Reliability Engineering and System Safety 200 (2020) 106913

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

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

Mélina Ribaud[a,b], Christophette Blanchet-Scalliet[a], Céline Helbert*[,a], Frédéric Gillot[b,c]

[a] Univ Lyon, École Centrale de Lyon, CNRS UMR 5208, Institut Camille Jordan, 36 avenue Guy de Collongue, Ecully, Cedex F-69134, France
[b] Univ Lyon, Ecole Centrale de Lyon, LTDS, CNRS UMR 5513, 36 avenue Guy de Collongue, Ecully Cedex F-69134, France
[c] INRIA Rennes, I4S Team, France

ARTICLE INFO

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

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 expectation and variance of a function have already been developed to answer to this question. However, efficient estimation in the framework of time-consuming simulation has not been completely explored. In this paper, a robust optimization procedure based on Taylor expansion, kriging prediction and a genetic NSGA-II algorithm is proposed. The two objectives are the Taylor expansion of expectation and variance. The kriging technique is chosen to surrogate the function and its derivatives. Afterwards, NSGA-II is performed on kriging response surfaces or kriging expected improvements to construct a Pareto front. One point or a batch of points is chosen carefully to enrich the learning set of the model. When the budget is reached the non-dominated points provide designs that make compromises between optimization and robustness. Seven relevant strategies based on this main procedure are detailed and compared in two test functions (2D and 6D). In each case, the results are compared 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.

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, though computational cost tends to explode even on state of the art super-computers. One main use of these simulations is to solve optimization problems. This work focuses on cases where the optimized solution is sensitive to input perturbations. For example, these perturbations are due to random dimensional 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 expectation and the second the variance. 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 of special interest. 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 **x** such that *f* (**x, U**) is minimal where **U** is a random vector (cf [4–7]). The authors in [4] propose to minimize the expectation of *f*(**x, 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 find x such that $f(x + H)$ is minimal

* Corresponding author.
E-mail address: celine.helbert@ec-lyon.fr (C. Helbert).

https://doi.org/10.1016/j.ress.2020.106913
Received 30 January 2019; Received in revised form 13 February 2020; Accepted 11 March 2020
Available online 13 March 2020
0951-8320/

with H a random vector accounting for perturbations such as manufacturing uncertainties. We introduce a multi-objective strategy to detect the whole set of robust solutions. The first objective is to quantify a level of the function in a neighborhood of a solution. The local expectation, $E[f(x+H)]$ is then considered. The second objective aims at measuring the robustness of a solution. Göhler et al. [8], Gabrel et al. [9] and Coco et al. [10] give some overviews of different robustness criteria. As our industrial partners quantify the robustness of a solution by the local variance of the output (see e.g. [6,11]), the second objective is $Var[f(x+H)]$.

In a context of a black-box optimization, these two objectives are computed by Monte-Carlo method which is unaffordable in practice even using a metamodel. In this paper, proxies based on the Taylor expansion as introduced by Darlington et al. [12], are proposed. These proxies are easily computed, their expression are analytical and involve the first and second derivatives. In the context of time-consuming simulations, these criteria are predicted by kriging. Kriging, form which the covariance structure between the GP model of the function and all the derivatives results, is well adapted. This structure is described in [13] and used again by Le Gratiet [14].

As the two objectives are accessible through kriging, a multi-objective optimization is performed to provide set of optimal solutions. In the literature, several approaches (see [15] for an overview) mixing a GP modeling and multi-objective optimization are proposed: the aggregation methods (see [16–18]), the hypervolume methods (see [19–21]), 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 has to perform 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 Jeong and Obayashi [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 Pronzato and Éric Thierry [26].

The paper is structured as follows. First the proxies of the two objectives are introduced in section 2. Then in section 3, the context of a Gaussian process metamodeling is presented. The general multi-objective optimization scheme is developed 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. Two proxies for the first moments of the function

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. Taking into account uncertainties into an optimization scheme needs the construction of specific criteria which quantify the local sensitivity to variabilities. A natural approach is to optimize jointly the local expectation and variance of the output. Let f be the studied function (a two-times differentiable function)

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

where p is the number of input variables, i.e. $x=(x_1,\ \dots,x_p)$. At a point $x \in D$, the quantities of interest - local expectation and variance - are $E(f(x+H))$ and $Var(f(x+H))$ where $\mathbb{H}$, representing the production errors, is a centered Gaussian vector with the covariance matrix Δ defined by:

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

In a black box context, the evaluation of $E(f(x+H))$ and $Var(f(x+H))$ requires a lot of calls to function f. In this paper we propose to consider a second order Taylor's approximation of function f, denoted $\tilde{f}$ and its associated expectation and variance. The objectives to be optimized are then :

- a Level Criterion of function f (LC_f) :

$$LC_f(\mathbf{x}){=}E(\tilde{f}(x+H)) = f(x) + \frac{1}{2}tr((\mathbb{H}_f(\mathbf{x})\Delta)) \tag{2}$$

- a Robustness Criterion of function f (RC_f) :

$$RC_f(\mathbf{x}){=}Var(\tilde{f}(x+H)) = tr(\nabla_f(\mathbf{x})\nabla_f(\mathbf{x})'\Delta) + \frac{1}{2}tr((\mathbb{H}_f(\mathbf{x})\Delta)^2) \tag{3}$$

where ∇_f is the gradient of f, $\mathbb{H}_f$ the Hessian matrix of f, tr is the matrix trace.

LC_f is to be minimized or maximized according to what is expected for function f. RC_f is to be minimized. 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 LC_f and RC_f.

Remarks:

- If the simulator provides the first derivatives with the output itself, the LC_f and RC_f criteria can be computed with only one call to the computer code.
- In the context of costly simulations, a robust optimization cannot be directly performed on LC_f and RC_f. 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 Eqs. (2) and (3), the criteria to be optimized depend 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 $k(\mathbf{x},\tilde{\mathbf{x}}) = \sigma^2 r_\theta(\mathbf{x}-\tilde{\mathbf{x}}),\ \forall\ (\mathbf{x},\tilde{\mathbf{x}}) \in D \times D$. 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 $(Y_{x_i,x_j}(\mathbf{x}))_{\mathbf{x}\in D} = \left(\frac{\partial^2 Y}{\partial x_i \partial x_j}(\mathbf{x})\right)_{\mathbf{x}\in D}$ 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 $(\mathbf{x}^1, \ldots, \mathbf{x}^n)$ be the initial design of experiments, where $\mathbf{x}^k \in D$, $1 \leq k \leq n$. 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:

$$\begin{aligned} \mathbf{y} &= (y^1, \ldots, y^n)' \\ \mathbf{y}_{x_i} &= \left(y^1_{x_i}, \ldots, y^n_{x_i}\right)' \\ \mathbf{y}_{x_i,x_j} &= \left(y^1_{x_i,x_j}, \ldots, y^n_{x_i,x_j}\right)' \end{aligned}$$

$(y^k, y^k_{x_1}, \ldots, y^k_{x_p}, y^k_{x_1,x_1}, \ldots, y^k_{x_i,x_j}, \ldots, y^k_{x_p,x_p})$, $k \in \{1, \ldots, n\}$ is then a realization of the following $d = 1 + \frac{3p}{2} + \frac{p^2}{2}$ dimensional GP:

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

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

3.2. Kriging predictions of the output and its derivatives

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 $e_{obs} = (1, 0_{\mathbb{R}^{d_{obs}-1}}, \ldots, 1, 0_{\mathbb{R}^{d_{obs}-1}})' \in \mathbb{R}^{nd_{obs}}$, $d_{obs} = \#u_{obs}$ and $e_{pred} = (1, 0_{\mathbb{R}^{d_{pred}-1}})' \in \mathbb{R}^{d_{pred}}$, $d_{pred} = \#u_{pred}$. The kriging mean is then given by the following equation:

$$\hat{\mathbf{z}}_{u_{pred}}(\mathbf{x}) = \hat{\mu} e_{pred} + \mathbf{c}_\theta(\mathbf{x})' \boldsymbol{\Sigma}_\theta^{-1} (\mathbf{z}_{u_{obs}} - \hat{\mu} e_{obs}), \quad \hat{\mathbf{z}}_{u_{pred}}(\mathbf{x}) \in \mathbb{R}^{d_{pred}} \tag{4}$$

