vector

$$Z_{u_{obs}} = Y$$
$$Z_{u_{pred}} = (Y, Y_{x_4}, Y_{x_5}, Yx_4, x_5, Y_{x_4,x_4}, Y_{x_5,x_5})$$

Like in previous section the initial design is composed of 18 points. In the same way as for the six-hump Camel more updates are added when derivatives are not affordable. Here 35 updates of 18 points are sequentially computed until a total budget of 648 points. We apply the best methods identified in Section 6.1.2: MyAlea, MyqEI and MyCL strategies.

The left part of Figure 10 shows that the three methods converge to the real front. At step 5, all methods have almost found the entire front. The bottom part of the front is difficult to localize even with 578 additional points. The right part of Figure 10 shows that the distance starts to converge to the expected value in the 100 first points. Since then the distances are a little bit unstable. For the IGD metric, the values are subject to little perturbations around the expected value zero, that is a good thing. For the HV measure, we observe high perturbations of the volumes. This phenomenon is explained by the fact that at some step a new area is explored and a new Nadir point appears. That is why, the volume can be small at some iterations until the method converges toward the theoretical non-dominated points. Figure 11 illustrates the case where a new reference point appears because the non-dominated set is updated. MyAlea takes 1h15min, MyqEI takes 1h40min and MyCL takes 1h04min for the 35 steps.

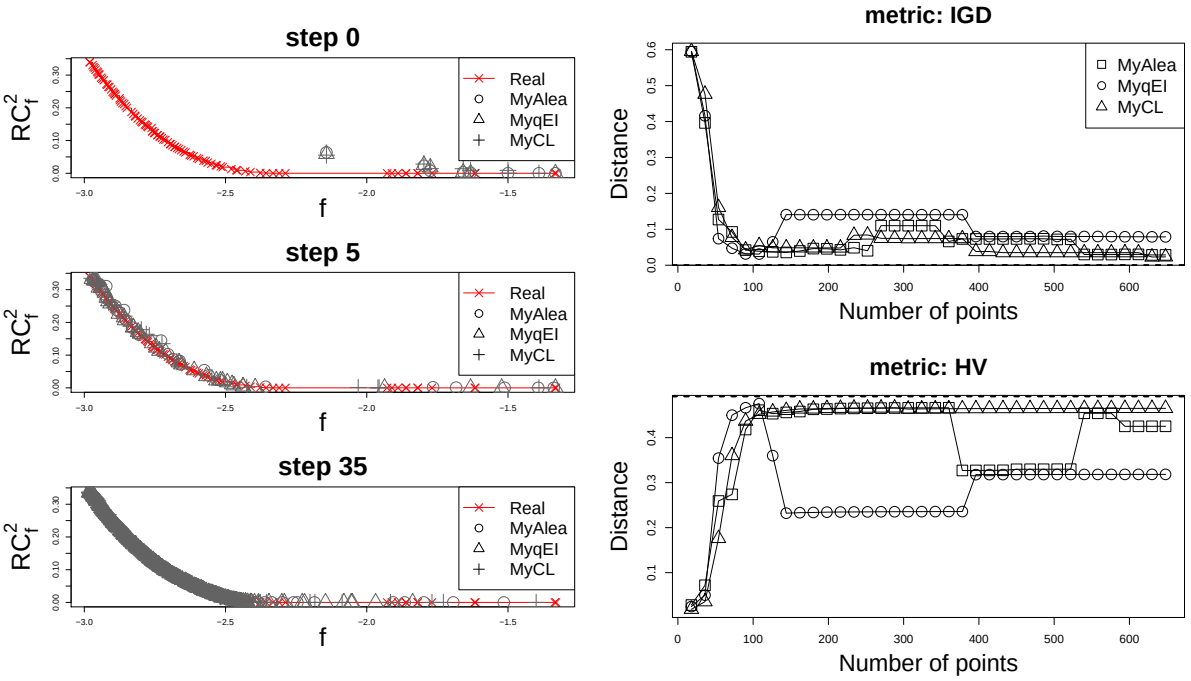

Figure 10: On the left: Pareto fronts obtained during the optimization procedure of the three strategies at initial step (step 0), step 5 and final step (step 35). On the right: Evolution of the metrics during the algorithm compute for all the methods in 100 simulations for the six-hump Camel function with no derivatives observation. The HV value of the theoretical front is represented by the doted line.

