Surrogate-Assisted Bounding-Box Approach for Optimization Problems with Approximated Objectives

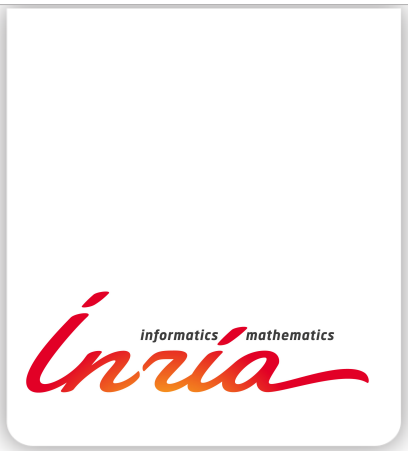
Mickael Rivier, Pietro Marco Congedo

Surrogate-Assisted Bounding-Box Approach for Optimization Problems with Approximated Objectives

Mickaël Rivier, Pietro Marco Congedo

RESEARCH REPORT
N° 9155
July 10, 2018
Project-Teams CARDAMOM

ISSN 0249-6399 ISRN INRIA/RR--9155--FR+ENG



Surrogate-Assisted Bounding-Box Approach for Optimization Problems with Approximated Objectives

Mickaël Rivier*†, Pietro Marco Congedo *‡

Project-Teams CARDAMOM

Research Report n° 9155 — July 10, 2018 — 37 pages

Abstract: In this work, we present a novel framework to perform multi-objective optimization, when considering an error on the objective functions. In many engineering optimization problems, the computation of the objective functions are affected by an error arising from the model employed for the computation. For example, in the case of uncertainty-based optimization the objective functions are statistics of a performance of interest which is uncertain due to the variability of the system input variables. These estimated objectives are affected by an error, which can be modeled with a confidence interval. The framework proposed here is general and aims at dealing with any error affecting the objective functions. The strategy is based on the extension of the Bounding-Box concept to the Pareto optima, where the error can be regarded with the abstraction of an interval (in one-dimensional problems) or a Bounding-Box (in multi-dimensional problems) around the estimated value. This allows the computation of an approximated Pareto front, whose accuracy is strongly dependent on the acceptable computational cost. This approach is then supplemented by the construction of a surrogate model on the objective functions, iteratively refined during the optimization process. This allows ultimately to further reduce the computation cost of the Pareto front with approximations of the objective functions at a negligible cost.
The validation of the proposed method is accomplished first by proving the mathematical convergence toward the true continuous Pareto front under some hypothesis. Secondly, a numerical algorithm is proposed and its performance is assessed on several numerical test-cases. Results are systematically compared to a simple Double-loop approach and to the classical Bounding-Box method.

Key-words: Multi-objective optimization, Error Bounding-Boxes, Cost reduction, Surrogate-assisting strategy, Uncertainty-based optimization

* 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 pour les problèmes d'optimisation avec des objectifs approximés

Résumé : Dans ce papier, nous présentons un nouveau cadre permettant la résolution d'optimisation multi-objectifs tout en considérant un bruit sur les fonctions objectifs. Dans la majorité des problèmes d'optimisation en ingénierie, le calcul des fonctions objectifs s'accompagne intrinsèquement d'une erreur provenant du modèle employé. Par exemple, les fonctions objectifs d'un problème d'optimisation sous incertitudes sont des moments statistiques estimés sur les fonctions de sortie, incertaines de par les variabilités d'entrée. Ces objectifs approximés sont donc affectés par une erreur, qui peut être représentée par un intervalle de confiance. Le cadre proposé ici reste général et aspire a pouvoir traiter avec des erreurs de n'importe quelle origine sur les fonctions objectifs. La stratégie repose sur l'extension du concept de Boîtes d'erreur conservatives aux optima de Pareto, avec une erreur interprétée comme un intervalle (dans les problèmes mono-objectif) ou une Boîte conservative (dans les problèmes multi-dimensions) autour de la valeur approximée. Cela permet d'approximer le front de Pareto, avec une précision fortement dépendante du coût de calcul acceptable. Cette approche est de plus couplée à la construction d'un modèle de substitution évolutif sur les fonctions objectifs, raffiné itérativement pendant le processus d'optimisation. Cela permet de réduire à nouveau la coût global de l'optimisation grâce à des approximations lues directement sur le modèle de substitution, pour un coût négligeable.

La validation de la méthode proposée se fera tout d'abord en prouvant formellement la convergence vers le vrai front de Pareto continu sous quelques suppositions. Ensuite, un algorithme est proposé et appliqué à plusieurs cas-test. Les résultats sont méthodiquement comparés à l'approche Double-boucle directe et à la méthode des Boîtes d'erreur conservatives classique.

Mots-clés : Optimisation multi-objectifs, Boîtes d'erreur conservative, Réduction du coût, Assistance par modèle de substitution, Optimisation dans l'incertain

1 Introduction

A key ingredient in any optimization method is the evaluation of the objective functions, which provides the metric to discriminate a good and feasible design, according to a set of objectives and constraints. In many real applications, the evaluation of the objective functions can be affected by a source of uncertainty, noise or error. A clear distinction has to be made among these elements. A possible classification is presented in [1], where four classes are identified: (i) noise, caused for example by sensor measurements or random simulations, (ii) uncertainty for robustness (or reliability) analysis, where the model/numerical method used to compute the objective functions includes stochastic variables, (iii) approximation error due to an approximate model of the exact objective functions (e.g. a response surface) and (iv) time-varying functions.