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

$$\hat{\mu} = (e'_{obs} \Sigma_\theta^{-1} e_{obs})^{-1} e'_{obs} \Sigma_\theta^{-1} \mathbf{z}_{u_{obs}}.$$

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

$$\hat{\mathbf{s}}^2_{u_{pred}}(\mathbf{x}) = \boldsymbol{\Sigma}_\theta(\mathbf{x}, \mathbf{x}) - (e_{pred} \quad \mathbf{c}_\theta(\mathbf{x})) \begin{pmatrix} 0 & e'_{obs} \\ e_{obs} & \boldsymbol{\Sigma}_\theta \end{pmatrix}^{-1} \begin{pmatrix} e'_{pred} \\ \mathbf{c}_\theta(\mathbf{x}) \end{pmatrix},$$

$$\hat{\mathbf{s}}^2_{u_{pred}}(\mathbf{x}) \in \mathcal{M}_{d_{pred} \times d_{pred}}$$

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

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

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

3.3. Gaussian processes conditionned by derivatives: 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:

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

The kriging covariance kernel is a tensor product one:

$$cov(Y(\mathbf{x}), Y(\tilde{\mathbf{x}})) = k(\mathbf{x} - \tilde{\mathbf{x}}) = \sigma^2 \prod_{j=1}^{p} \rho_{\theta_j}(|x_j - x'_j|), \quad \theta = (\theta_1, \ldots, \theta_p) \in \mathbb{R}_+^p \tag{5}$$

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 Fig. 1)
- 10 learning points where f and all the derivatives are observed (middle part of Fig. 1)
- 60 learning points where f is observed (right part of Fig. 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 Fig. 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 Fig. 1 shows that kriging without derivatives does better. However in industrial applications, computing derivatives is often more affordable.

4. Robust optimization procedure

The proposed robust optimization procedure based on the two criteria introduced above (see Eqs. (2) and (3)) is as follows:

$$\text{Find the Pareto set } \mathbb{X}_0\text{, the solution of the following multi-objective optimization } \min_{\mathbf{x} \in \mathbb{R}^p} \{LC_f(\mathbf{x}), RC_f(\mathbf{x})\} \tag{6}$$

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 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 $\{\hat{obj}_{LC_f}(x)\}$ and $\{\hat{obj}_{RC_f}(x)\}$ related to the two objectives $\{LC_f(x)\}$ 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

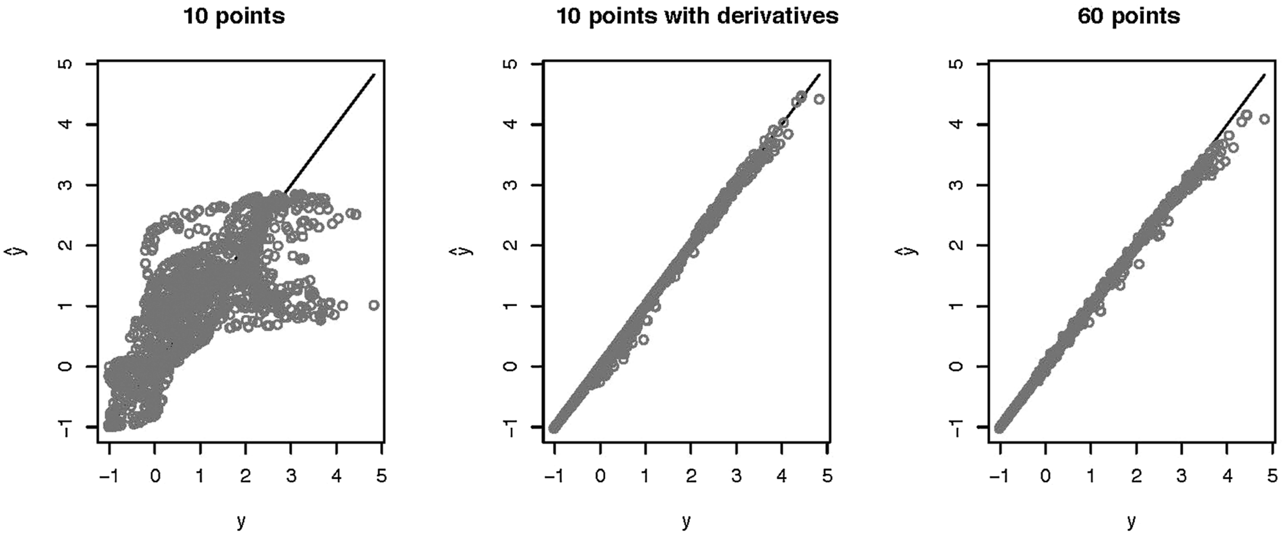


Fig. 1. Prediction plots for the six-hump Camel function: 10 points without observation of the derivatives (on the left), 10 points with 5 derivatives (in the middle) and 60 points without observation of the derivatives (on the right).

optimization method is applied to solve $\min_{\mathbf{x}\in\mathbb{R}^p}\{\hat{obj}_{LC_f}(x), \hat{obj}_{RC_f}(x)\}$. 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 criteria (level and robustness). This approach, quite crude, is denoted by the "plug in" approach in the following and is described below. The second approach is based on the famous Expected Improvement quantity in order to take into account prediction uncertainty. The third one is the most complex: it optimizes the multipoint Expected Improvement versions of $\{LC_f(x)\}$ and $\{RC_f(x)\}$.

4.1. The "plug in" response surfaces

We remind you that $\hat{z}(\mathbf{x})$ from Eq. (4) is

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

The prediction of the Level Criterion LC_f is given by

$$LC_{\hat{y}}(\mathbf{x}) = \hat{y}(\mathbf{x}) + \frac{1}{2}tr(\mathbb{H}_{\hat{y}}(\mathbf{x})\Delta) = B\hat{z}(\mathbf{x}). \tag{7}$$

where $B = (1, 0, \cdots, 0, \frac{\delta_1^2}{2}, 0 \cdots, 0, \frac{\delta_p^2}{2}, 0, \cdots, 0, \frac{\delta_p^2}{2}, 0, \cdots, 0)$. This formula corresponds to the exact conditional expectation since LC_f is linear in the derivatives.

The prediction of $RC_f(\mathbf{x})$ is defined by:

$$RC_{\hat{y}}(\mathbf{x}) = tr(\nabla_{\hat{y}}(\mathbf{x})\nabla_{\hat{y}}(\mathbf{x})'\Delta) + \frac{1}{2}tr((\mathbb{H}_{\hat{y}}(\mathbf{x})\Delta)^2) \tag{8}$$

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

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

$$\text{Find the Pareto set } \mathbb{X}_0, \quad \text{the solution of the following multi-objective optimization } \min_{\mathbf{x}\in\mathbb{R}^p}\{LC_{\hat{y}}(\mathbf{x}), RC_{\hat{y}}(\mathbf{x})\} \tag{9}$$

Remarks:

- The definition of the predicted robustness criterion corresponds to the definition of Eq. (3) 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 **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 **x** is:

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

where $\min(y(\mathbb{X})) = \min(y^1, ..., y^n)$.

