Surrogate-Assisted Bounding-Box Approach Applied to Constrained Multi-Objective Optimisation Under Uncertainty

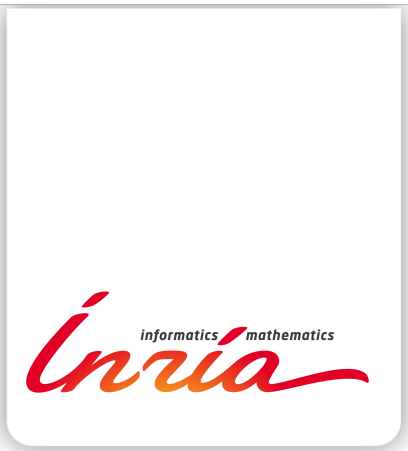
Mickael Rivier, Pietro Marco Congedo

Surrogate-Assisted Bounding-Box Approach Applied to Constrained Multi-Objective Optimisation Under Uncertainty

Mickaël Rivier , Pietro Marco Congedo

RESEARCH REPORT
N° 9214
October 17, 2018
Project-Teams CARDAMOM

ISSN 0249-6399 ISRN INRIA/RR--9214--FR+ENG



Surrogate-Assisted Bounding-Box Approach Applied to Constrained Multi-Objective Optimisation Under Uncertainty

Mickaël Rivier *†, Pietro Marco Congedo *‡

Project-Teams CARDAMOM

Research Report n° 9214 — October 17, 2018 — 48 pages

Abstract: This paper is devoted to tackling constrained multi-objective optimisation under uncertainty problems. In particular, the SABBa (Surrogate-Assisted Bounding-Box approach) framework is applied and extended to handle both robust and reliability-based constrained optimisation problems. This approach aims at efficiently dealing with uncertainty-based optimisation problems, with approximated robustness and reliability measures. A Bounding-Box (or conservative box) is defined as a multi-dimensional product of intervals centred on approximated objectives and constraints and containing the underlying true values. In SABBa, this approach is supplemented with a surrogate-assisting strategy, which is very effective to reduce the overall computational cost, notably during the last iterations of the optimisation. Intuitively, the algorithm applies an uncertainty quantification (UQ) refinement only on promising designs, which allows quick convergence toward the most promising area. The surrogate-assisting strategy model is then automatically refined and used at a negligible computational cost.
The focus of this work is on the formulation of an efficient SABBa framework to tackle robust and reliability-based optimisation problems. For this purpose, we propose a Constrained Boxed Pareto Dominance, and a Pareto Optimal Probability (POP) is computed for each box. We propose some specific choices for the surrogate-modelling method, the conservative error computation and the refinement criteria. Additional features such as coupled design-uncertainty space modelling, reuse strategy and Gaussian Processes (GP)-based box size are explored, compared and discussed.
We perform several analytical and applicative robust and reliability-based optimisation test cases comparing the proposed approach with the nested-loop strategy and a decoupling approach based on metamodels built *a priori*. We assess the performances using a probabilistic modified Hausdorff distance to the Pareto optimal set.

Key-words: Multi-objective optimisation, Uncertainty-based optimisation, Error bounding boxes, Imprecise Pareto front, Surrogate-assisting strategy

* Inria Bordeaux Sud-Ouest - Team CARDAMOM
† ArianeGroup, Le Haillan
‡ DeFI - CMAP - Ecole Polytechnique, Inria Saclay - Ile de France, Polytechnique - X, CNRS

RESEARCH CENTRE BORDEAUX – SUD-OUEST

351, Cours de la Libération
Bâtiment A 29
33405 Talence Cedex

Le framework SABBa appliqué à l'optimisation contrainte multi-objectif sous incertitudes

Résumé : Ce papier s'attaque aux problème d'optimisation contrainte multi-objectif sous incertitudes. Le framework SABBa (Surrogate-Assisted Bounding-Box approach) est appliqué et étendu aux problèmes d'optimisation robuste et sous contraintes fiabilistes. Cette approche vise à traiter efficacement les problèmes d'optimisation sous incertitudes, dont les mesures de robustesse et de fiabilité sont approximées. Une Boîte d'erreur (boîte conservative) est définie comme un produit multi-dimensionnel d'intervalles centrés sur les objectifs et contraintes approximés, contenant les valeurs exactes sous-jacentes. Dans SABBa, cette approche est couplée à une stratégie d'assistance par modèle de substitution, qui permet de réduire sensiblement le coût global de l'étude, notamment dans les dernières itérations d'optimisation. Intuitivement, l'algorithme raffine la quantification d'incertitudes (UQ) seulement sur les designs les plus prometteurs, permettant une rapide convergence vers la zone optimale. Le modèle de substitution est alors raffiné pour les itérations suivantes.

L'accent est ici mis sur les développements permettant une meilleure efficacité du framework SABBa dans le cadre des problèmes d'optimisation robuste et fiable. Pour cela, nous proposons une règle de domination de Pareto pour boîtes d'erreur sous contraintes ainsi qu'un calcul de la probabilité de Pareto optimalité (POP) de chaque boîte. Les choix du modèle de substitution, du calcul de l'erreur et du critère de raffinement sont aussi exposés. Enfin, quelques stratégies supplémentaires, telles que l'utilisation de l'espace couplé design-incertitudes pour la quantification d'incertitudes, une stratégie de réutilisation des sorties et une taille de boîte basée sur les modèles par processus Gaussiens sont explicitées, comparées et analysées.

Cette approche est appliquée sur des cas-tests d'optimisation robuste et fiable analytiques et d'ingénierie et comparée à la stratégie double-boucle et à une méthode découplée par génération *a priori* d'une surface de réponse en espace couplé. Les performances sont exprimées en proposant une distance de Hausdorff modifiée probabiliste à l'ensemble des designs Pareto optimaux.

Mots-clés : Optimisation multi-objectif, Optimisation sous incertitudes, Boîte d'erreur conservative, Front de Pareto imprécis, Assistance par modèle de substitution

1 Introduction

The topic of optimisation under uncertainty is of particular interest for companies nowadays as (i) the always increasing computational power now allows for multiple runs and reanalysis of simulations and because (ii) robustness, reliability and cost optimality are critical factors in assessing the quality of a product and the efficiency of a company.

In this context, two main areas are of primary interest: Robust Optimisation (RO) and Reliability-Based Optimisation (RBO). Robustness definition by Oxford dictionary is "the ability to withstand or overcome adverse conditions or rigorous testing". In other words, it is the ability to guarantee a performance which is less sensitive to off-design conditions, while the definition of reliability is "the quality of being trustworthy or of performing consistently well", meaning that uncertainties in the environment should not lead to the failure of a system.

One of the central issues in conducting such analysis is the computational cost of attaining a prescribed level of accuracy. Most of the time, these optimisations are performed making use of numerical simulations, with run times ranging from minutes to weeks. This prohibits direct simulation-based methods such as nested Monte Carlo Simulations (MCS), performing a full Uncertainty Quantification (UQ) at each optimisation iteration.

Contributions to reliability-based optimisation mostly rely on the computation of failure probability. A classical and widely used approach is the First-Order Reliability Method (FORM). Based on the computation of the Most Probable Failure Point (MPFP), an optimisation of the failure probability is a nested problem, thus very computationally-demanding. Such an approach has been compared to the Performance Measure Approach (PMA) in [1] and decoupled approaches in [2], the later showing significant improvements in the total cost of the reliability-based optimisation. Other decoupled approaches rely on local approximations of the failure probability and the gradient to perform line search optimisation, as in [3], based on Subset Simulation (SS) computations, or in [4], supplemented by a stabilisation strategy.

The idea of using a surrogate model to tackle the computational cost limitation has been extensively used and is usually a prerequisite for the use of Sample Average Approximation (SAA) methods. Model-based RBO is presented in [5], based on two Neural Networks (NN) methodologies. In [6], a non-intrusive and systematic approach for Topology Optimisation (TO) under uncertainty based on Polynomial Chaos Expansion (PCE) is presented. Low-cost computation of small probabilities as well as extreme quantiles with a multi-point enrichment strategy, through the use of PC-Kriging metamodels, is presented in [7]. Finally, in [8], Stepwise Uncertainty Reduction (SUR) strategies for failure probability estimation are proposed, based on the volume of the excursion set.

Another approach to mention is the one in Ref. [9], which aims at dealing with multiple probabilistic constraints by computing "safety factor" constraints, tuned from MCS or SS. A method designed for reliability assessment of dynamic systems is also depicted in [10], which is based on the computation of Extreme Value Distribution (EVD) through an estimator-corrector of the maximum entropy solution constrained by the fractional moments.

In literature, there are also several contributions to the topic of Robust Optimisation, in particular to the topic of structural optimisation and Topological Optimisation (TO). Instead of probabilistic uncertainty, many contributions deal with interval uncertainty. Worst-case optimisation with monotonic assumptions has been performed through min-max compositions [11] or with first-order Taylor approximations [12]. This worst-case paradigm has been extended to TO and is called Robust Topological Optimisation (RTO). Computational burden is reduced in [13] by reanalysis techniques for approximating solutions of similar linear systems. In [14], Chebyshev

transformation allows avoiding large overestimations during interval arithmetic computations.

Taguchi's robustness paradigm also provides a RO formulation. It aims at maximising the performance (or mean performance) while minimising the associated variance. Ref. [13] is an example of this paradigm for RTO problems. In [15], the uncertain problem is transformed into a deterministic one through advanced interval analysis approximations. Intrusive methods such as the stochastic finite element method [16] allow for direct computation of the statistical moments within the incremental path-dependent procedure. Adjoint methods coupled with a gradient descent algorithm in [17] allow for sequential optimisation of each objective, up to convergence to a Nash equilibrium.

To perform mean performance optimisation, Ref. [18] introduces the Stochastic Subset Optimisation (SSO), which is then supplemented by a local optimisation through Simultaneous Perturbation Stochastic Approximation (SPSA). SPSA was also used in [19], coupled with an adaptive Importance Sampling (IS) strategy. In this approach, there are adaptive approximations of both the optimal IS density and the relevant uncertain dimensions.

Surrogate-based methodologies are also massively exploited in the area of Robust Optimisation. A topological optimisation based on PCE in [6] allows for the computation of statistical moments on top of failure probability. In [20], the authors propose a kriging-based Sequential Approximate Optimisation (SAO) strategy. They build a kriging metamodel in a subdomain of the coupled design/uncertain spaces. At each cycle, gradient-based optimisation is performed, with a translation of the subdomain.

A general discussion over the use of metamodels in the context of uncertainty-based optimisation is in [21], which shows that such methods yield promising results, notably the kriging or Gaussian Process (GP) metamodels. In this context, [22] proposes a target performance and variance minimisation RO formulation. They make an approximation of the performance function with a kriging model in the coupled space, and low-cost MCS on the model allows to build a second kriging metamodel, on the statistical moments, to optimise. Quite similarly, Ref. [23] compares the one-layer and two-layer kriging model strategy (1L-KM and 2L-KM). The later shows better results, which encourages modelling in the coupled space. Such a comparison has also been conducted in [24] within the DAKOTA software. We draw similar conclusions from this work.

Some works take full advantage of the nature of Gaussian processes. In [25], the projection of a GP model is conducted analytically to perform mean performance optimisation. Optimisation in the design space is then performed through Bayesian Optimisation (BO). In [26], they exploit the ability of Gaussian process modelling to deal with noisy observations. They build a GP model on the 90% percentile, based on Monte Carlo estimation at some given designs. The associated standard deviation is taken into account through heterogeneous nugget effects. Finally, [27] uses conditional simulations in the context of multi-objective optimisation to sample Pareto optima. Vorob'ev expectation and deviation then allow to quantify and visualise uncertainty on the Pareto front.

More generally, there have been extensions of GP-based optimisation (called BO or Efficient Global Optimisation, EGO) to stochastic or noisy black-box evaluations. For example, in [28], the Sequential Kriging Optimisation (SKO) method extends the famous Expected Improvement (EI) criterion to stochastic systems by minimising a chosen percentile of the GP model. Noisy Expected Improvement, as well as on-line allocation, has been developed in [29]. It proposes a quantile-based EI criterion which takes into account the predictive noise variance of the next observation as well as a finite computational budget to balance exploration and local search accordingly.

In several contributions, [30, 31, 32], the authors introduce the concept of Bounding-Boxes and

the associated Pareto dominance rules. Such ideas have been applied to uncertainty-based optimisation to create a continuous and adaptive multi-fidelity strategy in the uncertainty quantification process. These concepts have been generalised in [33] to refinable noisy computations, and supplemented by a surrogate-assisting strategy to obtain the general and parsimonious SABBa framework.

The main contribution of this paper is the extension of the SABBa framework to the problem of optimisation under uncertainty, allowing to tackle both probabilistic objectives (RO) and constraints (RBO). Parsimony is improved through the use of GP surrogate models, allowing for natural computations of conservative errors and refinement criteria. Moreover, a *Constrained Boxed Pareto Dominance* is proposed to tackle approximated constraints, and the Pareto Optimal Probability (POP) is computed for each box to focus refinement on highly probable non-dominated boxes. Finally, additional features such as metamodeling in the coupled space to make use of the correlations in the design space, iteratively decreasing accuracy thresholds and redefining box sizes based on GP models are proposed and compared. Note that a quantitative indicator is used for all comparisons, based on a probabilistic modified Hausdorff distance explained in the following.

We systematically compare the results to two reference methods: i) The direct nested loop approach, with a full Uncertainty Quantification (UQ) analysis at each optimisation iteration, ii) The so-called A Priori MetaModel (APMM), with a nested loop approach performed on a surrogate model constructed beforehand, allowing negligible-cost UQ during the optimisation process.

We depict the general uncertainty-based optimisation problem in Section 2, with the classical robustness and reliability measures. Then, we present the SABBa framework extended to constrained optimisation in Section 3. Sections 4 and 5 illustrate some numerical choices and features, whereas we recall the global algorithm in Section 6. We present the analytical and physical test-cases in Section 7. Some conclusions and perspectives are drawn in Section 8.

2 Uncertainty-based optimisation: problem formulation

The objective of this section is to formulate a general expression for an Optimisation under Uncertainty problem including both Robust and Reliability-Based Optimisation (RO and RBO) aspects.

A deterministic constrained multi-objective optimisation problem can be described as follows:

$$\begin{aligned} \text{minimise/maximise:} \quad & \boldsymbol{f}\left(\boldsymbol{x}\right), \\ \text{satisfying:} \quad & \boldsymbol{g}\left(\boldsymbol{x}\right) \leq \boldsymbol{0}, \\ \text{by changing:} \quad & \boldsymbol{x} \in \mathcal{X}, \end{aligned}$$

where the m_1 objective functions are collected in a vector $\boldsymbol{f} \in \mathbb{R}^{m_1}$, the m_2 constraint functions in $\boldsymbol{g} \in \mathbb{R}^{m_2}$ and $\boldsymbol{x} \in \mathcal{X}$ are the n design variables included in the design domain $\mathcal{X} \subset \mathbb{R}^n$.

In the presence of uncertain parameters $\boldsymbol{\xi}$ (*e.g.* environmental, material or geometrical parameters), the objective and constraint functions are dependent on both $\boldsymbol{x}$ and $\boldsymbol{\xi}$. Hence, $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})$ and $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\xi})$ must be replaced with robustness and reliability measures, $\boldsymbol{\rho_f}(\boldsymbol{x})$ and $\boldsymbol{\rho_g}(\boldsymbol{x})$ respectively. These allow the formulation of a very general uncertainty-based constrained multi-objective optimisation problem, as follows:

$$\begin{aligned} \text{minimise/maximise:} \quad & \boldsymbol{\rho_f}(\boldsymbol{x}), \\ \text{satisfying:} \quad & \boldsymbol{\rho_g}(\boldsymbol{x}) \leq \boldsymbol{0}, \\ \text{by changing:} \quad & \boldsymbol{x} \in \mathcal{X}, \end{aligned}$$

with $\boldsymbol{\rho_f} \in \mathbb{R}^{m_1}$ and $\boldsymbol{\rho_g} \in \mathbb{R}^{m_2}$.

Many formulations are possible for $\boldsymbol{\rho_f}$ and $\boldsymbol{\rho_g}$. The following are tackled in this paper:

$$\begin{aligned}
\text{Expectation} \quad & \boldsymbol{\rho_f}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})], \\
\text{Variance} \quad & \boldsymbol{\rho_f}(\boldsymbol{x}) = \mathbb{V}_{\boldsymbol{\xi}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})], \\
\text{Min/Max} \quad & \boldsymbol{\rho_f}(\boldsymbol{x}) = \min_{\boldsymbol{\xi}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})] \text{ or } \max_{\boldsymbol{\xi}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})], \\
\text{Quantile} \quad & \boldsymbol{\rho_f}(\boldsymbol{x}) = \mathrm{q}^p_{\boldsymbol{\xi}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})], \ p \in [0, 1],
\end{aligned}$$

and the same for $\boldsymbol{\rho_g}$. Note that any other measure could also be treated similarly in the following.

Remark For conciseness, $\boldsymbol{\rho_f}$ and $\boldsymbol{\rho_g}$ are gathered in vector $\boldsymbol{\rho}$,

$$\forall \boldsymbol{x} \in \mathcal{X}, \ \boldsymbol{\rho}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{\rho_f}(\boldsymbol{x}) \\ \boldsymbol{\rho_g}(\boldsymbol{x}) \end{pmatrix}.$$

The main issue here is then the accurate computation of these integral quantities. In this work, we approximate statistical measures through Monte Carlo Simulations (MCS) on surrogate models. To improve the accuracy of these approximations, we can increase the training set used for building the surrogate.

Hence, the uncertainty-based optimisation problem reduces to a constrained multi-objective optimisation problem with approximated and refinable measures. Such an issue has been dealt with in [33] in the unconstrained case. Next section recalls the primary results on optimisation problems with refinable approximated objectives and describes the extension to constrained problems as well as further developments.

3 SABBa framework for uncertainty-based optimisation

In this section, we present the Surrogate-Assisted Bounding-Box approach. Notations and classical *Boxed Pareto dominance* introduced in [33] are recalled, then constrained extension is proposed. Finally, we remind the global algorithm alongside main assumptions and convergence results.

3.1 Bounding-Box definition

A m-dimensional box is defined as follows, with $\boldsymbol{a}$ the center and $\boldsymbol{r}$ the positive half-width vector:

$$\mathcal{B}(\boldsymbol{a}, \boldsymbol{r}) = \left\{ \boldsymbol{b} \in \mathbb{R}^m \mid \boldsymbol{b} \in [\boldsymbol{a} - \boldsymbol{r}, \boldsymbol{a} + \boldsymbol{r}] \right\} \in \wp(\mathbb{R}^m),$$

where $\wp(\mathbb{R}^m)$ is the power set of $\mathbb{R}^m$.

The Bounding-Box approach consists in approximating an unknown value, here statistical measures $\boldsymbol{\rho}$, by a conservative box containing the true values. More precisely, the unknown $\boldsymbol{\rho}$ is approximated by $\widetilde{\boldsymbol{\rho}}^k$, k being the level of refinement of the approximation. The exact approximation error $\boldsymbol{\varepsilon}^k = \boldsymbol{\rho} - \widetilde{\boldsymbol{\rho}}^k$ is conservatively approximated by $\widetilde{\boldsymbol{\varepsilon}}^k \geq |\boldsymbol{\varepsilon}^k|$. Hence it comes that $\boldsymbol{\rho} \in \mathcal{B}(\widetilde{\boldsymbol{\rho}}^k, \widetilde{\boldsymbol{\varepsilon}}^k)$ (see Fig. 1).

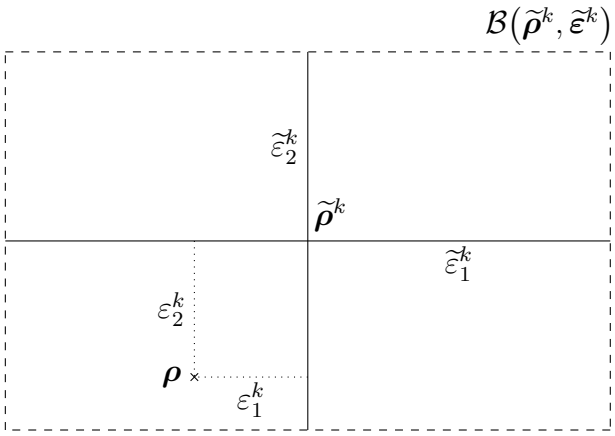


Figure 1: Bounding-Box approximation

3.1.1 Boxed-Pareto Dominance

With the aim to perform multi-objective optimisation on such objects, we define the *Boxed Pareto dominance*, for two boxes $\big(\mathcal{B}(\boldsymbol{a},\boldsymbol{r}),\mathcal{B}(\boldsymbol{b},\boldsymbol{r}')\big) \in \wp(\mathbb{R}^m)^2$, as follows

$$\begin{aligned}
\mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \underset{\mathcal{B}}{\succ} \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') &\iff \forall j \in \mathcal{I}_1^m,\ \pm a_j + r_j \leq \pm b_j - r'_j \quad \text{and} \\
&\qquad \exists j \in \mathcal{I}_1^m,\ \pm a_j + r_j < \pm b_j - r'_j; \\
\mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \underset{\mathcal{B}}{\succcurlyeq} \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') &\iff \forall j \in \mathcal{I}_1^m,\ \pm a_j + r_j < \pm b_j - r'_j; \\
\mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \underset{\mathcal{B}}{\sim} \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') &\iff \mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \underset{\mathcal{B}}{\not\succ} \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') \ \text{ and } \ \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') \underset{\mathcal{B}}{\not\succ} \mathcal{B}(\boldsymbol{a},\boldsymbol{r}).
\end{aligned}$$

$\forall j \in \mathcal{I}_1^m$, with $\mathcal{I}_1^m = [\![1, m]\!]$, the symbol $\pm$ (implicitly $\pm_j$) indicates the goal in the j^{th} dimension:

$$\pm = \begin{cases} + & \text{for minimisation} \\ - & \text{for maximisation.} \end{cases}$$

Intuitively, $\mathcal{B}(\boldsymbol{a},\boldsymbol{r})$ dominates $\mathcal{B}(\boldsymbol{b},\boldsymbol{r}')$ if the worst outcome of $\mathcal{B}(\boldsymbol{a},\boldsymbol{r})$ dominates in the classical sense the best outcome of $\mathcal{B}(\boldsymbol{b},\boldsymbol{r}')$. An example is given hereafter, in Fig. 2. In the case of bi-minimisation, the dominance are the following: $A \underset{\mathcal{B}}{\sim} B$, $A \underset{\mathcal{B}}{\sim} C$, $A \underset{\mathcal{B}}{\sim} D$, $B \underset{\mathcal{B}}{\sim} C$, $B \underset{\mathcal{B}}{\succcurlyeq} D$, $C \underset{\mathcal{B}}{\sim} D$. Hence, only A, B and C are non-dominated.