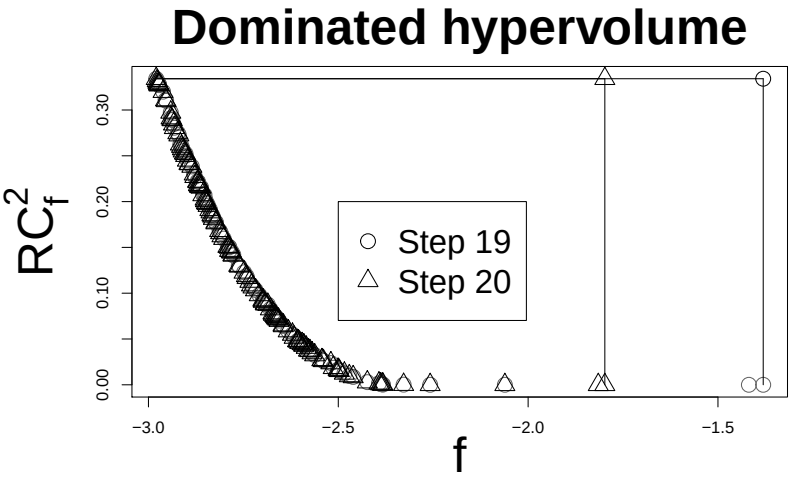

Figure 11: Dominated hypervolumes obtained during the MyAlea strategy at the step 19 (circles) and step 20 (triangle) that corresponds respectively to 360 points and 378 points.

7 Conclusion

In this article, we propose a robust optimization procedure based on the prediction of the function and its derivatives by a co-kriging model. First of all, we describe the robustness criterion based on the Taylor development. Then, the co-kriging model is introduced. Finaly, the robust optimization is performed on both the function and the robustness criterion. A Pareto front of the robust solutions is generated by a genetic algorithm named NSGA-II. We detail seven strategies and compare them for the same budget in two test functions (2D and 6D). In each case, we compare the results when the derivatives are observed and not.

In conclusion, the results show that the efficiency of the strategies is linked to the regularity of the function. Indeed, if the function is easy, the derivative observations are not essential and the strategies like MyCL and MyqEI are relevant. Functions we find in the real life are often smooth. However, when the function is more complicated like the six-hump Camel a most exploratory strategy like MyAlea is recommended. This strategy is easy to use because the understanding of the complex criterion like EI or qEI is not necessary. The study we propose is a first work that reveal efficient strategies based on kriging prediction rather than EI approaches.

Appendices

A Number of point for the estimation of RC^1

Let $\mathbf{x} \in D \subset \mathbb{R}^p$, an observation point. Let $\mathbf{H} \sim \mathcal{N}(0_{\mathbb{R}^p}, \Delta^2)$ be the random variable such as $\mathbf{x} + \mathbf{H} \sim \mathcal{N}(\mathbf{x}, \Delta^2)$ where Δ^2 is defined by:

$$\Delta^2 = \begin{pmatrix} \delta_1^2 & 0 & \dots & 0 \\ 0 & \delta_2^2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \delta_p^2 \end{pmatrix}$$

Let

$$\begin{array}{rccc} f: & \mathbb{R}^p & \longrightarrow & [a;b] \\ & \mathbf{x} & \longmapsto & f(\mathbf{x}) \end{array}$$

be a 2 times differentiable bounded function, where $a \in \mathbb{R}$ and $b \in \mathbb{R}$. Then all the moments of $f(\mathbf{x} + \mathbf{H})$ exist. Let $\mu = \mathbb{E}(f(\mathbf{x} + \mathbf{H}))$ and $\bar{f} = \frac{1}{n}\sum_{i=1}^n f(\mathbf{x} + \mathbf{h}^i)$ be the empirical estimator of μ. Let $v_f = Var(f(\mathbf{x} + \mathbf{H}))$ and $S^2 = \frac{1}{n}\sum_{i=1}^n (f(\mathbf{x} + \mathbf{h}^i) - \bar{f}(\mathbf{x} + \mathbf{H}))^2$ the empirical estimator of σ^2. Where $\mathbf{h}^1, \dots, \mathbf{h}^n$ are n realizations of the random variable $\mathbf{X}$. The aim of this section is to find the asymptotic law of the empirical estimator S^2.

