Robust optimization: a kriging-based multi-objective optimization approach

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

Robust optimization: a kriging-based multi-objective optimization approach

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 Collongue, F-69134 Ecully Cedex, France
[2]Univ Lyon, Ecole Centrale de Lyon, LTDS, CNRS UMR 5513, 36 avenue Guy de Collongue, F-69134 Ecully Cedex, France
[3]INRIA Rennes, I4S Team

September 10, 2020

Abstract

In the robust shape optimization context, the evaluation cost of numerical models is reduced by the use of a response surface. Multi-objective methodologies for robust optimization that consist in simultaneously minimizing the function and a robustness criterion (the second moment) have already been developed. However, efficient estimation of the robustness criterion in the framework of time-consuming simulation has not been greatly explored. A robust optimization procedure based on the prediction of the function and its derivatives by kriging is proposed. The second moment is replaced by an approximated version using Taylor expansion. A Pareto front is generated by a genetic algorithm named NSGA-II with a reasonable time of calculation.
Seven relevant strategies are detailed and compared with the same calculation time in two test functions (2D and 6D). In each case, we compare the results when the derivatives are observed and when they are not. The procedure is also applied to an industrial case study where the objective is to optimize the shape of a motor fan.

Keywords. Robust Optimization, Gaussian process modeling, Multi-objective optimization, Taylor expansion, Expected Improvement.

1 Introduction

Complex physical phenomena are increasingly studied through numerical simulations. These numerical models are able to mimic real experiments with a high degree of accuracy. They predict the physical measures of interest (outputs) very precisely, but are extremely costly to calculate. One main use of these simulations is to solve optimization problems. This work focuses on cases where the optimized

*celine.helbert@ec-lyon.fr

solution is sensitive to input perturbations. For example, these perturbations are due to random fluctuations during production. A robust solution is then sought. To solve the robust optimization problem, one way is to introduce a multi-objective optimization formulation where the first objective is the function itself and the second is a robustness criterion. These two objectives are often antagonistic. The issue of robust optimization is then to find a Pareto front that strikes the right balance between the optimization of the function and the impact of input perturbations (uncertainties). As the simulations provided by the numerical code are often time-consuming, only a few of them are then affordable. So, the computer code cannot be intensively exploited to provide the robust optimum. In this case, the optimization procedure is often run on a kriging model (see e.g. [1]) 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. The authors in [2] 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.

The literature contains a sample of works that handle robust optimization. Methodologies depend on the kind of uncertainties. The authors in [3] propose two classes of uncertainties: uncertainties that "are primitively linked to the environment and condition of use" and uncertainties that "are connected with the production/manufacturing process". In the first type of uncertainties, the aim is to find $\mathbf{x}$ such that $f(\mathbf{x}, \mathbf{U})$ is minimal where $\mathbf{U}$ a random vector (cf [4], [5], [6] and [7]). The authors in [4] propose to minimize the expectation of $f(\mathbf{x}, \mathbf{U})$ with a Gaussian process-based methodology. The authors in [5] propose an algorithm that minimizes the worst-case. In [7] a mono-objective solution based on the worst-case on the response surface is proposed. In all these sequential methods, the variables are clearly separated into two classes (design and uncertain) and the robust criterion is summed up either by the expectation or the worst-case.

In our context, the aim is to optimize the function f taking into account manufacturing tolerances without stochastic modeling. We introduce a multi-objective strategy to detect the whole set of robust solutions. The first objective is the function itself (not the mean nor the worst-case), while the second objective is a robustness criterion which needs to be described.

The quantification of the robustness is challenging. [8], [9] and [10] give some overviews of different robustness criteria. Our industrial partners quantify the variability of a solution by the local curvature of the output in a neighborhood of the solution (see e.g. [6] and [11]). In this paper, a criterion based on the first and second derivatives is proposed. In the particular case of fluctuations modeled by a centered Gaussian random vector, it can be interpreted as a local variance of the Taylor approximation of the function f, as proposed by [12]. In the context of time-consuming simulations, this criterion is predicted by kriging. Kriging is well adapted, since it can exploit the covariance structure between the GP model of the function and all the derivatives. This structure is described in [13] and used again by [14].

Then, the function and its robustness criterion are accessible through kriging. A multi-objective optimization is performed to provide solutions. In the literarure, several approaches (see [15] for an overview) mixing a GP modeling and multi-objective optimization are proposed: the aggregation methods (see [16], [17] and [18]), the hypervolume methods (see [19], [20] and [21]), the maximin method (see [22]) and the uncertainty reduction method (see [23]). [24] shows that the aggregation methods are

not efficient with a complex Pareto front. The hypervolume, maximin and uncertainty reduction algorithms need to need to preform multi-objective optimization on Gaussian processes. As the developed robustness criterion is no longer Gaussian, it could be costly to adapt these methods in our case. Some optimization procedures inspired by [25] are proposed. These procedures consist in applying an evolutionary algorithm on the kriging predictions and in taking into account kriging variance as suggested by [26].

The paper is structured as follows. Our robustness kriging-based criterion is introduced in section 2. In section 3, the context of a Gaussian process metamodeling is introduced. The general multi-objective optimization scheme is presented in section 4, and the different enrichment strategies in section 5. The quality criteria for comparing Pareto fronts are given in section 6. Finally, in section 7, the behavior of our methodology is studied on two toy functions and on an industrial test case.

2 Robustness criterion

Mass production involves manufacturing operations generating uncertainties on part properties, such as geometrical dimensions, material properties and so on. Part design accepts such uncertainties within a specified range, provided as tolerances, for the whole system to work when the considered part is integrated. Robust optimization needs the construction of a criterion which quantifies the local sensitivity to variabilities. One way is to minimize a local curvature which is linked to first and second derivatives. Let f be the objective function (a two-times differentiable function)

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

where p is the number of input variables, i.e. $x = (x_1, \ldots, x_p)$. Each variable j is assumed to vary in the interval $x_j \pm 2\delta_j$. We propose the following robustness criterion to be minimized

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

where ∇_f is the gradient of f, $\mathbb{H}_f$ the Hessian matrix of f, tr is the matrix trace and Δ is defined by:

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

This criterion is composed of two terms. The first part involves the gradient of f (first derivatives) and the second the Hessian matrix (second derivatives). Minimizing this criterion implies causing the gradient and the Hessian to vanish. This leads to flat local extrema. The associated designs are insensitive to production fluctuations. This criterion does not allow discrimination between maxima and minima or between two maxima. This is why we perform a multi-objective optimization on f and RC_f.

Remarks:

- If the output of a simulation provides the results of the function and the first derivatives, the RC_f

criterion can be computed with only one call to the computer code. However, in the context of costly simulations, a robust optimization cannot be directly performed on f and RC_f.

- In the particular case of fluctuations modeled by a centered Gaussian random vector $\mathbf{H}$, $RC_f(\mathbf{x})$ is equal to $Var\left(\widetilde{f}(\mathbf{x}+\mathbf{H})\right)$ where $\widetilde{f}(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + \nabla_f(\mathbf{x})\cdot\mathbf{h} + \frac{1}{2}\mathbf{h}'\mathbb{H}_f(\mathbf{x})\mathbf{h}$.

The next section presents how these quantities can be predicted using a kriging approach.

3 Gaussian process modeling for the function and its derivatives

As can be seen in Equation (2), the robustness criterion depends on the first and second derivatives of f. A Gaussian process metamodel (see [14]) is well suited to this context in the sense that all derivatives can easily be predicted. In this section, the model and the predictions are presented and illustrated on a toy example.

3.1 Kriging Model

Let us assume function f to be a realization of a Gaussian process $(Y(\mathbf{x}))_{\mathbf{x}\in D}$ with a constant mean, μ, and with a stationary covariance function $k(\mathbf{x},\tilde{\mathbf{x}}) = \sigma^2 r_{\boldsymbol{\theta}}(\mathbf{x}-\tilde{\mathbf{x}})$, $\forall(\mathbf{x},\tilde{\mathbf{x}}) \in D\times D$. This process is assumed to be two-times differentiable in mean square at point $(\mathbf{x},\tilde{\mathbf{x}})$.
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 first-order partial derivative of $(Y(\mathbf{x}))_{\mathbf{x}\in D}$ with respect to x_i, and by $\left(Y_{x_i,x_j}(\mathbf{x})\right)_{\mathbf{x}\in D} = \left(\frac{\partial^2 Y}{\partial x_i \partial x_j}(\mathbf{x})\right)_{\mathbf{x}\in D}$ the second-order partial derivative of $(Y(\mathbf{x}))_{\mathbf{x}\in D}$ with respect to x_i and x_j.

All the covariance structures between the process and its derivatives are then well-known and are given by:

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

Let $(x^1,\ldots,x^n)$ be the initial design of experiments, where $x^k \in D, 1\leq k\leq n$. The evaluation of the function (resp. first and second derivatives) at point $\mathbf{x}^k$ is denoted by $y^k \in \mathbb{R}$ (resp. $y^k_{x_i} \in \mathbb{R}$ and $y^k_{x_i,x_j} \in \mathbb{R}$), where $i \in \{1,\ldots,p\}$, $j \in \{i,\ldots,p\}$ and $k \in \{1,\ldots,n\}$. The collection of outputs $\mathbf{y}$, $\mathbf{y}_{x_i}$ and $\mathbf{y}_{x_i,x_j}$ is such that:

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

$(y^k, y^k_{x_1},\ldots,y^k_{x_p}, y^k_{x_1,x_1},\ldots,y^k_{x_i,x_j},\ldots,y^k_{x_p,x_p})$, $k\in\{1,\ldots,n\}$ is then a realization of the following

$d = 1 + \frac{3p}{2} + \frac{p^2}{2}$ dimensional GP:

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

at points $\mathbf{x}^1, \ldots, \mathbf{x}^n$.