Here, this paper aims at dealing with objective functions subject to approximation errors. This kind of errors on the objective functions could notably arise when performing optimization under uncertainties, requiring the computation of statistical moments, tail probabilities or other probabilistic measures. Such errors can also refer to the variability of the objective values in the context of noisy optimization.

In the context of noise- and uncertainty-based optimization, many ideas have been developed in the last decades to cope with uncertain parameters, assess the reliability of a system or improve the convergence of optimization algorithms despite noisy evaluations.

Several studies use Taguchi's robust design concept (see [2] for a review). A strong effort has been provided on the reformulation of optimization problems with probabilistic or worst-case constraints ([3, 4, 5, 6]) or robustness measures. Jin et al [7] introduced a trade-off between performance and robustness by adding a robustness measure, computed by means of First-Order Second-Moment (analyzed in [8]) or Monte-Carlo sampling. Similar ideas are exploited with a Decision Support Problem in [9], with a weighted single-objective problem in [10], with a probabilistic indicator in [11] and coupled with percentile constraints and inverse reliability in [3].

Another class of techniques to perform optimization in noisy or uncertain context are the stochastic search algorithm, such as Evolutionary Algorithm. Review papers [1, 12] depicted the main methods to deal with uncertainties in an EA environment, namely averaging, robust index selection or robustness measures. In this context, a robustness measure can be seen as a new fitness assessment, as in [13], or as an additional robustness index objective in [14]. Deb et al [15] introduced two concepts of robustness, by revising the objective function or applying an additional constraint on the problem, both computed by averaging. However, in literature, the main strategy to account for uncertainty on the objective functions is to use probabilistic ranking of the individuals. More precisely, Stochastic Pareto Genetic Algorithm, developed in [16], computes probabilistic dominations between individual, assuming normal distributions, and uses them for ranking. Hughes [17] assumes normal distributions and computes the probability of wrong decision when comparing two individuals to propose a probabilistic extension of the Non-dominated Sorting Genetic Algorithm ranking. This work was then supplemented by [18] with a Bayesian learning of the variances of input noises. Teich [19] followed an idea similar to [17] but with a uniform distribution of the objective functions, yielding a ranking which is computed by summing probabilistic dominance. Robust evolutionary optimization has also been performed with coevolutionary algorithms such as in [20], to perform worst-case optimization. Other heuristics like Simulated Annealing and Pure Random Search have been developed in the noisy optimization context in [21] and [22]. Model-based approaches have also been compared and extended to noisy optimization problems in [23]. Finally, some works [24, 25] introduced new dominance criteria for non-stochastic interval uncertainties, while [26] introduced a worst-case methodology. Epistemic uncertainties have also been treated by means of evidence theory in

[27, 28].

Other techniques used for robust optimization are based on Stochastic programming (see [29] for a detailed review). These works are mostly focused on choosing optimal robustness measures. However, the practical applications are based on Sample Average Approximations and require expensive black-box evaluations to be approximated by surrogate models. In scenario-based optimization, alternative robustness concepts have been introduced to enrich the decision maker and introduce human-like reasoning. Some of these concepts are discussed and compared in [30, 31, 32, 33].

In the review of Mlakar et al [34], several techniques of optimization under uncertainty have been compared, and the concept of Bounding-Boxes and the associated Pareto dominance rules have been introduced. In Fusi et al [35], some concepts related to deterministic robustness measures (stochastic programming) and probabilistic boxes in the original objective space (robust evolutionary algorithms) have been reformulated in order to associate Bounding-Boxes approximations to robustness measures, so that approximated measures can be taken into account before full convergence. In practice, this means that the boxes associated with non-efficient and dominated designs would not be refined while Pareto optimal boxes would be highly converged.

The first contribution of this paper relies on providing some mathematical evidence about the convergence of the Bounding-Box approach in the measures and design space for a general problem of multi-objective optimization with approximated objective functions. One drawback of the approach from [35] is that substantial gains tend only to be accrued in the early stages of the optimization, whereas when converging in the vicinity of the Pareto front the impact is less significant. This can be easily explained since the convergence toward Pareto optimal designs leads to more and more potential optima, with very refined but costly evaluations of the Bounding-Boxes.

For curing this issue, the second main contribution of this paper is the coupling of the Bounding-Box approach with a surrogate-assisting method which tends to further reduce the computational cost during the last iterations of the optimization algorithm. Accelerative strategies using surrogate models have been studied, notably within the optimization process, like in [36], and extensively in the Evolutionary Strategy context ([37, 38, 39, 40, 41]). The strategy chosen here is to iteratively refine a metamodel each time an evaluation is performed, in order to bypass costly evaluations where convergence is detected. This coupling has several *a priori* advantages over classical approaches. In fact, the Bounding-Box approach quickly drives the optimization process in the Pareto optimal area, meaning that the assisting metamodel would only need to be converged over this area.

The aim of this paper is to introduce a framework called SABBa (Surrogate-Assisted Bounding-Box approach), and to provide mathematical and numerical evidence about its convergence properties. After having further explained the strategy in Section 2, convergence of the Pareto front will be mathematically analyzed under several assumptions in Section 3. Then, the general pseudo-algorithm of the framework is presented in Section 4 and three analytical test-cases are fully analyzed in Section 5 in order to highlight the main behaviors of the coupling in terms of computational cost and accuracy. Finally, conclusive remarks and future works is discussed in Section 6.