The analytical expression of the EI for a Gaussian process is given by:

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

where $\hat{y}(\mathbf{x})$ is the kriging mean, $\hat{s}(\mathbf{x})$ 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 level and robustness criteria,
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 processes $(LC_Y(\mathbf{x}))_{\mathbf{x}\,\in\,D}$ and $(RC_Y(\mathbf{x}))_{\mathbf{x}\,\in\,D}$. From Eqs. (2) and (3), the processes are naturally defined by:

$$LC_Y(\mathbf{x}) = Y(x) + \frac{1}{2}tr(\mathbb{H}_Y(\mathbf{x})\Delta) \tag{10}$$

$$RC_Y(\mathbf{x}) = tr(\nabla_Y(\mathbf{x})\nabla_Y(\mathbf{x})'\Delta) + \frac{1}{2}tr((\mathbb{H}_Y(\mathbf{x})\Delta)^2) \tag{11}$$

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

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 LC_f and RC_f are then as follows:

$$\begin{aligned} EI_{LC_y}(\mathbf{x}) &= \mathbb{E}[(\max(LC_y(\mathbb{X}^*)) - LC_Y(\mathbf{x}))^+ | Z(\mathbb{X}) = \mathbf{z}_{u_{obs}}] \\ EI_{RC_y}(\mathbf{x}) &= \mathbb{E}[(\max(RC_y(\mathbb{X}^*)) - RC_Y(\mathbf{x}))^+ | Z(\mathbb{X}) = \mathbf{z}_{u_{obs}}] \end{aligned}$$

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

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

$$\text{Find the Pareto set } \mathbb{X}_0, \quad \text{the solution of the following multi-objective optimization} \\ \min_{\mathbf{x}\in\mathbb{R}^p} \{EI_{LC_y}(\mathbf{x}), EI_{RC_y}(\mathbf{x})\} \tag{12}$$

Remarks:

- A solution $\mathbf{x}^1$ dominates another solution $\mathbf{x}^2$ for the m objectives $g_1, ..., g_m$ if and only if $\forall\, i \in \{1, ..., m\}$ $g_i(\mathbf{x}^1) \leq g_i(\mathbf{x}^2)$ and $\exists\, i \in \{1, ..., m\}$ $g_i(\mathbf{x}^1) < g_i(\mathbf{x}^2)$. 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 level and the robustness criteria are not observed, $\max(LC_y(\mathbb{X}^*))$ and $\max(RC_y(\mathbb{X}^*))$ are predicted by kriging.
- As the process $LC_Y(\mathbf{x})$ is Gaussian, the expression of EI_{LC_y} is

$$EI_{LC_y}(\mathbf{x}) = (\max(LC_y(\mathbb{X}) - LC_{\hat{y}}(\mathbf{x}))\Phi\left(\frac{\max(LC_y(\mathbb{X})) - LC_{\hat{y}}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) \\ + \hat{s}(\mathbf{x})\phi\left(\frac{\max(LC_y(\mathbb{X})) - LC_{\hat{y}}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right)$$

  where $\hat{s}(\mathbf{x})$ is the kriging variance.
- 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 $\mathbf{X} = (\mathbf{x}^{n+1}, ..., \mathbf{x}^{n+q})'$ [29]. In a multi-objective context, the expressions of the q-EI are:

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

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

We note that q-EI involves $\min(LC_Y(\mathbf{x}^{n+1}),\ ..., LC_Y(\mathbf{x}^{n+q}))$ instead of $LC_Y(\mathbf{x}^{n+1})$. 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:

$$\text{Find the Pareto set } \mathbb{X}_0, \quad \text{the solution of the following multi-objective optimization} \\ \min_{\mathbf{X}\in\mathbb{R}^{p\times q}} \{-qEI_{LC_y}(\mathbf{X}), -qEI_{RC_y}(\mathbf{X})\}$$

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: $-EI_{LC_y}$ 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 $\{-EI_{LC_y}, -EI_{RC_y}\}$ 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 $-\,qEI_{LC_y} - qEI_{RC_y}$.

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 $-EI_{LC_y}$ 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. $z(\mathbf{x}_0^1)$ is then considered known and is assumed to be equal to $\hat{z}(\mathbf{x}_0^1)$. Another computation of $-EI_{LC_y}$ 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 $-EI_{LC_y}$ 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. $LC_y(\mathbf{x}_0^1)$ (resp. $RC_y(\mathbf{x}_0^1)$) is then considered known and is assumed to be equal to

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

$\min LC_y(\mathbb{X}^*)$ (resp. $\min RC_y(\mathbb{X}^*)$). Another computation of $-EI_{LC_y}$ 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.

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 [31–34]). The Inverted Generational Distance (IGD) and the Hypervolume (HV) are selected here to compare strategies. Let $\mathbf{f} = (f_1, \dots, f_m)$ 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.

Table 1
Minimization problems and methods for choosing the interesting points.

Method	Minimization	Interesting points	Updates
MyAlea	LC_y, RC_y	Random points on the Pareto front and the parameter space	Batch
MyEIClust	LC_y, RC_y	Cluster on EI_{LC_y} and EI_{RC_y}	Batch
MyqEI	LC_y, RC_y	Annealing algorithm on qEI_{LC_y} and qEI_{RC_y}	Batch
MyKB	LC_y, RC_y	Kriging believer	Batch
MyCL	LC_y, RC_y	Constant liar	Batch
MEIyAlea	EI_{LC_y}, EI_{RC_y}	Random point on the Pareto front	Seq
MqEIyAlea	qEI_{LC_y}, qEI_{RC_y}	Random point on the Pareto front	Batch

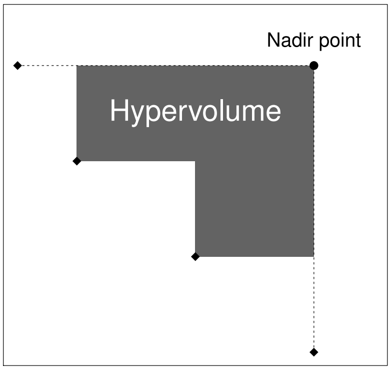


Fig. 2. Diamonds represent the individuals of the empirical Pareto front $\mathbb{X}^*$. The black circle is the Nadir point of the set $\mathbb{X}^*$.

- Hypervolume (HV) see [34]. Fig. 2 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 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 subject to manufacturing errors with $\delta_j = \frac{0.05}{4}(\max(x_j) - \min(x_j))$, $j = \{1, 2\}$.

Fig. 3 shows the four optimal areas for robust optimization in the objective and parameter space.

In order to perform a robust optimization, the function and all the first and second derivatives need to be predicted. The set of predicted indexes is $u_{pred} = \{1, \dots, 6\}$ and corresponds to the following vector:

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

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 $u_{obs} = \{1, \dots, 6\}$ that corresponds to the process vector:

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

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 Fig. 4 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: MqEIyAlea is really time-consuming and the metrics IGD and HV have not yet converged even if the number of

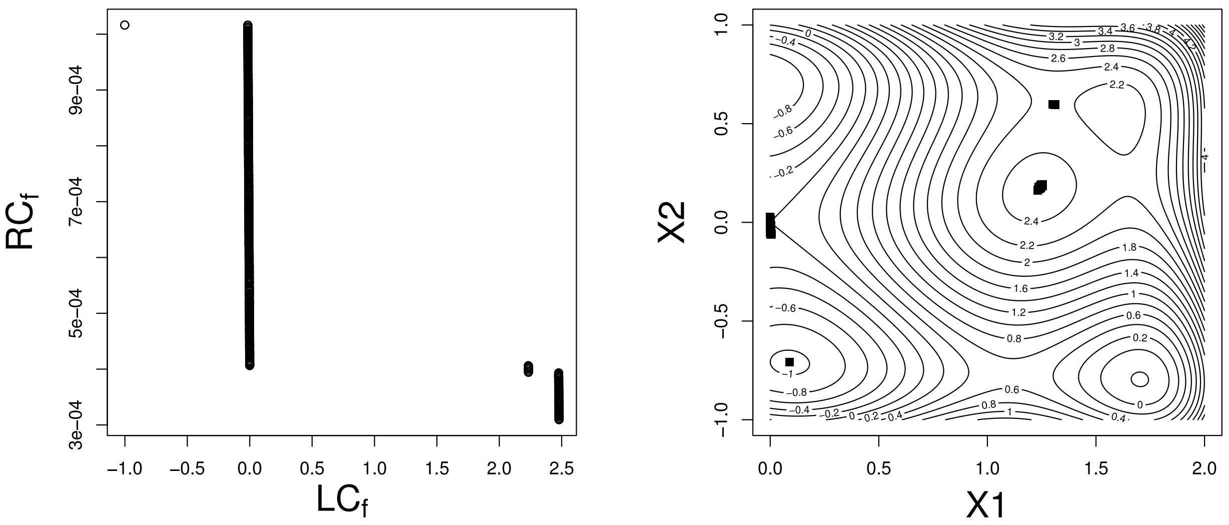


Fig. 3. Pareto front of the six-hump Camel function in the objective space (left) and in the parameter space (right).

found areas is the highest. MEIyAlea gives the worst results in terms of metrics and found areas with a high computation time. Among the five other methods, MyqEI, MyCL and MyKB give the best compromises in terms of metrics, areas and computation time.They are fast as they are based on the response surface which is easy to compute. They find the different areas due to an efficient exploration step. Fig. 5 shows the boxplots of these three methods for each distance. MyqEI gives the worst results in HV metric. 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.