We recall that

$$\mathbb{E}(\bar{f}) = \mu$$
$$\mathbb{E}(S^2) = \frac{n-1}{n} v_f$$

In order to be as intelligible as possible $f(\mathbf{x}^i + \mathbf{h}^i)$ is written f_i.

We can notice that

$$S^2 = \frac{1}{n}\sum_{i=1}^{n}(f_i - \mu) - (\bar{f} - \mu)^2 \tag{12}$$

Consider the vector $(f_i - \mu)^2, 1 \leq i \leq n$. It is a vector of iid random variables. Moreover $\mathbb{E}\left[(f_i - \mu)^2\right] = v_f$, $Var\left[(f_i - \mu)^2\right] = \mathbb{E}\left[(f - \mu)^4\right] - v_f^2 = \mu_4 - v_f^2$. Then the central limit theorem (CLT) implies:

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}(f_i - \mu)^2 - v_f]\right) \to \mathcal{N}(0, \mu_4 - v_f^2)$$

With Equation (12), we obtain that:

$$\sqrt{n}(S^2 - v_f) = \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}(f_i - \mu) - v_f\right) - \sqrt{n}(\bar{f} - \mu)^2$$

The CLT applied to $\bar{f}$ gives:

$$\sqrt{n}\left(\bar{f} - \mu\right) \to \mathcal{N}(0, v_f)$$

Law of large number, $\bar{f} \to^{p.s} \mu \Rightarrow \bar{f} \to^{\mathbb{P}} \mu$ then $\sqrt{n}\left(\bar{f} - \mu\right) \to^{\mathbb{P}} 0$ and $\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}(f_i - \mu)^2 - v_f\right) \to \mathcal{N}(0, \mu_4 - v_f^2)$ then by the theorem of Slutski $\sqrt{n}\left(S^2 - v_f\right) \to \mathcal{N}(0, \mu_4 - v_f^2) - 0$.
We obtain:

$$\sqrt{(n)}(S^2 - v_f) \longrightarrow^{\mathcal{L}} \mathcal{N}(0, \mu_4 - v_f^2)$$

Asymptotically

$$\frac{S^2 - v_f}{\sqrt{\mu_4 - v_f^2/n}} \sim \mathcal{N}(0, 1)$$

Let z the quantile of the standard normal distribution of a risk α, then:

$$\mathbb{P}\left(\left|\frac{S^2 - v_f}{\sqrt{\mu_4 - v_f^2/n}}\right| \leq z\right) = 1 - \alpha$$

The empirical estimator $\hat{\mu}_4 = \frac{1}{n}\sum_{i=1}^{n}(f_i - \bar{f})^4$ of μ_4 is plugged in the equation:

$$\mathbb{P}\left(\left|\frac{S^2 - v_f}{\sqrt{\hat{\mu}_4 - v_f^4/n}}\right| \leq z\right) = 1 - \alpha$$

$$\begin{aligned}\left|\frac{S^2 - v_f}{\sqrt{\hat{\mu}_4 - v_f^2/n}}\right| \leq z &\Leftrightarrow \frac{(S^2 - v_f)^2}{\hat{\mu}_4 - v_f^2/n} \leq z \\ &\Leftrightarrow ((S^2)^2 - 2S^2 v_f + v_f^2) - \frac{z}{n}(\hat{\mu}_4 - v_f^2) \leq 0 \\ &\Leftrightarrow \left(1 + \frac{z^2}{n}\right) v_f^2 - 2S^2 v_f + \left((S^2)^2 - \frac{z^2\hat{\mu}_4}{n}\right) \leq 0\end{aligned}$$

$$\begin{aligned}\Delta &= (-2S^2)^2 - 4\left(1 + \frac{z^2}{n}\right)\left(S^4 - \frac{z^2\hat{\mu}_4}{n}\right) \\ &= 4(S^2)^2 - 4(S^2)^2 + \frac{4\hat{\mu}_4 z^2}{n} - \frac{4z^2(S^2)^2}{n} + \frac{4z^4\hat{\mu}_4}{n^2} \\ &= \frac{4z^2}{n}\left(\hat{\mu}_4\left(1 + \frac{z^2}{n}\right) - (S^2)^2\right)\end{aligned}$$