2 Overview of the SABBa framework

This section illustrates the SABBa framework, which is based on the coupling between the Bounding-Box approach and the surrogate-assisting strategy. The Bounding-Box approach rely on the definitions of Bounding-Boxes and the associated Pareto dominance, given in Sections 2.1

and 2.2, respectively. This approach is then described in Section 2.3, followed by the description of the surrogate-assisting strategy. Finally, a summary of the proposed framework illustrated with a flowchart is given in Section 2.5.

2.1 Definition of a Bounding-Box in a multi-objective problem

A multi-objective optimization problem can be stated as

$$\begin{aligned} \text{minimize/maximize:} \quad & \boldsymbol{f}(\boldsymbol{x}) \\ \text{by changing:} \quad & \boldsymbol{x} \in \mathcal{X} \end{aligned} \tag{1}$$

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

In Eq. 1, $\boldsymbol{f}$ represent user-defined measures. The main hypothesis of the proposed framework consists in assuming that these measures can not be computed exactly, but only approximated. They yield a specific error, which can potentially be reduced through additional evaluations. The objective functions are approximated with a quantity $\widetilde{\boldsymbol{f}}^k$, which corresponds to a level of refinement of the error $\boldsymbol{\varepsilon}^k$, so that $\widetilde{\boldsymbol{f}}^k = \boldsymbol{f} - \boldsymbol{\varepsilon}^k$. Of course, $\boldsymbol{\varepsilon}^k$ has to be estimated as it depends on $\boldsymbol{f}$. This quantity is supposed to be estimated by $\widetilde{\boldsymbol{\varepsilon}}^k$, with $|\boldsymbol{\varepsilon}^k| \leq \widetilde{\boldsymbol{\varepsilon}}^k$, *i.e.* $\boldsymbol{f} \in \left[\widetilde{\boldsymbol{f}}^k - \widetilde{\boldsymbol{\varepsilon}}^k, \widetilde{\boldsymbol{f}}^k + \widetilde{\boldsymbol{\varepsilon}}^k\right]$. This m-dimensional interval containing $\boldsymbol{f}$ (represented in Figure 1) is named as a box, $\mathcal{B}\left(\widetilde{\boldsymbol{f}}^k, \widetilde{\boldsymbol{\varepsilon}}^k\right)$, using a notation introduced in the following definition.

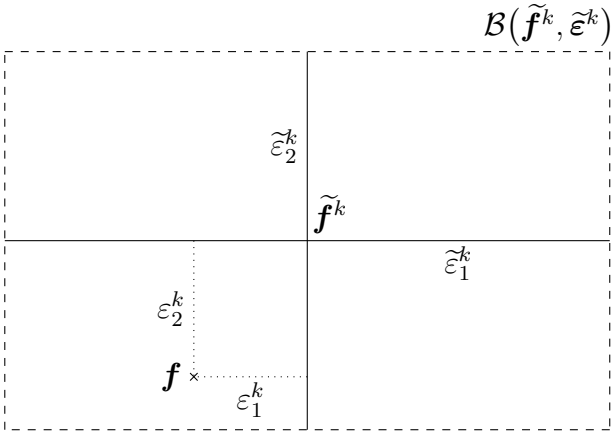


Figure 1: Bounding-Box approximation

Definition 1. A m-dimensional box is defined by its center and half-width vectors as follows:

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

$\forall (\boldsymbol{a}, \boldsymbol{r}) \in \mathbb{R}^m \times \mathbb{R}_+^m$, and with $\wp(\mathbb{R}^m)$ the power set of $\mathbb{R}^m$.

Remarks

- $\mathcal{B}(\boldsymbol{a}, \boldsymbol{0}) \equiv \boldsymbol{a}$.
- To lighten the notation, with $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^m$, we set:

$$\mathcal{B}_{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{r}) \equiv \mathcal{B}\left(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{r}\right), \quad \forall (\boldsymbol{x}, \boldsymbol{r}) \in \mathcal{X} \times \mathbb{R}^m.$$

2.2 Pareto dominance

In multi-objective optimization, objective functions are usually naturally conflicting. There is no one optimal design anymore but rather a set of trade-off optima, which are not worse than an other design in all objective dimensions simultaneously.These designs are said Pareto-optimal and the ensemble is called the Pareto front. These designs are non-dominated according to the following Pareto dominance rule:

Definition 2. For two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^m$, and with $\mathcal{I}_1^m = [\![1, m]\!]$:

$$\begin{aligned} \boldsymbol{a} \succ \boldsymbol{b} \ (\boldsymbol{a} \text{ dominates } \boldsymbol{b}) &\iff \forall j \in \mathcal{I}_1^m,\ \pm a_j \leq \pm b_j \quad \text{and} \\ &\quad\ \exists j \in \mathcal{I}_1^m,\ \pm a_j < \pm b_j, \\ \boldsymbol{a} \succcurlyeq \boldsymbol{b} \ (\boldsymbol{a} \text{ strictly dominates } \boldsymbol{b}) &\iff \forall j \in \mathcal{I}_1^m,\ \pm a_j < \pm b_j, \\ \boldsymbol{a} \sim \boldsymbol{b} \ (\boldsymbol{a} \text{ is indifferent to } \boldsymbol{b}) &\iff \boldsymbol{a} \not\succ \boldsymbol{b} \text{ and } \boldsymbol{b} \not\succ \boldsymbol{a}. \end{aligned}$$