From this Pareto dominance, we can define the following Pareto optimal sets. The first one is the Pareto front in a given set $\mathcal{A}$, and the other two are respectively the classical discrete approximation and the boxed discrete approximation.

$$\begin{aligned}
\mathcal{P}\big(\mathcal{A}\big) &= \big\{\boldsymbol{a} \in \mathcal{A} \mid \nexists \boldsymbol{b} \in \mathcal{A},\ \boldsymbol{b} \succ \boldsymbol{a}\big\}, \\
\widetilde{\mathcal{P}}\Big(\big\{\boldsymbol{a}_i\big\}_{i=1}^N\Big) &= \big\{\boldsymbol{a}_i,\ i \in \mathcal{I}_1^N \mid \nexists j \in \mathcal{I}_1^N,\ \boldsymbol{a}_j \succ \boldsymbol{a}_i\big\}, \\
\widetilde{\mathcal{P}}_{\mathcal{B}}\Big(\big\{\mathcal{B}_i\big\}_{i=1}^N\Big) &= \big\{\mathcal{B}_i,\ i \in \mathcal{I}_1^N \mid \nexists j \in \mathcal{I}_1^N,\ \mathcal{B}_j \underset{\mathcal{B}}{\succ} \mathcal{B}_i\big\}.
\end{aligned}$$

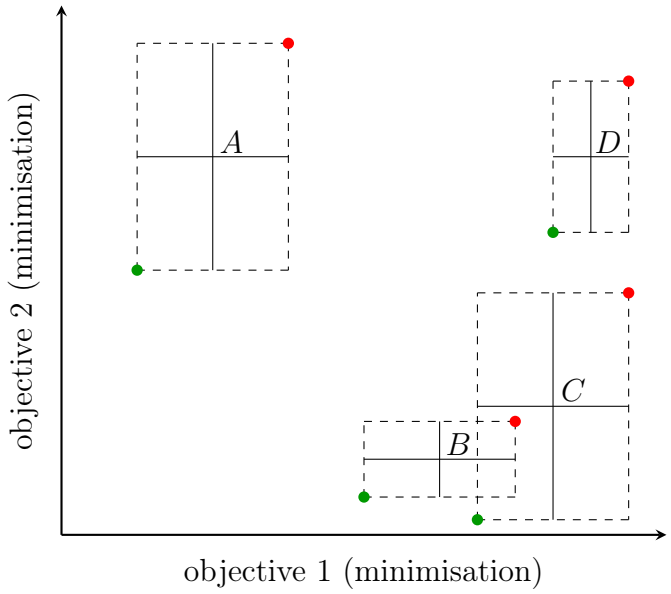


Figure 2: *Boxed Pareto Dominance*: Comparison of 4 boxes, best and worst outcomes in green and red, respectively.

The pre-image in the design space are also given as follows

$$\begin{aligned}
\mathcal{X}_{\mathcal{P}}^{\boldsymbol{\rho}} &= \left\{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{\rho}(\boldsymbol{x}) \in \mathcal{P}\right\},\\
\mathcal{X}_{\widetilde{\mathcal{P}}}^{\boldsymbol{\rho}}\left(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\right) &= \left\{\boldsymbol{x}_i,\ i \in \mathcal{I}_1^N \mid \boldsymbol{\rho}(\boldsymbol{x}_i) \in \widetilde{\mathcal{P}}\left(\left\{\boldsymbol{\rho}(\boldsymbol{x}_i)\right\}_{i=1}^N\right)\right\},\\
\mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\boldsymbol{\rho}}\left(\left\{\left(\boldsymbol{x}_i, \boldsymbol{r}_i\right)\right\}_{i=1}^N\right) &= \left\{\boldsymbol{x}_i,\ i \in \mathcal{I}_1^N \mid \mathcal{B}_{\boldsymbol{\rho}}(\boldsymbol{x}_i, \boldsymbol{r}_i) \in \widetilde{\mathcal{P}}_{\mathcal{B}}\left(\left\{\mathcal{B}_{\boldsymbol{\rho}}(\boldsymbol{x}_i, \boldsymbol{r}_i)\right\}_{i=1}^N\right)\right\},
\end{aligned}$$

with $\mathcal{B}_{\boldsymbol{\rho}}(\boldsymbol{x}_i, \boldsymbol{r}_i) = \mathcal{B}(\boldsymbol{\rho}(\boldsymbol{x}_i), \boldsymbol{r}_i)$.

3.1.2 Constrained Boxed Pareto dominance

The *Boxed Pareto dominance* allows comparing boxes in a multi-objective optimisation context. For dealing with robust objectives as well as reliability-based constraints, the Pareto dominance should allow taking into account constraint violation. To this extent, a *Constrained Pareto dominance* $\succ_c$ is first proposed, with $\succ$ the classical Pareto dominance.

Let us assume that elements to compare are in $\mathbb{R}^m$ and $m = m_1 + m_2$ with dimensions 1 to m_1 being the objective dimensions (subject to minimisation or maximisation) and the m_2 latest being the constraint dimensions (assumed as inferior or superior inequality constraints). Finally, we introduce the notation $\boldsymbol{a}_{\mathcal{D}}$, with $\mathcal{D} = \{d_1, \ldots, d_n\}$, for the projection of $\boldsymbol{a}$ on the d_i-th dimensions. Formally, $\boldsymbol{a}_{\mathcal{D}}$ is defined as:

$$\boldsymbol{a}_{\mathcal{D}} = \sum_{i=1}^{n} a_{d_i} \boldsymbol{e}_{d_i},$$

with $\boldsymbol{e}_k$ the unit vector in the k-th direction.

In the multi-objective constrained problem, vector $\boldsymbol{a}_{\mathcal{I}_1^{m_1}}$ represents the objective values and $\boldsymbol{a}_{\mathcal{I}_{m_1+1}^{m}}$ the constraint values, with again $\mathcal{I}_a^b = [\![a, b]\!]$. Note that the *Constrained Pareto dominance* compares boxes from the coupled objective/constraint spaces while the classical Pareto dominance only compares elements of the objective space. Dominance rules can be defined as follows

$$\begin{aligned}
\boldsymbol{a} \succ_c \boldsymbol{b} &\iff \boldsymbol{a}_{\mathcal{I}_1^{m_1}} \succ \boldsymbol{b}_{\mathcal{I}_1^{m_1}} \text{ and } \boldsymbol{a} \in \mathcal{S} \quad \text{or} \\
&\qquad \boldsymbol{a} \in \mathcal{F}^{\complement} \text{ and } \boldsymbol{b} \in \mathcal{F} \\
\boldsymbol{a} \succcurlyeq_c \boldsymbol{b} &\iff \boldsymbol{a}_{\mathcal{I}_1^{m_1}} \succcurlyeq \boldsymbol{b}_{\mathcal{I}_1^{m_1}} \text{ and } \boldsymbol{a} \in \mathcal{S} \quad \text{or} \\
&\qquad \boldsymbol{a} \in \mathcal{F}^{\complement} \text{ and } \boldsymbol{b} \in \mathcal{F} \\
\boldsymbol{a} \sim_c \boldsymbol{b} &\iff \boldsymbol{a} \not\succ_c \boldsymbol{b} \text{ and } \boldsymbol{b} \not\succ_c \boldsymbol{a}
\end{aligned}$$

where the admissible set $\mathcal{S}$ contains all points satisfying all constraints while any point of the failure set $\mathcal{F}$ disrespects at least one constraint and the subscript $\cdot^{\complement}$ stands for complement. One can see that any point in its complement dominates all points of the failure set and that for $\boldsymbol{a}$ to dominate $\boldsymbol{b}$, $\boldsymbol{a}$ must be in the admissible set. Note that here, $\mathcal{S} = \mathcal{F}^{\complement}$. This will not be the case in the following, in the context of boxed objective and constraint approximations.

Remark Assuming the m_2 constraints on $\boldsymbol{a}$ are of the form

$$\forall j \in \mathcal{I}_{m_1+1}^{m},\ \pm a_j \leq \pm c_j,$$

with $\boldsymbol{c}$ the constraint thresholds and where

$$\pm = \begin{cases} + & \text{for inferiority constraints} \\ - & \text{for superiority constraints,} \end{cases}$$

the admissible and failure sets are defined as follows

$$\begin{aligned}
\boldsymbol{a} \in \mathcal{S} &\iff \forall j \in \mathcal{I}_{m_1+1}^{m},\ \pm a_j \leq \pm c_j, \\
\boldsymbol{a} \in \mathcal{F} &\iff \exists j \in \mathcal{I}_{m_1+1}^{m},\ \pm a_j > \pm c_j.
\end{aligned}$$

When comparing boxes of $\mathbb{R}^m$, the boxed admissible $\mathcal{S}_{\mathcal{B}}$ and failure $\mathcal{F}_{\mathcal{B}}$ sets are defined as follows,

$$\begin{aligned}
\mathcal{B}(\boldsymbol{a}, \boldsymbol{r}) \in \mathcal{S}_{\mathcal{B}} \subseteq \wp(\mathbb{R}^m) &\iff \forall \boldsymbol{a}' \in \mathcal{B}(\boldsymbol{a}, \boldsymbol{r}),\ \boldsymbol{a}' \in \mathcal{S} \in \wp(\mathbb{R}^m) \\
\mathcal{B}(\boldsymbol{a}, \boldsymbol{r}) \in \mathcal{F}_{\mathcal{B}} \subseteq \wp(\mathbb{R}^m) &\iff \forall \boldsymbol{a}' \in \mathcal{B}(\boldsymbol{a}, \boldsymbol{r}),\ \boldsymbol{a}' \in \mathcal{F} \in \wp(\mathbb{R}^m)
\end{aligned}$$

Hence, one can see that in the general case, $\mathcal{S}_{\mathcal{B}} \neq \mathcal{F}_{\mathcal{B}}^{\complement}$. A more computationally friendly formulation is the following, using again the best and worst outcomes:

$$\begin{aligned}
\mathcal{B}(\boldsymbol{a}, \boldsymbol{r}) \in \mathcal{S}_{\mathcal{B}} &\iff \forall j \in \mathcal{I}_{m_1+1}^{m},\ \pm a_j + r_j \leq \pm c_j, \\
\mathcal{B}(\boldsymbol{a}, \boldsymbol{r}) \in \mathcal{F}_{\mathcal{B}} &\iff \exists j \in \mathcal{I}_{m_1+1}^{m},\ \pm a_j - r_j > \pm c_j.
\end{aligned}$$

Finally, the *Boxed Constrained Pareto dominance* is defined as follows:

$$\begin{aligned}
\mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \underset{\mathcal{B}}{\succ}_c \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') &\iff \mathcal{B}(\boldsymbol{a},\boldsymbol{r})_{\mathcal{I}_1^{m_1}} \underset{\mathcal{B}}{\succ} \mathcal{B}(\boldsymbol{b},\boldsymbol{r}')_{\mathcal{I}_1^{m_1}} \text{ and } \mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \in \mathcal{S}_{\mathcal{B}} \quad \text{or} \\
&\qquad \mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \in \mathcal{F}_{\mathcal{B}}^{\complement} \text{ and } \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') \in \mathcal{F}_{\mathcal{B}}, \\
\mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \underset{\mathcal{B}}{\succ\!\succ}_c \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') &\iff \mathcal{B}(\boldsymbol{a},\boldsymbol{r})_{\mathcal{I}_1^{m_1}} \underset{\mathcal{B}}{\succ\!\succ} \mathcal{B}(\boldsymbol{b},\boldsymbol{r}')_{\mathcal{I}_1^{m_1}} \text{ and } \mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \in \mathcal{S}_{\mathcal{B}} \quad \text{or} \\
&\qquad \mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \in \mathcal{F}_{\mathcal{B}}^{\complement} \text{ and } \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') \in \mathcal{F}_{\mathcal{B}}, \\
\mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \underset{\mathcal{B}}{\sim}_c \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') &\iff \mathcal{B}(\boldsymbol{a},\boldsymbol{r}) \underset{\mathcal{B}}{\not\succ}_c \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') \text{ and } \mathcal{B}(\boldsymbol{b},\boldsymbol{r}') \underset{\mathcal{B}}{\not\succ}_c \mathcal{B}(\boldsymbol{a},\boldsymbol{r}),
\end{aligned}$$

with the notation $\mathcal{B}(\boldsymbol{a},\boldsymbol{r})_{\mathcal{I}_1^{m_1}} = \mathcal{B}(\boldsymbol{a}_{\mathcal{I}_1^{m_1}}, \boldsymbol{r}_{\mathcal{I}_1^{m_1}})$.

With this Pareto dominance rule, any box entirely in the failure set is considered dominated. Else, for a box to be dominated by another one, the non-constrained *Boxed Pareto Dominance* must be fulfilled in the objective dimensions and the dominant box must lie entirely in the admissible set. An example is given in Fig. 3 with a constrained mono-objective minimisation. Here, $A \in \mathcal{F}_{\mathcal{B}}$, $B \in \mathcal{F}_{\mathcal{B}}^{\complement} \cap \mathcal{S}_{\mathcal{B}}^{\complement}$, $C \in \mathcal{S}_{\mathcal{B}}$ and $D \in \mathcal{S}_{\mathcal{B}}$. The dominance are then as follows: $A \underset{\mathcal{B}}{\prec\!\prec}_c B$, $A \underset{\mathcal{B}}{\prec\!\prec}_c C$, $A \underset{\mathcal{B}}{\prec\!\prec}_c D$, $B \underset{\mathcal{B}}{\sim}_c C$, $B \underset{\mathcal{B}}{\sim}_c D$, $C \underset{\mathcal{B}}{\succ\!\succ}_c D$. Only B and C are non-dominated. A is dominated by being entirely in the failure set and D is dominated by C in the classical *Boxed Pareto Dominance* sense in 1D as C is entirely in the admissible set.

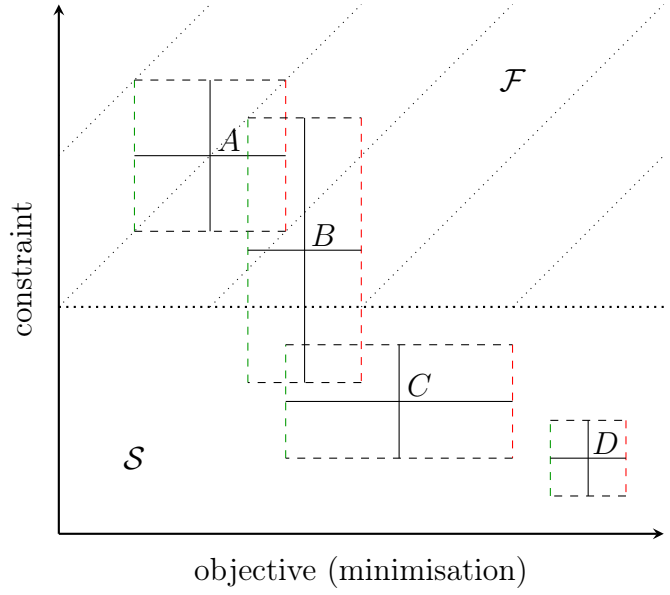


Figure 3: *Constrained Boxed Pareto Dominance*: Comparison of 4 boxes, best and worst outcomes in green and red, respectively.

The classical and boxed Pareto optimal sets $\mathcal{P}$ and $\mathcal{P}_{\mathcal{B}}$ are approximated by their discrete counterpart:

$$\begin{aligned}
\widetilde{\mathcal{P}}\left(\{\boldsymbol{a}_i\}_{i=1}^N\right) &= \{\boldsymbol{a}_i,\ i \in \mathcal{I}_1^N \mid \nexists j \in \mathcal{I}_1^N,\ \boldsymbol{a}_j \succ_c \boldsymbol{a}_i \ \&\ \boldsymbol{a}_i \in \mathcal{F}^{\complement}\}, \\
\widetilde{\mathcal{P}}_{\mathcal{B}}\left(\{\mathcal{B}_i\}_{i=1}^N\right) &= \{\mathcal{B}_i,\ i \in \mathcal{I}_1^N \mid \nexists j \in \mathcal{I}_1^N,\ \mathcal{B}_j \underset{\mathcal{B}}{\succ}_c \mathcal{B}_i \ \&\ \mathcal{B}_i \in \mathcal{F}_{\mathcal{B}}^{\complement}\}.
\end{aligned}$$

These newly defined sets can be used within SABBa, to focus refinements on non-dominated boxes, thus on boxes with optimal objective values and that remain at least partially out of the failure set. Note that these sets can be empty if all boxes are in the failure set, meaning that computational power remains focused on exploration and not on refinement.

3.2 SABBa algorithm

The objective of the SABBa framework is to lower the computational cost associated with returning a converged Pareto front when dealing with approximated and refinable objectives and constraints. The unconstrained version of this framework is developed in [33] and relies on the coupling between a Bounding-Box approach proposed in [32] and a surrogate-assisting strategy built on the objectives that bypass some function evaluations, notably at the end of the optimisation process.

SABBa can be coupled to a whatever optimisation algorithm and surrogate model for the assisting strategy. This feature allows the approach to be very readily applicable and to benefit from new optimisation or metamodeling algorithms. In the following, the chosen optimisation algorithm is NOMAD [34] for its reliable management of multiple objectives and constraints within a derivative-free framework. All surrogate models will be handled with Gaussian Processes (using the python Gaussian Process package GPy [35]).

In this work, SABBa is extended to deal with approximated constraints and applied to the problem of optimisation under uncertainty. Hence, the approximated objectives and constraints are statistics computed over the uncertain dimensions $\boldsymbol{\xi}$ (see Section 2). We make several numerical choices to ensure and increase SABBa overall parsimony. These ones are introduced in this section and are more precisely explained in Sections 4 and 5.

The significant steps of the Algorithm 1 are presented her and details are given afterwards. We denote some specific stages of the algorithm using letters between parenthesis, recalled in the description below.

Algorithm 1 Algorithm overview

```
Choose or loop over values of s_1 and s_2                          ▷ (a)
while Optimisation running do
    Read new designs x
    Compute local surrogate errors at x                            ▷ (b)
    if Surrogate error is below s_1 then
        Use surrogate value to approximate measures ρ
    else
        Compute a first approximation of ρ → function evaluations  ▷ (c,d)
        while Refinement needed do                                 ▷ (e)
            Choose which box to refine                             ▷ (f)
            Compute finer approximation of ρ → function evaluations ▷ (c,d)
        end while
    end if
end while
Return non-dominated boxes                                         ▷ (f)
Compute quality of the optimisation output                         ▷ (g)
```

SABBa relies on two user-defined thresholds $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. These can be fixed to a small value to enforce accuracy. An iterative lowering of these values is proposed in this work, allowing for both cost reduction and intermediate results for the user. We refer to this strategy to as the

Reuse strategy (denoted with *(a)* in Algorithm 1) and is more deeply explained in Section 5.2. We quantitatively investigate its impact in Section 7.

For all new designs $\boldsymbol{x}$ given by the optimiser, we approximate the surrogate model error at these $\boldsymbol{x}$ using the surrogate-assisting strategy. An error approximation $\hat{\boldsymbol{r}}$ is proposed in [33] when using the surrogate-assisting model:

$$\hat{\boldsymbol{r}}(\boldsymbol{x}_i) = \widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i) + \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i)$$

with $\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i)$ the error associated to the surrogate-assisting model and $\widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i)$ the one arising from the impreciseness of the training data on which the model is built. Indeed, the surrogate-assisting model is built on the center of the boxes, which are a noised approximation of the robustness and reliability-based statistics $\boldsymbol{\rho}$. In this work, we use a heteroscedastic Gaussian Process (GP) as surrogate-assisting model, and its predictive variance gives an error estimation *(b)*. Such a model naturally deals with noised training data, as presented in Section 5.3, and the usefulness of $\widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i)$ in the above equation is investigated in the test-cases.

If the above error is below the threshold $\boldsymbol{s}_1$, the predictive values of the uncertainty-based objectives and constraints provided by the surrogate model are returned to the optimiser. Else, a first approximation of the statistical measures $\boldsymbol{\rho}$ must be computed. In this case, we choose to use an interpolating GP to model the behaviour of the quantities of interest ($\boldsymbol{f}$ and $\boldsymbol{g}$ from Section 2) in the uncertain dimensions $\boldsymbol{\xi}$. Two approaches are proposed here *(c)*, and compared in the test-cases:

- One denoted as *Separated space*: One GP model is built for each design $\boldsymbol{x}$, thus giving as many models $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})$ as designs, but of reduced input dimension.
- One denoted as *Coupled space*: Only one global GP is built on both the design and uncertain dimensions, thus resulting in a single model $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$ allowing for correlations in the design space but of higher input dimension.

Once one of the above strategies is applied, the robustness and reliability-based measures can be approximated with a Monte Carlo Sampling at a negligible-cost. This computation gives the center of the associated box, and the width of the box can be obtained (denoted with *(d)* in Algorithm 1) following the computation details given in Section 4.2.