$\Delta > 0$ if $\hat{\mu}_4\left(1 + \frac{z^2}{n}\right) > (S^2)^2$.
The square function is convex, $\left((f_1 - \bar{f})^2, \ldots, (f_n - \bar{f})^2\right)$ is a real n-uplet and $\sum_{i=1}^{n}\frac{1}{n} = 1$. Thanks to the Jensen Inequality (convexity):

$$\frac{1}{n}\sum_{i=1}^{n}(f_i - \bar{f})^4 \geq \left(\frac{1}{n}\sum_{i=1}^{n}(f_i - \bar{f})^2\right)^2 \Leftrightarrow \hat{\mu}_4 \geq (S^2)^2$$

Then $\Delta > 0$ and

$$v_f \in \left[\frac{2S^2 - \sqrt{\Delta}}{2\left(1 + \frac{z^2}{n}\right)}; \frac{2S^2 + \sqrt{\Delta}}{2\left(1 + \frac{z^2}{n}\right)}\right]$$

$$\begin{aligned}\frac{2S^2 - \sqrt{\Delta}}{2\left(1 + \frac{z^2}{n}\right)} &= \frac{2S^2 - \sqrt{\frac{4z^2}{n}\left(\hat{\mu}_4\left(1 + \frac{z^2}{n}\right) - (S^2)^2\right)}}{2\left(1 + \frac{z^2}{n}\right)} \\ &= S^2 - \frac{z}{\sqrt{n}}\sqrt{\hat{\mu}_4 - (S^2)^2} + o\left(\frac{1}{n}\right)\end{aligned}$$

$$\frac{2S^2 + \sqrt{\Delta}}{2\left(1 + \frac{z^2}{n}\right)} = S^2 + \frac{z}{\sqrt{n}}\sqrt{\hat{\mu}_4 - (S^2)^2} + o\left(\frac{1}{n}\right)$$

Then approximatively,

$$v_f \in \left[S^2 - \frac{z}{\sqrt{n}}\sqrt{\hat{\mu}_4 - (S^2)^2}; S^2 + \frac{z}{\sqrt{n}}\sqrt{\hat{\mu}_4 - (S^2)^2} \right]$$

In order to obtain an approximation error lower or equal to e_n, we choose:

$$n > \frac{z^2}{e_n^2}(\hat{\mu}_4 - (S^2)^2)$$

B Taylor

Proposition 1. *Let $\mathbf{H} \sim \mathcal{N}(0_{\mathbb{R}^d}, \Delta^2)$ where $\Delta^2 = diag\{\delta_1, \ldots, \delta_p\} \in \mathcal{M}_{p\times p}$ and a function $f : D \subset \mathbb{R}^p \to \mathbb{R}$ to times differentiable. $\tilde{f}$ is the Taylor approximation trunked at the order two then:*

$$Var\left(\tilde{f}(x+H)\right) = tr\left(\nabla_f \nabla_f' \Delta^2\right) + \frac{1}{2} tr\left(\mathbb{H}_f^2(\delta_1^2, \ldots, \delta_p^2)(\delta_1^2, \ldots, \delta_p^2)'\right)$$

where $\nabla_f \in \mathbb{R}^p$ the vector of the gradient of f and $\mathbb{H}_f \in \mathcal{M}_{p,p}$ the Hessian matrix and tr the matrix trace.