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

$$\pm = \begin{cases} + & \text{for minimization,} \\ - & \text{for maximization.} \end{cases}$$

Definition 3. For two designs $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$, where $\mathcal{X}$ is the design space, and $\boldsymbol{f}$ the objective functions, we introduce the following notations:

$$\begin{aligned} \boldsymbol{x} \overset{\boldsymbol{f}}{\succ} \boldsymbol{y} &\iff \boldsymbol{f}(\boldsymbol{x}) \succ \boldsymbol{f}(\boldsymbol{y}), \\ \boldsymbol{x} \overset{\boldsymbol{f}}{\succcurlyeq} \boldsymbol{y} &\iff \boldsymbol{f}(\boldsymbol{x}) \succcurlyeq \boldsymbol{f}(\boldsymbol{y}), \\ \boldsymbol{x} \overset{\boldsymbol{f}}{\sim} \boldsymbol{y} &\iff \boldsymbol{f}(\boldsymbol{x}) \sim \boldsymbol{f}(\boldsymbol{y}). \end{aligned}$$

Remark Note that the objective functions appear explicitly in the above notation.

Definition 4. A design $\boldsymbol{x} \in \mathcal{X}$ is said to be non-dominated (or efficient) regarding a set $\mathcal{A} \subseteq \mathcal{X}$ if and only if:

$$\nexists \boldsymbol{y} \in \mathcal{A},\ \boldsymbol{y} \overset{\boldsymbol{f}}{\succ} \boldsymbol{x}.$$

Then, $\boldsymbol{x}$ is said to be Pareto-optimal iff $\boldsymbol{x}$ is non-dominated regarding $\mathcal{X}$.

However, this definition of Pareto dominance can not be retained in the following since $\boldsymbol{f}$ is assumed to be unknown. Only the approximate value $\widetilde{\boldsymbol{f}}^k$ and its conservative error $\widetilde{\boldsymbol{\varepsilon}}^k$ are used, considering that $\boldsymbol{f} \in \mathcal{B}\big(\widetilde{\boldsymbol{f}}^k, \widetilde{\boldsymbol{\varepsilon}}^k\big)$. A new dominance criterion is then introduced.

Following [34] and [35], this dominance is denoted as *Boxed Pareto dominance*. The basic criterion yields that a box $\mathcal{B}(\boldsymbol{a}, \boldsymbol{r})$ dominates a box $\mathcal{B}(\boldsymbol{b}, \boldsymbol{r}')$ if and only if any point of $\mathcal{B}(\boldsymbol{a}, \boldsymbol{r})$ dominates (in the classical way) every point of $\mathcal{B}(\boldsymbol{b}, \boldsymbol{r}')$. The boxed dominance rule can be interpreted as a classical dominance rule between the least performing point of $\mathcal{B}(\boldsymbol{a}, \boldsymbol{r})$ and the most performing one of $\mathcal{B}(\boldsymbol{b}, \boldsymbol{r}')$, as seen in the following definition.

Definition 5. For two boxes $\big(\mathcal{B}(\boldsymbol{a}, \boldsymbol{r}), \mathcal{B}(\boldsymbol{b}, \boldsymbol{r}')\big) \in \wp(\mathbb{R}^m)^2$,

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

The following equivalence is verified:

$$\forall \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^m,\ \mathcal{B}(\boldsymbol{a}, 0) \underset{\mathcal{B}}{\succ} \mathcal{B}(\boldsymbol{b}, 0) \iff \boldsymbol{a} \succ \boldsymbol{b}, \tag{2}$$

which traduces the consistency between the two domination rules when the size of the box tends to zero.

Remark It can be noted that the relation between $\mathcal{B}(\boldsymbol{a}, \boldsymbol{r})$ and $\mathcal{B}(\boldsymbol{b}, \boldsymbol{r}')$ is the same as between $\mathcal{B}(\boldsymbol{a}, \boldsymbol{r}')$ and $\mathcal{B}(\boldsymbol{b}, \boldsymbol{r})$.

2.3 Bounding-Box approach

Based on the above definitions, the Bounding-Box approach proposed in [35] is depicted here. The basic assumption, as mentioned in Section 3, is that the values of the objective function can be approximated with an error, which can be eventually reduced at the price of an additional computational cost. Practically, this approach is constituted by the following steps:

- Each time new designs are provided by the optimizer, rough first approximations are computed, and these boxes are compared in terms of the *Boxed Pareto dominance*;
- Then, before returning values to the optimizer, non-dominated boxes are further refined up to a user-defined threshold, while dominated ones are frozen.

After the refinement loop, only efficient boxes are completely refined while the dominated ones remain roughly approximated, thus saving computational power. This algorithm naturally yields a multi-fidelity approximation of the objectives, by linking the errors on the objective functions, which can be potentially reduced, with the Boxed Pareto optimality defined above. Such a multi-fidelity approximation of the objectives can be observed in Figure 2, where only non-dominated (red) boxes show high accuracy.