These boxes are compared to each other and to the ones coming from the surrogate-assisting strategy through the *Constrained Boxed Pareto dominance (e)* presented in Section 3.1.2 to only retain the non-dominated boxes. These efficient boxes are then sequentially refined until they reach a level of accuracy driven by the threshold $\boldsymbol{s}_2$. Practically, we find the most promising box (not coming from the surrogate-assisting strategy) *(f)* by computing the Pareto Optimal Probability of all boxes (computation details are given in Section 4.1). Then, given the design at which measures should gain in accuracy, criteria presented in Section 4.2 drive the selection of the coordinates where computing the quantities of interest *(d)*. Note that when using a model in the coupled space, this refinement will impact all surrounding boxes *(c)*. Note also that the comparison through the *Constrained Boxed Pareto dominance* is performed after each refinement to retain as few efficient boxes as possible. This approach, introduced in [33], follows the set sequence of non-dominated boxes, which cardinality decreases with refinements. This sequence is recalled here, for $\left\{\boldsymbol{x}_i\right\}_{i=1}^N \in \mathcal{X}^N$, $\boldsymbol{\rho}$ the objective and constraints functions, $\widetilde{\boldsymbol{\rho}}^k$ refinable approximations and $\widetilde{\boldsymbol{\varepsilon}}^k$ their conservative errors, from given $\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{\rho},0} = \left\{\boldsymbol{x}_i\right\}_{i=1}^N$ and $\left\{\widetilde{\boldsymbol{\varepsilon}}^0(\boldsymbol{x}_i)\right\}_{i=1}^N$,

$$\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{\rho},k+1} = \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\widetilde{\boldsymbol{\rho}}^k}\left(\left\{\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_i}^{\boldsymbol{\rho},k}, \widetilde{\boldsymbol{\varepsilon}}^k\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_i}^{\boldsymbol{\rho},k}\right)\right)\right\}_{i=1}^{N^k}\right) \tag{1}$$

with N^k the decreasing number of elements in $\widetilde{\mathcal{X}}^{\boldsymbol{\rho},k}_{\widetilde{\mathcal{P}}}$.

Finally, when the optimiser has reached convergence, the optimisation output is returned to the user. By computing the Pareto Optimal Probability of all boxes, SABBa gives an accurate representation of the most probable optimal design *(f)*. This representation is finally supplemented by the computation of a quantitative indicator, which measures the expected modified Hausdorff distance between the true optimal designs and the optima found by SABBa *(g)*. Section 6.2 provides practical details for this computation.

We picture the structure of the framework in the flowchart given in Figure 4. One can indeed see that the values returned to the optimiser come either from true computations or the surrogate model depending on the surrogate error at $\boldsymbol{x}$ compared to $\boldsymbol{s}_1$. The notation $\widetilde{\boldsymbol{\rho}}^{k_{min}}$ refers to the first measures approximation at $\boldsymbol{x}$ so that the error goes below $\boldsymbol{s}_2$ or the box gets dominated. On the other hand, $\widetilde{\boldsymbol{\rho}}^l_d$ is simply the value of the predictive measure from the surrogate model. Note that this predictive value may change if the surrogate gets updated. Finally, $\hat{\boldsymbol{\rho}}$ refers to the value returned to the optimisation process.

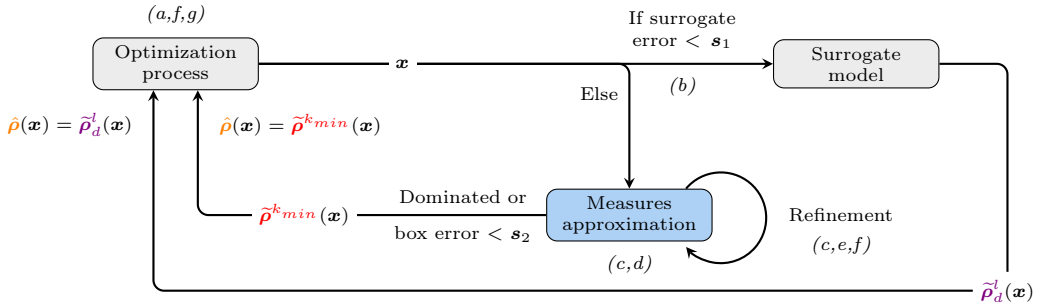


Figure 4: Structure of the SABBa framework

We provide the numerical details associated with all steps depicted in the above in the following sections.

4 Numerical strategies

This section intends to introduce the features developed and used in the SABBa framework. First, we give the definition and then the practical way of computing the Pareto Optimal Probability. Then, we explain the management of boxes approximations through a procedure for conservative errors computation and the associated refinements.

4.1 Pareto Optimal Probability

As seen in the preceding section, the SABBa framework is supplemented by the computation for each Bounding-Box of its probability of being non-dominated. The knowledge of the Pareto Optimal Probability (POP) of all boxes allows a ranking of the designs within the non-dominated set and has two primary applications: (i) boxes with a higher POP are refined before less promising ones to optimise the number of black-box evaluations; (ii) final solutions in the design space can be post-treated to return not only non-dominated designs but also the distribution of POP over the design space, thus efficiently guiding toward the most likely optimal area.

To illustrate the challenges associated to the POP computation, three examples are used (represented in terms of boxes in Figures 6 and 7). In particular, we compare the different POP

computational variants proposed hereafter on (i) a test-case with highly clustered boxes to highlight a specific issue associated to some POP computations. Then, more classical examples are illustrated: (ii) a bi-minimization problem and (iii) a constrained mono-objective minimization problem. Quantitative results are summarized in Table 1.

The objective and constraint functions are denoted by $\boldsymbol{\rho}$ where $\boldsymbol{\mathcal{D}}_{all}$ represents all studied designs, each of which has a box approximation. The exact POP computation follows the formula below:

$$POP_{true}(\boldsymbol{x}_i) = \mathbb{P}_{\{\boldsymbol{\rho}(\boldsymbol{x}_k)\}_k}\left[\bigcap_{\substack{\boldsymbol{x}_j \in \boldsymbol{\mathcal{D}}_{all} \\ j \neq i}} \boldsymbol{x}_j \overset{\boldsymbol{\rho}}{\nsucc}_c \boldsymbol{x}_i\right].$$

Remark It is made explicit in the above that the probability is computed over the values $\boldsymbol{\rho}(\boldsymbol{x}_k)$ for all k, which are all assumed to follow independent uniform distributions within their associated Bounding-Box. Moreover, it is recalled that

$$\boldsymbol{x}_j \overset{\boldsymbol{\rho}}{\succ}_c \boldsymbol{x}_i \iff \boldsymbol{\rho}(\boldsymbol{x}_j) \succ_c \boldsymbol{\rho}(\boldsymbol{x}_i).$$

Note however that this POP computation yields a combinatorial complexity that profoundly limits its calculation in closed form and requires a Monte-Carlo approximation. Moreover, as this computation relies on a discrete sampling, heterogeneity in the spreading of the box in the measures space induces a heterogeneity in the associated Pareto Optimal Probabilities. Figure 6 illustrates an example with clustered boxes (C to F). Geometrically, all these boxes are equally close to the Pareto front, whereas box B is sensibly less efficient than A, hence further from the Pareto front. However, one can see in Table 1 that the POP_{true} computation yields higher values of POP for box B than for boxes C to F.

Hence, we propose two POP approximations in the following, compared afterwards in Table 1.

Ref. [30] proposes the use of a Box probabilistic ranking as the fitness function for the ESPEA algorithm. The score of a given box is computed by averaging the one-to-one domination probability with regards to all other boxes. Such an idea is used here, replacing the domination probability with the POP, as follows:

$$POP_{av}(\boldsymbol{x}_i) = \frac{1}{N-1} \sum_{\substack{\boldsymbol{x}_j \in \boldsymbol{\mathcal{D}}_{all} \\ j \neq i}} \mathbb{P}_{\boldsymbol{\rho}(\boldsymbol{x}_j),\boldsymbol{\rho}(\boldsymbol{x}_i)}\left[\boldsymbol{x}_j \overset{\boldsymbol{\rho}}{\nsucc}_c \boldsymbol{x}_i\right],$$

with $\boldsymbol{\rho}$ of dimension $m = m_1 + m_2$ where m_1 is the number of objectives and m_2 the number of constraints. Hence, $\boldsymbol{\rho}_{\mathcal{I}_1^{m_1}} = \boldsymbol{\rho_f}$ are the objectives values and $\boldsymbol{\rho}_{\mathcal{I}_{m_1+1}^{m}} = \boldsymbol{\rho_g}$ the constraints values. The probability in the above expression is computed as follows, using the independence assumptions between boxes and between dimensions:

$$\begin{aligned}
\mathbb{P}_{\boldsymbol{\rho}(\boldsymbol{x}_j),\boldsymbol{\rho}(\boldsymbol{x}_i)}\left[\boldsymbol{x}_j \overset{\boldsymbol{\rho}}{\nsucc}_c \boldsymbol{x}_i\right] &= 1 - \mathbb{P}_{\boldsymbol{\rho}(\boldsymbol{x}_j),\boldsymbol{\rho}(\boldsymbol{x}_i)}\left[\boldsymbol{x}_j \overset{\boldsymbol{\rho}}{\succ}_c \boldsymbol{x}_i\right] \\
&= 1 - \Big(\mathbb{P}_{\boldsymbol{\rho_g}(\boldsymbol{x}_i)}\left[\boldsymbol{\rho_g}(\boldsymbol{x}_i) > \mathbf{0}\right] \\
&\quad + \mathbb{P}_{\boldsymbol{\rho_g}(\boldsymbol{x}_i)}\left[\boldsymbol{\rho_g}(\boldsymbol{x}_i) \leq \mathbf{0}\right]\mathbb{P}_{\boldsymbol{\rho_g}(\boldsymbol{x}_j)}\left[\boldsymbol{\rho_g}(\boldsymbol{x}_j) \leq \mathbf{0}\right]\mathbb{P}_{\boldsymbol{\rho}(\boldsymbol{x}_j),\boldsymbol{\rho}(\boldsymbol{x}_i)}\left[\boldsymbol{x}_j \overset{\boldsymbol{\rho_f}}{\succ} \boldsymbol{x}_i\right]\Big) \\
&= \mathbb{P}_{\boldsymbol{\rho_g}(\boldsymbol{x}_i)}\left[\boldsymbol{\rho_g}(\boldsymbol{x}_i) \leq \mathbf{0}\right]\Big(1 - \mathbb{P}_{\boldsymbol{\rho_g}(\boldsymbol{x}_j)}\left[\boldsymbol{\rho_g}(\boldsymbol{x}_j) \leq \mathbf{0}\right]\mathbb{P}_{\boldsymbol{\rho}(\boldsymbol{x}_j),\boldsymbol{\rho}(\boldsymbol{x}_i)}\left[\boldsymbol{x}_j \overset{\boldsymbol{\rho_f}}{\succ} \boldsymbol{x}_i\right]\Big)
\end{aligned} \tag{2}$$

where

$$\mathbb{P}_{\boldsymbol{\rho_g}(\boldsymbol{x})}\left[\boldsymbol{\rho_g}(\boldsymbol{x}) \leq \mathbf{0}\right] = \prod_{k=m_1+1}^{m} \max\left(0, \min\left(1, \frac{-\widetilde{\rho_k^-}(\boldsymbol{x})}{2\widetilde{\varepsilon}_k(\boldsymbol{x})}\right)\right) = \prod_{k=m_1+1}^{m} \left[\frac{-\widetilde{\rho_k^-}(\boldsymbol{x})}{2\widetilde{\varepsilon}_k(\boldsymbol{x})}\right]_0^1,$$

with again $\boldsymbol{\rho_g} = \boldsymbol{\rho}_{\mathcal{I}^m_{m_1+1}}$, and where the notation $\left[\,\cdot\,\right]_0^1$ means that maximal and minimal values vary between 0 and 1. Moreover, the lower bound of the box is written as $\widetilde{\rho}_k^-(\boldsymbol{x}) = \widetilde{\rho}_k(\boldsymbol{x}) - \widetilde{\varepsilon}_k(\boldsymbol{x})$. The second probability involved in the last line of Equation (2) is computed as follows, with again $\boldsymbol{\rho_f} = \boldsymbol{\rho}_{\mathcal{I}_1^{m_1}}$:

$$\begin{aligned}
\mathbb{P}_{\boldsymbol{\rho}(\boldsymbol{x}_j),\boldsymbol{\rho}(\boldsymbol{x}_i)}\big[\boldsymbol{x}_j \overset{\boldsymbol{\rho_f}}{\not\succ} \boldsymbol{x}_i\big] &= 1 - \mathbb{P}_{\boldsymbol{\rho}(\boldsymbol{x}_j),\boldsymbol{\rho}(\boldsymbol{x}_i)}\big[\boldsymbol{x}_j \overset{\boldsymbol{\rho_f}}{\succ} \boldsymbol{x}_i\big] \\
&= 1 - \prod_{k=1}^{m_1} \mathbb{P}_{\rho_k(\boldsymbol{x}_j),\rho_k(\boldsymbol{x}_i)}\big[\rho_k(\boldsymbol{x}_j) \leq \rho_k(\boldsymbol{x}_i)\big] \\
&= 1 - \prod_{k=1}^{m_1} \left(\left[\frac{L_{1,k}}{2\widetilde{\varepsilon}_k(\boldsymbol{x}_j)}\right]_0^1 + \left[\frac{L_{2,k}}{2\widetilde{\varepsilon}_k(\boldsymbol{x}_j)}\right]_0^1 \left(\frac{1}{2}\left[\frac{L_{2,k}}{2\widetilde{\varepsilon}_k(\boldsymbol{x}_i)}\right]_0^1 + \left[\frac{L_{3,k}}{2\widetilde{\varepsilon}_k(\boldsymbol{x}_i)}\right]_0^1 \right) \right),
\end{aligned}$$

where

$$\begin{aligned}
L_{1,k} &= \widetilde{\rho}_k^-(\boldsymbol{x}_i) - \widetilde{\rho}_k^-(\boldsymbol{x}_j), \\
L_{2,k} &= \min(\widetilde{\rho}_k^+(\boldsymbol{x}_j), \widetilde{\rho}_k^+(\boldsymbol{x}_i)) - \max(\widetilde{\rho}_k^-(\boldsymbol{x}_j), \widetilde{\rho}_k^-(\boldsymbol{x}_i)), \\
L_{3,k} &= \widetilde{\rho}_k^+(\boldsymbol{x}_i) - \widetilde{\rho}_k^+(\boldsymbol{x}_j).
\end{aligned}$$

In each dimension, $L_{1,k}$ can be interpreted as the portion of the box associated to $\boldsymbol{x}_j$ dominating the one associated to $\boldsymbol{x}_i$, $L_{3,k}$ the portion of the second box dominated by the first one and $L_{2,k}$ the overlapping area. We depict these lengths in Fig. 5. One can note from the above that they can be negative and that the computation is not symmetric between $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$.

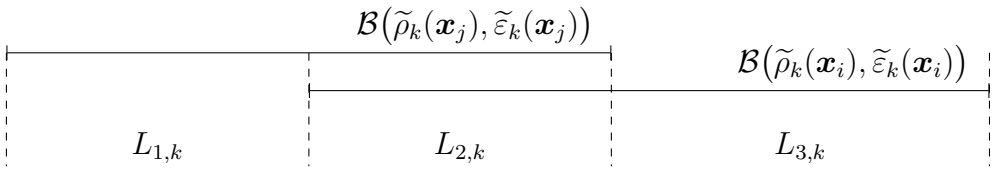


Figure 5: Computational details

Since we compute each probability only between two boxes, the computational burden is extremely low. However, as one can observe in Table 1, this approximation still suffers from heterogeneously spread boxes in Example 1 and returns non-null values of POP for strictly dominated boxes in Examples 2 and 3.

Finally, another measure, denoted as POP_{min}, is defined as follows, with $\boldsymbol{\mathcal{D}}$ the set of non-dominated designs (with the *Constrained Boxed Pareto dominance*):

$$POP_{min}(\boldsymbol{x}_i) = \min_{\substack{\boldsymbol{x}_j \in \boldsymbol{\mathcal{D}} \\ j \neq i}} \Big(\mathbb{P}_{\boldsymbol{\rho}(\boldsymbol{x}_j),\boldsymbol{\rho}(\boldsymbol{x}_i)}\big[\boldsymbol{x}_j \overset{\boldsymbol{\rho}}{\not\prec}_c \boldsymbol{x}_i\big] \Big).$$

This approximation, like the previous one, yields a good relative ranking between boxes and is efficiently computable. Moreover, the Pareto Optimal Probabilities computed in regions of different spread densities are now comparable as only one POP value between two boxes is retained in the POP_{min} approximation. Table 1 pictures the three approaches by giving the POP of all boxes from Fig. 6 and 7.

For its simplicity, interpretability and very low computational burden, POP_{min} will be used in the following.

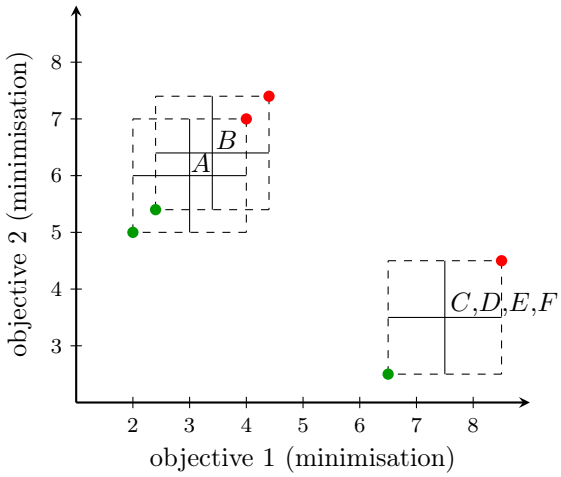
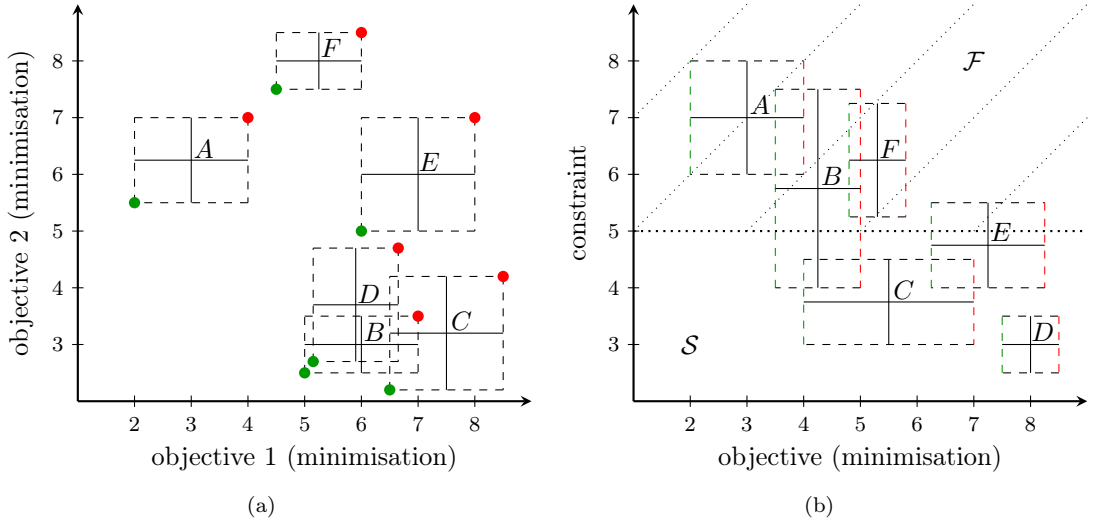


Figure 6: Example 1, with boxes C to F superimposed. Best and worst outcomes in green and red respectively. Associated POPs in Table 1.



(a) (b)

Figure 7: (a) Example 2 with a bi-minimization problem ; (b) Example 3 with a constrained mono-objective minimization. Best and worst outcomes in green and red respectively. Associated POPs in Table 1.

4.2 Box computation and refinement

We provide now some computational details on how to compute conservative errors on both surrogate-assisting model and measure approximations. As mentioned in Section 3.2, we calculate these approximations with Gaussian Processes, which give both a predictive value and a predictive variance of the model. These conservative errors are then based on the GP predictive variance, relaxing the conservative requirement to a $\pm 3\sigma$ paradigm.

Practically, with $\boldsymbol{\sigma}^2_{\boldsymbol{f}}(\boldsymbol{z})$ the predictive variance of a GP model built on functions $\boldsymbol{f}$, the

	Example 1			Example 2			Example 3		
	POP_{true}	POP_{av}	POP_{min}	POP_{true}	POP_{av}	POP_{min}	POP_{true}	POP_{av}	POP_{min}
A	0.898	0.98	0.898	1.0	1.0	1.0	0.0	0.0	0.0
B	0.538	0.908	0.538	0.902	0.98	0.912	0.254	0.279	0.254
C	0.521	0.85	0.75	0.39	0.828	0.419	0.724	0.943	0.746
D	0.521	0.85	0.75	0.621	0.924	0.622	0.0	0.628	0.0
E	0.521	0.85	0.75	0.017	0.508	0.07	0.022	0.483	0.031
F	0.521	0.85	0.75	0.0	0.735	0.0	0.0	0.0	0.0

Table 1: POPs comparison on examples 1 to 3 (Figs. 6 and 7)

conservative error is computed as:

$$\widetilde{\varepsilon}_{\boldsymbol{f}}(\boldsymbol{z}) = 3\boldsymbol{\sigma}_{\boldsymbol{f}}(\boldsymbol{z})$$

which does not imply $\widetilde{\varepsilon}_{\boldsymbol{f}}(\boldsymbol{z}) \geq |\varepsilon(\boldsymbol{z})|$ with ε the true error from the surrogate. However, the probability of dissatisfying the conservative assumption is $\mathbb{P}\big[\widetilde{\varepsilon}(\boldsymbol{z}) < |\varepsilon(\boldsymbol{z})|\big] < 0.3\%$.