Proof. The Taylor approximation trunked at the order two is:

$$\tilde{f}(\mathbf{x}+H) = f(\mathbf{x}) + \nabla_f'(\mathbf{x})H + \frac{1}{2}H'\mathbb{H}_f(\mathbf{x})H$$

and

$$\begin{aligned}
Var\left(\tilde{f}(\mathbf{x}+\mathbf{H})\right) &= Var\left(f(\mathbf{x}) + \nabla_f'\mathbf{H} + \frac{1}{2}\mathbf{H}'\mathbb{H}_f\mathbf{H}\right) \\
&= Var\left({}^t\nabla_f\mathbf{H} + \frac{1}{2}\,{}^t\mathbf{H}\mathbb{H}_f\mathbf{H}\right) \\
&= \mathbb{E}\left(\left({}^t\nabla_f\mathbf{H} + \frac{1}{2}\,{}^t\mathbf{H}\mathbb{H}_f\mathbf{H}\right)^2\right) - \mathbb{E}\left({}^t\nabla_f\mathbf{H} + \frac{1}{2}\,{}^t\mathbf{H}\mathbb{H}_f\mathbf{H}\right)^2 \\
&= \mathbb{E}\left(({}^t\nabla_f\mathbf{H})^2\right) + \mathbb{E}\left({}^t\nabla_f\mathbf{H}\,{}^t\mathbf{H}\mathbb{H}_f\mathbf{H}\right) + \frac{1}{4}\mathbb{E}\left(({}^t\mathbf{H}\mathbb{H}_f\mathbf{H})^2\right) \\
&\quad - \mathbb{E}\left({}^t\nabla_f\mathbf{H}\right)^2 - \mathbb{E}\left({}^t\mathbf{H}\mathbb{H}_f\mathbf{H}\right)\mathbb{E}\left({}^t\nabla_f\mathbf{H}\right) - \frac{1}{4}\mathbb{E}\left({}^t\mathbf{H}\mathbb{H}_f\mathbf{H}\right)^2
\end{aligned}$$

Let's calculate each terms

1. $\mathbb{E}\left[\nabla_f'\mathbf{H}\right] = \sum_i (\nabla_f)_i \mathbb{E}\left[\mathbf{h}_i\right] = 0$
2. $\mathbb{E}\left[\mathbf{H}'\mathbb{H}_f\mathbf{H}\right] = \sum_i \sum_j (\mathbb{H}_f)_{i,j}\mathbb{E}\left[\mathbf{h}_i\mathbf{h}_j\right] = \sum_i (\mathbb{H}_f)_{i,i}\delta_i^2 = tr\left(\mathbb{H}_f\Delta^2\right)$
3. $\mathbb{E}\left[\left(\nabla_f'H\right)^2\right] = \sum_i \sum_j (\nabla_f)_i(\nabla_f)_j\mathbb{E}\left[\mathbf{h}_i\mathbf{h}_j\right] = \sum_i (\nabla_f)_i(\nabla_f)_i\delta_i^2 = tr\left(\nabla_f\nabla_f'\Delta^2\right)$
4. $\mathbb{E}\left[\left((\nabla_f'H)'H'\mathbb{H}_f H\right)^2\right] = \sum_i \sum_j \sum_k (\nabla_f)_i(\mathbb{H}_f)_{jk}\mathbb{E}\left[\mathbf{h}_i\mathbf{h}_j\mathbf{h}_k\right] = \sum_i (\nabla_f)_i(\mathbb{H}_f)_{ii}\mathbb{E}\left[\mathbf{h}_i^3\right] = 0$

5.