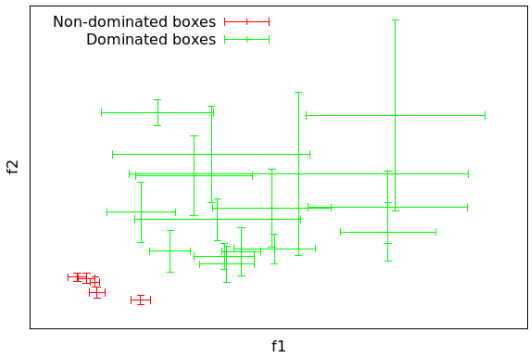


Figure 2: Bounding-Boxes refinement strategy

2.4 Surrogate-assisting strategy

In this paper, the Bounding-Box approach is coupled with a surrogate-assisting strategy, aiming at building and iteratively refining a surrogate model built on the objective functions, evaluated here as the center of the boxes associated to each design. It is based on the following steps:

- When the optimizer provides a new design $\boldsymbol{x}$ at which the objective values must be approximated, the local metamodel approximation error is compared to a user-defined threshold to determine whether a new box must be computed and refined (implying several simulation calls) or the surrogate can be used;
- If this approximation error is low enough, the predictive value of the surrogate at the new design $\boldsymbol{x}$ is directly returned to the optimizer;
- Else, the center of the newly computed box at $\boldsymbol{x}$ is used to refine the assisting surrogate model for the following iterations.

The chosen surrogate model should allow for computing the error approximation in order to detect the convergence of the surrogate. For example, this can easily done when using statistical learning such as Gaussian process regression, by means of predictive variance. Moreover, some works have been conducted to compute surrogate model variability from the Leave-One Out Cross Validation errors (see [42] for example). Note that the choice of a specific surrogate method is not made here, as the SABBa framework can be used regardless of the surrogate model, which should be chosen by the user.

The next paragraph is devoted to the coupling between the Bounding-Box approach and the Surrogate-Assisting strategy within the SABBa framework.

2.5 Structure of the framework

The proposed framework aims at obtaining a converged Pareto front with a low computational cost by exploiting at best the information of the error, associated with the computation of the objective functions through the refinement of only non-dominated boxes (using the concept of *Boxed Pareto dominance*). It couples the Bounding-Box approach defined in [35] and the surrogate-assisting strategy designed to lower the overall computational cost.

Two thresholds are defined in the following. The first, $\boldsymbol{s}_1$, mentioned in Section 2.4, is compared to the actual surrogate approximation error to decide whether to bypass the box computation and refinement by using the metamodel or not. If the error is larger than $\boldsymbol{s}_1$, the approximation is then refined until the associated error (box width) becomes lower than a second user-defined threshold $\boldsymbol{s}_2$, mentioned in Section 2.3. As seen in the preceding sections, whenever a box is dominated following the *Boxed Pareto Dominance* rule, it is no longer refined. Consequently, only non-dominated boxes are refined up to $\boldsymbol{s}_2$.

The structure of the framework is presented in the flowchart given in Figure 3. One can see that the approximation of the objective function values comes either from true computations or from the surrogate model depending on the surrogate error at $\boldsymbol{x}$ and on $\boldsymbol{s}_1$. These two approximations are denoted as $\widetilde{\boldsymbol{f}}^{k_{min}}$ and $\widetilde{\boldsymbol{f}}_d^l$, and will be properly explained in the next sections. Note that the notation $\boldsymbol{f}_{OBJ}$ simply refers to the value returned to the optimization process.

Intuitively, the Bounding-Box approach allows a quick convergence toward the Pareto front through rough approximation of non-efficient boxes. Then, the surrogate-assisting strategy be locally very accurate in this optimal area which will be highly sampled.

Note that this framework aims to tackle any multi-objective optimization problem with approximated information. It is independent from the optimization algorithm, the method to estimate conservative box errors, the metrics, etc.

The following section deals with mathematical aspects of the surrogate assisted Bounding-Box approach. From a theoretical point of view, it aims at proving that the framework permits the convergence toward the real Pareto-optimal set when the number of designs and the refinement of the boxes tend to infinity. From a more practical point of view, we aim to show also that the

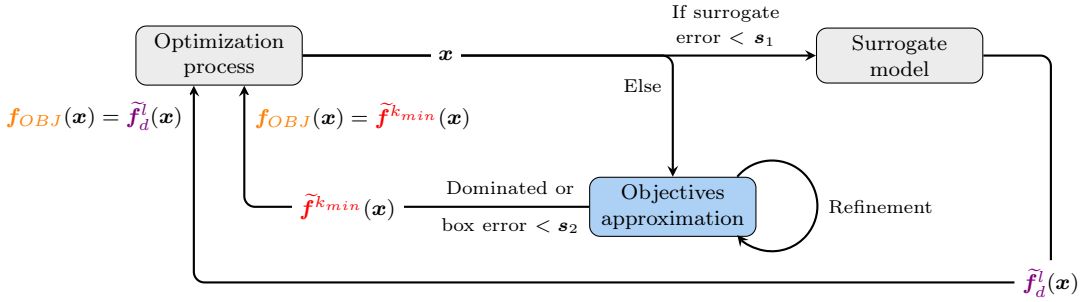


Figure 3: Structure of the SABBa framework

strategy is robust, meaning that among a discrete set of designs, all the truly non-dominated ones stay within the boxed Pareto-optimal set of the framework at any refinement level. Moreover, the strategy should systematically find an efficient and robust box size to assign to interpolated designs, thus building a deep coupling between the surrogate and the Bounding-Boxes.