3.2 Kriging predictions

The problem is to predict Z considering observations at points $\mathbf{x}^1, \ldots, \mathbf{x}^n$. However, the entire vector Z is not always observable. Let $u_{obs} \subset \{1, \ldots, d\}$ be the components that are observable. For example, only the function and its first derivatives can be affordable. Likewise, it is not always necessary to predict the whole vector Z. Let $u_{pred} \subset \{1, \ldots, d\}$ be the components that need to be predicted.
In the following we assume that $1 \in u_{obs}$ and we denote $f_{obs} = (1, 0_{\mathbb{R}^{d_{obs}-1}}, \ldots, 1, 0_{\mathbb{R}^{d_{obs}-1}})' \in \mathbb{R}^{nd_{obs}}$, $d_{obs} = \#u_{obs}$ and $f_{pred} = (1, 0_{\mathbb{R}^{d_{pred}-1}})' \in \mathbb{R}^{d_{pred}}$, $d_{pred} = \#u_{pred}$. The kriging mean is then given by the following equation:

$$\widehat{\mathbf{z}}_{u_{pred}}(\mathbf{x}) = \widehat{\mu} f_{pred} + \mathbf{c}_{\boldsymbol{\theta}}(\mathbf{x})' \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} (\mathbf{z}_{u_{obs}} - \widehat{\mu} f_{obs}),\ \ \widehat{\mathbf{z}}_{u_{pred}}(\mathbf{x}) \in \mathbb{R}^{d_{pred}} \tag{3}$$

where $\mathbf{z}_{u_{obs}} = \begin{pmatrix} z^1_{obs} \\ \vdots \\ z^n_{obs} \end{pmatrix}$ the observation vector. $\widehat{\mathbf{z}}_{u_{pred}}(\mathbf{x})$ is the prediction vector and

$$\widehat{\mu} = (f'_{obs} \Sigma_\theta^{-1} f_{obs})^{-1} f'_{obs} \Sigma_\theta^{-1} \mathbf{z}_{u_{obs}}.$$

The mean square error (MSE) at point $\mathbf{x} \in D$ is given by:

$$\widehat{\mathbf{s}}^2_{u_{pred}}(\mathbf{x}) = \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) - \begin{pmatrix} f_{pred} & \boldsymbol{c}_{\boldsymbol{\theta}}(\mathbf{x}) \end{pmatrix} \begin{pmatrix} 0 & f'_{obs} \\ f_{obs} & \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \end{pmatrix}^{-1} \begin{pmatrix} f'_{pred} \\ \boldsymbol{c}_{\boldsymbol{\theta}}(\mathbf{x}) \end{pmatrix},\ \ \widehat{\mathbf{s}}^2_{u_{pred}}(\mathbf{x}) \in \mathcal{M}_{d_{pred} \times d_{pred}}$$

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

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

and

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

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

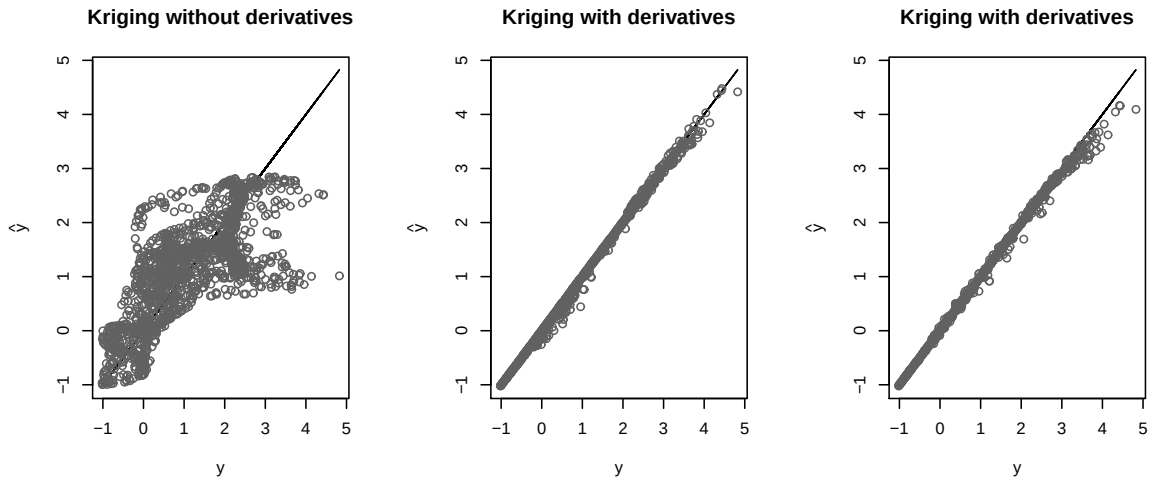


Figure 1: Prediction plots for the six-hump Camel function: 10 points without observation of the derivatives (on the left), 10 points with 5 derivatives (in the middle) and 60 points without observation of the derivatives (on the right).

3.3 Illustration with the six-hump Camel function

In this section, different kriging-based response surfaces conditioning or not on derivatives are compared. The chosen toy function is the six-hump Camel function defined by:

$$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1x_2 + \left(-4 + 4x_2^2\right) x_2^2, \mathbf{x} \in [-2;2] \times [-1;1]$$

The kriging covariance kernel is a tensor product one:

$$cov\left(Y(\mathbf{x}), Y(\tilde{\mathbf{x}})\right) = k(\mathbf{x} - \tilde{\mathbf{x}}) = \sigma^2 \prod_{j=1}^{p} \rho_{\theta_j}\left(|x_j - x'_j|\right), \boldsymbol{\theta} = (\theta_1, \ldots, \theta_p) \in \mathbb{R}_+^p \tag{4}$$

where ρ_{θ_j} is a correlation function which only depends on the one dimensional range parameter θ_j, see e.g. [1] and [27]. A Matern 5/2 kernel is used because the output is assumed to be two-times continuously differentiable:

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

Kriging predictive quality has been compared in different learning situations:

- 10 learning points where f is observed (left part of Figure 1)
- 10 learning points where f and all the derivatives are observed (middle part of Figure 1)
- 60 learning points where f is observed (right part of Figure 1)

The learning sets composed of 10 or 60 points are maximin latin hypercube samplings. The test set is a latin hypercube sampling of 1500 points. As expected, the left and middle parts of Figure 1 show that kriging with derivatives performs much better than without. While computing one derivative costs as

much as computing a new point, the right part of Figure 1 shows that kriging without derivatives does better. However in industrial applications, computing derivatives is often more affordable.

4 Robust optimization procedure

In this section, the robust optimization procedure that uses our criterion (see Equation (6)) is presented. The robust optimization problem is written as:

$$\text{Find the Pareto set } \mathbb{X}_0, \text{ the solution of the following multi-objective optimization}$$
$$\min_{\mathbf{x}\in\mathbb{R}^p} \{f(\mathbf{x}), RC_f(\mathbf{x})\} \tag{5}$$

The approach to solve it in the context of time-consuming simulations is based on a classical black-box optimization scheme (see [2]). The optimization scheme (see Figure 2) is based on the following steps:

- **Initialization.** The costly function and possibly its derivatives are evaluated on a well-chosen initial design of experiments. A kriging model is adjusted on this first set of outputs. Two response surfaces $\{\hat{obj}_f(x)\}$ and $\{\hat{obj}_{RC_f}(x)\}$ related to the two objectives $\{f(x)\}$ and $\{RC_f(x)\}$ are predicted.
  Remarks: in the different case studies, the chosen initial design is a maximin Latin Hypercube Sampling (maximin HLS) (see [28]).

- **Loop until the budget is reached**

  1. **Multi-objective optimization.** A multi-objective global optimization method is applied to solve $\min_{\mathbf{x}\in\mathbb{R}^p}\{\hat{obj}_f(x), \hat{obj}_{RC_f}(x)\}$. A Pareto front is identified.
     Remarks: The NSGA II algorithm is chosen for its good performances in finding complex Pareto fronts.
  2. **Enrichment.** A set of q points is selected from the Pareto front. The function and possibly its derivatives are evaluated on these new points. The Gaussian process model and the two response surfaces are updated.

The aim of this section is to define the two response surfaces to be optimized. The next section focuses on different strategies for selecting good points from the Pareto front.

Three different response surfaces have been studied to run the multi-objective methodology. The first approach consists in optimizing the predicted version of the function and the robustness criterion. This approach, quite crude, is denoted by the "plug in" approach in the following and is described below. The second aoproach is based on the famous Expected Improvement quantity in order to take into account prediction uncertainty. The third approach is the most complex: it optimizes the multipoint Expected Improvement versions of $\{f(x)\}$ and $\{RC_f(x)\}$.