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.

Method	Updates	Computation time	Nb areas
MyAlea	Batch	2 min	2.60
MyEIClust	Batch	2 min	2.96
MyqEI	Batch	6 min 45 sec	3.22
MyKB	Batch	3 min	3.19
MyCL	Batch	3 min	3.28
MEIyAlea	Seq	1 h 21 min	1.83
MqEIyAlea	Batch	3 h 16 min	3.63

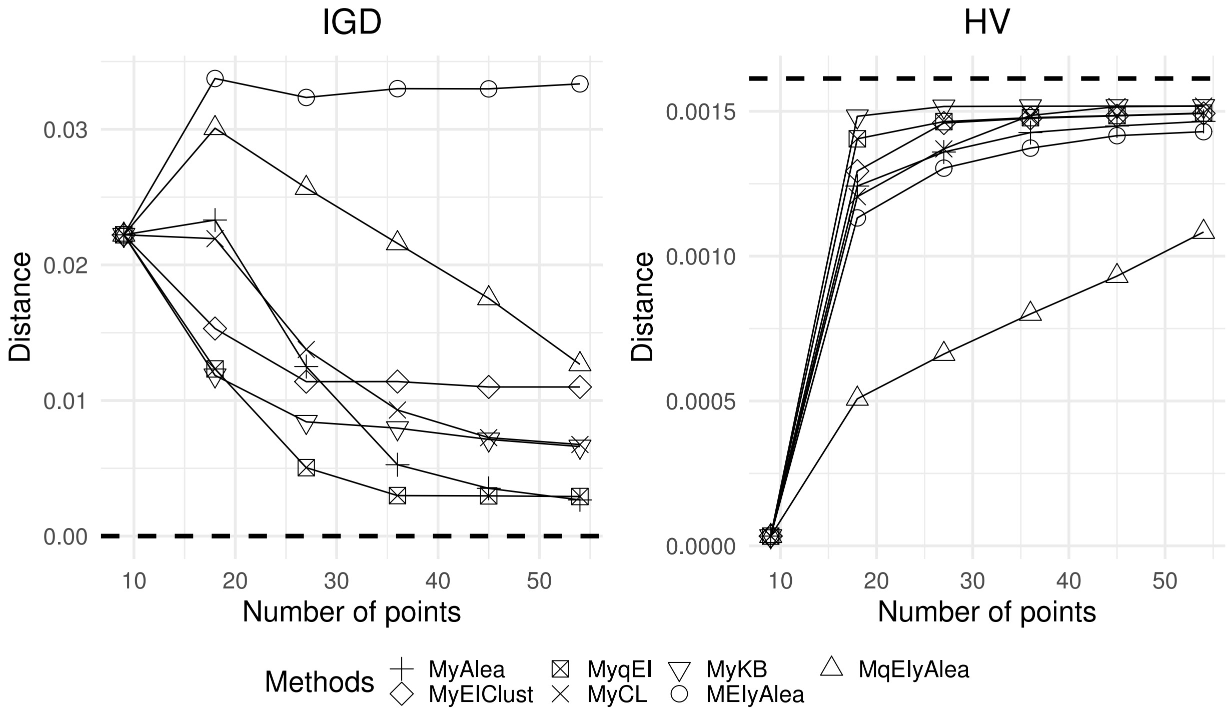
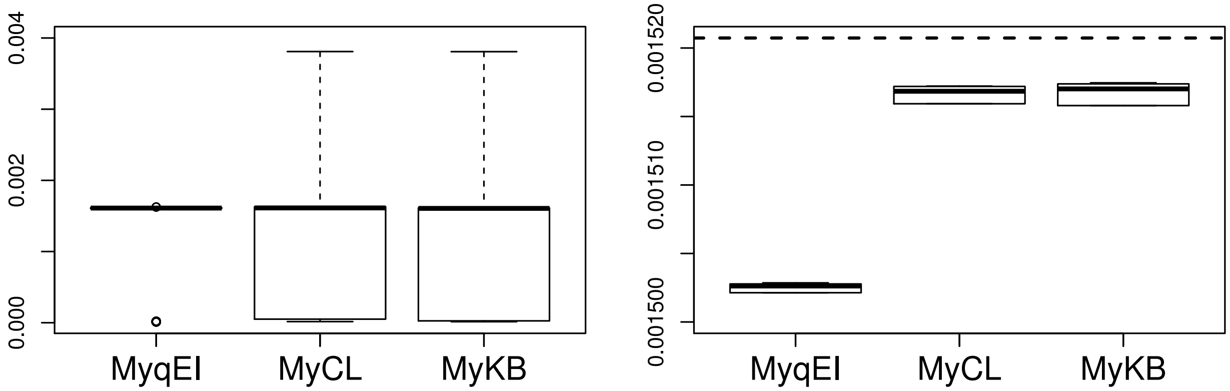


Fig. 4. Six-hump Camel function with derivative observations. Evolution of the Pareto metrics with the number of points computed for all the methods over 100 different runs of the algorithm. The HV value of the theoretical front is represented by the dotted line.



(a) IGD metric

(b) HV metric

Fig. 5. Boxplots of the metrics computed for the three best methods over 100 simulations for the six-hump Camel function with derivative observations.

The observed set of indexes is $u_{obs} = \{1\}$, while the predicted set of indexes is $u_{pred} = \{1, \dots, 6\}$ that corresponds to the process vectors:

$$\begin{aligned} Z_{u_{obs}} &= Y \\ Z_{u_{pred}} &= \left(Y,\ Y_{x_1},\ Y_{x_2},\ Y_{x_1,x_2},\ Y_{x_1,x_1},\ Y_{x_2,x_2}\right) \end{aligned}$$

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 Fig. 6 and Table 3. Our analysis is as follows: the six-hump Camel function is difficult to approximate without derivative observations. MqEIyAlea gives the best results, but it is far too

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.

Method	Updates	Computation time	Nb areas
MyAlea	Batch	14 min	3.69
MyEIClust	Batch	42 min	2.20
MyqEI	Batch	1 h 08	2.22
MyCL	Batch	19 min	2.85
MyKB	Batch	19 min	2.29
MEIyAlea	Seq	8 h 41	1.68
MqEIyAlea	Batch	26 h 47	3.94

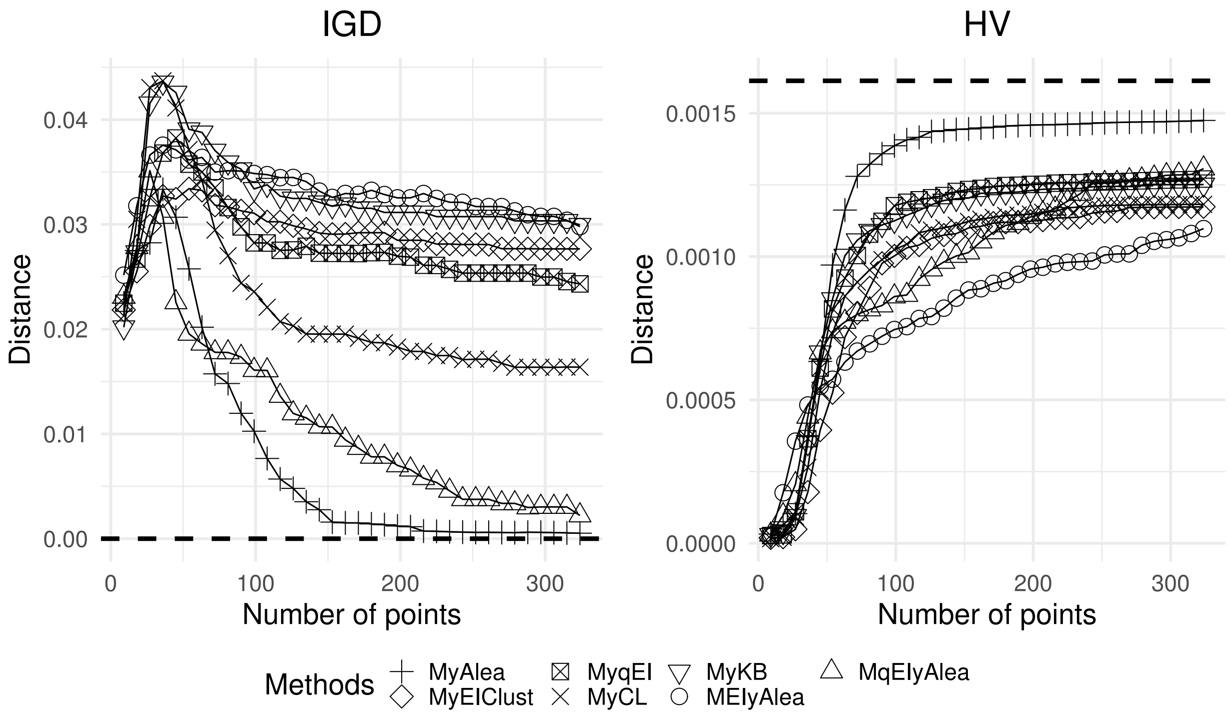


Fig. 6. Six-hump Camel function without derivatives' observations. Evolution of the Pareto metrics with the number of points compute for all the methods over 100 different runs of the algorithm. The HV value of the theoretical front is represented by the dotted line.