3 Mathematical analysis of the proposed approach

The goal of this section is twofold. First, we want to prove the convergence of a set of interest, defined in Def. 10, toward the exact Pareto-optimal area when the number N of visited designs and the refinement k of the objective functions tend to infinity.

Secondly, the convergence of the method is demonstrated when the surrogate-assisting strategy is used.

3.1 Bounding-Box definition and convergence analysis

Based on the Pareto dominance defined in the preceding section, some sets are introduced that represent Pareto-optimal sets. The set $\mathcal{P}$ represents the true continuous Pareto front of the problem. Its discrete counterpart is labeled as $\widetilde{\mathcal{P}}$. Finally, we indicate with $\widetilde{\mathcal{P}}_{\mathcal{B}}$ the set of Pareto-optimal boxes.

Definition 6. Let us introduce the following sets, with $\mathcal{I}_1^N = [\![1, N]\!]$, $\mathcal{A} \subset \mathbb{R}^m$, designs $\boldsymbol{a}_i \in \mathbb{R}^m$ and boxes $\mathcal{B}_i \in \wp(\mathbb{R}^m)$:

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

In the case of multi-objective Pareto analysis (with $\boldsymbol{f}$ the objective functions and $\mathcal{X}$ the design space), $\mathcal{A}$ represents $\boldsymbol{f}(\mathcal{X})$ and designs $\boldsymbol{a}_i$ as well as the center of boxes $\mathcal{B}_i$ are elements of $\boldsymbol{f}(\mathcal{X})$. Note that $\mathcal{P}(\boldsymbol{f}(\mathcal{X}))$ can simplify to $\mathcal{P}$, *i.e.* the Pareto front of the problem, while $\widetilde{\mathcal{P}}$ is always linked to a given set of designs.

Definition 7. With $\boldsymbol{f}$ the objective functions, $\mathcal{X}$ the design space, $\boldsymbol{x}_i \in \mathcal{X}$ and $\boldsymbol{r}_i \in \mathbb{R}^m$, let us define

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

Practically, $\mathcal{X}^{\boldsymbol{f}}_{\mathcal{P}}$, $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\big\{\boldsymbol{x}_i\big\}_{i=1}^N\Big)$ and $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}\Big(\big\{(\boldsymbol{x}_i,\boldsymbol{r}_i)\big\}_{i=1}^N\Big)$ represent the ensemble of design coordinates associated with the Pareto Front, $\mathcal{P}$, $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{P}}_{\mathcal{B}}$, respectively.

Remark $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}\Big(\big\{(\boldsymbol{x}_i,\boldsymbol{r}_i)\big\}_{i=1}^N\Big)$ can be explicitly defined as follows:

$$\boldsymbol{x}_i \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}\Big(\big\{(\boldsymbol{x}_i,\boldsymbol{r}_i)\big\}_{i=1}^N\Big) \iff \forall k \in \mathcal{I}_1^N, \exists j \in \mathcal{I}_1^m,\ \pm f_j(\boldsymbol{x}_i) - r_{ij} < \pm f_j(\boldsymbol{x}_k) + r_{kj}. \tag{3}$$

Proposition 1. *Note that* $\forall\big\{(\boldsymbol{x}_i,\boldsymbol{r}_i)\big\}_{i=1}^N \in \big(\mathcal{X}\times\mathbb{R}^m\big)^N$,

$$\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\big\{\boldsymbol{x}_i\big\}_{i=1}^N\Big) \subseteq \big\{\boldsymbol{x}_i\big\}_{i=1}^N \text{ and } \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}\Big(\big\{(\boldsymbol{x}_i,\boldsymbol{r}_i)\big\}_{i=1}^N\Big) \subseteq \big\{\boldsymbol{x}_i\big\}_{i=1}^N.$$

The proof is provided in A.

Note that the inclusion becomes strict whenever a design does not satisfy the non-domination condition.

Finally, two last sets are introduced, $\mathcal{P}_c$ and $\widetilde{\mathcal{P}}_c$, which represent the continuous extension of $\mathcal{P}$ and $\widetilde{\mathcal{P}}$, respectively. These sets contain all the dominated but not strictly dominated points of $\mathbb{R}^m$ with respect to $\mathcal{P}$ and $\widetilde{\mathcal{P}}$.