4.1 The "plug in" response surfaces

We remind you that $\widehat{z}(\mathbf{x})$ from Equation (3) is

$$\widehat{z}(\mathbf{x}) = \left(\widehat{y}(\mathbf{x}), \ldots, \widehat{y}_{x_p,x_p}(\mathbf{x})\right)'$$

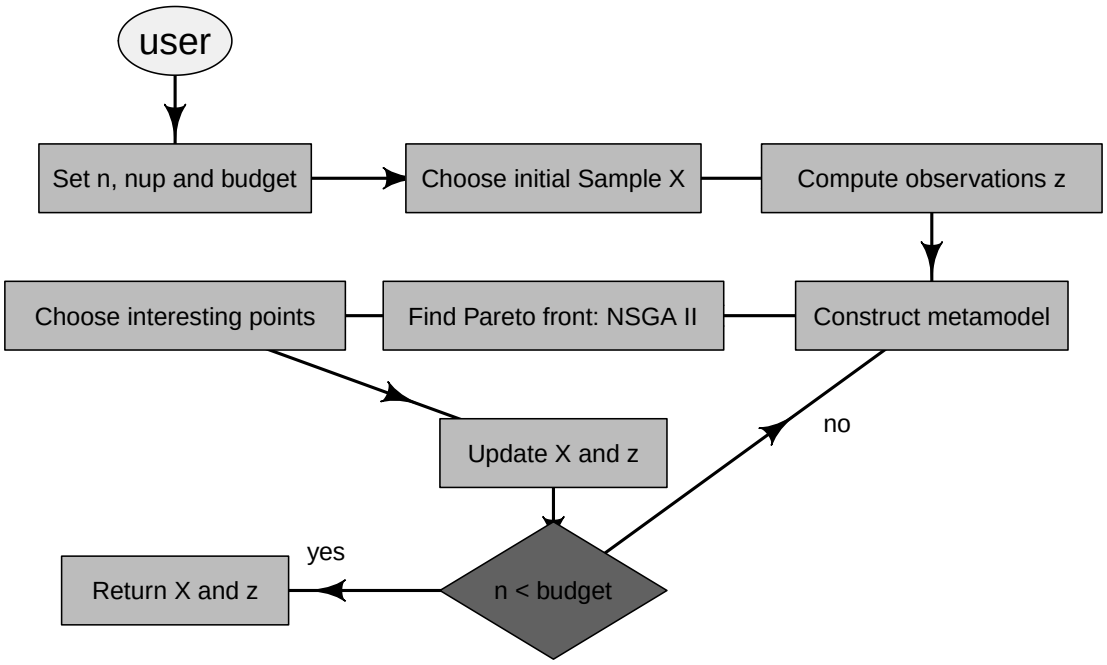


Figure 2: The robust optimization procedure.

The prediction of the true function f is given by the first coordinate of the vector $\widehat{z}(\mathbf{x})$.
The prediction of $RC_f(\mathbf{x})$ is defined by:

$$RC_{\widehat{y}}(\mathbf{x}) = tr\left(\nabla_{\widehat{y}}(\mathbf{x})\nabla_{\widehat{y}}(\mathbf{x})'\Delta^2\right) + \frac{1}{2}tr\left(\mathbb{H}^2_{\widehat{y}}(\mathbf{x})(\delta_1^2,\ldots,\delta_p^2)'(\delta_1^2,\ldots,\delta_p^2)\right) \tag{6}$$

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 $\begin{pmatrix}\widehat{y}_{x_1,x_1} & \cdots & \widehat{y}_{x_1,x_p}\\ \vdots & \ddots & \vdots \\ \widehat{y}_{x_p,x_1} & \cdots & \widehat{y}_{x_p,x_p}\end{pmatrix}$ and corresponds to the prediction of the Hessian matrix. $\nabla_{\widehat{y}}$ and $\mathbb{H}_{\widehat{y}}$ are obtained from different components of $\widehat{z}(x)$.

The **"plug in" formulation** is then:

Find the Pareto set $\mathbb{X}_0$, the solution of the following multi-objective optimization

$$\min_{\mathbf{x}\in\mathbb{R}^p}\{\widehat{y}(\mathbf{x}), RC_{\widehat{y}}(\mathbf{x})\} \tag{7}$$

Remarks:

- The definition of the predicted robustness criterion corresponds to the definition of Equation (2) where the derivatives have been replaced by their prediction.
- These response surfaces are easy to compute. While NSGA II runs quickly on these quantities, prediction uncertainty is not taken into account at this stage.

4.2 The "expected improvement" response surfaces

Unlike the previous case, in this approach we take into account the kriging variance in the optimization scheme. The best way to do this is to optimize the expected improvement.

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 (see [2]) 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:

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

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

In our case, these formulas have to be adapted:

i) to the robustness criterion,

ii) to a larger set of observations that may include derivatives,

iii) to a multi-objective optimization context.

To answer to i, we need to define the process $(RC_Y(\mathbf{x}))_{\mathbf{x}\in D}$. From Equation 2, the process is naturally defined by:

$$RC_Y(\mathbf{x}) = tr\left(\nabla_Y(\mathbf{x})\nabla_Y(\mathbf{x})'\Delta^2\right) + \frac{1}{2}tr\left(\mathbb{H}_Y^2(\mathbf{x})(\delta_1^2, \ldots, \delta_p^2)'(\delta_1^2, \ldots, \delta_p^2)\right) \tag{8}$$

where ∇_Y is the vector $\begin{pmatrix} Y_{x_1} \\ \vdots \\ Y_{x_p} \end{pmatrix}$ and $\mathbb{H}_Y$ is the matrix $\begin{pmatrix} Y_{x_1,x_1} & \ldots & Y_{x_1,x_p} \\ \vdots & \ddots & \vdots \\ Y_{x_p,x_1} & \ldots & Y_{x_p,x_p} \end{pmatrix}$.

To answer to point ii, conditional expectations are considered over observations of vector z that includes derivatives when they are available.

Finally to answer to iii, the authors in [25] show that, in the context of multi-objective optimization, the usual reference value, which is the current observed minimum, is too constraining. To continue to allow improvement, this reference value is rather taken as the worst value on the current Pareto front. The expressions of EI for f and RC_f are then as follows:

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

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

The **"expected improvement" formulation** is then:

Find the Pareto set $\mathbb{X}_0$, the solution of the following multi-objective optimization

$$\min_{\mathbf{x}\in\mathbb{R}^p}\{EI_y(\mathbf{x}), EI_{RC_y}(\mathbf{x})\} \tag{9}$$

Remarks:

- 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 solutions $\mathbb{X}$, the solutions of 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 $\max(RC_y(\mathbb{X}^*))$.
- As the link between $RC_Y(\mathbf{x})$ and $Z(\mathbf{x})$ is 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.

4.3 The "multi-point expected improvement" response surfaces

While the EI strikes a good balance between exploration and minimization, it computes the improvement of a single point. The multi-point EI (q-EI) is used to measure the improvement of q points $\mathbf{X} = (\mathbf{x}^{n+1},\ldots,\mathbf{x}^{n+q})'$ ([29]). In a multi-objective context, the expressions of the q-EI are:

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

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

We note that q-EI involves $\min(Y(\mathbf{x}^{n+1}),\ldots,Y(\mathbf{x}^{n+q}))$ instead of $Y(\mathbf{x}^{n+1})$. The improvement is provided by the set of q points simultaneously chosen. Besides, in the context of multi-optimization, the reference value is the maximum of the Pareto front outputs.

The **"multi-points expected improvement" formulation** is then:

Find the Pareto set $\mathbb{X}_0$, the solution of the following multi-objective optimization

$$\min_{\mathbf{X}\in\mathbb{R}^{p\times q}}\{-qEI_y(\mathbf{X}), -qEI_{RC_y}(\mathbf{X})\}$$

5 Sequential stategy for enrichment

Seven enrichment strategies have been developed based on the three approaches described above. Once the Pareto front has been found (NGSAII algorithm), points are chosen to enrich the set of observations. Different strategies can be studied. They are described below.

5.1 Enrichment for the "plug in" formulation

With this approach, it is not costly to find the Pareto front since the response surfaces are easily computed. However, the kriging variance has never been considered. If kriging predictions turn out to be of poor

quality, some interesting areas can be missed. Hence the first strategy consists in choosing part of the points from the Pareto front but also part of the points randomly in the parameter space. Other strategies consist in using information from the kriging variance, for example through an expected improvement criterion.

More precisely, five enrichment approaches have been benchmarked and are described below:

1. MyAlea: $\lfloor \frac{q}{2} \rfloor$ [1] points are selected randomly on the Pareto front, while $q - \lfloor \frac{q}{2} \rfloor$ 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 in [30] 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 [29] can be used to replace the q-EI in order 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 Believer strategy. This strategy consists of the following steps: 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. $\widehat{y}(\mathbf{x}_0^1)$ 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. This strategy consists of the following steps: 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 kriging variance is not taken into account during multi-objective optimization. Except if the MyAlea strategy is used, some interesting areas can be missed. The second approach solves this issue by conducting multi-objective optimization directly on the EI.

5.2 Enrichment for the "expected improvement" formulation

Multi-objective optimization is performed on the EI of the output and the robustness criterion. This approach takes into account the kriging variance right from the start of the procedure. For this approach, one enrichment strategy is proposed to add one points one by one:

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

Because this strategy adds points sequentially one by one ($q = 1$), the last formulation is introduced to add points by batch.

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

Method	Minimization	Interesting points	Updates
MyAlea	y, RC_y	Random points on the Pareto front and the parameter 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 for choosing the interesting points.

5.3 Enrichment for the "multi-point expected improvement" formulation

One last enrichment approach is proposed to add q points simultaneously:

7. MqEIyAlea: one point is randomly extracted from the Pareto front. This point will provide q points in the parameter space for the next optimization step.

The seven methods for performing the enrichment are summarized in Table 1.

6 Quality criteria for Pareto fronts

The seven strategies based on three different response surfaces are compared through the quality of the resulting Pareto front. Several measures exist to quantify the quality of a Pareto front (cf [31], [32], [33] and [34]). The Inverted Generational Distance (IGD) and the Hypervolume (HV) are selected here to compare strategies. Let $\mathbf{f} = (f_1, \ldots, f_m)$ be the objective function, $\mathcal{P}$ the theoretical Pareto front, and $\mathbb{X}^*$ the empirical Pareto front where $M = \#\mathcal{P}$. The chosen performance metrics are:

- Inverted Generational Distance (IGD) see [35]:

$$IGD(\mathbb{X}^*) = \sqrt{\frac{1}{M}\sum_{i=1}^{M} d_i^2}$$

  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 [34]. Figure 3 shows the Hypervolume (HV) of a Pareto front. In [36] the authors introduce an algorithm to compute this volume. The empirical HV is compared to the theoretical one. The Hypervolume depends on the reference point. Whenever possible the nadir point of the true Pareto front is used. The Hypervolume then enables the comparaison of two empirical fronts.