$$\begin{aligned}
\mathbb{E}\left[(H'\mathbb{H}_f H)^2\right] &= \sum_i \sum_j \sum_k \sum_l (\mathbb{H}_f)_{ij}(\mathbb{H}_f)_{kl}\mathbb{E}\left[\mathbf{h}_i\mathbf{h}_j\mathbf{h}_k\mathbf{h}_l\right] \\
&= \sum_i \sum_{k\neq i} (\mathbb{H}_f)_{ii}(\mathbb{H}_f)_{kk}\mathbb{E}\left[\mathbf{h}_i^2\mathbf{h}_k^2\right] + 2\sum_i \sum_{j\neq i} (\mathbb{H}_f)_{ij}(\mathbb{H}_f)_{ij}\mathbb{E}\left[\mathbf{h}_i^2\mathbf{h}_j^2\right] + \sum_i (\mathbb{H}_f)_{ii}^2\mathbb{E}\left[\mathbf{h}_i^4\right] \\
&= \sum_i \sum_{k\neq i} (\mathbb{H}_f)_{ii}(\mathbb{H}_f)_{kk}\delta_i^2\delta_k^2 + 2\sum_i \sum_{j\neq i} (\mathbb{H}_f)_{ij}(\mathbb{H}_f)_{ij}\delta_i^2\delta_j^2 + 3\sum_i (\mathbb{H}_f)_{ii}^2\delta_i^2 \\
&= \sum_i \sum_{k} (\mathbb{H}_f)_{ii}(\mathbb{H}_f)_{kk}\delta_i^2\delta_k^2 - \sum_i (\mathbb{H}_f)_{ii}^2\delta_i^2 + 2\sum_i \sum_{j} (\mathbb{H}_f)_{ij}(\mathbb{H}_f)_{ij}\delta_i^2\delta_j^2 \\
&- 2\sum_i (\mathbb{H}_f)_{ii}^2\delta_i^2 + 3\sum_i (\mathbb{H}_f)_{ii}^2\delta_i^2 \\
&= tr\left(\mathbb{H}_f\Delta^2\right)^2 + 2tr\left(\mathbb{H}_f^{\cdot 2}(\delta_1^2,\ldots,\delta_p^2)(\delta_1^2,\ldots,\delta_p^2)'\right)
\end{aligned}$$

Finally

$$\begin{aligned}
Var\left(\tilde{f}(\mathbf{x}+H)\right) &= tr\left(\nabla_f\nabla_f'\Delta^2\right) + 0 + \frac{1}{4}tr\left(\mathbb{H}_f\Delta^2\right)^2 - 0 - tr\left(\mathbb{H}_f\Delta^2\right)\times 0 - \frac{1}{4}tr\left(\mathbb{H}_f\Delta^2\right)^2 \\
&- \frac{2}{4}tr\left(\mathbb{H}_f^{\cdot 2}(\delta_1^2,\ldots,\delta_p^2)(\delta_1^2,\ldots,\delta_p^2)'\right) \\
&= tr\left(\nabla_f\nabla_f'\Delta^2\right) + \frac{1}{2}tr\left(\mathbb{H}_f^{\cdot 2}(\delta_1^2,\ldots,\delta_p^2)(\delta_1^2,\ldots,\delta_p^2)'\right)
\end{aligned}$$

□

References

[Apley et al., 2006] Apley, D. W., Liu, J., and Chen, W. (2006). Understanding the effects of model uncertainty in robust design with computer experiments. *Journal of Mechanical Design*, 128(4):945–958.

[Beyer and Sendhoff, 2007] Beyer, H.-G. and Sendhoff, B. (2007). Robust optimization a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33):3190 – 3218.

[Coco et al., 2014] Coco, A. A., Solano-Charris, E. L., Santos, A. C., Prins, C., and de Noronha, T. F. (2014). Robust optimization criteria: state-of-the-art and new issues. *Technical Report UTT-LOSI-14001, ISSN: 2266-5064*.

[Darlington et al., 1999] Darlington, J., Pantelides, C., Rustem, B., and Tanyi, B. (1999). An algorithm for constrained nonlinear optimization under uncertainty. *Automatica*, 35(2):217 – 228.

[Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

[Emmerich et al., 2011] Emmerich, M. T., Deutz, A. H., and Klinkenberg, J. W. (2011). Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2147–2154. IEEE.

[Fonseca et al., 2006] Fonseca, C. M., Paquete, L., and López-Ibáñez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, pages 1157–1163. IEEE Press, Piscataway, NJ.

[Gabrel et al., 2014] Gabrel, V., Murat, C., and Thiele, A. (2014). Recent advances in robust optimization: An overview. *European journal of operational research*, 235(3):471–483.

[Ginsbourger et al., 2010] Ginsbourger, D., Le Riche, R., and Carraro, L. (2010). Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Springer.