This is used as conservative error for the surrogate-assisting model, but also for the approximated measures, following the formulas proposed hereafter:

$$\begin{aligned}
\text{Expectation } \widetilde{\boldsymbol{\mu}} &: \widetilde{\varepsilon}_{\mu}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}\big[\widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})\big] \\
\text{Variance } \widetilde{\boldsymbol{\sigma}^2} &: \widetilde{\varepsilon}_{\sigma^2}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}\big[\big(\widetilde{\varepsilon}_{\mu}(\boldsymbol{x}) + \widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})\big)^2 + 2\big|\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{\mu}}(\boldsymbol{x})\big|\big(\widetilde{\varepsilon}_{\mu}(\boldsymbol{x}) + \widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})\big)\big] \\
\text{Min } \widetilde{\boldsymbol{m}} &: \widetilde{\varepsilon}_{\min}(\boldsymbol{x}) = \max\big(\big|\widetilde{\boldsymbol{m}}(\boldsymbol{x}) - \min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^{-}_{\boldsymbol{x}}(\boldsymbol{\xi})]\big|, \big|\widetilde{\boldsymbol{m}}(\boldsymbol{x}) - \min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^{+}_{\boldsymbol{x}}(\boldsymbol{\xi})]\big|\big) \\
\text{Max } \widetilde{\boldsymbol{M}} &: \widetilde{\varepsilon}_{\max}(\boldsymbol{x}) = \max\big(\big|\widetilde{\boldsymbol{M}}(\boldsymbol{x}) - \max_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^{-}_{\boldsymbol{x}}(\boldsymbol{\xi})]\big|, \big|\widetilde{\boldsymbol{M}}(\boldsymbol{x}) - \max_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^{+}_{\boldsymbol{x}}(\boldsymbol{\xi})]\big|\big) \\
\text{Quantile } \widetilde{\mathbf{q}^p} &: \widetilde{\varepsilon}_{\mathrm{q}^p}(\boldsymbol{x}) = \max\big(\big|\widetilde{\mathbf{q}^p}(\boldsymbol{x}) - \mathrm{q}^p_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^{-}_{\boldsymbol{x}}(\boldsymbol{\xi})]\big|, \big|\widetilde{\mathbf{q}^p}(\boldsymbol{x}) - \mathrm{q}^p_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^{+}_{\boldsymbol{x}}(\boldsymbol{\xi})]\big|\big)
\end{aligned} \tag{3}$$

where $\widetilde{\boldsymbol{f}}^{+}_{\boldsymbol{x}}(\boldsymbol{\xi}) = \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) + \widetilde{\varepsilon}_{\boldsymbol{f}\boldsymbol{x}}(\boldsymbol{\xi})$ and $\widetilde{\boldsymbol{f}}^{-}_{\boldsymbol{x}}(\boldsymbol{\xi}) = \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\varepsilon}_{\boldsymbol{f}\boldsymbol{x}}(\boldsymbol{\xi})$. Note that in the *separated space* case, the notation $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})$ is straightforward, but in the *coupled space* case, one should read $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})$ as $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$ (see subsection 5.1). In practice, the expected values above are approximated by means of Monte Carlo Sampling (MCS) at negligible cost on surrogate model. Justification of these box sizes are given in A.

Details must also be given on how to refine this surrogate model $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})$. The refinement could be based on a space-filling paradigm, by geometrically filling the biggest hole in the sampling. However, we choose here to use GP model information fully and to refine at the maximum of a computed criterion. A partial criterion is computed for each measure of the uncertainty-based

problem, with the following formulas:

$$\begin{aligned}
\text{Expectation } \widetilde{\boldsymbol{\mu}} &: \boldsymbol{c}_{\mu}(\boldsymbol{\xi}) = \widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \\
\text{Variance } \widetilde{\boldsymbol{\sigma}^2} &: \boldsymbol{c}_{\sigma^2}(\boldsymbol{\xi}) = \widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \\
\text{Min } \widetilde{\boldsymbol{m}} &: \boldsymbol{c}_{\min}(\boldsymbol{\xi}) = \left[\frac{\widetilde{\boldsymbol{m}} - \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{-}(\boldsymbol{\xi})}{2\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})} \right]_+ \\
\text{Max } \widetilde{\boldsymbol{M}} &: \boldsymbol{c}_{\max}(\boldsymbol{\xi}) = \left[\frac{\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{+}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{M}}}{2\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})} \right]_+ \\
\text{Quantile } \widetilde{\mathbf{q}^p} &: \boldsymbol{c}_{\mathrm{q}^p}(\boldsymbol{\xi}) = \left[\frac{\widetilde{\mathbf{q}^p} - \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{-}(\boldsymbol{\xi})}{2\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})} \right]_+ \left[\frac{\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{+}(\boldsymbol{\xi}) - \widetilde{\mathbf{q}^p}}{2\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})} \right]_+ p(\boldsymbol{\xi})
\end{aligned} \tag{4}$$

with the same definition of $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{+}(\boldsymbol{\xi})$ and $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{-}(\boldsymbol{\xi})$ as before and where $\left[\,\cdot\,\right]_+ = \max(0, \cdot)$ refers to the value if positive, 0 either. Note that we heuristically choose here to multiply the criteria for the mean, variance and quantile measures by the input Probability Density Function (PDF) $p(\boldsymbol{\xi})$ to put more weight on the most likely area. Justifications for these formulas are given in B.

These partial criteria are then scalarised into the final refinement criterion through the following weighted sum:

$$c(\boldsymbol{\xi}) = \sum_{i=1}^{m} w_i \bar{c}_i(\boldsymbol{\xi}) \tag{5}$$

with $\bar{c}_i(\boldsymbol{\xi})$ the normalised partial criteria, m the number of measures and w_i the weights. The normalised partial criteria are computed as follows:

$$\bar{c}_i(\boldsymbol{\xi}) = \frac{c_i(\boldsymbol{\xi}) - \min\limits_{\boldsymbol{\xi}}[c_i(\boldsymbol{\xi})]}{\max\limits_{\boldsymbol{\xi}}[c_i(\boldsymbol{\xi})] - \min\limits_{\boldsymbol{\xi}}[c_i(\boldsymbol{\xi})]} \in [0, 1]$$

We propose here to compute the weights as the ratio between the conservative error associated with measure i and the target accuracy. In this manner, any partial criterion associated with high error compared to the target accuracy will profoundly influence the final criterion. Practically, to emphasise this behaviour, the dependence to the conservative error is chosen quadratic:

$$w_i = \left(\frac{\widetilde{\varepsilon}_i}{s_{2_i}} \right)^2.$$

Note that in the case of multi-point refinement, we conduct a greedy sequential approach by assuming that previous refinements are performed, fixing the predictive values of the GP model and updating the predictive variance to compute the new refinement criterion. We can then perform Black-box evaluations in parallel on these points. This approach allows performing multi-point refinement efficiently without the need of any clustering heuristic. However, this strategy makes a lot of assumptions and may not yield excellent results with many refinements in parallel.

Now that the significant computation details have been made explicit, we present in the following section some additional features.

5 Further features

The previous section illustrated essential ingredients of the proposed framework: the POP and boxes management. Contrarily, this section presents additional features which may impact the

overall computational cost beneficially. The use of surrogate modelling in the coupled space, the reuse strategy looping on the values of $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ and the computation of the errors associated to surrogate-assisted boxes have been mentioned in Section 3.2. We present more deeply each of these three features here, with a highlight on the different variants used in the applicative section (Section 7).

5.1 Coupled space formulation

In the context of uncertainty-based optimisation, many classical approaches rely on the surrogate modelling of $\boldsymbol{f}$ in the coupled space. This so-called *coupled space* regroups the design space $\mathcal{X}$ and the uncertain space Ξ. One must note that in the following, $\boldsymbol{f}$ will regroup all underlying functions (Quantities of Interest) on which we compute objective and constraint measures, instead of two different notations $\boldsymbol{f}$ and $\boldsymbol{g}$. Once such a surrogate is built, it allows integrating on the uncertain dimensions to compute robustness or reliability measures at any given design. Formally, by approximating $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})$ with $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$, the measures can be directly computed, most importantly at any given $\boldsymbol{x}$, by sampling in Ξ. For example, expected performances can then be approximated by:

$$\widetilde{\boldsymbol{\rho}}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi}_i).$$

On the contrary, in [33], the computation of each box and the associated refinements are performed locally, at a fixed $\boldsymbol{x}$, in the design space. In the case of uncertainty-based measures, the approach would reduce to build, for each studied design $\boldsymbol{x}$, a model $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}$ in the uncertain space Ξ to sample at a negligible cost for measure computation. The expected performance would then be computed by:

$$\widetilde{\boldsymbol{\rho}}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}_i),$$

with the difference that one should build a model $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}$ from scratch at each new design.

One can see that the *coupled space* exploitation yields a much more efficient use of each computation to reduce the overall number of evaluation compared to the more straightforward approach that is denoted as *separated space*. However, within the SABBa framework, the use of previous computations to reduce the overall cost is performed through the surrogate-assisting strategy, which uses correlations between approximated measures $\widetilde{\boldsymbol{\rho}}(\boldsymbol{x})$ instead of between the underlying $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})$. While $\widetilde{\boldsymbol{\rho}}$ is built on approximated measures, which may affect the quality, it has the significant advantage to lie only on the design space, often low-dimensional compared to the uncertain space. On the other hand, the computation of a new box will always require to start a model $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}$ from scratch, unlike when the coupled space model is used.

Based on these observations, we propose a quantitative comparison of the following strategies:

- SS (Separated-space): a model $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}$ at each new design and surrogate-assisting strategy to lower cost. This is the most straightforward application of [33].
- CO (Coupled-space only): a model $\widetilde{\boldsymbol{f}}$ in both the design and uncertain dimensions. No surrogate-assisting strategy.
- CS (Coupled-space): a model $\widetilde{\boldsymbol{f}}$ in the coupled space but allowing the surrogate-assisting strategy to bypass computations if the model is converged.

For the comparison to be more comprehensive, it should be performed both when the coupled space model is particularly precise and when it shows poor accuracy. Practically, the comparison is carried out in the following cases:

- With a low-dimensional uncertain space, where the surrogate model is able to capture well the behaviour in all dimensions.
- In a harder case where the surrogate-assisting strategy will converge faster than the coupled space model, because of its inability to capture both design and uncertainty impacts precisely.

We conduct these comparisons in Section 7, where all different variants are recalled.

5.2 Reuse strategy

Two user-defined thresholds $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ have been introduced in [33] and explained in Section 3.2. The first one, $\boldsymbol{s}_1$, is the maximum surrogate-assisting error allowing for bypassing the true approximation of $\boldsymbol{\rho}$, and using the surrogate $\widetilde{\boldsymbol{\rho}}_d$ instead. Note that the subscript $\cdot_d$ refers to the surrogate-assisting model. The second threshold, $\boldsymbol{s}_2$, represents the box size (approximation conservative error $\widetilde{\boldsymbol{\varepsilon}}$) under which the approximation is considered converged and is not further refined.

Different values for these thresholds give different levels of accuracy in the optimisation output. They should be chosen *a priori* and have an essential impact on the global computational cost. Hence, the SABBa framework would greatly benefit from automation of the choice of these thresholds.

Moreover, while lower thresholds allow for high accuracy, they force a high convergence on all non-dominated boxes, even at the beginning of the optimisation process. The best designs from the initial set will, therefore, be highly refined, inducing a massive computational cost, and may be quickly dominated in the next iterations. This behaviour yields a non-negligible waste of time that will be illustrated in the following sections.

We propose a reuse capability within the SABBa framework to overcome both the need for automation and the waste of time on early boxes. Practically, it allows for reusing the set of boxes at the end of a run of SABBa to launch another run with different thresholds. For instance, we depict a double minimisation problem in Figure 8. The reuse iterations allow for rough convergence of the three first boxes (with low-cost $3c_1$), before the adding of the fourth one. The latest being the only non-dominated one, the costly refinement ($c_2 > c_1$) can be performed only on the fourth box during the next reuse iteration. This capability allows to get rough results quickly and to iteratively refine all boxes, thus getting highly converging designs from the initial set only when they show good performances *a posteriori*.

The user can here give a predefined sequence of pairs of thresholds or choose them iteratively at each iteration. The reuse capability surely has a significant impact concerning adaptability and allows the user not to choose a given value of $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ *a priori*. We explore the impact on the computational cost quantitatively in Section 7 for each test-case.

5.3 Gaussian Process greedy boxes

As mentioned in Section 3.2, the surrogate-assisting GP model is built on training data with heterogeneous gaussian noises, corresponding to the computed box sizes. This procedure allows linking the surrogate-assisting model variance to the box refinement on which the model is built. Practically, one must give the variance associated with each approximated measure $\widetilde{\boldsymbol{\rho}}$ to make a representative model $\widetilde{\boldsymbol{\rho}}_d$ in the design space. Following the link between variance and error

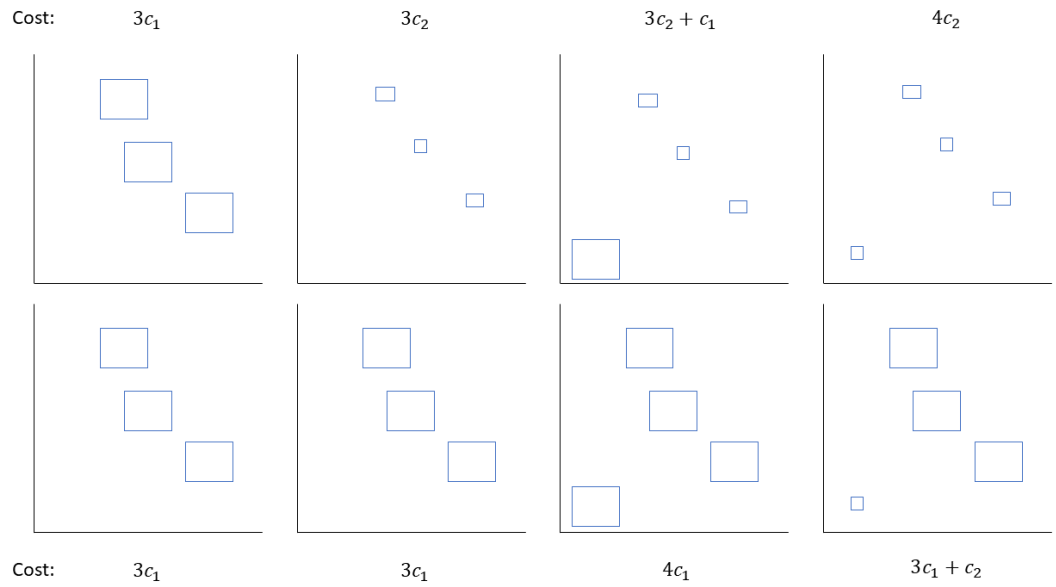


Figure 8: Cost comparison ($c_1 < c_2$). First line: No reuse capability. Second line: With reuse capability

proposed in Section 4.2, the noise associated to the training data of each box when building the GP model is defined from the conservative error of those boxes as follows:

$$\boldsymbol{\sigma}_n^2(\boldsymbol{x}) = \left(\frac{\widetilde{\boldsymbol{\varepsilon}}(\boldsymbol{x})}{3}\right)^2.$$

A heteroscedastic GP model is then built, where predictive mean $\widetilde{\boldsymbol{\rho}}_d$ and covariance Σ_d are defined for each objective or constraint dimension following the classical definitions:

$$\begin{aligned} \widetilde{\boldsymbol{\rho}}_d &= K_*^T (K + \Delta)^{-1} \widetilde{\boldsymbol{\rho}} \\ \Sigma_d &= K_{**} - K_*^T (K + \Delta)^{-1} K_* \end{aligned} \tag{6}$$

with K the covariance matrix between training points, K_{**} the one between test points and K_* the covariance matrix between test and training points. The diagonal matrix $\Delta = diag\big(\{\sigma_n^2(\boldsymbol{x}_i)\}_i\big)$ represents the nugget effect, referred to as heterogeneous gaussian noises in the above. For each dimension, the predictive variance σ_d^2 at each test point can be interpreted as the diagonal value of the matrix Σ_d. More details on GP models can be found in [36].

The coupling of this surrogate-assisting strategy with the Bounding-Box approach is performed through the redefinition of the box size as seen in Section 3. To tackle both the approximation errors on the training set for the surrogate-assisting model and the approximation error of the model itself, the box size is defined as

$$\hat{\boldsymbol{r}}(\boldsymbol{x}_i) = \widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i) + \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i).\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) \leq \boldsymbol{s}_1}(\boldsymbol{x}_i).$$

However, in this formulation, $\widetilde{\boldsymbol{\varepsilon}}^{k_{min}}$ must be known for every $\boldsymbol{x}_i$. This is not the case here as this error comes from the computation and refinement of the box and is not known *a priori*.

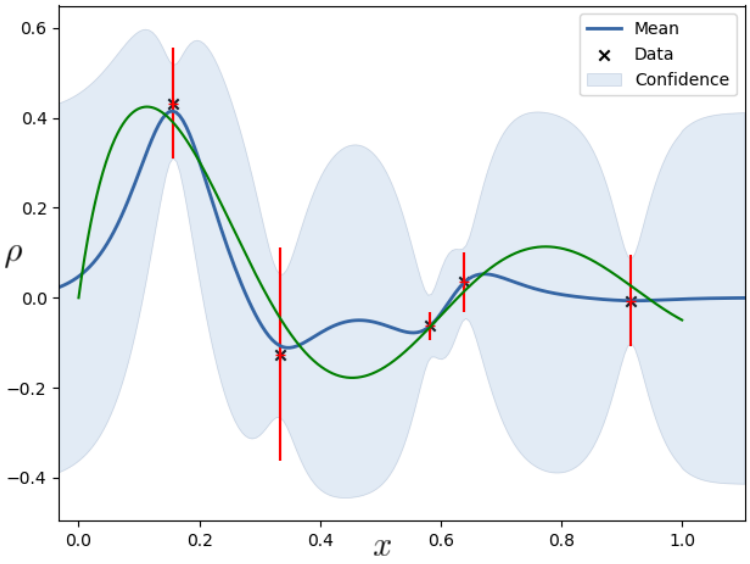


Figure 9: Heteroscedastic Gaussian Process. Underlying function in green, input noise in red and GP model in blue.

It was proposed in [33] to replace $\widetilde{\varepsilon}^{k_{min}}(\boldsymbol{x}_i)$ with $\boldsymbol{s}_2$, based on the assumption that in the Pareto optimal area, $\widetilde{\varepsilon}^{k_{min}}(\boldsymbol{x}_i) \leq \boldsymbol{s}_2$ (because the boxes are fully converged). This formulation has the advantage of being very general but usually overestimates the error when the surrogate-assisting model tackles noisy evaluations. This is notably the case when using a Gaussian Process (GP) modelling, which has the ability to deal with heterogeneous gaussian noises as seen above.

Practically, when modeling an underlying function f from observations $y_i(\boldsymbol{x}) = f_i(\boldsymbol{x}) + \epsilon_i(\boldsymbol{x})$ with $\epsilon_i(\boldsymbol{x})$ random noises, the model presented above retrieves the underlying function f and gives an associated predictive variance.

In this paper, we build the surrogate-assisting strategy on a set of approximated and disparately refined boxes. An error on the underlying true statistical measures can then be computed from a GP regression model and proposes the new "greedy" formulation for box size as follows:

$$\hat{\boldsymbol{r}}(\boldsymbol{x}_i) = \widetilde{\varepsilon}^{k_{min}}(\boldsymbol{x}_i).\mathbb{1}_{\widetilde{\varepsilon}_d^{l_{in}}(\boldsymbol{x}_i) > \boldsymbol{s}_1}(\boldsymbol{x}_i) + \widetilde{\varepsilon}_d^{l_{in}}(\boldsymbol{x}_i).\mathbb{1}_{\widetilde{\varepsilon}_d^{l_{in}}(\boldsymbol{x}_i) \leq \boldsymbol{s}_1}(\boldsymbol{x}_i) \tag{7}$$

This formulation is very straightforward algorithmically:

- If the box is computed (MCS on the model in separated or coupled space), the size of the box is the conservative error proposed in Section 4.2.

- If the box comes from the surrogate-assisting GP model because the model conservative error is low enough, this error is used as box size (following the 3σ paradigm).

This formulation should avoid high overestimation but has a higher risk of underestimation of the error. We quantify its impact on the overall framework in Section 7.

6 Algorithm

In this section, we provide all the algorithmic details for implementing the SABBa framework. The general pseudo-algorithm for low-cost optimisation under uncertainty is now presented, alongside the chosen quantitative indicator for comparing the different approaches in Section 7.

6.1 General pseudo-algorithm

We give the overall detailed algorithm with an explicit formulation of each step depicted in Algorithm 1. In Algorithm 2, $\boldsymbol{\mathcal{D}}$ represents the set of all designs studied during the optimisation process. This set is then reinitialised at each reuse iteration. $\boldsymbol{\mathcal{D}}$ is incremented by N new designs $\boldsymbol{\mathcal{D}}_{new}$ at each optimisation iteration. For these new designs, the subroutine `compareDesignMM` allows differentiating measures that can be computed with the surrogate-assisting model from the ones that must be computed from true evaluations, which are then gathered in the set $\boldsymbol{\mathcal{D}}'$. The only set which is not initialised at each reuse iteration is $\boldsymbol{\mathcal{D}}''$, keeping track of all designs which measures are computed from true evaluations (as opposed to measures from the surrogate-assisting strategy). This set is used to initialise the next optimisation processes.