7 Applications

This section compares the strategies on two toy functions and one industrial test case. The toy functions are the six-hump Camel in two dimensions and the Hartmann in six dimensions. Two cases are considered depending on whether the derivatives are affordable or not. For efficiency's sake, only three

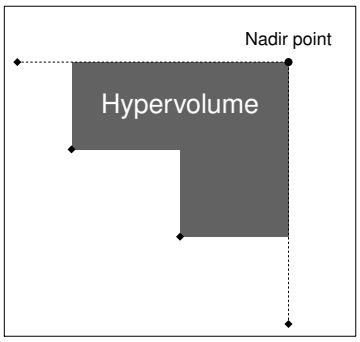


Figure 3: Diamonds represent the individuals of the empirical Pareto front $\mathbb{X}^*$. The black circle is the Nadir point of the set $\mathbb{X}^*$.

of the best strategies are applied on the Hartmann function and on the industrial test case. For these applications NSGA II is performed with populations of a hundred points. Each generation is computed with a crossed probability of 1 and a mutation probability of $\frac{1}{p}$, where p is the number of inputs.

7.1 Six-hump Camel function: 2D

In this application, the six-hump Camel function is considered. The two input variables are affected by uncertainties that are modeled using a Gaussian distribution with a standard deviation of $\delta_j = \frac{0.05}{4}(\max(x_j) - \min(x_j)), j = \{1, 2\}$. Then:

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

Figure 4 shows the four optimal areas for robust optimization in the objective and parameter space.

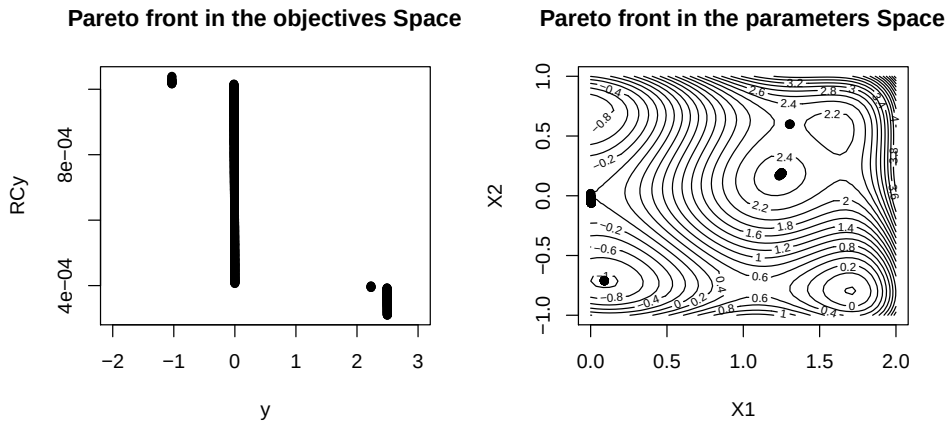


Figure 4: Pareto front of the six-hump Camel function in the objective space (left) and in the parameter space (right).

In order to perform a robust optimization, the function and all the first and second derivatives need to be

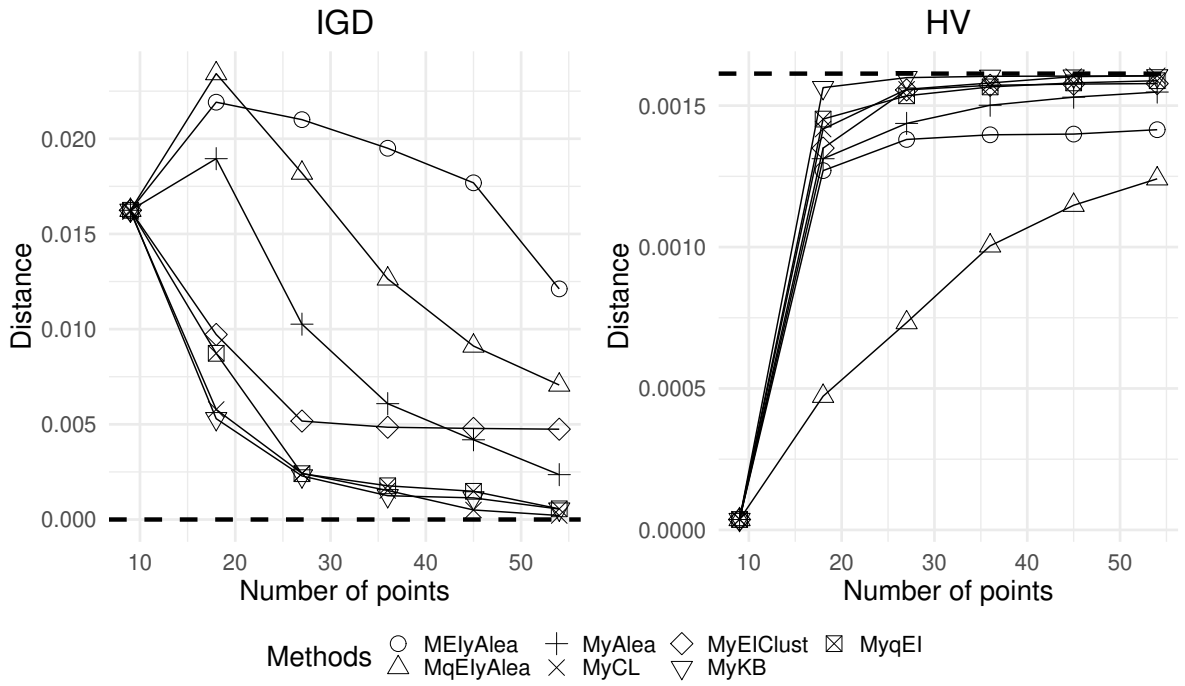


Figure 5: Six-hump Camel function with derivative observations. Evolution of the Pareto metrics with the number of points computed for all the methods over 100 different runs of the algorithm. The HV value of the theoretical front is represented by the dotted line.

predicted. The set of predicted indexes is $u_{pred} = \{1, \ldots, 6\}$ and corresponds to the following vector:

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

7.1.1 Derivative observations

In this first part of the study, the function and all the derivatives are available at each evaluated point. The set of observed indexes is $u_{obs} = \{1, \ldots, 6\}$ that corresponds to the process 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 learning set is a maximin LHS of nine points. Nine updates of five 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 5 and Table 2. In the table, two criteria are used to compare the methods: the computation time and the number of areas found after 54 evaluations. In the figure, the methods are compared through two Pareto front performance metrics.

Our analysis is as follows: MyKB and MyCL are the two most efficient strategies in terms of metrics, found areas, and computation time. Then MyqEI, MEIClust and MqEIyAlea give good results for the metrics and the areas, even though MyqEI is far better in metrics and MqEIyAlea in areas. Finally, MyAlea and MEIyAlea are the least efficient methods 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: Summary of the results obtained with the seven strategies on 100 simulations on the six-hump Camel function with derivative observation. The true number of areas is 4.

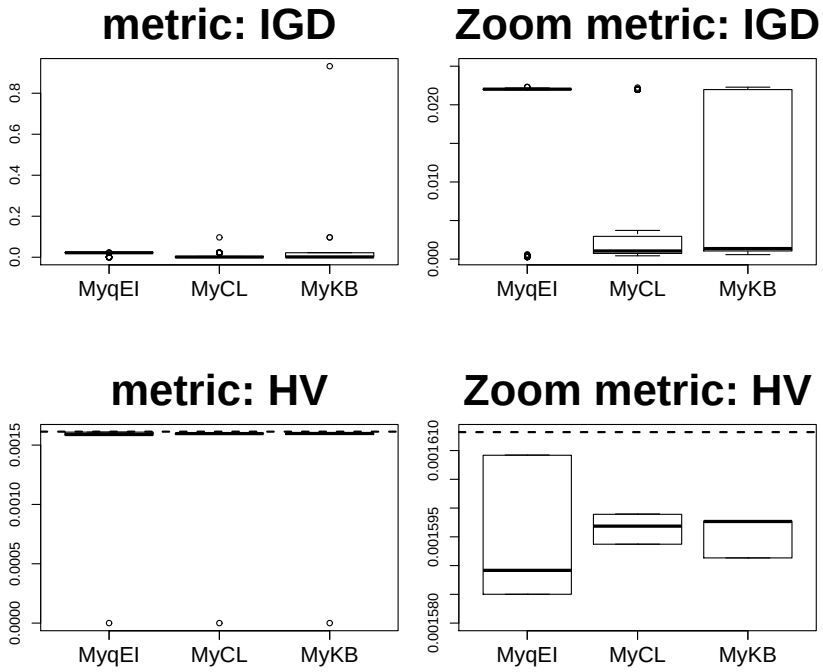


Figure 6: Boxplots of the metrics computed for the three best methods over 100 simulations for the six-hump Camel function with derivative observations.

MqEIyAlea are really time-consuming. Then, the best methods selected to be used for robust optimization in limited budget applications are MyqEI, MyCL and MyKB, which fully exploit batch computation of EI without excessive computational costs. Figure 6 shows the boxplots of these three methods for each distance. MyqEI gives the worst results in mean. This is due to the annealing simulation of the strategy that is difficult to tune.