time-consuming. MyAlea strategy, which is partially based on a random search, gives comparable results. In this context, too much reliance should not be placed upon kriging. On the contrary, MEIyAlea provides bad results by any criterion. The MyCL and MyKB, which exploit the kriging variance, also give good results. Finally, the MyEIClust, MyqEI and strategies that use the EI criterion provide less accurate results. The best strategy is MyAlea, but MyKB and MyCL are also retained to test them in a higher dimension.

7.2. Hartmann function: 6D

In this section, the three best strategies (MyqEI, MyKB and MyCL) 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 Eq. (5)). The studied function is the six-dimensional Hartmann function defined by:

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

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 two input variables x_4 and x_5 are assumed to vary in the interval $x_j \pm 2\delta_j$ where $\delta_j = \frac{0.05}{4}(\max(x_j) - \min(x_j))$, $j = \{1, 2\}$. 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:

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

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 right part of Fig. 7 shows that the three methods converge to the true front. MyqEI is the fastest. On the simulation presented on the left part of Fig. 7, it is the only method which finds at step 5 the all Pareto front. MyKB takes 8 minutes and MyCL takes 9 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 $u_{pred} = \{1, 5, 6, 20, 26, 27\}$.

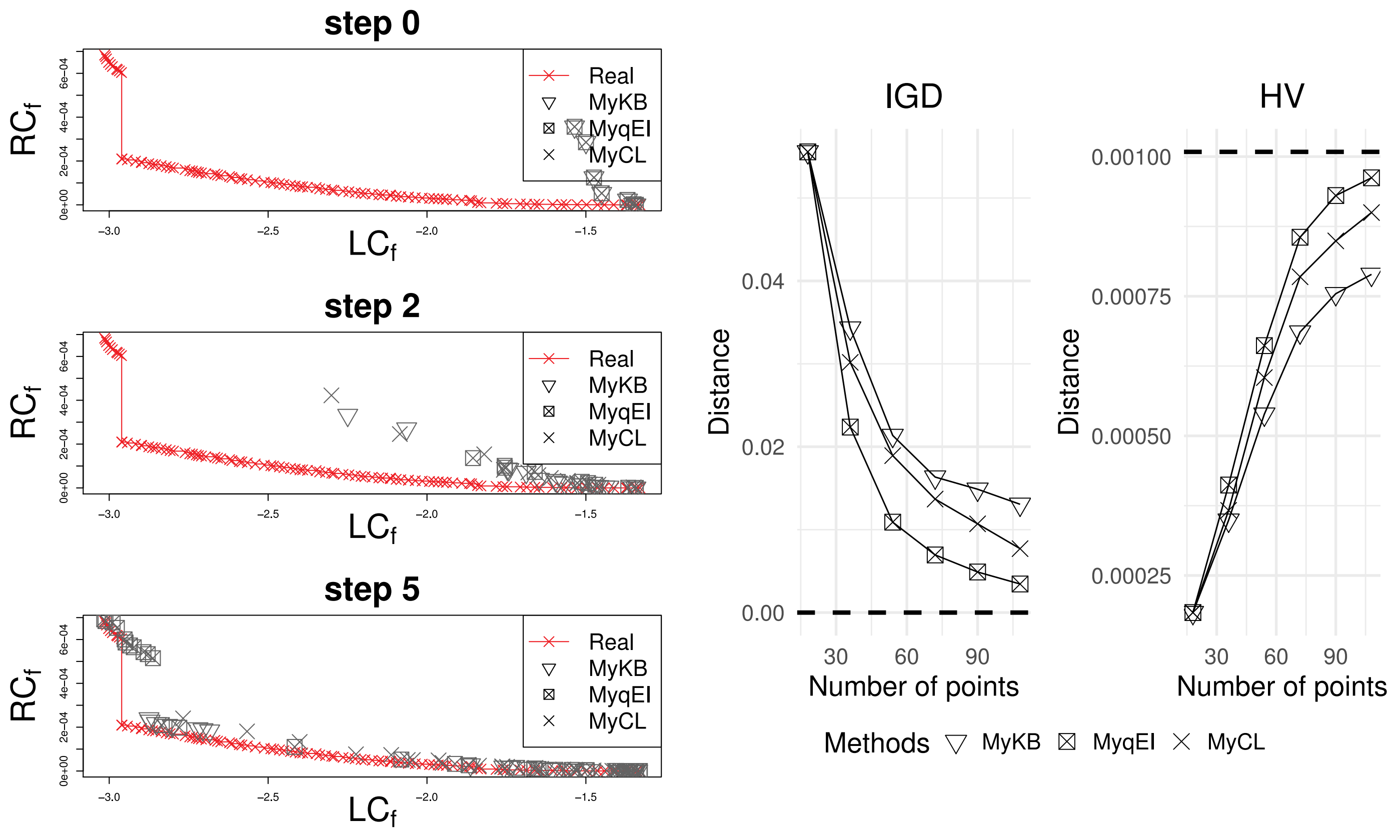


Fig. 7. On the left: Pareto fronts obtained during the optimization procedure of the three strategies at the initial step (step 0), middle step (step 2) and final step (step 5). On the right: evolution of the metrics computed during the algorithm for all the methods over 100 simulations for the Hartmann function with derivative observations. The HV value of the theoretical front is represented by the dotted line.