Algorithm 2 Detailed pseudo-algorithm for uncertainty-based optimisation

Initialise $\boldsymbol{\mathcal{D}}''$ empty
for $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ from the reuse loop **do**
  Get $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$
  Read previous $\boldsymbol{\mathcal{D}}''$ and $\boldsymbol{\mathcal{B}}_{\hat{\boldsymbol{\rho}}}\big(\boldsymbol{\mathcal{D}}'', \hat{\boldsymbol{r}}(\boldsymbol{\mathcal{D}}'')\big)$
  $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}''$
  Launch Optimisation
  while Optimisation running **do**
    Read new designs $\boldsymbol{\mathcal{D}}_{new} = \big\{\boldsymbol{x}_i\big\}_{i=1}^{N}$
    $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}} \cup \boldsymbol{\mathcal{D}}_{new}$
    Initialise $\boldsymbol{\mathcal{D}}'$ empty
    Call `compareDesignMM`($\boldsymbol{\mathcal{D}}_{new}$, $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\widetilde{\boldsymbol{\rho}}_d$, $\widetilde{\boldsymbol{\varepsilon}}_d$, $\widetilde{\boldsymbol{\rho}}^0$, $\widetilde{\boldsymbol{\varepsilon}}^0$, $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{r}}$, $\boldsymbol{\mathcal{D}}'$) (Alg. 3)
    $\boldsymbol{\mathcal{D}}'' = \boldsymbol{\mathcal{D}}'' \cup \boldsymbol{\mathcal{D}}'$
    Compute $\boldsymbol{\mathcal{D}} = \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\hat{\boldsymbol{\rho}},0} = \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\hat{\boldsymbol{\rho}}}\big(\big(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{r}}(\boldsymbol{\mathcal{D}})\big)\big)$
    $\boldsymbol{\mathcal{D}}' = \boldsymbol{\mathcal{D}} \cap \boldsymbol{\mathcal{D}}'$
    $k = 0$
    while $\max\limits_{\boldsymbol{x}\in\boldsymbol{\mathcal{D}}'}\big[\widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x})\big] > \boldsymbol{s}_2$ **do**
      $k = k + 1$
      Select $\boldsymbol{x}_r = \underset{\boldsymbol{x}\in\boldsymbol{\mathcal{D}}'}{\operatorname{argmax}}\big[POP_{min}(\boldsymbol{x}).\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x})>\boldsymbol{s}_2}\big]$ to refine
      Call `iterateRefinement`($\boldsymbol{\mathcal{D}}'$, $\boldsymbol{x}_r$, $\widetilde{\boldsymbol{\rho}}^{k-1}$, $\widetilde{\boldsymbol{\varepsilon}}^{k-1}$, $\widetilde{\boldsymbol{\rho}}^k$, $\widetilde{\boldsymbol{\varepsilon}}^k$) (Alg. 4)
      for each $\boldsymbol{x} \in \boldsymbol{\mathcal{D}}'$ **do**
        $\hat{\boldsymbol{\rho}}(\boldsymbol{x}) = \widetilde{\boldsymbol{\rho}}^k(\boldsymbol{x})$
        $\hat{\boldsymbol{r}}(\boldsymbol{x}) = \widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x})$
      end for
      Compute $\boldsymbol{\mathcal{D}} = \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\hat{\boldsymbol{\rho}},k} = \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\hat{\boldsymbol{\rho}}}\Big(\Big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\hat{\boldsymbol{\rho}},k-1}, \hat{\boldsymbol{r}}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\hat{\boldsymbol{\rho}},k-1}\big)\Big)\Big)$
      $\boldsymbol{\mathcal{D}}' = \boldsymbol{\mathcal{D}} \cap \boldsymbol{\mathcal{D}}'$
    end while
    Update $\widetilde{\boldsymbol{\rho}}_d$ and $\widetilde{\boldsymbol{\varepsilon}}_d$
  end while
  Save $\boldsymbol{\mathcal{D}}''$ and $\boldsymbol{\mathcal{B}}_{\hat{\boldsymbol{\rho}}}\big(\boldsymbol{\mathcal{D}}'', \hat{\boldsymbol{r}}(\boldsymbol{\mathcal{D}}'')\big)$
end for
Return $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{B}}_{\hat{\boldsymbol{\rho}}}\big(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{r}}(\boldsymbol{\mathcal{D}})\big)$

In Algorithm 2, one can see that $\boldsymbol{\mathcal{D}}$ represents the non-dominated set sequence defined in Equation (1). Both $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}}'$ only contain non-dominated designs and are updated after every refinement at each optimisation iteration.

Algorithm 3 deals with the first approximation of the uncertainty-based measures and the use (or not) of the surrogate-assisting model. If the surrogate error is below the user-defined threshold $\boldsymbol{s}_1$ for all objective and constraint dimensions, boxes can be approximated, with the classical or the greedy box size. Else, in the case of *separated space*, N_{first} evaluations are performed to build an initial model to compute approximated measures. In the case of a *coupled space*, the existing model is directly used for Monte Carlo measure approximation.

Algorithm 3 `compareDesignMM`($\boldsymbol{\mathcal{D}}_{new}$, $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\widetilde{\boldsymbol{\rho}}_d$, $\widetilde{\boldsymbol{\varepsilon}}_d$, $\widetilde{\boldsymbol{\rho}}^0$, $\widetilde{\boldsymbol{\varepsilon}}^0$, $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{r}}$, $\boldsymbol{\mathcal{D}}'$)

for each $\boldsymbol{x} \in \boldsymbol{\mathcal{D}}_{new}$ **do**
  if $\widetilde{\boldsymbol{\varepsilon}}_d(\boldsymbol{x}) \leq \boldsymbol{s}_1$ **then**
    $\hat{\boldsymbol{\rho}}(\boldsymbol{x}) = \widetilde{\boldsymbol{\rho}}_d(\boldsymbol{x})$
    if greedy boxes **then**
      $\hat{\boldsymbol{r}}(\boldsymbol{x}) = \widetilde{\boldsymbol{\varepsilon}}_d(\boldsymbol{x})$
    else
      $\hat{\boldsymbol{r}}(\boldsymbol{x}) = \boldsymbol{s}_2 + \widetilde{\boldsymbol{\varepsilon}}_d(\boldsymbol{x})$
    end if
  else
    if separated spaces **then**
      Evaluate $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})$ at N_{first} LHS different $\boldsymbol{\xi}$
      Build $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})$
      Compute $\widetilde{\boldsymbol{\rho}}^0(\boldsymbol{x})$ and $\widetilde{\boldsymbol{\varepsilon}}^0(\boldsymbol{x})$ from $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})$
    else
      if coupled model does not exist yet **then**
        Evaluate $\boldsymbol{f}(\boldsymbol{x}_{init}, \boldsymbol{\xi})$ at N_{first} LHS different $\boldsymbol{\xi}$ for all $\boldsymbol{x}_{init} \in \boldsymbol{\mathcal{D}}_{new}$
        Build $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$ from previous evaluations
      end if
      Get $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$
      Compute $\widetilde{\boldsymbol{\rho}}^0(\boldsymbol{x})$ and $\widetilde{\boldsymbol{\varepsilon}}^0(\boldsymbol{x})$ from $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$
    end if
    $\hat{\boldsymbol{\rho}}(\boldsymbol{x}) = \widetilde{\boldsymbol{\rho}}^0(\boldsymbol{x})$
    $\hat{\boldsymbol{r}}(\boldsymbol{x}) = \widetilde{\boldsymbol{\varepsilon}}^0(\boldsymbol{x})$
    $\boldsymbol{\mathcal{D}}' = \boldsymbol{\mathcal{D}}' \cup \boldsymbol{x}$
  end if
end for
Return $\widetilde{\boldsymbol{\rho}}^0$, $\widetilde{\boldsymbol{\varepsilon}}^0$, $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{r}}$ and $\boldsymbol{\mathcal{D}}'$

The subroutine `iterateRefinement` is depicted in Algorithm 4. It performs N_{new} evaluations at a given design $\boldsymbol{x}_r$ maximising the Pareto Optimal Probability (as seen in Alg. 2). The coordinates $\boldsymbol{\xi}$ in the uncertain space at which evaluations are performed are given by maximising the refinement criterion defined in Equation (5). Note that in the case of a *separated space* paradigm, the other measure approximations are unchanged while in the case of *coupled space*, the global model is updated and measures are recomputed using MCS at their respective designs.

Algorithm 4 `iterateRefinement`($\mathcal{D}'$, $\boldsymbol{x}_r$, $\widetilde{\boldsymbol{\rho}}^{k-1}$, $\widetilde{\boldsymbol{\varepsilon}}^{k-1}$, $\widetilde{\boldsymbol{\rho}}^{k}$, $\widetilde{\boldsymbol{\varepsilon}}^{k}$)

for each $\boldsymbol{x} \in \mathcal{D}'$ **do**
  if $\boldsymbol{x} = \boldsymbol{x}_r$ **then**
    Compute N_{new} locations $\boldsymbol{\xi}_i$ maximising the refinement criterion
    Evaluate $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi})$ at these $\boldsymbol{\xi}_i$
    Refine the surrogate $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})$ or $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$
    Compute $\widetilde{\boldsymbol{\rho}}^k(\boldsymbol{x})$ and $\widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x})$ from the new model
  else
    if separated spaces **then**
      $\widetilde{\boldsymbol{\rho}}^k(\boldsymbol{x}) = \widetilde{\boldsymbol{\rho}}^{k-1}(\boldsymbol{x})$
      $\widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x}) = \widetilde{\boldsymbol{\varepsilon}}^{k-1}(\boldsymbol{x})$
    else
      Compute $\widetilde{\boldsymbol{\rho}}^k(\boldsymbol{x})$ and $\widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x})$ from the new coupled model after refinement at $\boldsymbol{x}_r$
    end if
  end if
end for
Return $\widetilde{\boldsymbol{\rho}}^k$ and $\widetilde{\boldsymbol{\varepsilon}}^k$

Remark We recall that in the above, functions $\boldsymbol{f}$ refer to all deterministic outputs of the black-box on which both robustness and reliability-based measures are computed.

One can note that the algorithm relies on a quite small set of parameters, which are:

- The predefined sequence of pair of thresholds $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ for the reuse strategy;
- The number of function evaluations N_{first} for the first measures computations and N_{new} for measures refinements;
- The number of designs N at each optimisation iteration and N_{init} at the first iteration. The optimiser may impose these parameters.

The SABBa framework for optimisation under uncertainty problems is applied to analytical and applicative test-cases to assess its performance against classical approaches. The quality of optimisation outputs must be quantified to conduct the comparisons. For this reason, we define a quantitative quality indicator in the following section.

6.2 Quantitative quality indicator

The comparison of two Pareto fronts could be challenging since several metrics could be used to judge whether a front is well approximated or not. This section aims to propose a quantitative indicator, which is built to assess and compare the convergence of the framework in analytical test-cases where the Pareto optimal designs are known.

6.2.1 Modified Hausdorff distance

A classical technique to quantify the convergence of a multi-objective optimiser is the computation of the Hausdorff distance between the algorithm optima and the true ones. The Hausdorff distance can geometrically be interpreted as a measure of the distance between two subsets of a given metric space. In its most general form, we can write the distance as follows:

$$d_H(\mathcal{A}, \mathcal{B}) = f\big(d_1(\mathcal{A}, \mathcal{B}), d_1(\mathcal{B}, \mathcal{A})\big)$$

with the function f usually being the max function, $f(u, v) = \max(u, v)$. The distance d_1 is defined in the classical Hausdorff distance as the maximum distance from an element of the first set to the second:

$$d_1(\mathcal{A}, \mathcal{B}) = \max_{a \in \mathcal{A}}[d_2(a, \mathcal{B})],$$

and lastly the distance d_2 is the minimal distance between a given point and an element of a set:

$$d_2(a, \mathcal{B}) = \min_{b \in \mathcal{B}}[d(a, b)].$$

The distance d here is the Euclidian norm, *i.e.* $d(a, b) = \|a - b\|_2$.

However, d_1 being a maximum, the classical Hausdorff distance cannot capture the overall similarities as long as an element of a set remains far from the other set. A paper of Dubuisson et al. [37] deals with this issue and proposes the so-called modified Hausdorff distance, after comparing twenty-four (24) variations. The modified Hausdorff distance d'_H uses a new definition of d_1 and replaces the maximum by a sample average:

$$d'_1 = \frac{1}{N} \sum_{a \in \mathcal{A}} d_2(a, \mathcal{B}).$$

Therefore, in cases where the Pareto optimal designs $\mathcal{X}_\mathcal{P}$ are known and approximated by the optimiser optimal design set $\mathcal{X}_{\widetilde{\mathcal{P}}}$, a quantitative indicator of the quality of the output of the optimisation process is the modified Hausdorff distance between those two sets. This indicator should converge toward zero, and its logarithm is usually used:

$$Q = \log\left(d'_H(\mathcal{X}_\mathcal{P}, \mathcal{X}_{\widetilde{\mathcal{P}}})\right).$$

Since the SABBa framework returns boxed objective values, with non-binary Pareto Optimal Probability, an extension of this quality indicator to the context of boxed values is required, which is presented in the next section.

6.2.2 Aleatory Pareto optima

In the context of boxed uncertainty-based measures approximation, we recall that the true values of the objective and constraint functions are assumed to be aleatory. To minimise the a priori assumptions - or maximise the entropy - we assume the true values to follow a uniform distribution within their associated bounding box. In other words, $\forall \boldsymbol{x}, \forall j \in \mathcal{I}_1^m$,

$$\rho_j(\boldsymbol{x}) \sim \mathcal{U}\left([\hat{\rho}_j(\boldsymbol{x}) - r_j(\boldsymbol{x}), \hat{\rho}_j(\boldsymbol{x}) + r_j(\boldsymbol{x})]\right).$$

Hence, the Pareto front approximation $\widetilde{\mathcal{P}} = \left\{\boldsymbol{\rho}(\boldsymbol{x}_i), i \in \mathcal{I}_1^N \;\middle|\; \nexists k \in \mathcal{I}_1^N, \boldsymbol{x}_j \overset{\boldsymbol{\rho}}{\succ}_c \boldsymbol{x}_i \;\&\; \boldsymbol{\rho}(\boldsymbol{x}_i) \in \mathcal{F}^{\complement}\right\}$ is not deterministic, as well as its design counterpart. However, many $\widetilde{\mathcal{P}}$ can be sampled by taking for each design $\boldsymbol{x}$ independent samples $\rho_j(\boldsymbol{x})$ for all j from the previous uniform law. Each set of draws defines a given Pareto front discrete approximation, and its associated Pareto optimal designs $\mathcal{X}_{\widetilde{\mathcal{P}}}$. In light of the possibility of sampling $\widetilde{\mathcal{P}}$ and $\mathcal{X}_{\widetilde{\mathcal{P}}}$, the boxed — or probabilistic — quality indicator is computed as the Monte Carlo approximation of the following expected modified Hausdorff distance:

$$Q_\mathcal{B} = \log\left(\mathbb{E}_{\mathcal{X}_{\widetilde{\mathcal{P}}}}\left[d'_H(\mathcal{X}_\mathcal{P}, \mathcal{X}_{\widetilde{\mathcal{P}}})\right]\right). \tag{8}$$

Remark It is immediate to see that this probabilistic indicator, when applied to deterministic — non-boxed — data, is consistent with the indicator Q. Indeed, as a deterministic value is equivalent to a box centred on this value and of size 0, each Pareto front sample is the deterministic one and $Q_{\mathcal{B}}$ reduces to $Q_{\mathcal{B}} = \log\left(\mathbb{E}_{\mathcal{X}_{\tilde{\mathcal{P}}}}\left[10^{Q}\right]\right) = Q$.

The next section deals with the application of the SABBa framework to analytical and engineering test-cases. The defined quality indicator is heavily employed to compare the different strategies. Examples of indicator values with associated Pareto front visualisations are given in Figs 14 to 16 and 19 to 21 (indicator values are given in the text).

7 Applications and comparisons

We apply here SABBa to two analytical test-cases and a physical engineering application. We quantitatively compare the framework, and all its variations (described in Section 5) to the classical *Double-loop*, and the *A Priori MetaModel* approaches employing the quality indicator defined in Section 6.2.

The SABBa framework should be able to deal with multi-objective and constrained optimisation problems, and also to deal efficiently with both low- high-dimensional problems. For this reason, the first analytical test-case is low-dimensional and deals with the Taguchi multi-objective robustness formulation (mean optimisation and variance minimisation). The second test-case features a higher number of dimensions in the context of a mean optimisation under quantile constraint. Finally, the engineering test-case deals with a dozen dimensions and is a mean mass density minimisation under a maximal constraint for the temperature in the uncertain dimensions.

We systematically compare the performance of the proposed framework to two reference strategies. In the Double-loop approach, also called Nested-loop or one-layer kriging model approach, a full uncertainty propagation is performed at each optimisation iteration. The associated cost for N_{opt} optimisation iterations and N_{UQ} evaluations to compute the statistical measures at each design $\boldsymbol{x}$ is therefore $N_{opt} \times N_{UQ}$ times the unitary cost of a black-box evaluation. The second reference approach is called A Priori MetaModel. In this case, we build a surrogate model of the black-box output in the *coupled space* before the optimisation process. The Double-loop is then performed on the surrogate instead of the true evaluations at a negligible cost. The overall computational cost N_{mm} times the unitary cost is here chosen *a priori* and can be adapted to the available budget, with a substantial impact on the accuracy of the results.

We compare here several variants formulated in Section 5, denoted as follows:

- CS-REU: when using coupled space model for measure approximations and looping over a predefined sequence of thresholds;
- SS-REU: when using many models in separated spaces and the reuse strategy;
- CS-REU-G: when using coupled space model, reuse strategy and greedy box error approximation;
- SS-REU-G: when using separated spaces models, reuse strategy and greedy box error approximation;
- CS-G: when using coupled space model and greedy box error approximation but no reuse strategy;
- CO-REU: when using coupled space model only (no Surrogate-Assisting strategy) and the reuse strategy.

Among the above variants, one can expect CS to give better results than SS and CO, taking advantage of both the low-dimensional surrogate-assisting model and the coupled space correlation to speed up box refinement. REU should also allow for more excellent intermediate results and an increased overall parsimony. Finally, G takes advantage of the Gaussian processes inherent capability of dealing with noisy data and should also lower the computational cost. Hence, one should expect CS-REU-G to give the most parsimonious results in the following.

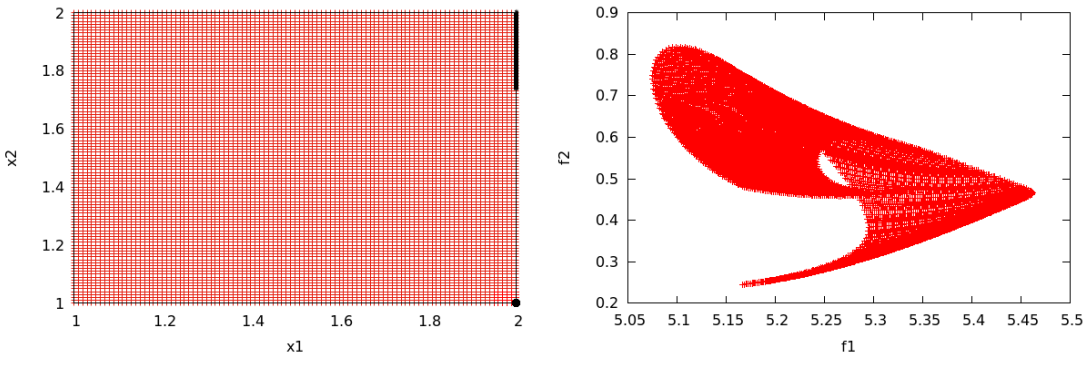
Note that ten runs are performed for each strategy to capture both the mean convergence curve and the associated variability, represented as a translucent band around the mean. We consider different seeds for each run in terms of initial samplings, both for the N_{init} coordinates in the design space and the N_{first} evaluations at each design point. The log distance is assumed to show a Gaussian distribution over the repeated runs, implying a log-normal distribution over the actual distance. High variability in the following figures will mean a lack of reliability of the associated approaches. The optimisation runs are considered converged when the log probabilistic modified Hausdorff distance $Q_{\mathcal{B}}$ reaches roughly a value of -2.

7.1 Test-case 1: Unconstrained Taguchi optimisation

This problem is a bi-objective robust optimisation proposed in [32]. There are two design variables $\boldsymbol{x}$ and one uncertain parameter ξ, and the problem reads:

$$\begin{aligned}
\text{minimise:} \quad & \boldsymbol{\rho_f}(\boldsymbol{x}) = \begin{pmatrix} \mu(\boldsymbol{x}) \\ \sigma^2(\boldsymbol{x}) \end{pmatrix} \\
\text{where:} \quad & \mu(\boldsymbol{x}) = \mathbb{E}_\xi[f(\boldsymbol{x},\xi)] \\
& \sigma^2(\boldsymbol{x}) = \mathbb{V}_\xi[f(\boldsymbol{x},\xi)] \\
\text{with:} \quad & f(\boldsymbol{x},\xi) = \xi - x_1\xi^5 + \cos(2\pi x_2 \xi) + 5 \\
& \xi \sim \mathcal{U}([0,1]) \\
\text{by changing:} \quad & (x_1, x_2) \in [1,2]^2
\end{aligned} \tag{9}$$

One can see in the following that the Pareto front associated with this problem is discontinuous, and the optimal set in the design space is the union of a segment and a point (see Fig. 10).



(a) (b)

Figure 10: Test-case 1: a) Discretisation of the design space in red and Pareto optimal sets in black. b) Image of the discretised points in the objective (μ,σ^2) space.

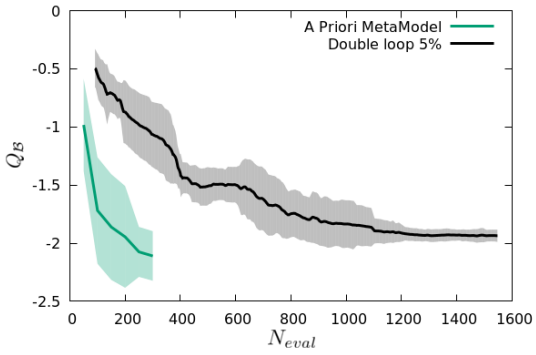


Figure 11: Test-case 1: Cost comparison between the A Priori MetaModel and Double loop approach.

The two reference strategies are compared quantitatively in Figure 11. The Double-loop approach shows a significantly slower convergence, which is not surprising as most of the recent uncertainty-based optimisation is performed using the A Priori MetaModel (APMM) strategy. For this reason, we compare the SABBa framework to the APMM strategy in the following.

For this first test-case, we consider two different surrogate-modelling capabilities. The aim is to illustrate SABBa performances both when the surrogate is able to represent the underlying functions accurately and when the surrogate shows poor convergence. Note that the quality of the surrogate yields consequences on the APMM model $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$, the surrogate-assisting model $\widetilde{\boldsymbol{\rho}}_d(\boldsymbol{x})$ and the separated or coupled space model $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})$ and $\widetilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{\xi})$. With a high-quality surrogate model, SABBa can make use of the *coupled space* model to quickly converge toward the optimal area. In the other case, the surrogate-assisting strategy, which has fewer dimensions than the *coupled space*, should be the greatest source of cost reduction and SABBa should converge much faster than the APMM strategy.