7.1.2 No derivative observations

The aim of this section is to analyze the behavior of the seven strategies when the derivative observations are not available.
The observed set of indexes is $u_{obs} = \{1\}$, while the predicted set of indexes is $u_{pred} = \{1, \ldots, 6\}$ that

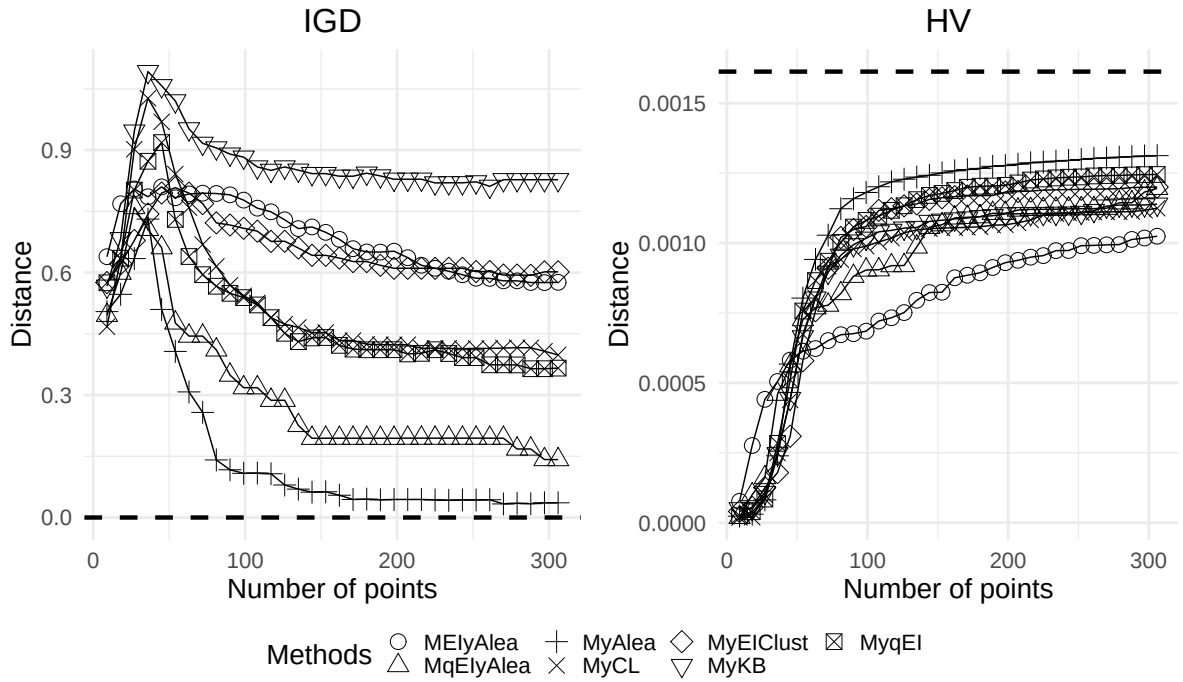


Figure 7: Six-hump Camel function without derivatives' observations. Evolution of the Pareto metrics with the number of points compute for all the methods over 100 different runs of the algorithm. The HV value of the theoretical front is represented by the dotted line.

corresponds to the process 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 maximin LHS of 9 points. The available information is poorer, and detection of the front need to add more points. For this reason, 35 updates of 5 points are performed up to a total budget of 324 points. The optimization scheme is carried out 100 times with different initial learning sets to compare the seven strategies.

Results are provided in Figure 7 and Table 3. Our analysis is as follows: the six-hump Camel function

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: Summary of the results obtained with the seven strategies on 100 simulation on the six-hump Camel function without derivative observation. The true number of areas is 4.

is difficult to approximate without derivative observations. MyAlea strategy, which is partially based on a random search, gives the best results. In this context, too much reliance should not be placed upon

kriging. MyqEI and MqEIyAlea provide quite good results because they use the qEI criterion. This takes into account the improvement provided by a batch of points of the front. However, MqEIyAlea is too time-consuming. The MyCL strategy, which does not trust the response surface, also gives quite good results, unlike the MyKB. Finally, the MyEIClust and MEIyAlea strategies that use the EI criterion provide poor results, even though the MyEIClust strategy is rather better thanks to the clustering used to enrich the set. The best strategy is MyAlea, but MyqEI and MyCL are also retained in order to test them in a higher dimension.

7.2 Hartmann function: 6D

In this section, the three best strategies identified in Section 7.1.1 are benchmarked in a higher dimension (six). A Gaussian process model is built with a tensor product kernel using the Matern5_2 covariance function (see Equation 4). The studied function is the six-dimensional Hartmann function defined by:

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

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

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

and

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

The random variables are x_4 and x_5 and follow a centered Gaussian distribution with a standard deviation of $\delta_j = \frac{0.05}{4}(\max(x_j) - \min(x_j))$, $j = \{4, 5\}$.

As above, two cases are considered depending on whether or not derivative observations are provided.

7.2.1 Derivative observations

The sets of indexes are $u_{obs} = u_{pred} = \{1, 5, 6, 20, 26, 27\}$ which correspond to the process vectors:

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

The initial sample set is a maximin LHS composed of 18 points. Five updates are made and 18 points are added by update for a total budget of 108 points. The best methods found in the previous test case with derivative information, MyqEI, MyCL and MyKB strategies, are applied.

The left part of Figure 8 shows that the three methods converge to the true front. At step 2, MyqEI gives the more advanced front. At the final step, the three methods perform very well (see the right part of

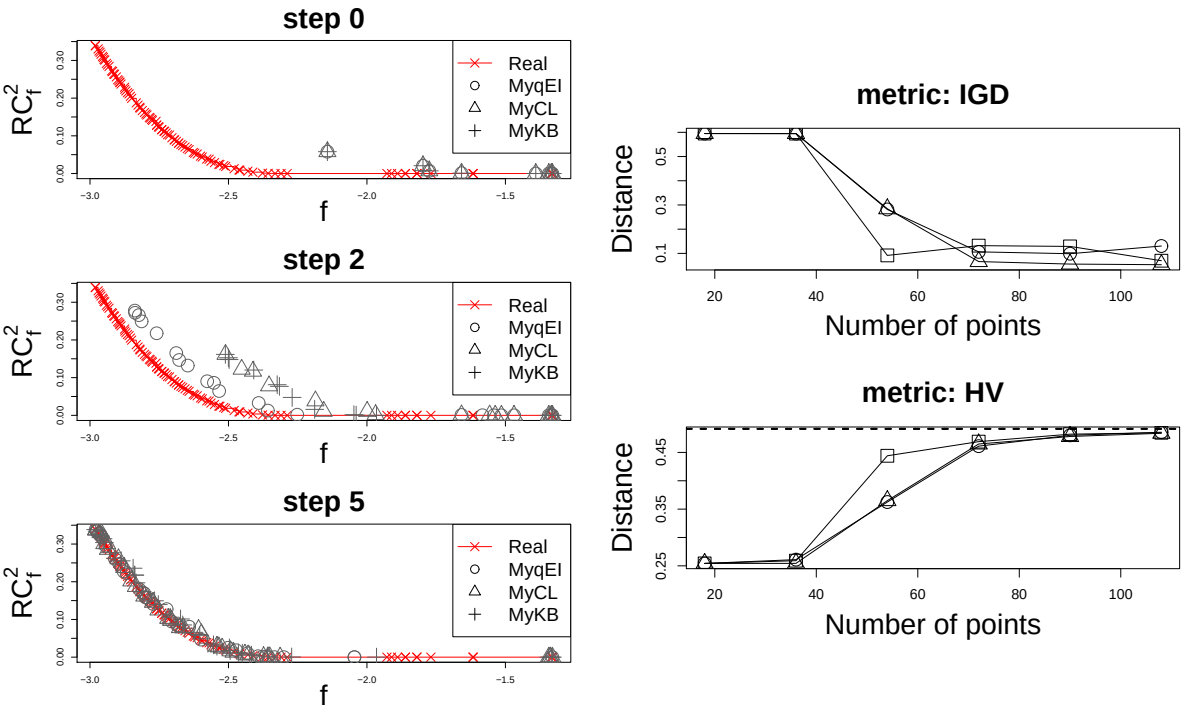


Figure 8: On the left: Pareto fronts obtained during the optimization procedure of the three strategies at the initial step (step 0), middle step (step 2) and final step (step 5). On the right: evolution of the metrics computed during the algorithm for all the methods over 100 simulations for the Hartmann function with derivative observations. The HV value of the theoretical front is represented by the dotted line.

Figure 8). MyKB and MyCL take 10 minutes for the five steps when MyqEI takes 12 minutes.

7.2.2 No derivative observations

The sets of indexes are $u_{obs} = \{1\}$ and $u_{pred} = \{1, 5, 6, 20, 26, 27\}$. They correspond to the process vectors:

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

The initial design is still a maximin LHS composed of 18 points. More updates are provided since derivatives are not affordable. Here 35 updates of 18 points are sequentially computed up to a total budget of 648 points. The best methods identified previously, MyAlea, MyqEI and MyCL strategies, are applied.

The left part of Figure 9 shows that the three methods converge to the true 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 9 shows that the distance starts to converge to the expected value within the first 100 points. For the IGD metric, the values are subject to few perturbations around the expected value zero. For the HV measure, the three methods converge to the theoretical value with only 100 points that correspond to 6 updates. MyAlea takes 1h15min, MyqEI takes 1h40min, and MyCL takes 1h04min for the 35 steps.