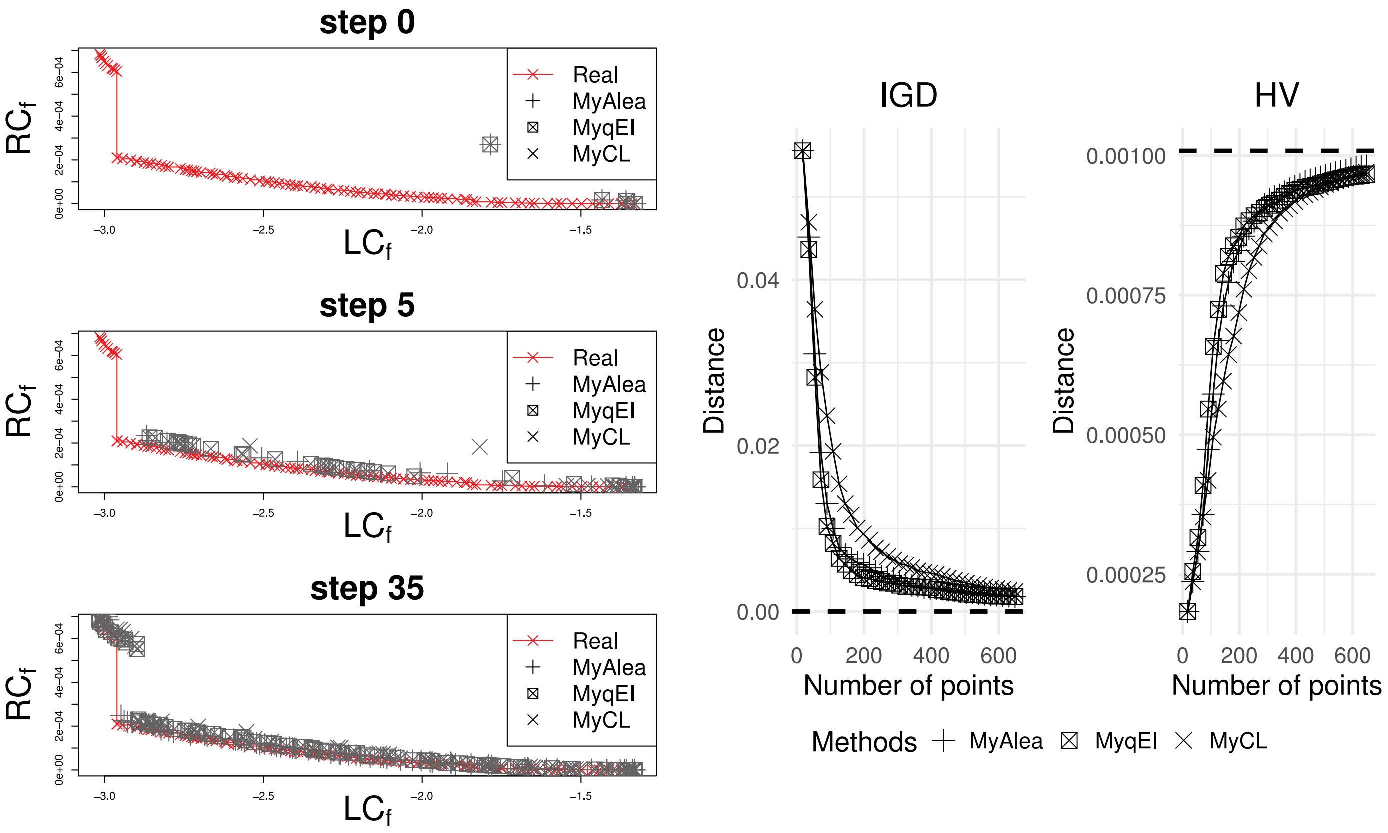


Fig. 8. On the left: Pareto fronts obtained during the optimization procedure of the three strategies at the initial step (step 0), step 5 and final step (step 35). On the right: evolution of the metrics during the algorithm computed for all the methods in 100 simulations for the Hartmann function with no derivative observation. The HV value of the theoretical front is represented by the dotted line.

They correspond to the process vectors:

$$\begin{aligned} Z_{u_{obs}} &= Y \\ Z_{u_{pred}} &= \left(Y,\ Y_{x_4},\ Y_{x_5},\ Y_{x_4,x_5},\ Y_{x_4,x_4},\ Y_{x_5,x_5}\right) \end{aligned}$$

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.

Fig. 8 shows that the three methods converge to the true front. At step 5, only one part of the front has been detected by two of the methods (MyCl and MyqEI). The top left part of the front is difficult to localize. On this simulation at step 35 it is reached (with 578 additional points). The right part of Fig. 8 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 56 min, MyqEI takes 1h19 min, and MyCL takes 1h02min for the 35 steps.

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.

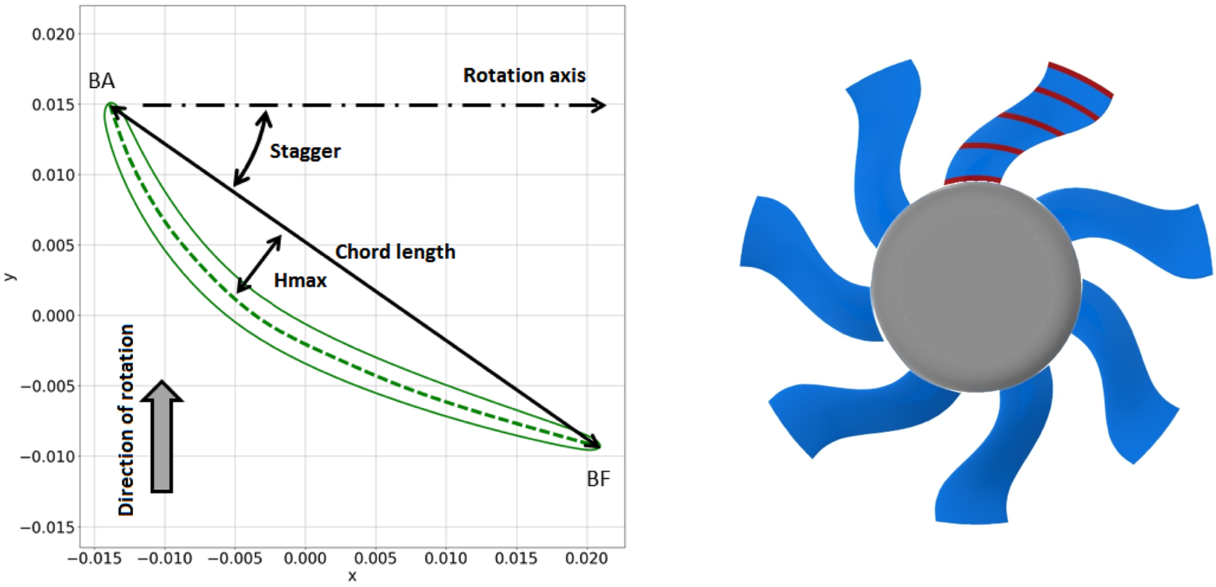


Fig. 9. Blade section with the three input parameters on the left. Sections are represented on the right by the red lines along one blade. Section 1 is the closest to the disc and section 5 the most far away.

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 Fig. 9 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 Fig. 9 on the left. As a result, each fan blade is

Table 5
Fixed physical parameters and values

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

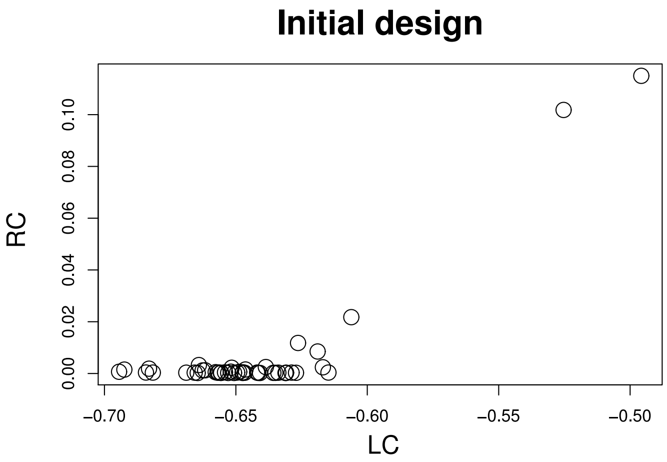


Fig. 10. The 46 initial observation points in the true objective space: opposite level criterion ($-LC_\eta$) and robustness criterion calculated on the efficiency (RC_η).

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.

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

	Update	1	2	3	4	5	Total
MyKB	Time	0h21	0h35	0h 48	0h58	1h15	3h57
MyqEI	Time	0h08	0h13	0h 19	0h24	0h32	1h36
MyCL	Time	0h19	0h29	0h 42	0h56	1h26	3h52