7.1.1 With high-quality surrogate model

The unconstrained Taguchi optimisation problem is first performed with a high-quality meta-modeling approach. Practically, we build the GP model by means of a classical RBF kernel (also called squared-exponential, exponential quadratic or Gaussian), with one lengthscale by dimension:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{r_{\boldsymbol{l}}^2}{2} \right)$$
$$\text{with } r_{\boldsymbol{l}} = \sqrt{\sum_{i=1}^{m} \left(\frac{x_i - x_i'}{l_i} \right)^2}$$

We recall that the kernel function allows computing the covariance matrix in Equation (6).

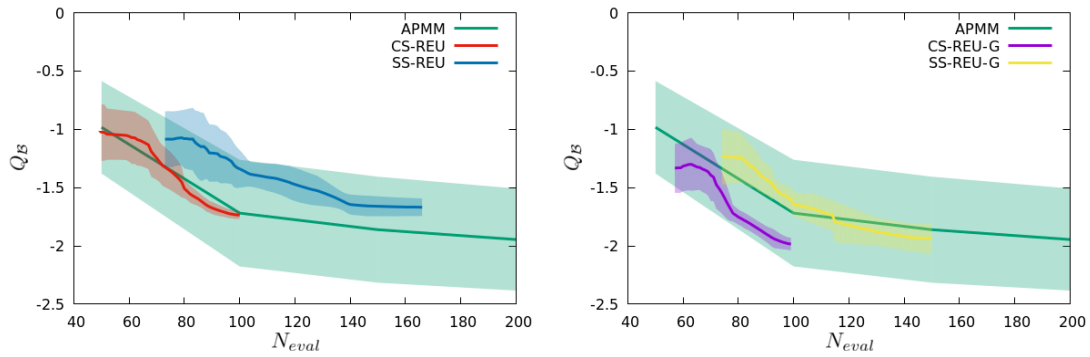
This model requires $m+1$ hyperparameters $\{\sigma^2, l_1, \ldots, l_m\}$ to be optimised but captures the characteristic length scale associated with each input dimension. Hence, such a model performs significantly better when using the APMM strategy or the SABBa framework with *coupled space* model, which heavily rely on the quality of the surrogate model.

The convergence curve concerning log-scaled probabilistic modified Hausdorff distance to the optimum, defined in Equation (8), is plotted in Figure 12. In these figures, we compare the

convergence of the APMM strategy, SABBa with coupled space model (CS-REU and CS-REU-G) and SABBa with separated spaces models (SS-REU and SS-REU-G). One can see in both Figures 12(a) and 12(b) that SABBa shows a very reduced variability compared to the APMM strategy. However, only SABBa CS-REU-G (purple curve in Fig. 12(b)) shows a significant improvement for the mean convergence. Note that here, the reuse capability is used 5 times for each run. With h_i the range covered in the i^{th}-dimension, namely,

$$\forall i, h_i = \max_{\boldsymbol{x}}[\rho_i(\boldsymbol{x})] - \min_{\boldsymbol{x}}[\rho_i(\boldsymbol{x})],$$

the thresholds $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are given in percentage of the ranges $\boldsymbol{h}$. More precisely, $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are chosen equal at each reuse iteration and are sequentially taken as 50%, 40%, 30%, 20%, 10% and 5%. Hence, at the final level of refinement, boxes widths are inferior to 5% of the spanned range h_i in each dimension. Note also that the parameters from Section 6.1 are chosen as follows: $N_{init} = 10$, $N = 1$ (sequential optimiser), $N_{first} = 5$ and $N_{new} = 1$.



(a) a (b) b

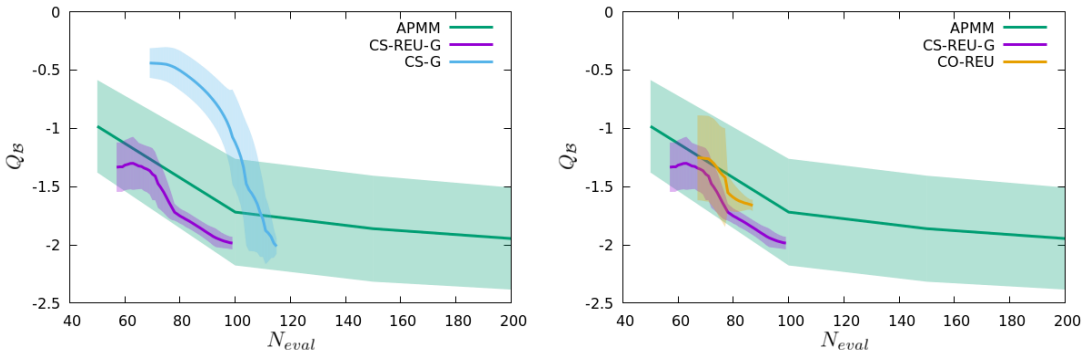
Figure 12: Test-case 1-a: Cost comparison between APMM and SABBa in separated or coupled space with a) classical surrogate boxes (CS-REU and SS-REU) or b) greedy boxes (CS-REU-G and SS-REU-G).

The most efficient strategy from Figure 12, *i.e.* SABBa CS-REU-G, is then compared in Figure 13 alongside the APMM strategy to other variants from Section 5: SABBa CS-G and CO-REU. Note that there is no difference between standard and greedy boxes in CO-REU as the SA strategy is not used.

In Figure 13(a), the approach is compared when using or not the reuse strategy alongside coupled space model and greedy boxes (CS-REU-G and CS-G). When reuse is not employed, $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are directly fixed to 5%. One can see that the use of the reuse capability allows for two main improvements. Firstly, it yields intermediate results, that is to say, well-converged optima before the 5% refinement. For instance, at $N_{eval} = 80$, SABBa with reuse capability gives a log-scaled distance of approximately -1.75 against -0.5 without reuse. Secondly, as seen in Figure 8, the reuse capability further increases the parsimonious behaviour of the SABBa framework. Here, final mean convergence is reached before $N_{eval} = 100$ with reuse against $N_{eval} = 115$ without reuse.

Then, Figure 13(b) illustrates the behaviour of the framework without the SA strategy (CS-REU-G against CO-REU). One can see that in the context of a high-quality surrogate model, the BB approach without surrogate-assisting strategy gives results quite similar to the SABBa framework, with a low variability when 5% refinement reached. However, the opportunity to

correlate directly the statistical measures $\boldsymbol{\rho}$ in the design space when using the SA strategy yields in average less expensive first approximations ($N_{eval} = 57$ against $N_{eval} = 67$) for a slightly better quality indicator (-1.33 against -1.25). The use of the SA strategy also shows a higher final convergence of -2 against -1.66. The slightly superior behaviour of the SABBa framework compared to the BB approach only is not surprising as SABBa uses the SA strategy to accelerate the convergence whenever it is possible. Hence, SABBa should always be at least as effective as the BB approach only. The consistent gain provided by the SA strategy is illustrated in the next section when considering a surrogate model unable to capture the underlying function with high accuracy (see Fig. 18).



(a) (b)

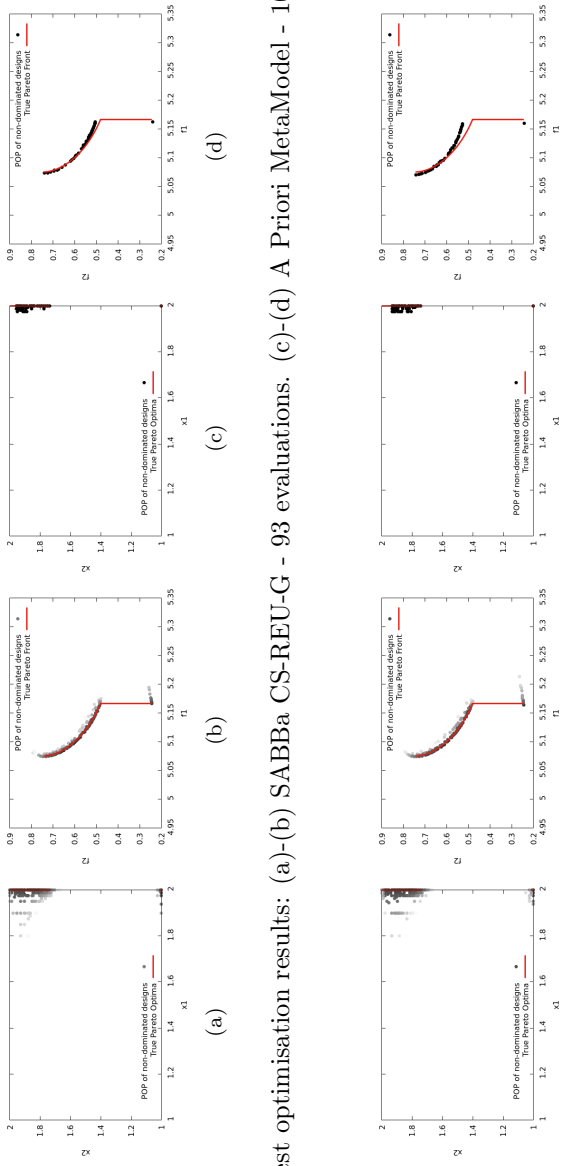
Figure 13: Test-case 1-a: Cost comparison between APMM, SABBa CS-REU-G and a) CS-G or b) CO-REU.

As stated previously, the log-normal distribution of the probabilistic modified Hausdorff distance deriving from the normality of the $Q_{\mathcal{B}}$ indicator makes approaches with high variability very unreliable. This fact is due to the heavy tail characteristic of the log-normal distribution.

In the above, one can see that the APMM strategy shows a fairly good average convergence, but has the highest variance. It makes this approach very risky, as it can be seen in Figures 14, 15 and 16, where we plot the optimisation outputs of the best, median and worst runs out of ten for the APMM strategy and the SABBa CS-REU-G. We draw these plots considering a number of evaluations of the order of $N_{eval} = 100$.

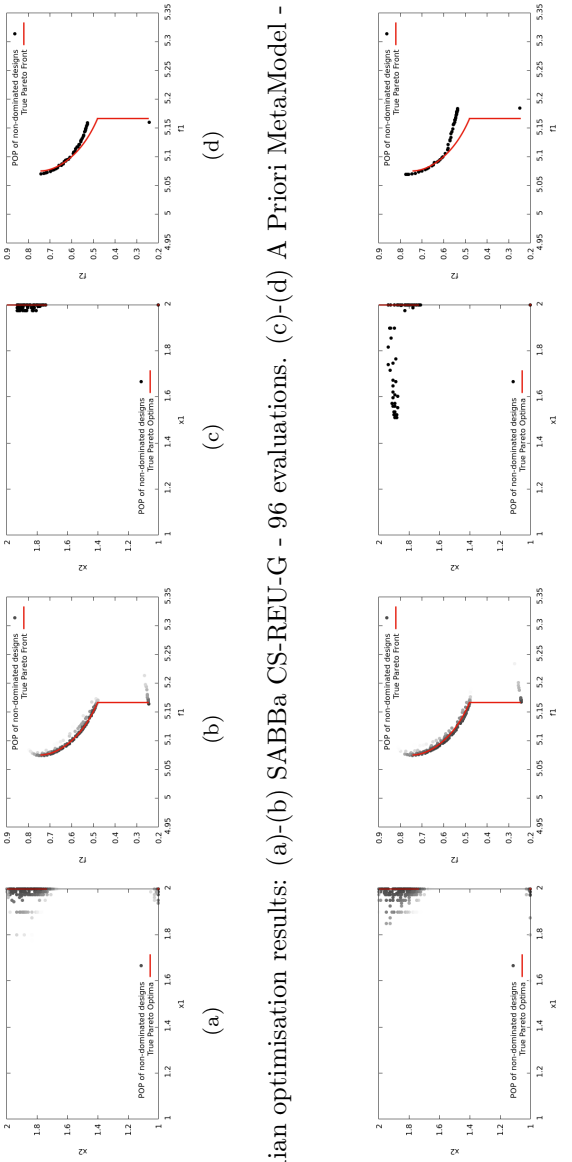
In the following, optimal designs returned by SABBa are plotted in greyscale while the ones from the APMM strategy are black. Indeed, the Pareto Optimal Probability associated with designs within the SABBa framework ranges continuously from 0 to 1 (with associated greyscale) based on the POP_{min} computation. Designs from the APMM strategy are deterministic and are either dominated or Pareto-optimal, *i.e.* $POP = 0$ (invisible individual) or 1 (black dot).

The best, median and worst runs using SABBa are associated with $Q_{\mathcal{B}} = -2.07$, -1.99 and -1.89 respectively. The values for the APMM strategy are -2.12, -1.89 and -0.72. The SABBa framework appears to be far more robust to its initial steps than the APMM strategy, which matches the observations made in Figures 12 and 13 about the high variability of the APMM approach. This fact is mainly due to the adaptive characteristic of SABBa, which allows for a consistent high refinement in the vicinity of the Pareto front while the APMM is very dependent to the Design Of Experiment used for the surrogate model construction.



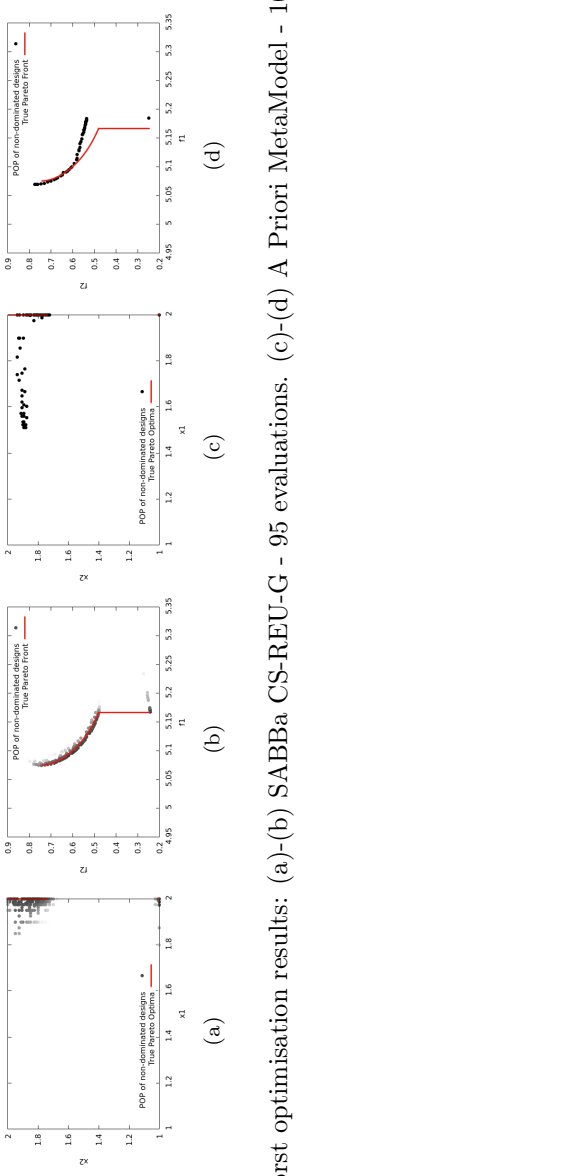
(a) (b) (c) (d)

Figure 14: Best optimisation results: (a)-(b) SABBa CS-REU-G - 93 evaluations. (c)-(d) A Priori MetaModel - 100 evaluations.



(a) (b) (c) (d)

Figure 15: Median optimisation results: (a)-(b) SABBa CS-REU-G - 96 evaluations. (c)-(d) A Priori MetaModel - 100 evaluations.



(a) (b) (c) (d)

Figure 16: Worst optimisation results: (a)-(b) SABBa CS-REU-G - 95 evaluations. (c)-(d) A Priori MetaModel - 100 evaluations.

7.1.2 With a low-quality surrogate model

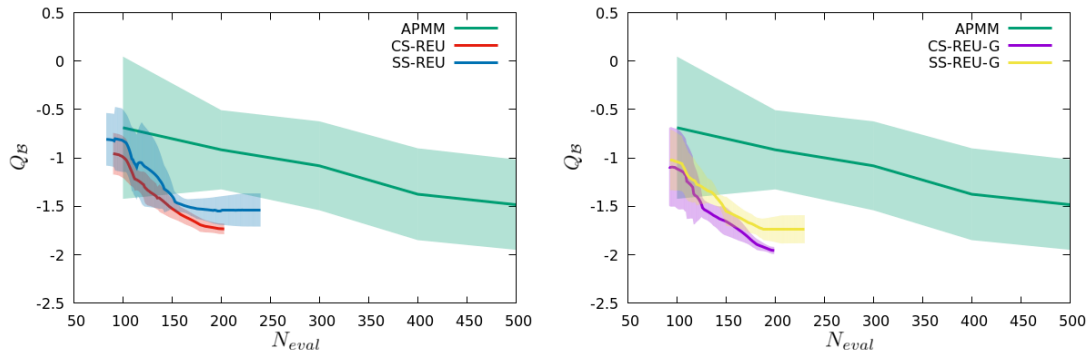
In the following, the same optimisation problem is solved using a low-quality surrogate model both with SABBa and the APMM strategy. Contrarily to the previous formulation, we build the kernel of the GP model with only one length scale, that has to account for all dimension. Practically,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{r_l^2}{2} \right)$$
$$\text{with } r_l = \frac{1}{l}\sqrt{\sum_{i=1}^{m} \left(x_i - x_i'\right)^2}$$

In this case, we should optimise only two hyperparameters $\{\sigma^2, l\}$. However, this model will notably fail when the characteristic lengths of the different dimensions are very disparate. This test-case aims at modelling problems where the coupled space surrogate is hard to converge as a whole. One can expect here the SA strategy to yield a significant cost improvement through low-dimensional measures representation (only in the design space). This kind of behaviour, simulated here through an inefficient surrogate model, would naturally arise when the number of uncertain dimensions is far greater than the number of design dimension so that the *coupled space* model is harder to converge than the SA model. It should be noted that such problems (with $d_{design} \ll d_{uncertainty}$) are very common.

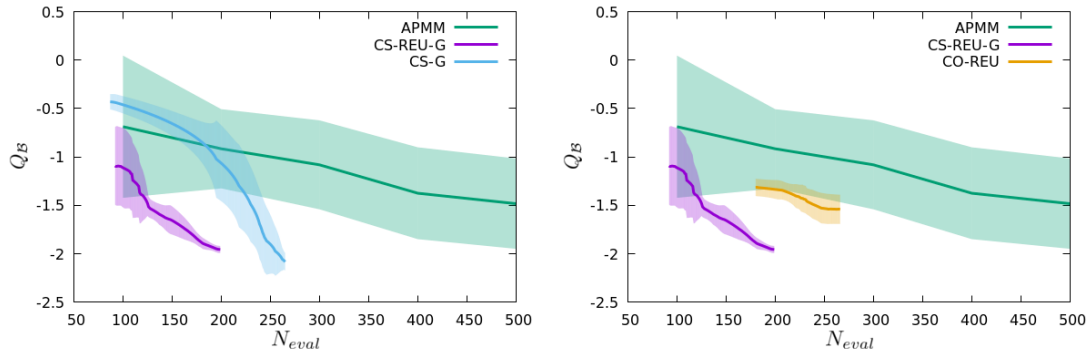
The results are given in Figures 17 and 18. The graphs follow the same pattern as above (Figs. 12 and 13). In Figure 17, the separated spaces and coupled space SABBa are compared to the APMM strategy. We can first note that the use of a low-quality surrogate model induces, for all approaches, a significant increase in the number of evaluations needed for optimisation convergence. The APMM strategy notably needs 500 evaluations to roughly reach an average log-scaled modified Hausdorff distance of -1.5 against -2 in 200 evaluations (see Figure 12(a)). SABBa CS-REU-G needs around 200 evaluations to converge instead of 100 previously but remains the most efficient approach. The difference with SS-REU-G is significantly lower compared to the previous application. This fact is due to the noticeable impact of the SA strategy in the fast convergence of both the coupled space and separated spaces approaches. We can observe this impact in Figure 18(b), where CO-REU shows relatively poor performances compared to the SABBa framework. Finally, one can see in Figure 18(a) that when the reuse capability is unused, the SABBa framework lacks accurate intermediate optima and needs additional evaluations to reach convergence.

As for the first application, with high-quality surrogates, the best, median and worst optimisation outputs are plotted both for the coupled space SABBa with greedy boxes and reuse strategy and for the APMM approach. One can see in Figures 19 to 21 that SABBA (CS-REU-G) allow for high consistency and high accuracy in capturing the Pareto-optimal area and the associated Pareto front. On the contrary, at $N_{eval} = 200$, even the best APMM strategy output shows high discrepancy with the real Pareto optima, notably in the measures space. The quality indicator associated with the SABBa outputs (APMM approach) are -2.02 (-1.31), -1.96 (-0.81) and -1.91 (-0.10), respectively.



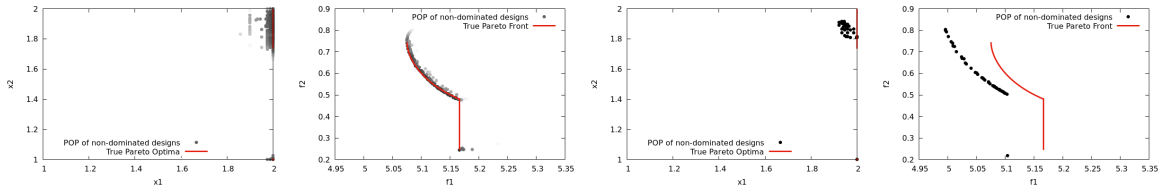
(a) (b)

Figure 17: Test-case 1-b: Cost comparison between APMM and SABBa in separated or coupled space with a) classical surrogate boxes (CS-REU and SS-REU) or b) greedy boxes (CS-REU-G and SS-REU-G).



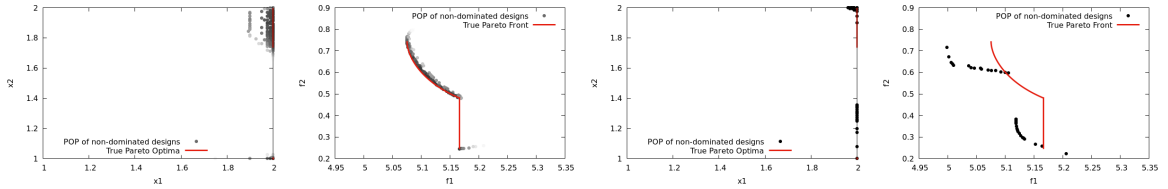
(a) (b)

Figure 18: Test-case 1-b: Cost comparison between APMM, SABBa CS-REU-G and a) CS-G or b) CO-REU.