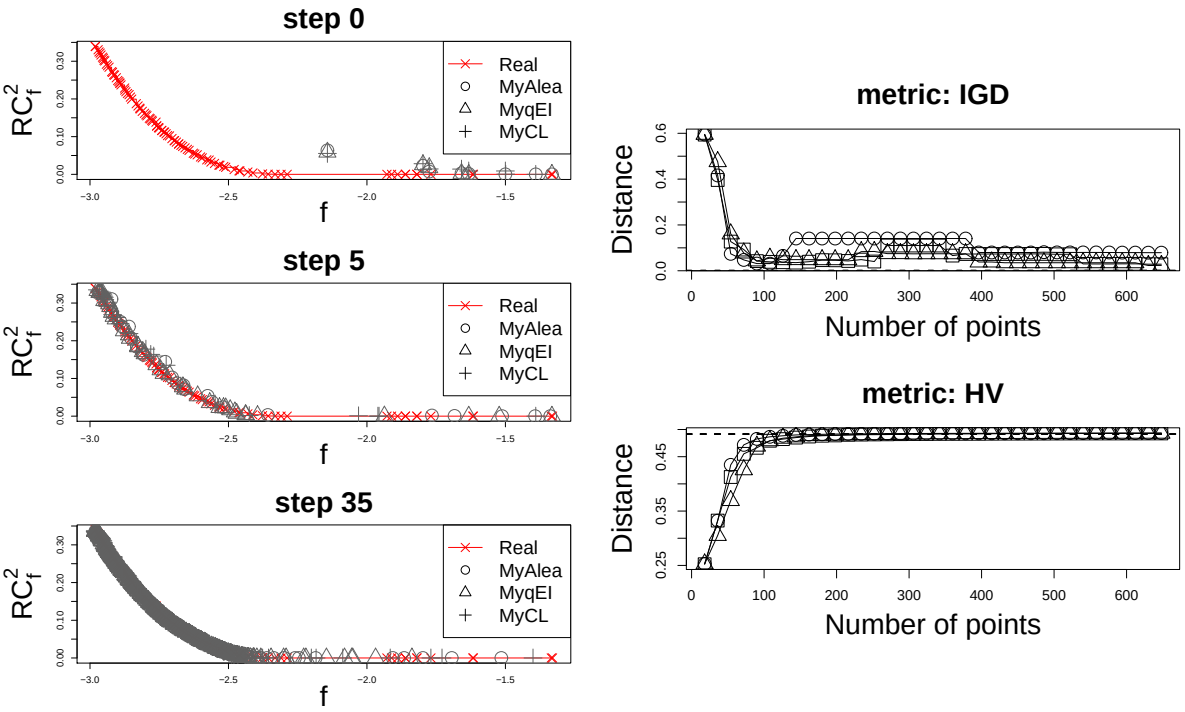


Figure 9: On the left: Pareto fronts obtained during the optimization procedure of the three strategies at the initial step (step 0), step 5 and final step (step 35). On the right: evolution of the metrics during the algorithm computed for all the methods in 100 simulations for the Hartmann function with no derivative observation. The HV value of the theoretical front is represented by the dotted line.

7.3 Industrial test case

7.3.1 Context

The chosen application is a front motor cooling fan design. Within daily use, uncertainties under operating conditions are mainly due to external parameters. In automotive application, it would be the design of the car, its air entrance conditions, the size and shape of the engine, the temperature, humidity, etc. As such part is usually provided by automotive supplier, these parameters are complete unknown, and the OEMs generally take the responsibility to validate their car as a system for all these conditions. However, in order to ensure the qualification of the product, the specification that are given are very strict and aims to compare fairly the fans between them. For instance, tests are made for a fixed rotational speed on standard test rig (see [37]). Therefore, the remaining variabilities are coming from the geometrical changes and the measurement uncertainties.

The use of numerical simulation with a very well controlled workflow (repeatability, mesh independency, controlled numerical convergence, etc.) help suppliers to reduce the measurement uncertainties. The geometry changes are an actual issue with production process, which involves plastic injection with glass fiber. It is well known in the state of the art that the plastic component that goes out of the mold does not have the same shape than the mold cavity. In particular, shrinking, warpage and residual stress distribution can yield plastic deformation, even long time after the production if we consider effect of temperature, humidity and aging. Sometimes, the blade modification is so important that the mold must be reworked, which is obviously an additional cost that could be saved with a robust optimization approach.

These phenomena are observed on fans for a long time, and previous experience with retro-engineering on used fans has allow suppliers to quantify the blade deformation: it can be easily converted into modification of the stagger angles, the chord, the camber, etc. However, the parameters that were selected in the present investigation are those which are varying the more, because the maximum freedom for change is located far from the hub and far from the ring (hub and ring are solid and massive cylinders that retains the blade at their attachments). If the robust optimization sorts out the more robust design according to these parameters which are at risk, it would without no doubt reduce the uncertainties due to supplier production process.

7.3.2 Numerical chain

We choose a low fidelity turbomachinery predimensioning tool named $TurboConcept^{TM}$ as main simulation part for the proposed design optimization .This code is developed and maintained by the Laboratoire de Mécanique des Fluides et d'Acoustique (LMFA) at Ecole Centrale de Lyon. The principal equations of Aerothermodynamics used to construct $TurboConcept^{TM}$ are described in [38] and [39]. It can be used according to two modes of execution. Theses are *inverse design*, a mode that find the most suitable fan geometry for specific input operating conditions and that is described in appendix A of [40], and *direct*, a mode that calculates performance criteria associated with specific input fan geometry and specific input operating conditions.This second mode is used to perform robust optimization.

A fan blade is divided into five sections of vane height. These are highlighted in red curves on Figure 10 on the right. A blade profile is parameterized according to three parameters of chord length, stagger and maximum camber (H_{max}). Their geometrical definition is represented in Figure 10 on the left. As a result, each fan blade is characterized by fifteen geometrical parameters, namely five chord, stagger

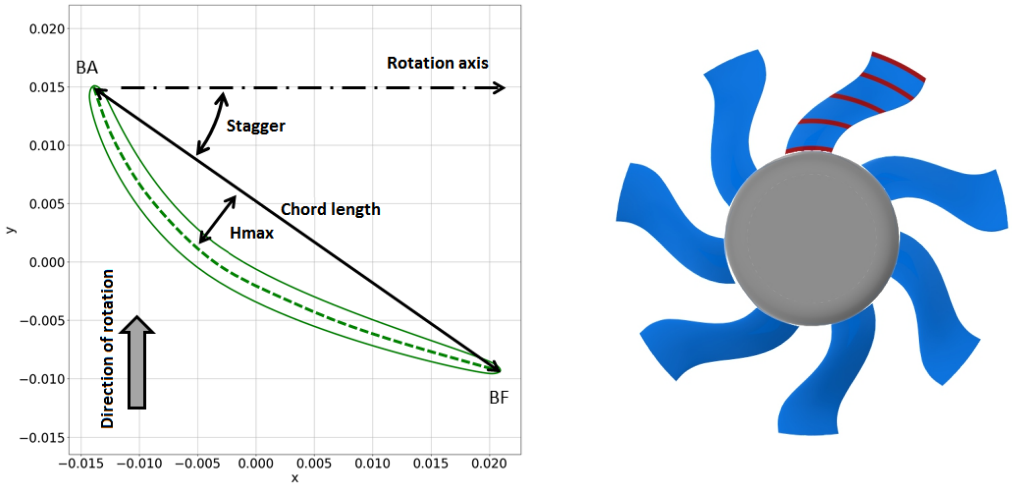


Figure 10: Blade section with the three input parameters on the left. Sections are represented on the right by the red lines along one blade. Section 1 is the closest to the disc and section 5 the most far away.

and maximum camber. They are denoted $\mathbf{x} = (x_1, \ldots, x_{15}) \in D$ and can vary within a specific range $[Min; Max]$ shown in Table 4. Among these inputs, only intermediate staggers (x_7, x_8, x_9) are subjected to manufacturing tolerances $x_i \pm 2\delta_i$, $i = \{7, 8, 9\}$. The values of δ_i are given by the industrial experts (see Table 4). The first and second derivatives of the uncertain variables are provided by the numerical code. The operating conditions of the fan have been set to the specific values in Table 5.

Input	Chord length					Stagger					Hmax				
Section	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Notation	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
Min	0.04	0.06	0.08	0.09	0.11	-50.67	-59.68	-65.87	-70.29	-73.58	3.82	3.82	3.82	2.86	1.91
Max	0.07	0.09	0.11	0.14	0.16	-45.85	-54	-59.59	-63.6	-66.57	5.73	5.73	5.73	4.29	2.86
δ	0	0	0	0	0	0	1.16	1.28	1.36	0	0	0	0	0	0

Table 4: Inputs of the numerical code. H_max is the maximal camber height. These inputs are considered at 5 different sections from sections 1 to 5.

Physical parameter name	fixed value
Rotation speed Ω $(rad.s^{-1})$	277.5
Volume flow rate Q $(m^3.s^{-1})$	0.833

Table 5: Fixed physical parameters and values

The performance criterion to be optimized is based on the static efficiency of the fan, defined by:

$$\eta = \frac{Q \times \Delta P}{C \times \Omega} \tag{10}$$

As the rotational speed Ω $(rad.s^{-1})$ and the volume flow Q $(m^3.s^{-1})$ are fixed, fan efficienty η (.) depends on two outputs of $TurboConcept^{TM}$. The first one is the delta of static pressure ΔP (Pa) between the output and the fan input. This pressure energy is provided by fan rotation and is necessary to counterbalance the pressure loss induced by the frictional forces acting on the fluid as it flows through the radiator fins. The second one is the resistive torque C $(N.m)$, corresponding to the moment of pressure and the viscous forces applied by the air on the fan.

7.3.3 Results

The initial sample set D_1 is a maximin LHS of 46 observations. Figure 11 shows the learning sample set in the true objective space $\{\eta, RC_\eta\}$. η represents the real costly efficiency function, while RC_η the robustness criterion calculated on η given by Equation (2). The total budget is composed of 136 points, and 90 points are added to the initial design with 5 updates of 18 points. The best three methods (MyCL, MyKB and MyqEI) used in Section 7.2.1 are selected to perform robust optimization.