Definition 8. With $\mathcal{A} \subset \mathbb{R}^m$ and $\forall i, \boldsymbol{a}_i \in \mathbb{R}^m$, let us define $\mathcal{P}_c$ and $\widetilde{\mathcal{P}}_c$, as follows

$$\begin{aligned}
\mathcal{P}_c(\mathcal{A}) &= \mathcal{P}(\mathcal{A}) \cup \big\{\boldsymbol{b} \in \mathbb{R}^m \mid \exists \boldsymbol{z} \in \mathcal{P}(\mathcal{A}),\ \boldsymbol{z} \succ \boldsymbol{b}\ \&\ \nexists \boldsymbol{z} \in \mathcal{P}(\mathcal{A}),\ \boldsymbol{z} \succcurlyeq \boldsymbol{b}\big\},\\
\widetilde{\mathcal{P}}_c\Big(\big\{\boldsymbol{a}_i\big\}_{i=1}^N\Big) &= \widetilde{\mathcal{P}}\Big(\big\{\boldsymbol{a}_i\big\}_{i=1}^N\Big) \cup \Big\{\boldsymbol{b} \in \mathbb{R}^m \big| \exists \boldsymbol{z} \in \widetilde{\mathcal{P}}\Big(\big\{\boldsymbol{a}_i\big\}_{i=1}^N\Big), \boldsymbol{z} \overset{\mathcal{P}}{\succ} \boldsymbol{b}\ \&\ \nexists \boldsymbol{z} \in \widetilde{\mathcal{P}}\Big(\big\{\boldsymbol{a}_i\big\}_{i=1}^N\Big), \boldsymbol{z} \overset{\mathcal{P}}{\succcurlyeq} \boldsymbol{b}\Big\}.
\end{aligned}$$

Usually, as for Def. 6, $\mathcal{A} \equiv \boldsymbol{f}(\mathcal{X})$ and $\mathcal{P}_c(\mathcal{A})$ is denoted as $\mathcal{P}_c$.

3.1.1 Mathematical formulation for error-based boxes

As stated in Section 2, the proposed framework is developed in the context of approximated objective functions, assuming that the error can be estimated and potentially reduced. The goal of this section is to provide some definitions and to introduce the so-called Bounding-Box recursive strategy.

Definition 9. An error is assumed on the computation of each objective function f_j. The approximated result $\widetilde{f}_j$ differs from the function f_j by an additive error, as follows:

$$\forall \boldsymbol{x} \in \mathcal{X}, \forall j \in \mathcal{I}_1^m,\ f_j(\boldsymbol{x}) = \widetilde{f}_j(\boldsymbol{x}) + \varepsilon_j(\boldsymbol{x}).$$

Assumption 1. Here, let us assume that a conservative error $\widetilde{\boldsymbol{\varepsilon}}$ can be computed on $\widetilde{\boldsymbol{f}}$, meaning that:

$$\forall \boldsymbol{x} \in \mathcal{X}, \forall j \in \mathcal{I}_1^m,\ |\varepsilon_j(\boldsymbol{x})| \leq \widetilde{\varepsilon}_j(\boldsymbol{x}).$$

Then, $\forall \boldsymbol{x} \in \mathcal{X}, \boldsymbol{f}(\boldsymbol{x}) \in \mathcal{B}_{\widetilde{\boldsymbol{f}}}\big(\boldsymbol{x}, |\boldsymbol{\varepsilon}(\boldsymbol{x})|\big) \subseteq \mathcal{B}_{\widetilde{\boldsymbol{f}}}\big(\boldsymbol{x}, \widetilde{\boldsymbol{\varepsilon}}(\boldsymbol{x})\big)$.

As shown in the following, assuming that this error is conservative is necessary for demonstrating the convergence of the proposed approach.

It is assumed that $\widetilde{\boldsymbol{f}}$ can be potentially improved by reducing the error through a refinement procedure. This conservative error is supposed to converge to zero when the refinement tends to infinity.

Assumption 2. The sequence $\left(\widetilde{\boldsymbol{\varepsilon}}^k\right)_{k\in\mathbb{N}_+}$ is considered in the following, where k represent the "degree of refinement". No constraint is assigned on its monotony but it is assumed that:

$$\forall j \in \mathcal{I}_1^m, \lim_{k\to+\infty} \widetilde{\varepsilon}_j^k = 0.$$

This states that an infinite refinement would lead to an error equal to zero. Note that in engineering applications, this refinement is often the biggest source of computational cost.

The goal of this error-driven optimization is to obtain a refined discrete Pareto-front, where the output of interest is $\mathcal{X}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}}$. The direct computation of this set is impossible without the knowledge of $\boldsymbol{f}$, this set is then approximated by means of approximating sets based only on the current $\widetilde{\boldsymbol{f}}^k$ and $\widetilde{\boldsymbol{\varepsilon}}^k$, which can be refined.

Let us now define the classical and recursive (developed in [35]) approaches, which are introduced to obtain an approximation of the refined discrete front. Note that the following development is written in the design space, but can be extended easily to the objective space.

Definition 10. For $\{\boldsymbol{x}_i\}_{i=1}^N \in \mathcal{X}^N$, $\boldsymbol{f}$ the objective functions and $\widetilde{\boldsymbol{f}}^k$ refinable approximations, $\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}},\text{classical}}^{\boldsymbol{f},0} = \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}},\text{recursive}}^{\boldsymbol{f},0} = \{\boldsymbol{x}_i\}_{i=1}^N$ and the associated initial conservative errors $\{\widetilde{\boldsymbol{\varepsilon}}^0(\boldsymbol{x}_i)\}_{i=1}^N$, let us introduce a decreasing sequence $\left(N^k\right)_{k\in\mathbb{N}_+}$, with $N^0 = N$, so that we define $\forall k \in \mathbb{N}_+$:

$$\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}},\text{classical}}^{\boldsymbol{f},k+1} = \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\widetilde{\boldsymbol{f}}^k}\left(\left\{\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}},\text{classical}_i}^{\boldsymbol{f},0}, \widetilde{\boldsymbol{\varepsilon}}^k\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}},\text{classical}_i}^{\boldsymbol{f},0}\right)\right)\right\}_{i=1}^{N}\right),$$
$$\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}},\text{recursive}}^{\boldsymbol{f},k+1} = \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\widetilde{\boldsymbol{f}}^k}\left(\left\{\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}},\text{recursive}_i}^{\boldsymbol{f},k}, \widetilde{\boldsymbol{\varepsilon}}^k\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}},\text{recursive}_i}^{\boldsymbol{f},k}\right)\right)\right\}_{i=1}^{N^k}\right).$$