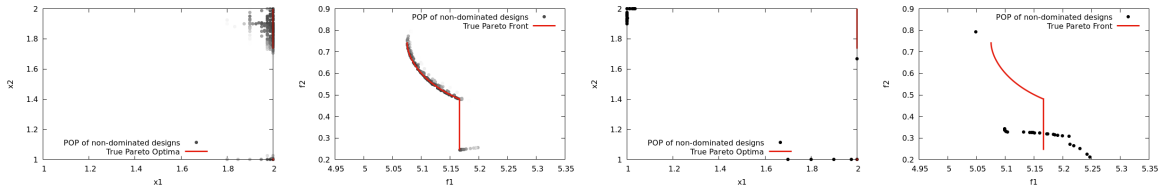
(a) (b) (c) (d)

Figure 19: Best optimisation results: (a)-(b) SABBa CS-REU-G - 197 evaluations. (c)-(d) A Priori MetaModel - 200 evaluations.



(a) (b) (c) (d)

Figure 20: Median optimisation results: (a)-(b) SABBa CS-REU-G - 196 evaluations. (c)-(d) A Priori MetaModel - 200 evaluations.



(a) (b) (c) (d)

Figure 21: Worst optimisation results: (a)-(b) SABBa CS-REU-G - 196 evaluations. (c)-(d) A Priori MetaModel - 200 evaluations.

These two applications of the first test-case, with high-quality and low-quality surrogates, allow drawing some conclusions on the behaviour of the SABBa framework. Because of the superior response of the APMM strategy over the Double loop, we have chosen to compare SABBa to APMM only.

From the results, we state that the most efficient variant of the framework consists in using SABBa CS-REU-G, with coupled space surrogate models (described in Section 5.1), with greedy boxes size (described in Section 5.3), and utilising the reuse capability (presented in Section 5.2). With these choices, the framework performs consistently better than the APMM strategy when averaging over several runs. Most of all, SABBa shows far higher consistency and robustness than APMM, with very low variability over repeated runs, and thus similar best and worst optimisation outputs. Finally, SABBa, and notably the SA strategy, shows the most significant improvements compared to the APMM approach when dealing with stiff-to-model underlying functions. This stiffness can happen when using a low-quality surrogate model or equivalently when coping with a high-dimensional uncertainty space.

7.2 Test-case 2: Quantile-constrained mean performance optimisation

We propose this second test-case to assess the performance of SABBa in a higher dimensional case and with the presence of a reliability-based constraint (derived from the Six-Hump Camel function) in addition to a robust objective (derived from a simplified Rosenbrock function). We consider here four design variables and three uncertain parameters. The problem is stated as follows:

$$
\begin{aligned}
\text{minimise:}\quad & \boldsymbol{\rho_f}(\boldsymbol{x}) = \mu(\boldsymbol{x}) \\
\text{satisfying:}\quad & \boldsymbol{\rho_g}(\boldsymbol{x}) = \mathrm{q}^{95\%}(\boldsymbol{x}) \leq 1 \\
\text{where:}\quad & \mu(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[f(\boldsymbol{x},\boldsymbol{\xi})] \\
& \mathrm{q}^{95\%}(\boldsymbol{x}) = \mathrm{q}^{95\%}_{\boldsymbol{\xi}}[g(\boldsymbol{x},\boldsymbol{\xi})] \\
\text{with:}\quad & f(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{i=1}^{3}\left[\big(1-x_i\big) + 3\bigg(1+\frac{\arctan\big(5(\xi_i-0.5)\big)}{2}\bigg)\big(x_{i+1}-x_i^2\big)^2\right] \\
& g(\boldsymbol{x},\boldsymbol{\xi}) = \bigg(4-2.1x_1^2+\frac{x_1^4}{3}\bigg)x_1^2 + x_1x_2 + \big(-4+4x_2^2\big)x_2^2 \\
& \qquad\qquad + \frac{\cos(2\pi\xi_1)-\sin(\frac{\pi}{2}\xi_1)-\xi_1-\big(\cos(2\pi 0.05)-\sin(\frac{\pi}{2}0.05)-0.05\big)}{5} \\
& \boldsymbol{\xi} \sim \mathcal{U}([0,1]^3) \\
\text{by changing:}\quad & \boldsymbol{x} \in [-0.2, 1.2]^4
\end{aligned}
\tag{10}
$$

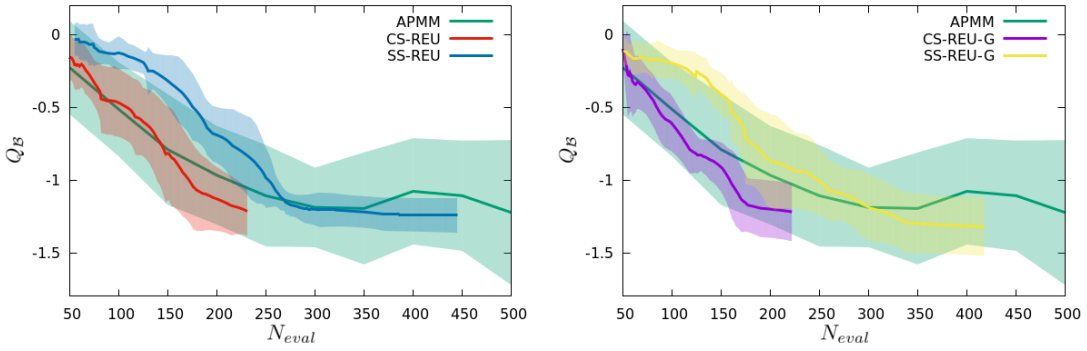
The optimum of the problem is easy to find as the robustness measure reduces to the classical formulation of the 4D Rosenbrock function and the reliability measures are precisely the Six-Hump Camel function. We can derive analytically that

$$
\begin{aligned}
\mu(\boldsymbol{x}) &= \sum_{i=1}^{3}\left[\big(1-x_i\big) + 3\big(x_{i+1}-x_i^2\big)^2\right], \\
\mathrm{q}^{95\%}(\boldsymbol{x}) &= \bigg(4-2.1x_1^2+\frac{x_1^4}{3}\bigg)x_1^2 + x_1x_2 + \big(-4+4x_2^2\big)x_2^2.
\end{aligned}
\tag{11}
$$

The optimum of this deterministic problem is found at $\boldsymbol{x}_* = (0.7033, 0.7035, 0.6212, 0.3859)$ with $\mathrm{q}^{95\%}(\boldsymbol{x}_*) = 1$ and $\mu(\boldsymbol{x}_*) = 0.4981$.

For this test case, both $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are sequentially taken as 50%, 40%, 30%, 20%, 10%, 5%, 3%, 2%, 1% and 0.5%. As for the other parameters, $N_{init} = 10$, $N = 1$ (sequential optimiser), $N_{first} = 5$ and $N_{new} = 1$.

First, Figure 22 presents the comparison of separated (SS-REU and SS-REU-G) and coupled (CS-REU and CS-REU-G) SABBa with the APMM strategy. Because of the use of a surrogate model able to efficiently deal with the coupled space variations of the underlying functions, SABBa shows better convergence again with the use of a coupled space model. In this case, the use of greedy boxes still increases the probabilistic proximity to the optimum, but the impact is quite slight. This fact is mainly because the problem here is mono-objective, meaning that the number of non-dominated boxes is significantly reduced. Using the greedy approach over the classical one for surrogate-based boxes do not impact the overall cost and accuracy as much as in a multi-objective context.



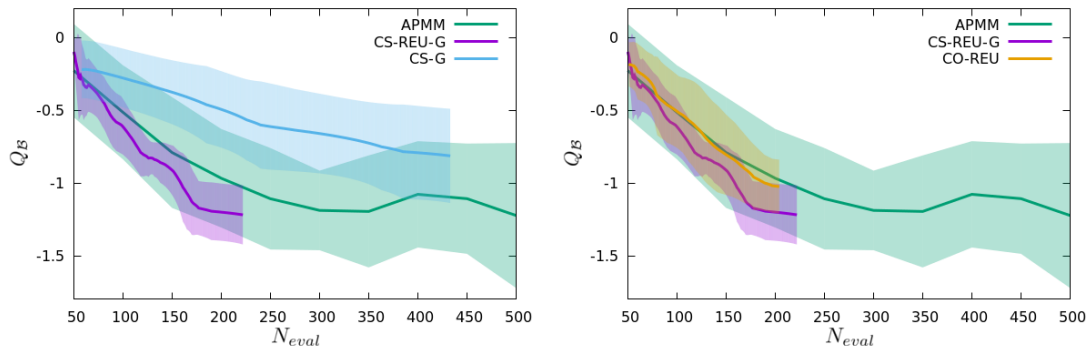
(a) (b)

Figure 22: Test-case 2: Cost comparison between APMM and SABBa in separated or coupled space with a) classical surrogate boxes (CS-REU and SS-REU) or b) greedy boxes (CS-REU-G and SS-REU-G).

We can see in Figure 23(a) that the reuse capability is still of significant importance to improve SABBa cost efficiency. In Figure 23(b), the CO-REU approach is nearly as cost-efficient as the SABBa CS-REU-G strategy, which seems to indicate that the SA strategy has a minor impact on the overall cost in this test-case.

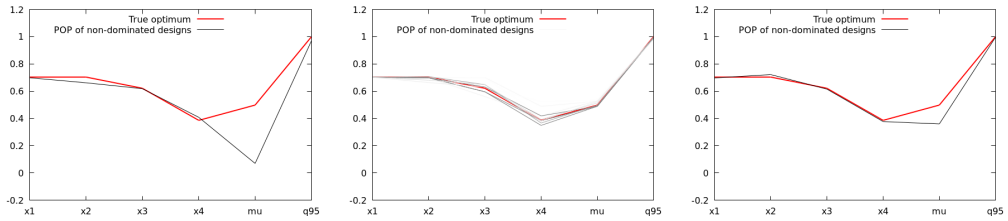
Finally, best, median and worst optimisation outputs are given in Figures 24 to 26. In this case, the 4D input space proscribes a direct representation such as in Figures 14 to 16 and 19 to 21. Alternatively, we compare parallel coordinates plots of the non-dominated designs, coloured by their POP_{min}. These plots represent each efficient individual by a grey curve and the true optimum in red. The value along each input or output axis can be read at the associated abscissa. We can see in Figure 24(a) and 24(c) that the best outputs using the APMM strategy (150 and 200 evaluations), with associated scores of -1.32 and -1.66, show a good proximity to the real optimum in the four input dimensions but not in the two output ones. This bias is not taken into account in the chosen distance as we emphasise finding the optimal designs. In Figure 24(b), the best SABBa CS-REU-G output is plotted, with an associated score of -1.51 for 161 evaluations. This output is blurrier around the true optimum compared to APMM-200 because of the greyscale of POP_{min} values. However, both the input and output dimensions are well converged using the SABBa framework, contrarily to both APMM strategies.

These parallel coordinates plots are also given in Figures 25 and 26 for the median and worst



(a) (b)

Figure 23: Test-case 1-b: Cost comparison between APMM, SABBa CS-REU-G and a) CS-G or b) CO-REU.



(a) (b) (c)

Figure 24: Best optimisation results: (a) A Priori MetaModel - 150 evaluations. (b) SABBa CS-REU-G - 161 evaluations. (c) A Priori MetaModel - 200 evaluations.

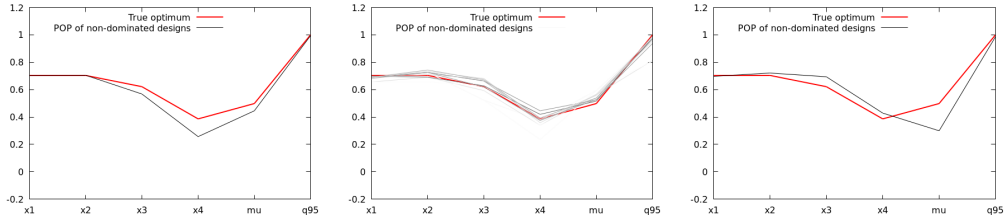
outcomes within the ten optimisation runs. The results for the APMM-150 strategy, as shown in Figures 25(a) and 26(a) have scores of -0.85 and 0.0, respectively. With 200 evaluations, the APMM strategy (Figures 25(c) and 26(c)) gives a median score of -1.12 and a worst score of -0.49. Both APMM-150 and APMM-200 show medium to low accuracy (notably Figure 26(a), which has converged far from the optimal design). On the contrary, median and worst runs from SABBa show greater robustness (Figures 25(b) and 26(b)), with $Q_\mathcal{B}$ scores of -1.29 and -0.95 respectively, with an associated overall cost of 170 and 161 evaluations.

7.3 Physical application

The SABBa framework is then applied here to an engineering uncertainty-based optimisation problem to assess its applicability and make use of its parsimonious behaviour.

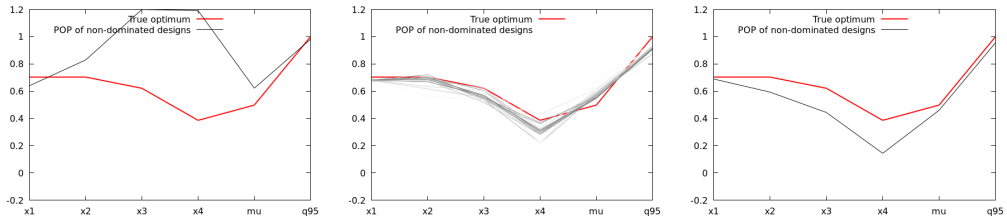
Here, we study the thermal response of a Thermal Protection System (TPS) during an hypothetic reentry, from entry interface to cool down, employing the PATO code, *i.e.* the Porous material Analysis Toolbox based on OpenFOAM, developed by Lachaud in [38]. As in the state-of-the-art approach, we assume that the problem is locally monodimensional and we study the material response at the stagnation point, which reaches the highest temperature during the reentry.

In particular, we consider a multilayered material, composed by an ablator of 5.8 cm high,



(a) (b) (c)

Figure 25: Median optimisation results: (a) A Priori MetaModel - 150 evaluations. (b) SABBa CS-REU-G - 170 evaluations. (c) A Priori MetaModel - 200 evaluations.



(a) (b) (c)

Figure 26: Worst optimisation results: (a) A Priori MetaModel - 150 evaluations. (b) SABBa CS-REU-G - 161 evaluations. (c) A Priori MetaModel - 200 evaluations.

an insulating layer (0.14 cm high) and an external part of the spacecraft (represented as a 1.27 cm material), for an overall width of 7.21 cm.

Figure 27 illustrates the typical evolution of the temperature at a depth of 1.5 cm inside the material and at the heated surface.

The analysis is performed using the properties of the Theoretical Ablative Composite for Open Testing (TACOT). In volume, TACOT is made of 10% of carbon fibers, 10% of phenolic resin, and is 80% porous. We assume adiabatic conditions at the bond line and a convective boundary condition at the surface of the ablative material. Ref. [39] has provided the surface total pressure and heat flux.

Formally, the optimisation problems reads as follows:

$$\begin{aligned}
\text{minimise:} \quad & \boldsymbol{\rho_f}(\boldsymbol{x}) = \mu(\boldsymbol{x}) \\
\text{satisfying:} \quad & \boldsymbol{\rho_g}(\boldsymbol{x}) = M(\boldsymbol{x}) \leq 473.15 \\
\text{where:} \quad & \mu(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\sigma(\boldsymbol{x}, \boldsymbol{\xi})] \\
& M(\boldsymbol{x}) = \max_{\boldsymbol{\xi}}[T_b(\boldsymbol{x}, \boldsymbol{\xi})] \\
\text{with:} \quad & \boldsymbol{\xi} \sim \mathcal{U}(\Xi) \\
\text{by changing:} \quad & \boldsymbol{x} \in [0.01, 0.1] \times [3.5, 7]
\end{aligned} \tag{12}$$

with x_1 the resin volume fraction (originally 10%) an x_2 the overall width of the system (originally 7.21 cm). We observe in Figure 27 that with initial parameters, the bottom temperature stays merely constant during the whole reentry. The search space is therefore translated to allow both low x_1 and low x_2.

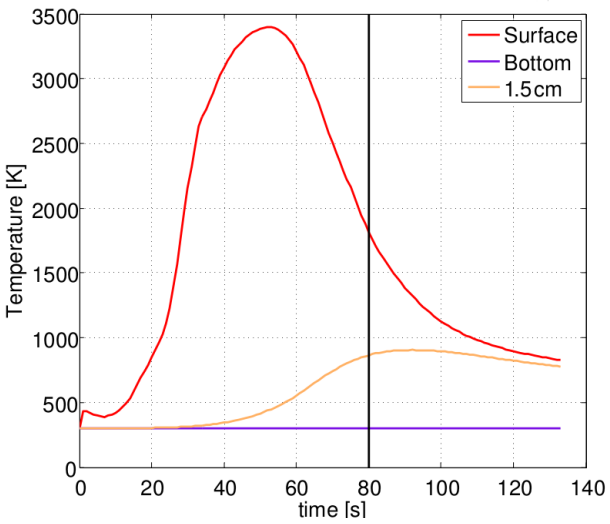


Figure 27: Temperature trend of the reference point at a depth of 1.5 cm inside TACOT material compared to the one of the heated surface, obtained with nominal material parameters.

In this test-case, Ξ is of dimension 10. The choice of the uncertain parameters relies on a sensitivity analysis performed on this test-case in a previous paper [40]. Here, retained uncertain parameters are :

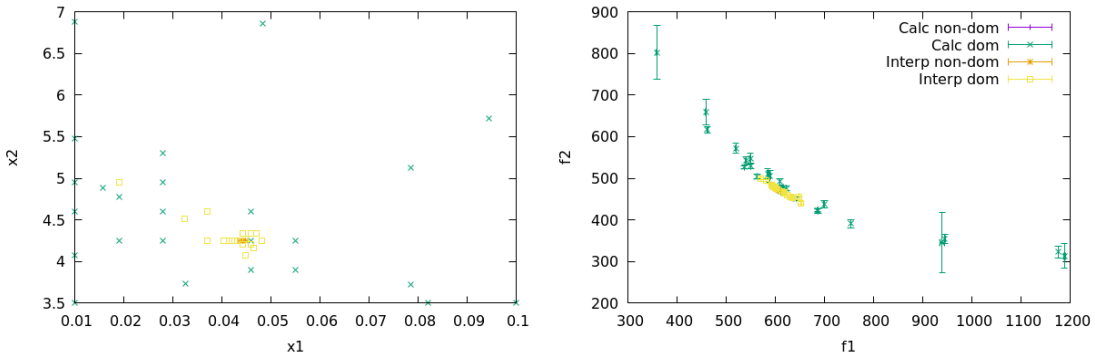
- Density and volume fraction of the fibrous preform (5% uncertainty each)
- Density and volume fraction of the phenolic resin (5% uncertainty each)
- Thermal properties of the charred material: heat capacity, conductivity and emissivity (5% uncertainty each)
- Oxygen fraction in the pyrolysis gases (10% uncertainty)
- Pyrolysis reaction activation energy (10% uncertainty)
- Overall width of the system (0.1 cm uncertainty)

Note that both design parameters are also affected by an uncertainty (± 0.5 for x_1 and ± 0.1 for x_2).

For this test case, again, $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are sequentially taken as 50%, 40%, 30%, 20%, 10%, 5%, 3%, 2%, 1% and 0.5%. This time, as the behavior should be simpler than in the previous test-cases, we consider $N_{init} = 5$, $N = 1$ (sequential optimiser), $N_{first} = 4$ and $N_{new} = 1$.

Outputs of the optimisation problem are plotted in Figure 28, with a design found at $\boldsymbol{x} = (0.044, 4.25)$ having the highest Pareto Optimal Probability (POP) with over 60%. Note that despite the dozen of dimensions, SABBa only ran **40** function evaluations to conduct the uncertainty-based optimisation. One can see in green in Figure 28(a) the designs that did not use the SA strategy and on which the surrogate assisting GP model is built. The designs where the SA strategy bypasses the computation are depicted in yellow when the measures are dominated and in orange when they are non-dominated. We report the objective and constraint values associated with each design in Figure 28(b), where one can note a dense concentration of points

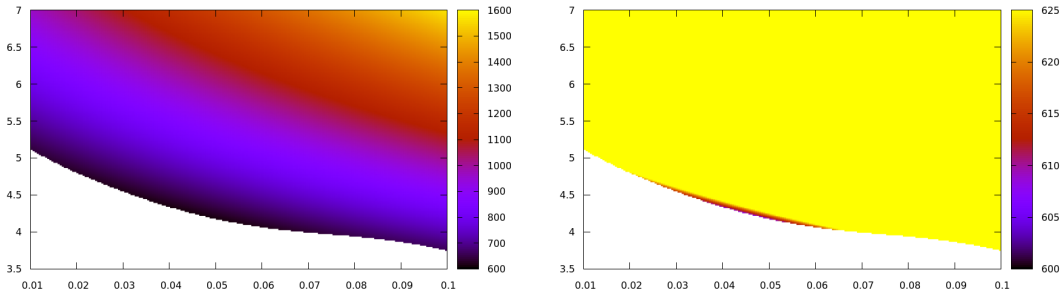
around (606.1, 472.6), where the objective is minimal, and the constraint is below the threshold 473.15.



(a) (b)

Figure 28: Outputs of the SABBa framework in (a) the input space, (b) the objective/constraint space.

We provide a representation of the final GP models in Figure 29. More precisely, we plot the mean surface density $\mu(\boldsymbol{x})$ from Equation (12) wherever the constraint is satisfied. Within this area, one can verify with the tight scale of Figure 29(b) that the optimum of the model is represented in Figure 28(a). The linear behaviour of the robust objective and simple behaviour of the reliable constraint explain the very low cost associated with the uncertainty-based optimisation. Such simple behaviours allow the SA strategy to bypass true evaluations and drastically reduce the overall cost quickly.