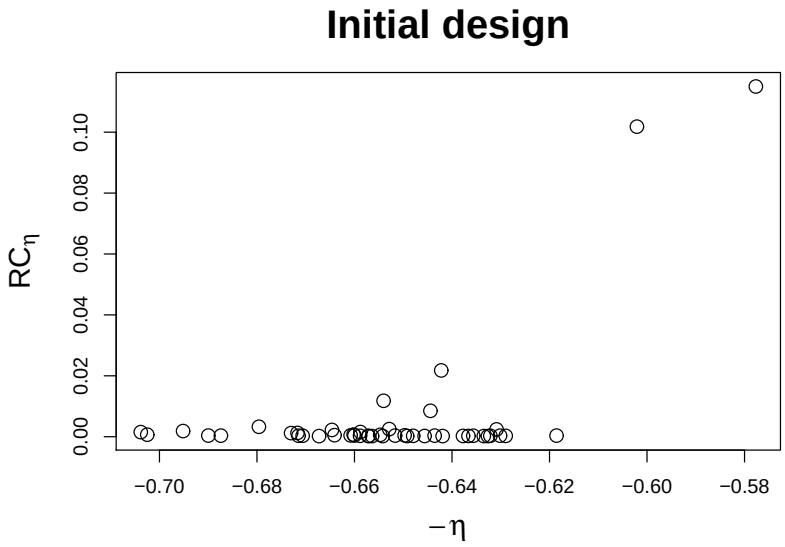


Figure 11: The 46 initial observation points in the true objective space: opposite efficiency ($-\eta$) and robustness criterion calculated on the efficiency (RC_η).

It can be seen in Figure 12 that at the final step, MyCL, MyKB and MyqEI have added points in the same interesting area. MyCL provides the worst progress in the objective space, while MyqEI gives the most dispersed areas and MyKB the most advanced ones. These differences stem form where strategies add points along updates. As can be seen on Figure 14, the three methods progress in the same interesting area. However, the MyqEI adds points in two different areas at the first update (in the middle and at the bottom right), which explains why the MyqEI strategy gives the most dispersed front. MyCL and MyKB progress in the same way, simply MyCL is slower.

	Update	1	2	3	4	5	Total
MyCL	Time	0h18	0h30	0h 40	1h00	1h00	3h28
MyKB	Time	0h18	0h31	0h 44	1h00	1h01	3h34
MyqEI	Time	0h16	0h25	0h 36	0h48	1h00	3h05

Table 6: Computation time for the three strategies, MyCL, MyKB and MyqEI.

Table 6 shows that MyqEI is the fastest strategy. To conclude, the three strategies give the same interesting non-dominated points that are compromises between efficiency and robustness. The shapes of two of these compromises (see the big square and triangle on Figure 12) are represented on Figure 13.

Four new optimization runs have been performed from new initial LHS designs (D_2, D_3, D_4, D_5). The initial number of points for the five models is $\{46, 48, 48, 48, 49\}$, a difference that stems from

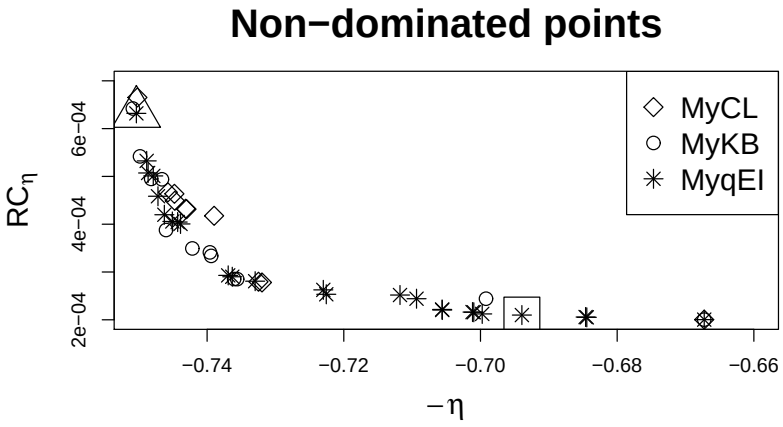
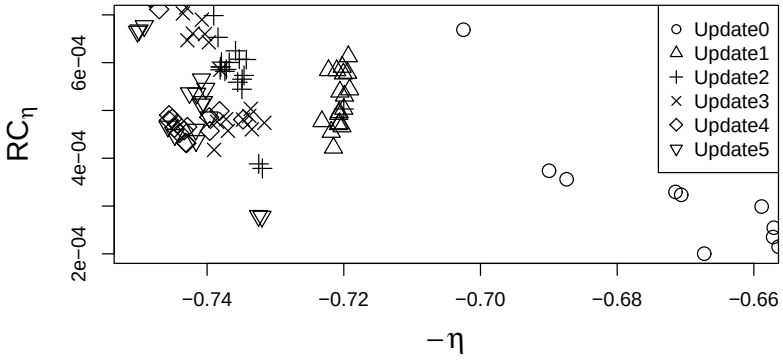


Figure 12: Non-dominated points of the final design for methods MyCL, MyKB and MyqEI in the true objective space: opposite efficiency ($-\eta$) and robustness criterion calculated on the efficiency (RC_η).

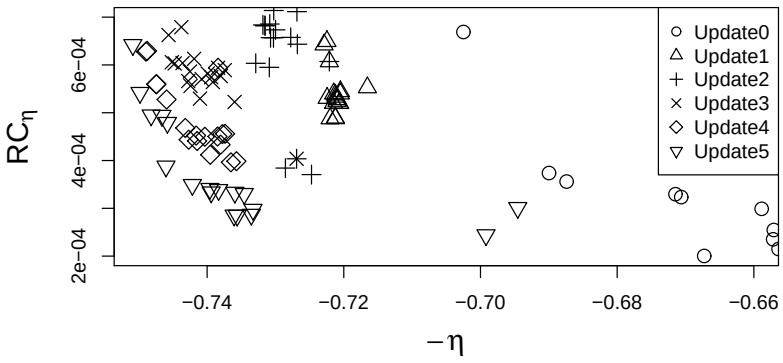
Figure 13: The shape on the left corresponds to the square of Figure 12, while that on the right to the triangle.

$TurboConcept^{TM}$. Fifty points were launched but not all of them converged. Nevertheless, at the end of the optimization all designs have 136 points. Figure 16 shows the five sets of learning samples in the true objective space $\{\eta, RC_\eta\}$.

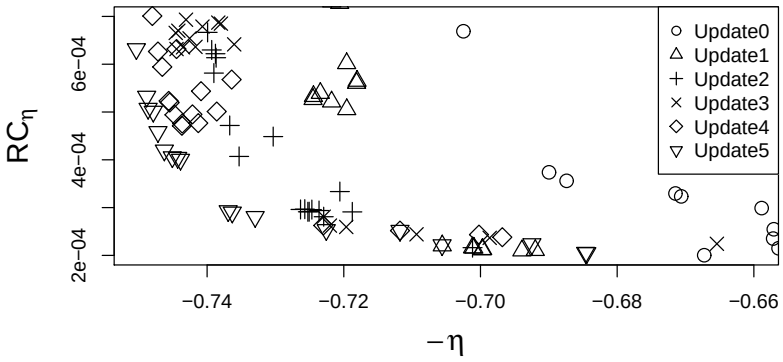
Figure 17 shows that the five different initial designs converge to the same Pareto front with all three methods. The choice of the optimized initial LHS has no impact on the final result. Moreover, the MyqEI method is the most reliable, as the five Pareto front coincide perfectly.



(a) MyCL Method



(b) MyKB Method



(c) MyqEI Method

Figure 14: Progression of the algorithm for the method MyCL (a), MyKB (b), MyqEI (c) in the true objectives space: opposite efficiency ($-\eta$) and robustness criterion calculated on the efficiency (RC_η).

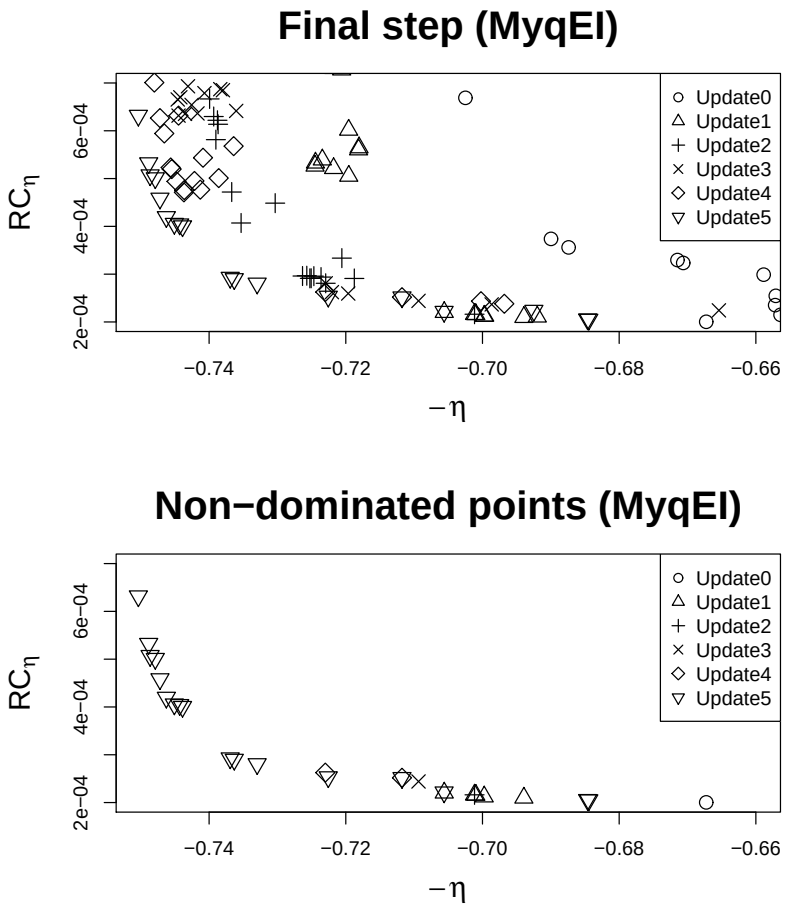


Figure 15: Zoom (top) and non-dominated (bottom) points of the final design for the MyCL method in the true objective space: opposite efficiency ($-\eta$) and robustness criterion calculated on the efficiency (RC_η).