[Göhler et al., 2016] Göhler, S. M., Eifler, T., and Howard, T. J. (2016). Robustness metrics: Consolidating the multiple approaches to quantify robustness. *Journal of Mechanical Design*, 138(11):111407.

[Hartigan and Wong, 1979] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

[Henkenjohann and Kunert, 2007] Henkenjohann, N. and Kunert, J. (2007). An efficient sequential optimization approach based on the multivariate expected improvement criterion. *Quality Engineering*, 19(4):267–280.

[Janusevskis and Le Riche, 2013] Janusevskis, J. and Le Riche, R. (2013). Simultaneous kriging-based estimation and optimization of mean response. *Journal of Global Optimization*, 55(2):313–336.

[Jeong and Obayashi, 2005] Jeong, S. and Obayashi, S. (2005). Efficient global optimization (ego) for multi-objective problem and data mining. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2138–2145. IEEE.

[Jones et al., 1998] Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492.

[Knowles, 2006] Knowles, J. (2006). Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.

[Le Gratiet, 2013] Le Gratiet, L. (2013). *Multi-fidelity Gaussian process regression for computer experiments*. PhD thesis, Université Paris-Diderot-Paris VII.

[Lelièvre et al., 2016] Lelièvre, N., Beaurepaire, P., Mattrand, C., Gayton, N., and Otsmane, A. (2016). On the consideration of uncertainty in design: optimization - reliability - robustness. *Structural and Multidisciplinary Optimization*, 54(6):1423–1437.

[Liu et al., 2007] Liu, W., Zhang, Q., Tsang, E., Liu, C., and Virginas, B. (2007). On the performance of metamodel assisted moea/d. In *International Symposium on Intelligence Computation and Applications*, pages 547–557. Springer.

[Marzat et al., 2013] Marzat, J., Walter, E., and Piet-Lahanier, H. (2013). Worst-case global optimization of black-box functions through kriging and relaxation. *Journal of Global Optimization*, 55(4):707–727.

[Picheny, 2015] Picheny, V. (2015). Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280.

[Ponweiser et al., 2008] Ponweiser, W., Wagner, T., Biermann, D., and Vincze, M. (2008). Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection. In *International Conference on Parallel Problem Solving from Nature*, pages 784–794. Springer.

[Pronzato and Éric Thierry, 2003] Pronzato, L. and Éric Thierry (2003). Robust design with nonparametric models: prediction of second-order characteristics of process variability by kriging1. *IFAC Proceedings Volumes*, 36(16):537 – 542. 13th IFAC Symposium on System Identification (SYSID 2003), Rotterdam, The Netherlands, 27-29 August, 2003.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA.

[Santner et al., 2003] Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The design and analysis of computer experiments*. Springer Series in Statistics. Springer-Verlag, New York.

[Schott, 1995] Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH.

[Stein, 1999] Stein, M. L. (1999). *Interpolation of spatial data*. Springer Series in Statistics. Springer-Verlag, New York. Some theory for Kriging.

[Svenson and Santner, 2016] Svenson, J. and Santner, T. (2016). Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics & Data Analysis*, 94:250–264.

[Troian et al., 2016] Troian, R., Shimoyama, K., Gillot, F., and Besset, S. (2016). Methodology for the design of the geometry of a cavity and its absorption coefficients as random design variables under vibroacoustic criteria. *Journal of Computational Acoustics*, 24(02):1650006.

[ur Rehman et al., 2014] ur Rehman, S., Langelaar, M., and van Keulen, F. (2014). Efficient kriging-based robust optimization of unconstrained problems. *Journal of Computational Science*, 5(6):872–881.

[Van Veldhuizen, 1999] Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Wright Patterson AFB, OH, USA. AAI9928483.

[Wagner et al., 2010] Wagner, T., Emmerich, M., Deutz, A., and Ponweiser, W. (2010). On expected-improvement criteria for model-based multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 718–727. Springer.

[Zhang et al., 2010] Zhang, Q., Liu, W., Tsang, E., and Virginas, B. (2010). Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474.

[Zitzler and Thiele, 1999] Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.