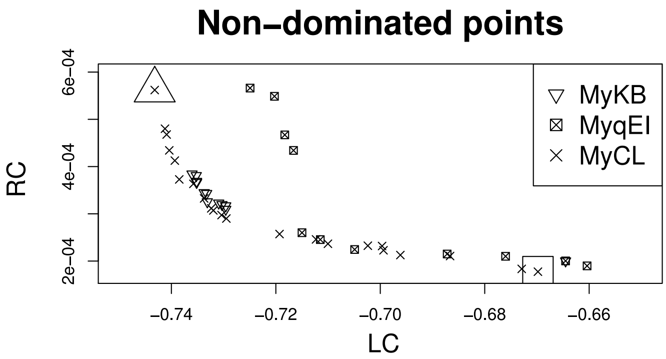


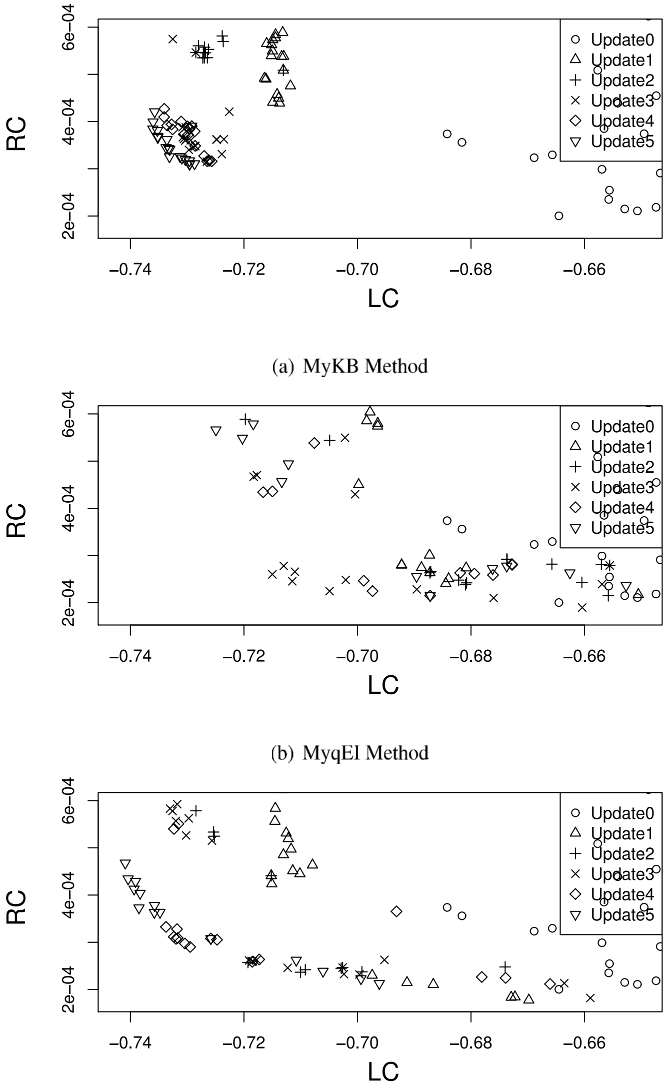
Fig. 11. Non-dominated points of the final design for methods MyCL, MyKB and MyqEI in the true objective space: opposite level criterion (– LC_η) and robustness criterion calculated on the efficiency (RC_η).

characterized by fifteen geometrical parameters, namely five chord, stagger 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.

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

$$\eta = \frac{Q \times \Delta P}{C \times \Omega} \tag{13}$$

As the rotational speed Ω ($rad.\ s^{-1}$) and the volume flow Q ($m^3.\ s^{-1}$) are fixed, fan efficiency η (.) depends on two outputs of *TurboConcept™*. 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.



(a) MyKB Method

(b) MyqEI Method

(c) MyCL Method

Fig. 13. Progression of the algorithm for methods MyKB (a), MyqEI (b), MyCL (c) in the true objectives space: opposite level criterion (– LC_η) and robustness criterion calculated on the efficiency (RC_η).

7.3.3. Results

The initial sample set D_1 is a maximin LHS of 46 observations. Fig. 10 shows the learning sample set in the true objective space $\{LC_\eta, RC_\eta\}$. LC_η and RC_η are the level and robustness criteria calculated on η given by Eq. (2) and (3). The total budget is composed of 136 points, and 90 points are added to the initial design with 5 updates of 18 points. The three best methods (MyKB, MyqEI and MyCL) used in Section 7.2.1 are selected to perform robust optimization.

As it can be seen MyKB, MyqEI and MyCL have well progressed

Fig. 12. The shape on the left corresponds to the square of Fig. 11, while that on the right to the triangle.

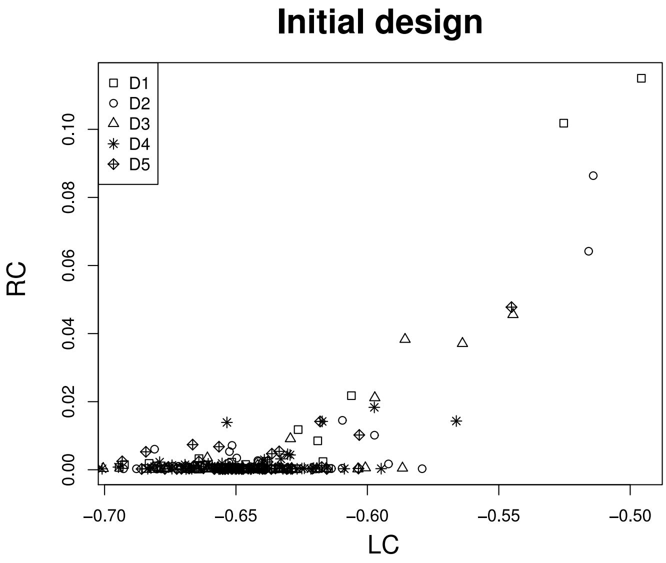


Fig. 14. The initial observation points for five LHS designs in the true objective space: opposite level criterion (− LC_η) and robustness criterion calculated on the efficiency (RC_η).

between the initial step (cf Fig. 10) and the final step (cf Fig. 11). They gives interesting non-dominated points that are compromises between efficiency and robustness. MyqEI provides the worst progress in the objective space, while MyKB gives the less dispersed areas and MyCL the most advanced and dispersed ones. These differences stem form where strategies add points along updates. As it can be seen on Fig. 13, all the points added by MyKB are in the same area (middle left). MyqEI adds points in different areas from the first update but at the end, it has not finished the progression. MyCL progresses better than the others. Since the second update, it puts a point in the bottom right area. Table 6 shows that MyqEI is the fastest strategy. Indeed MyCL and MyKB need q-updates of the kriging model to select the batches.

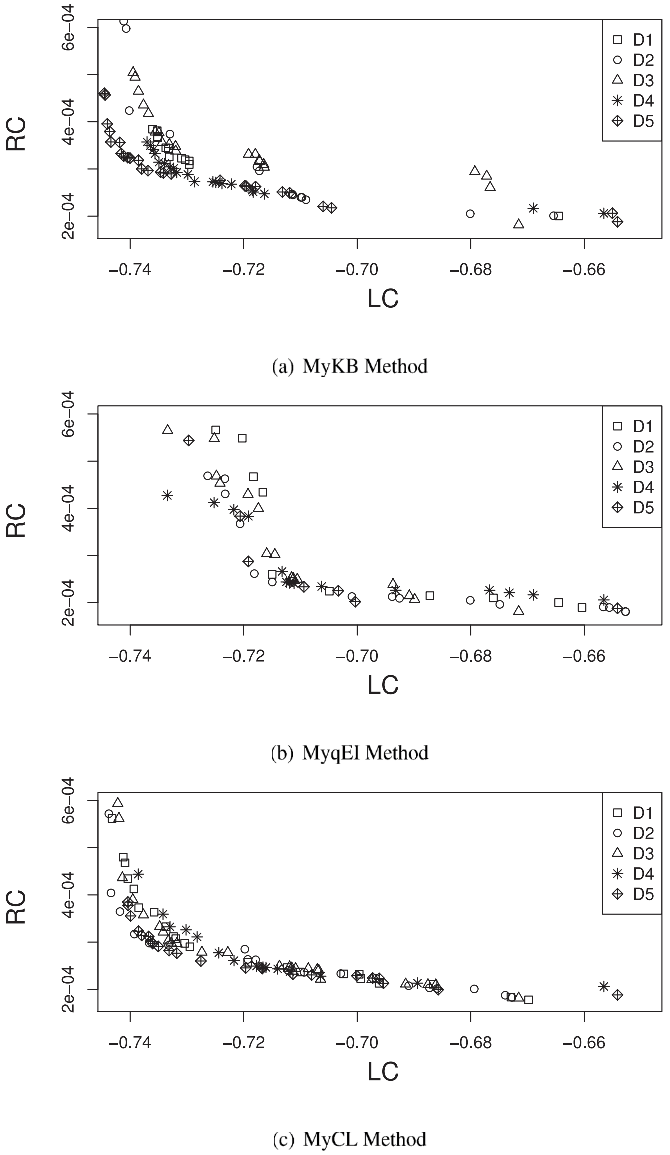
The shapes of two of these compromises (see the big square and triangle on Fig. 11) are represented on Fig. 12.

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 *TurboConcept*^{TM}. Fifty points were launched but not all of them converged. Nevertheless, at the end of the optimization all designs have 136 points. Fig. 14 shows the five sets of learning samples in the true objective space $\{LC_\eta, RC_\eta\}$.

Fig. 15 shows that the five different initial designs converge to the same Pareto front with the three methods. The choice of the optimized initial LHS has little impact on the final result. Moreover, the MyCL method is the most reliable, as the five Pareto fronts coincide perfectly.