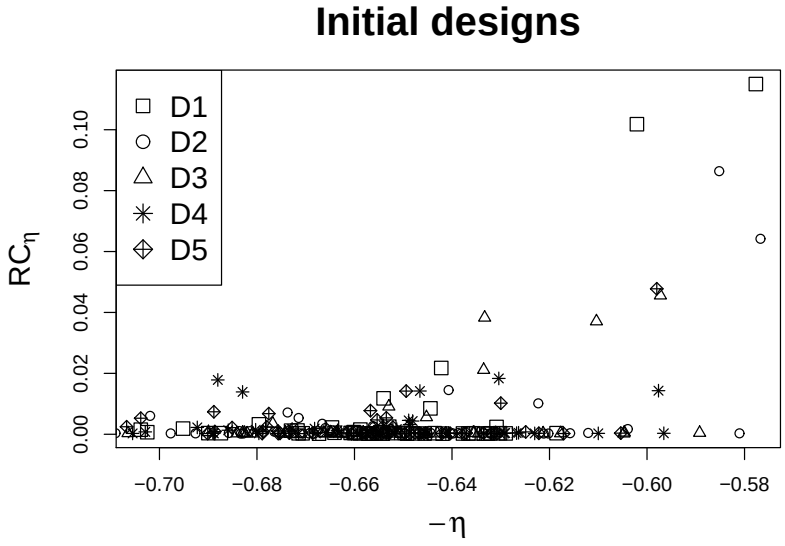
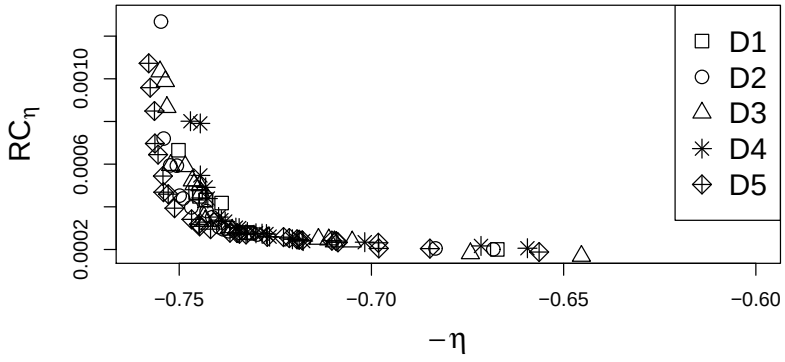
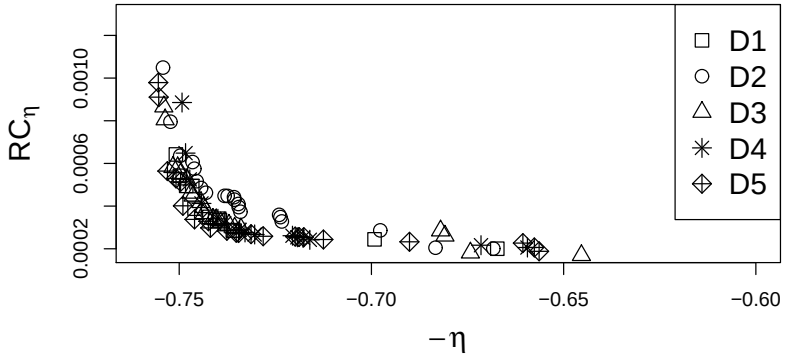


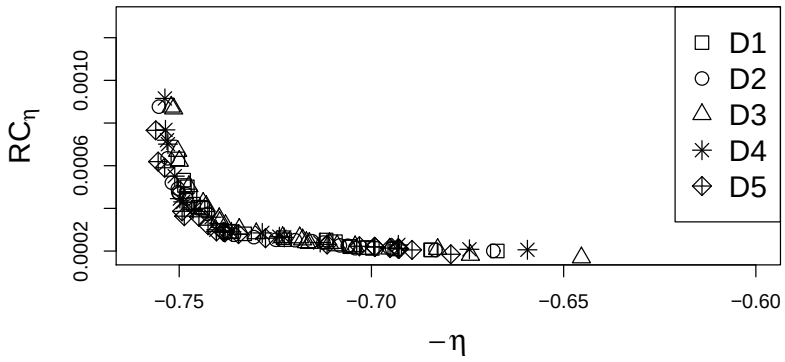
Figure 16: The initial observation points for five LHS designs in the true objective space: opposite efficiency ($-\eta$) and robustness criterion calculated on the efficiency (RC_η).



(a) MyCL Method



(b) MyKB Method



(c) MyqEI Method

Figure 17: Non-dominated points of the five final designs for the methods MyCL (a), MyKB (b), MyqEI (c) in the true objective space: opposite efficiency ($-\eta$) and robustness criterion calculated on the efficiency (RC_η).

8 Conclusion

In this paper, we propose an efficient kriging-based robust optimization procedure. The methodology is based on a multi-objective optimization of the function and a robustness criterion simultaneously. The robustness criterion is defined as a Taylor expansion of the local variance. This expression using derivatives has the advantage of being easily predicted under Gaussian process modeling. The introduced multi-objective strategies are iterative and based on two steps: a NSGA-II algorithm performed on predicted versions of the two objectives and a relevant enrichment composed of a batch of points carefully chosen from the Pareto front. Seven strategies have been compared on two toy functions. The study reveals that it is far more efficient computer-wise to optimize the plug in versions of kriging prediction rather than EI. When points are selected using kriging variance, the procedure detects all the diversity of the robust solutions. Finally, the methodology is applied on an industrial problem that consists in optimizing motor fan shape taking into account production uncertainties. Interesting shapes are provided to ensure robust optimization of turbomachinary efficiency, which strike the right balance between efficiency and robustness.

Acknowledgments

This work benefited from the financial support of the French ANR project "PEPITO" (ANR-14-CE23-0011). We also thank the LMFA (Laboratory of Fluid, Mechanics and Acoustics from Ecole Centrale de Lyon) that provided the numerical codes of the industrial test case, as well as Manuel Henner from VALEO who helped us understand the context of the industrial test case.

References

[1] Thomas J. Santner, Brian J. Williams, and William I. Notz. *The design and analysis of computer experiments*. Springer Series in Statistics. Springer-Verlag, New York, 2003.

[2] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[3] Nicolas Lelièvre, Pierre Beaurepaire, Cécile Mattrand, Nicolas Gayton, and Abdelkader Otsmane. On the consideration of uncertainty in design: optimization - reliability - robustness. *Structural and Multidisciplinary Optimization*, 54(6):1423–1437, Dec 2016.

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

[5] Julien Marzat, Eric Walter, and Hélène Piet-Lahanier. Worst-case global optimization of black-box functions through kriging and relaxation. *Journal of Global Optimization*, 55(4):707–727, 2013.

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

[7] Samee Ur Rehman, Matthijs Langelaar, and Fred van Keulen. Efficient kriging-based robust optimization of unconstrained problems. *Journal of Computational Science*, 5(6):872–881, 2014.

[8] Simon Moritz Göhler, Tobias Eifler, and Thomas J Howard. Robustness metrics: Consolidating the multiple approaches to quantify robustness. *Journal of Mechanical Design*, 138(11):111407, 2016.

[9] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European journal of operational research*, 235(3):471–483, 2014.

[10] Amadeu Almeida Coco, Elyn L Solano-Charris, Andréa Cynthia Santos, Christian Prins, and Thiago Ferreira de Noronha. Robust optimization criteria: state-of-the-art and new issues. *Technical Report UTT-LOSI-14001, ISSN: 2266-5064*, 2014.

[11] Renata Troian, Koji Shimoyama, Frédéric Gillot, and Sébastien Besset. 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, 2016.

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

[13] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006.

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

[15] Tobias Wagner, Michael Emmerich, André Deutz, and Wolfgang Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 718–727. Springer, 2010.

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

[17] Wudong Liu, Qingfu Zhang, Edward Tsang, Cao Liu, and Botond Virginas. On the performance of metamodel assisted moea/d. In *International Symposium on Intelligence Computation and Applications*, pages 547–557. Springer, 2007.

[18] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2010.

[19] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. 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, 2008.

[20] Mickael Binois. *Uncertainty quantification on pareto fronts and high-dimensional strategies in bayesian optimization, with applications in multi-objective automotive design*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2015.

[21] Michael TM Emmerich, André H Deutz, and Jan Willem Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2147–2154. IEEE, 2011.

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

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

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

[25] Shinkyu Jeong and Shigeru Obayashi. 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, 2005.

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

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

[28] Delphine Dupuy, Céline Helbert, Jessica Franco, et al. Dicedesign and diceeval: Two r packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):1–38, 2015.

[29] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Springer, 2010.

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

[31] David Allen Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Wright Patterson AFB, OH, USA, 1999. AAI9928483.

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

[33] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

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

[35] Carlos A Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican International Conference on Artificial Intelligence*, pages 688–697. Springer, 2004.

[36] Carlos M. Fonseca, Luís Paquete, and Manuel López-Ibáñez. 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, July 2006.

[37] ISO 5801:2007(E) - Industrial fans — Performance testing using standardized airways. Standard, International Organization for Standardization, Geneva, CH, December 2007.

[38] Xavier Ottavy. Cours de turbomachine à fluide compressible. Laboratoire de Mécanique des Fluides et d'Acoustique, 2007.

[39] Xavier Ottavy. Basic design of fans and compressors - endwall flows and losses. Laboratoire de Mécanique des Fluides et d'Acoustique, 2015.

[40] Quentin Rendu. *Modélisation des écoulements transsoniques décollés pour l'étude des intéractions fluide-structure*. PhD thesis, Université Claude Bernard Lyon 1, December 2016. Thèse de doctorat dirigée par Jacob, Marc C. et Aubert, Stéphane Mécanique des fluides Lyon 2016.