(a) (b)

Figure 29: Final GP model built within the optimisation process representing $\boldsymbol{\rho_f}(\boldsymbol{x}) = \mu(\boldsymbol{x})$ on the area where the constraint is satisfied with (a) classical scale, (b) tight scale.

Note however that in this application, while the porosity is correctly updated when the matrix volume fraction is changed, the thermal properties of both virgin and charred materials are not updated when the resin volume fraction is reduced.

8 Conclusions

In this work, we present SABBa, an innovative and efficient framework for optimisation under uncertainty problems, permitting to handle robustness and reliability-based measures, such as the Taguchi (mean and variance) robust optimisation or the quantile constrained optimisation problems. The efficiency of SABBa relies on several new features introduced in this paper. Parsimony is notably achieved through Pareto Optimal Probability (POP) computation, Gaussian Process (GP) greedy error approximation, coupled space modelling and the reuse strategy.

We compute the measures through Monte Carlo Sampling (MCS) on a GP model built on the uncertain variables. We can refine this approximation by increasing the size of the training set of the model. Such a GP model yields a predictive mean and variance for the function value at each $\boldsymbol{x}$, and a 3σ paradigm is assumed to fulfil the conservativeness assumptions made in [33] weakly. Note that the surrogate-assisting model predictive value and approximation error are computed similarly. We propose here also the POP, which allows choosing the most promising boxes to refine and returning a quantitative score for each box at the end of the optimisation process.

We have proposed some additional features and quantified their impact in the applicative section. We investigated the effect of building one surrogate model in the coupled space instead of many surrogates, for each design, in the uncertain space only. We also introduced the so-called reuse capability to limit the computational cost during the exploratory phase and to return optimal sets of incremental accuracy. Finally, we proposed a new, "greedy", computation of the box size when the SA strategy is used to avoid error overestimation.

We apply the framework to both analytical and applicative test cases. To evaluate the performances of the methods, we have formulated a specific indicator based on a modified Hausdorff distance. The performances of the framework are then assessed quantitatively on the analytical cases and compared to classical approaches, namely the Double loop approach, and the so-called A Priori MetaModel approach. The latest shows very superior behaviour compared to the Double loop approach but is still outperformed by the proposed method. Overall, SABBa also shows a dramatic increase in the robustness of the optimisation process, through a minimal variance in the optimisation outputs over repeated runs.

This approach remains very general and broadly applicable to any optimisation process on statistical measures. It shows excellent results within a manageable computational cost, which is of primary importance in the case of engineering applications, where black-box evaluations can take several days to run. Moreover, the optimiser is considered autonomous, allowing for easy coupling to any existing deterministic process. Several steps in this work remain improvable, such as the refinement criterion, which here follows local information instead of the more new integral criteria. Following actions could be the application of the uncertainty-based SABBa to other engineering-heavy cases, the inclusion of Value at Risk and Conditional Value at Risk (VaR and CVaR) as possible measures and a more intrusive optimisation method through the development of uncertainty-based Bayesian Optimisation.

A Justification for box sizes

Some developments are given here to explain the choice of the conservative error in Equation 3 associated with measure approximations.

The expectation approximation conservative error $\widetilde{\varepsilon}_\mu$ is quite straightforward:

$$\begin{aligned}
\left|\varepsilon_\mu(\boldsymbol{x})\right| = \left|\boldsymbol{\mu}(\boldsymbol{x}) - \widetilde{\boldsymbol{\mu}}(\boldsymbol{x})\right| &= \left|\mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})]\right| \\
&= \left|\mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) + \boldsymbol{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})]\right| \\
&= \left|\mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})]\right| \\
&\leq \mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})] \\
&= \widetilde{\boldsymbol{\varepsilon}}_\mu(\boldsymbol{x})
\end{aligned}$$

with $\boldsymbol{\mu}(\boldsymbol{x})$ the true statistical moment on $\boldsymbol{f}$ at a given $\boldsymbol{x}$ and $\widetilde{\boldsymbol{\mu}}(\boldsymbol{x})$ the approximated one, computed on the surrogate model $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}$. The true error $\boldsymbol{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}$ is defined from $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\xi}) = \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) + \boldsymbol{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})$ and is conservatively approximated by $\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) \geq |\boldsymbol{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})|$.

The same can be conducted for the variance measure, with a bit more risks of overestimation:

$$\begin{aligned}
\left|\varepsilon_{\sigma^2}(\boldsymbol{x})\right| = \left|\boldsymbol{\sigma}^{\boldsymbol{2}}(\boldsymbol{x}) - \widetilde{\boldsymbol{\sigma}^{\boldsymbol{2}}}(\boldsymbol{x})\right| &= \left|\mathbb{E}_{\boldsymbol{\xi}}[(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\xi}) - \boldsymbol{\mu}(\boldsymbol{x}))^2] - \mathbb{E}_{\boldsymbol{\xi}}[(\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{\mu}}(\boldsymbol{x}))^2]\right| \\
&= \left|\mathbb{E}_{\boldsymbol{\xi}}[(\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{\mu}}(\boldsymbol{x}) + \boldsymbol{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) - \boldsymbol{\varepsilon}_\mu(\boldsymbol{x}))^2] - \mathbb{E}_{\boldsymbol{\xi}}[(\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{\mu}}(\boldsymbol{x}))^2]\right| \\
&= \left|\mathbb{E}_{\boldsymbol{\xi}}[(\boldsymbol{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) - \boldsymbol{\varepsilon}_\mu(\boldsymbol{x}))^2 + 2(\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{\mu}}(\boldsymbol{x}))(\boldsymbol{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) - \boldsymbol{\varepsilon}_\mu(\boldsymbol{x}))]\right| \\
&\leq \mathbb{E}_{\boldsymbol{\xi}}[(\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) + \widetilde{\boldsymbol{\varepsilon}}_\mu(\boldsymbol{x}))^2 + 2|\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{\mu}}(\boldsymbol{x})|(\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) + \widetilde{\boldsymbol{\varepsilon}}_\mu(\boldsymbol{x}))] \\
&= \widetilde{\boldsymbol{\varepsilon}}_{\sigma^2}(\boldsymbol{x})
\end{aligned}$$

The conservative error associated with the expectation approximation could also be explained based on the monotonicity of the expectation operator. With $f \geq g \iff \forall \boldsymbol{x}, f(\boldsymbol{x}) \geq g(\boldsymbol{x})$, we have by definition $\boldsymbol{f}(\boldsymbol{x},\cdot) \in [\widetilde{\boldsymbol{f}}_{\boldsymbol{x}} - \widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}, \widetilde{\boldsymbol{f}}_{\boldsymbol{x}} + \widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}]$. The monotonicity of the expectation operator then gives $\mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\xi})] \in \left[\mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})], \mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) + \widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})]\right]$, or equivalently $\boldsymbol{\mu}(\boldsymbol{x}) \in \left[\widetilde{\boldsymbol{\mu}}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})], \widetilde{\boldsymbol{\mu}}(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})]\right]$, thus giving the conservative error found in the above $\widetilde{\boldsymbol{\varepsilon}}_\mu(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})]$.

By defining $\widetilde{\boldsymbol{f}}^+_{\boldsymbol{x}}(\boldsymbol{\xi}) = \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) + \widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}\boldsymbol{x}}(\boldsymbol{\xi})$ and $\widetilde{\boldsymbol{f}}^-_{\boldsymbol{x}}(\boldsymbol{\xi}) = \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{\varepsilon}}_{\boldsymbol{f}\boldsymbol{x}}(\boldsymbol{\xi})$, the monotonicity of the minimum, maximum and quantile operators can also be exploited to obtain conservative error approximations. For the case of the minimum operator, with $\widetilde{\boldsymbol{m}} = \min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi})]$,

$$\begin{aligned}
\boldsymbol{m}(\boldsymbol{x}) = \min_{\boldsymbol{\xi}}[\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\xi})] &\in \left[\min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^-_{\boldsymbol{x}}(\boldsymbol{\xi})], \min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^+_{\boldsymbol{x}}(\boldsymbol{\xi})]\right] \\
&= \left[\widetilde{\boldsymbol{m}}(\boldsymbol{x}) - \left|\widetilde{\boldsymbol{m}}(\boldsymbol{x}) - \min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^-_{\boldsymbol{x}}(\boldsymbol{\xi})]\right|, \widetilde{\boldsymbol{m}}(\boldsymbol{x}) + \left|\widetilde{\boldsymbol{m}}(\boldsymbol{x}) - \min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^+_{\boldsymbol{x}}(\boldsymbol{\xi})]\right|\right] \\
&\in \left[\widetilde{\boldsymbol{m}}(\boldsymbol{x}) - \widetilde{\boldsymbol{\varepsilon}}_{\min}(\boldsymbol{x}), \widetilde{\boldsymbol{m}}(\boldsymbol{x}) + \widetilde{\boldsymbol{\varepsilon}}_{\min}(\boldsymbol{x})\right]
\end{aligned}$$

where $\widetilde{\boldsymbol{\varepsilon}}_{\min}(\boldsymbol{x}) = \max\left(\left|\widetilde{\boldsymbol{m}}(\boldsymbol{x}) - \min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^-_{\boldsymbol{x}}(\boldsymbol{\xi})]\right|, \left|\widetilde{\boldsymbol{m}}(\boldsymbol{x}) - \min_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^+_{\boldsymbol{x}}(\boldsymbol{\xi})]\right|\right)$.

The same idea can be followed for the maximum and quantile approximations, resulting in the following errors:

$$\begin{aligned}
\widetilde{\boldsymbol{\varepsilon}}_{\max}(\boldsymbol{x}) &= \max\left(\left|\widetilde{\boldsymbol{M}}(\boldsymbol{x}) - \max_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^-_{\boldsymbol{x}}(\boldsymbol{\xi})]\right|, \left|\widetilde{\boldsymbol{M}}(\boldsymbol{x}) - \max_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^+_{\boldsymbol{x}}(\boldsymbol{\xi})]\right|\right) \\
\widetilde{\boldsymbol{\varepsilon}}_{\mathrm{q}^p}(\boldsymbol{x}) &= \max\left(\left|\widetilde{\mathbf{q}^p}(\boldsymbol{x}) - \mathrm{q}^p_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^-_{\boldsymbol{x}}(\boldsymbol{\xi})]\right|, \left|\widetilde{\mathbf{q}^p}(\boldsymbol{x}) - \mathrm{q}^p_{\boldsymbol{\xi}}[\widetilde{\boldsymbol{f}}^+_{\boldsymbol{x}}(\boldsymbol{\xi})]\right|\right)
\end{aligned}$$

B Justification for refinement criteria

The chosen partial criteria for GP-model refinement given in Equation 4 are also explained in the following.

Both the expectation and variance are global measures over the whole domain. For this reason, it has been chosen to iteratively add a point at the maximum predictive conservative error, to converge the model on the entire space. This strategy is usually called Maximum Mean Square Predictive Error or MMSPE. Note that the partial criteria are multiplied by the input pdf to weight the predictive error according to the probability of occurrence. Thus, the final partial criteria are:

$$\boldsymbol{c}_{\mu}(\boldsymbol{\xi}) = \widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) p(\boldsymbol{\xi})$$
$$\boldsymbol{c}_{\sigma^2}(\boldsymbol{\xi}) = \widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi}) p(\boldsymbol{\xi})$$

The minimum (resp maximum) measure partial criteria is simply the probability of exceeding the current minimal (resp. maximal) value $\widetilde{\boldsymbol{m}}$ (resp. $\widetilde{\boldsymbol{M}}$). This is performed with an assumption of uniform distribution with the conservative error box. Hence, the criteria can be written as such:

$$\boldsymbol{c}_{\min}(\boldsymbol{\xi}) = \left[\frac{\widetilde{\boldsymbol{m}} - \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{-}(\boldsymbol{\xi})}{2\widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})} \right]_{+}$$
$$\boldsymbol{c}_{\max}(\boldsymbol{\xi}) = \left[\frac{\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{+}(\boldsymbol{\xi}) - \widetilde{\boldsymbol{M}}}{2\widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})} \right]_{+}$$

where $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{+}(\boldsymbol{\xi}) = \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) + \widetilde{\varepsilon}_{\boldsymbol{f}\boldsymbol{x}}(\boldsymbol{\xi})$ and $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{-}(\boldsymbol{\xi}) = \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}(\boldsymbol{\xi}) - \widetilde{\varepsilon}_{\boldsymbol{f}\boldsymbol{x}}(\boldsymbol{\xi})$ and with $\left[\cdot \right]_{+} = \max(0, \cdot)$ referring to the value if positive, 0 either.

Finally, for the case of the quantile measure, it has been chosen to compute the product of the aforementioned probability of exceeding with regards to the quantile value, multiplied by the input density:

$$\boldsymbol{c}_{\mathrm{q}^p}(\boldsymbol{\xi}) = \left[\frac{\widetilde{\mathbf{q}^p} - \widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{-}(\boldsymbol{\xi})}{2\widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})} \right]_{+} \left[\frac{\widetilde{\boldsymbol{f}}_{\boldsymbol{x}}^{+}(\boldsymbol{\xi}) - \widetilde{\mathbf{q}^p}}{2\widetilde{\varepsilon}_{\boldsymbol{f}_{\boldsymbol{x}}}(\boldsymbol{\xi})} \right]_{+} p(\boldsymbol{\xi})$$

This product is maximised on the hyperplane $\widetilde{\boldsymbol{f}}_{\boldsymbol{x}} = \mathrm{q}^p$. Multiplying the criteria by $p(\boldsymbol{\xi})$ puts more weight according to the probability of occurrence. The spread may not be optimal, but the optimisation of the criteria being performed on a fixed sampling, the tightening of the conservative error is very likely to decrease the value of the criterion in the surrounding area because samples are very unlikely to be localised exactly on the isoline of the function.

We can note that the chosen criteria are far from being optimal. However, these choices give a fast determination of the refinement point, and the number of samples is usually low enough so that the choice of the training point is not of significant importance compared to the quality of the metamodeling strategy.

References

[1] Jae-Ohk Lee, Young-Soon Yang, and Won-Sun Ruy. A comparative study on reliability-index and target-performance-based probabilistic structural design optimization. *Computers & Structures*, 80(3):257 – 269, 2002.

[2] Qinghai Zhao, Xiaokai Chen, Zhengdong Ma, and Yi Lin. A comparison of deterministic, reliability-based topology optimization under uncertainties. *Acta Mechanica Solida Sinica*, 29(1):31 – 45, 2016.

[3] H.A. Jensen, M.A. Valdebenito, G.I. Schuëller, and D.S. Kusanovic. Reliability-based optimization of stochastic systems using line search. *Computer Methods in Applied Mechanics and Engineering*, 198(49):3915 – 3924, 2009.

[4] M.A. Valdebenito and G.I. Schu˙Efficient strategies for reliability-based optimization involving non-linear, dynamical structures. *Computers & Structures*, 89(19):1797 – 1811, 2011. Civil-Comp.

[5] Manolis Papadrakakis and Nikos D Lagaros. Reliability-based structural optimization using neural networks and monte carlo simulation. *Computer Methods in Applied Mechanics and Engineering*, 191(32):3491 – 3507, 2002.

[6] Vahid Keshavarzzadeh, Felipe Fernandez, and Daniel A. Tortorelli. Topology optimization under uncertainty via non-intrusive polynomial chaos expansion. *Computer Methods in Applied Mechanics and Engineering*, 318:120 – 147, 2017.

[7] R. Schöbi, B. Sudret, and S. Marelli. Rare event estimation using polynomial-chaos kriging. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 3(2):D4016002, 2017.

[8] Julien Bect, David Ginsbourger, Ling Li, Victor Picheny, and Emmanuel Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, May 2012.

[9] Jianye Ching and Wei-Chih Hsu. Approximate optimization of systems with high-dimensional uncertainties and multiple reliability constraints. *Computer Methods in Applied Mechanics and Engineering*, 198(1):52 – 71, 2008. Computational Methods in Optimization Considering Uncertainties.

[10] Jun Xu, Wangxi Zhang, and Rui Sun. Efficient reliability assessment of structural dynamic systems with unequal weighted quasi-monte carlo simulation. *Computers & Structures*, 175:37 – 51, 2016.

[11] I. Elishakoff, R.T. Haftka, and J. Fang. Structural design under bounded uncertainty—optimization with anti-optimization. *Computers & Structures*, 53(6):1401 – 1405, 1994.

[12] Stewart McWilliam. Anti-optimisation of uncertain structures using interval analysis. *Computers & Structures*, 79(4):421 – 430, 2001.

[13] Oded Amir, Ole Sigmund, Boyan S. Lazarov, and Mattias Schevenels. Efficient reanalysis techniques for robust topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 245-246:217 – 231, 2012.

[14] Jinglai Wu, Jie Gao, Zhen Luo, and Terry Brown. Robust topology optimization for structures under interval uncertainty. *Advances in Engineering Software*, 99:36 – 48, 2016.

[15] Ning Chen, Dejie Yu, Baizhan Xia, and Zhengdong Ma. Topology optimization of structures with interval random parameters. *Computer Methods in Applied Mechanics and Engineering*, 307:300 – 315, 2016.

[16] Ioannis Doltsinis, Zhan Kang, and Gengdong Cheng. Robust design of non-linear structures using optimization methods. *Computer Methods in Applied Mechanics and Engineering*, 194(12):1779 – 1795, 2005. Special Issue on Computational Methods in Stochastic Mechanics and Reliability Analysis.

[17] Zhili Tang and Jacques Périaux. Uncertainty based robust optimization method for drag minimization problems in aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 217-220:12 – 24, 2012.

[18] Alexandros A. Taflanidis and James L. Beck. An efficient framework for optimal robust stochastic system design using stochastic simulation. *Computer Methods in Applied Mechanics and Engineering*, 198 (1):88 – 101, 2008.

[19] Juan Camilo Medina and Alexandros A. Taflanidis. Adaptive importance sampling for optimization under uncertainty problems. *Computer Methods in Applied Mechanics and Engineering*, 279:133 – 162, 2014.

[20] J. Zhang, A.A. Taflanidis, and J.C. Medina. Sequential approximate optimization for design under uncertainty problems utilizing kriging metamodeling in augmented input space. *Computer Methods in Applied Mechanics and Engineering*, 315:369 – 395, 2017.

[21] Yaochu Jin and Bernhard Sendhoff. *Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach*, pages 237–251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[22] Kwon-Hee LEE and Gyung-Jin PARK. A global robust optimization using kriging based approximation model. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 49(3):779–788, 2006.

[23] Gabriella Dellino, Jack P. C. Kleijnen, and Carlo Meloni. Robust optimization in simulation: Taguchi and krige combined. *INFORMS Journal on Computing*, 24(3):471–484, 2012.

[24] Michael Eldred, Anthony Giunta, Steven Wojtkiewicz, and Timothy Trucano. Formulations for Surrogate-Based Optimization Under Uncertainty. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Multidisciplinary Analysis Optimization Conferences. American Institute of Aeronautics and Astronautics, sep 2002.

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

[26] Rodolphe Le Riche, Victor Picheny, Andre Meyer, Nam-Ho Kim, and David Ginsbourger. Gears design with shape uncertainties using controlled monte carlo simulations and kriging. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA No*, page 2257, 2009.

[27] Mickaël Binois, David Ginsbourger, and Olivier Roustant. Quantifying uncertainty on pareto fronts with gaussian process conditional simulations. *European journal of operational research*, 243(2):386–394, 2015.

[28] D Huang, T T Allen, W I Notz, and N Zeng. Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models. *Journal of Global Optimization*, 34(3):441–466, mar 2006.

[29] Victor Picheny, David Ginsbourger, and Yann Richet. Noisy Expected Improvement and on-line computation time allocation for the optimization of simulators with tunable fidelity. working paper or preprint, June 2010.

[30] Jürgen Teich. *Pareto-Front Exploration with Uncertain Objectives*, pages 314–328. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[31] Miha Mlakar, Tea Tusar, and Bogdan Filipic. Comparing solutions under uncertainty in multiobjective optimization. *Mathematical Problems in Engineering*, 2014:1–10, 2014.

[32] Francesca Fusi and Pietro Marco Congedo. An adaptive strategy on the error of the objective functions for uncertainty-based derivative-free optimization. *Journal of Computational Physics*, 309:241–266, February 2016.

[33] Mickael Rivier and Pietro Marco Congedo. Surrogate-Assisted Bounding-Box Approach for Optimization Problems with Approximated Objectives. Research Report RR-9155, Inria, February 2018.

[34] Sébastien Le Digabel. Nomad: Nonlinear optimization with the mads algorithm. 37:44, 01 2011.

[35] GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

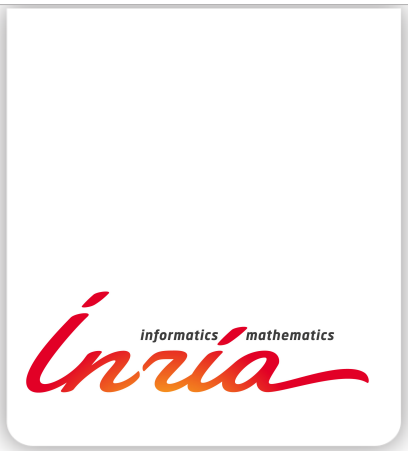
[36] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

[37] M. P. Dubuisson and A. K. Jain. A modified hausdorff distance for object matching. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 566–568 vol.1, Oct 1994.

[38] J. Lachaud and N. N. Mansour. Porous material analysis toolbox based on openfoam and applications. *Journal of Thermophysics and Heat Transfer*, 28(2):191–202, 2014. doi: 10.2514/1.T4262.

[39] K. A. Trumble, I. Cozmuta, S. Sepka, and P. Jenniskens. Post-flight aerothermal analysis of the stardust sample return capsule. *AIAA paper*, 1201:1–15, 2008.

[40] Mickael Rivier, Jean Lachaud, and Pietro Marco Congedo. Ablative thermal protection system under uncertainties: effect of pyrolysis gas composition. Research Report RR-9175, Inria Bordeaux Sud-Ouest, May 2018.



RESEARCH CENTRE
BORDEAUX – SUD-OUEST

351, Cours de la Libération
Bâtiment A 29
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399