8. Conclusion

In this paper, an efficient kriging-based robust optimization procedure is proposed. The methodology is based on a multi-objective optimization of the two first moments (expectation and variance). Proxies are given based on a Taylor expansion and for Gaussian errors. These expressions using derivatives have 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 kriging response surfaces or kriging expected improvements and relevant enrichment methods composed of one point or a batch of points carefully chosen on the Pareto front. Seven strategies have been compared on two toy functions. The study reveals that it is far more computer-wise efficient 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.



(a) MyKB Method

(b) MyqEI Method

(c) MyCL Method

Fig. 15. Non-dominated points of the five final designs for methods MyKB (a), MyqEI (b), MyCL (c) in the true objective space: opposite level criterion (− LC_η) and robustness criterion calculated on the efficiency (RC_η).

Declaration of Competing Interest

The authors declare that they do not have any financial or non-financial conflict of interests.

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.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ress.2020.106913

References

[1] Santner TJ, Williams BJ, Notz WI. The design and analysis of computer experiments. Springer Series in Statistics Springer-Verlag, New York0-387-95420-1; 2003. https://doi.org/10.1007/978-1-4757-3799-8.

[2] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Global Optim 1998;13(4):455–92.

[3] Lelièvre N, Beaurepaire P, Mattrand C, Gayton N, Otsmane A. On the consideration of uncertainty in design: optimization - reliability - robustness. Struct Multidiscip Optim 2016;54(6):1423–37. https://doi.org/10.1007/s00158-016-1556-5.

[4] Janusevskis J, Le Riche R. Simultaneous kriging-based estimation and optimization of mean response. J Global Optim 2013;55(2):313–36.

[5] Marzat J, Walter E, Piet-Lahanier H. Worst-case global optimization of black-box functions through kriging and relaxation. J Global Optim 2013;55(4):707–27.

[6] Apley DW, Liu J, Chen W. Understanding the effects of model uncertainty in robust design with computer experiments. J Mech Des 2006;128(4):945–58.

[7] Ur Rehman S, Langelaar M, van Keulen F. Efficient kriging-based robust optimization of unconstrained problems. J Comput Sci 2014;5(6):872–81.

[8] Göhler SM, Eifler T, Howard TJ. Robustness metrics: Consolidating the multiple approaches to quantify robustness. J Mech Des 2016;138(11):111407.

[9] Gabrel V, Murat C, Thiele A. Recent advances in robust optimization: an overview. Eur J Oper Res 2014;235(3):471–83.

[10] Coco AA, Solano-Charris EL, Santos AC, Prins C, de Noronha TF. Robust optimization criteria: state-of-the-art and new issues. Technical Report UTT-LOSI-14001, ISSN: 2266-5064 2014.

[11] Troian R, Shimoyama K, Gillot F, Besset S. Methodology for the design of the geometry of a cavity and its absorption coefficients as random design variables under vibroacoustic criteria. J Comput Acoust 2016;24(02):1650006.

[12] Darlington J, Pantelides C, Rustem B, Tanyi B. An algorithm for constrained nonlinear optimization under uncertainty. Automatica 1999;35(2):217–28. https://doi.org/10.1016/S0005-1098(98)00150-2. https://www-sciencedirect-com.bibliopam.ec-lyon.fr/science/article/pii/S0005109898001502

[13] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Adaptive Computation and Machine Learning MIT Press, Cambridge, MA978-0-262-18253-9; 2006.

[14] Le Gratiet L. Multi-fidelity gaussian process regression for computer experiments. Université Paris-Diderot-Paris VII; 2013. Ph.D. thesis.

[15] Wagner T, Emmerich M, Deutz A, Ponweiser W. On expected-improvement criteria for model-based multi-objective optimization. International conference on parallel problem solving from nature. Springer; 2010. p. 718–27.

[16] Knowles J. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans Evolut Comput 2006;10(1):50–66.

[17] Liu W, Zhang Q, Tsang E, Liu C, Virginas B. On the performance of metamodel assisted moea/d. International symposium on intelligence computation and applications. Springer; 2007. p. 547–57.

[18] Zhang Q, Liu W, Tsang E, Virginas B. Expensive multiobjective optimization by moea/d with gaussian process model. IEEE Trans Evolut Comput 2010;14(3):456–74.

[19] Ponweiser W, Wagner T, Biermann D, Vincze M. Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection. International conference on parallel problem solving from nature. Springer; 2008. p. 784–94.

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

[21] Emmerich MT, Deutz AH, Klinkenberg JW. Hypervolume-based expected improvement: Monotonicity properties and exact computation. Evolutionary computation (CEC), 2011 IEEE congress on. IEEE; 2011. p. 2147–54.

[22] Svenson J, Santner T. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. Comput Stat Data Anal 2016;94:250–64.

[23] Picheny V. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. Stat Comput 2015;25(6):1265–80.

[24] Henkenjohann N, Kunert J. An efficient sequential optimization approach based on the multivariate expected improvement criterion. Qual Eng 2007;19(4):267–80.

[25] Jeong S, Obayashi S. Efficient global optimization (ego) for multi-objective problem and data mining. Evolutionary computation, 2005. The 2005 IEEE congress on. 3. IEEE; 2005. p. 2138–45.

[26] Pronzato L, Éric Thierry. Robust design with nonparametric models: prediction of second-order characteristics of process variability by kriging1. IFAC Proc. Vol. 2003;36(16):537–42. https://doi.org/10.1016/S1474-6670(17)34817-6. 13th IFAC Symposium on System Identification (SYSID 2003), Rotterdam, The Netherlands, 27-29 August, 2003; http://www.sciencedirect.com/science/article/pii/S1474667017348176

[27] Stein ML. Interpolation of spatial data. Springer Series in Statistics Springer-Verlag, New York0-387-98629-4; 1999. https://doi.org/10.1007/978-1-4612-1494-6. Some theory for Kriging

[28] Dupuy D, Helbert C, Franco J, et al. Dicedesign and diceeval: two r packages for design and analysis of computer experiments. J Stat Softw 2015;65(11):1–38.

[29] Ginsbourger D, Le Riche R, Carraro L. Kriging is well-suited to parallelize optimization. Computational intelligence in expensive optimization problems. Springer; 2010. p. 131–62.

[30] Hartigan JA, Wong MA. Algorithm as 136: a k-means clustering algorithm. J R Stat Soc Ser C (Appl Stat) 1979;28(1):100–8.

[31] Van Veldhuizen DA. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Wright Patterson AFB, OH, USA; 1999. Ph.D. thesis. AAI9928483

[32] Schott JR. Fault tolerant design using single and multicriteria genetic algorithm optimization. Tech. Rep.. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH; 1995.

[33] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evolut Comput 2002;6(2):182–97.

[34] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evolut Comput 1999;3(4):257–71. https://doi.org/10.1109/4235.797969.

[35] Coello CAC, Sierra MR. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. Mexican international conference on artificial intelligence. Springer; 2004. p. 688–97.

[36] Fonseca CM, Paquete L, López-Ibáñez M. An improved dimension-sweep algorithm for the hypervolume indicator. Proceedings of the 2006 congress on evolutionary computation (CEC 2006). Piscataway, NJ: IEEE Press; 2006. p. 1157–63. https://doi.org/10.1109/CEC.2006.1688440.

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

[38] Ottavy X. Cours de turbomachine à fluide compressible. Laboratoire de Mécanique des Fluides et d’Acoustique; 2007https://www.researchgate.net/publication/271191275_cours_Turbomachines.

[39] Ottavy X. Basic design of fans and compressors - endwall flows and losses. Laboratoire de Mécanique des Fluides et d’Acoustique; 2015. https://doi.org/10.13140/RG.2.2.33942.65608https://www.researchgate.net/publication/325645425_Basic_design_of_fans_and_compressors_-_Endwall_flows_and_losses.

[40] Rendu Q. Modélisation des écoulements transsoniques décollés pour l’étude des intéractions fluide-structure. Université Claude Bernard Lyon 1; 2016. Ph.D. thesis. Thèse de doctorat dirigée par Jacob, Marc C. et Aubert, Stéphane Mécanique des fluides Lyon 2016; http://www.theses.fr/2016LYSE1328