As the recursive approach is developed in the following, the notation is slightly lightened as follows:

$$\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k+1} = \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\widetilde{\boldsymbol{f}}^k}\left(\left\{\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_i}^{\boldsymbol{f},k}, \widetilde{\boldsymbol{\varepsilon}}^k\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_i}^{\boldsymbol{f},k}\right)\right)\right\}_{i=1}^{N^k}\right), \tag{4}$$

where, as it might be expected, $\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}$ represents the approximation of $\mathcal{X}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}}$ at the k^{th} refinement of the boxes, by using $\widetilde{\boldsymbol{f}}^k$ instead of $\boldsymbol{f}$. Note that this set remains a boxed Pareto-optimal set.

The so-called classical approach refers to a double-loop or nested optimization, *i.e.* all boxes are fully refined before checking the Pareto-optimality. The difference between these approaches lies in the recursive aspect of the second sequence, which is of major importance. As seen in Assumption 2, the most time-consuming step consists in decreasing the conservative error $\widetilde{\boldsymbol{\varepsilon}}^k$. In the recursive Bounding-Box approach, Proposition 1 states that $\forall k \in \mathbb{N}_+$, $\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k} \subseteq \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k-1} \subseteq ... \subseteq \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},0}$. And, as seen in the proof of this proposition, any non-satisfied non-domination condition implies a strict inclusion between two sets of the sequence. Hence, if any box is dominated, the size N^k $\left(= Card\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\right)\right)$ of the set of interest starts decreasing, meaning that the refinement of $\widetilde{\boldsymbol{\varepsilon}}^k$ is required only on $N^k < N$ designs. On the contrary, with the classical approach, all N designs are refined at each iteration.

The next section is devoted to the convergence of the recursive set sequence.

3.1.2 Convergence analysis

This section aims to demonstrate that the sequence $\left(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}\right)_{k\in\mathbb{N}_+}$ theoretically converges to the real Pareto-optimal set. More precisely, the goal is to demonstrate the convergence of the continuous Pareto front $\widetilde{\mathcal{P}}_c$ built on the approximated measures $\widetilde{\boldsymbol{f}}^k\left(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}\right)$ toward the real continuous front $\mathcal{P}_c$.

To this extent, one more assumption is needed: it is assumed that there is no singular value in the Pareto-optimal set associated with the objective functions $\boldsymbol{f}$. Then, the designs $\boldsymbol{x}_i$ are assumed to be spread enough around the optimal set when N increases.

Assumption 3. For a given multi-objective optimization, it is assumed that:

$$\forall \boldsymbol{y} \in \mathcal{X}^{\boldsymbol{f}}_{\mathcal{P}}, \exists \mathcal{D} \subseteq \mathcal{X} \text{ so that } \boldsymbol{y} \in \mathcal{D}, \mathring{\mathcal{D}} \neq \emptyset \text{ and } \forall j \in \mathcal{I}_1^m,\ f_j \in \mathcal{C}^0(\mathcal{D}).$$

Let $\mathcal{D}(\boldsymbol{y})$ be a set satisfying the previous condition for a given $\boldsymbol{y} \in \mathcal{X}$.

$$\forall \epsilon \in \mathbb{R}_+^*, \exists M \in \mathbb{N}_+, \forall \boldsymbol{y} \in \mathcal{X}^{\boldsymbol{f}}_{\mathcal{P}}, \exists k \in \mathcal{I}_1^M,\ \boldsymbol{x}_k \in \mathcal{D}(\boldsymbol{y}), \left\|\boldsymbol{x}_k - \boldsymbol{y}\right\| \leq \epsilon.$$

This notably traduces the fact that the optimizer converges on the whole Pareto front to refine the discrete front on the integrity of $\mathcal{X}^{\boldsymbol{f}}_{\mathcal{P}}$ and that there exists a non-empty part of $\mathcal{X}$ around each efficient design where the restriction of all f_j to it is continuous.

Remarks The following relation is trivial:

$$\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\Big)\Big) = \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\Big).$$

More generally, $\forall \mathcal{S}$ s.t. $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\Big) \subseteq \mathcal{S} \subseteq \left\{\boldsymbol{x}_i\right\}_{i=1}^N$,

$$\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}(\mathcal{S}) = \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\Big). \tag{5}$$

Note that the same can be said for the Pareto fronts $\widetilde{\mathcal{P}}$.

Proposition 2. $\forall \boldsymbol{y} \in \left\{\boldsymbol{x}_i\right\}_{i=1}^N,\quad \boldsymbol{y} \notin \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\Big) \iff \exists \boldsymbol{y}' \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\Big),\ \boldsymbol{y}' \succ \boldsymbol{y}.$

The proof is given in B.

Lemma 1. $\forall \left\{\left(\boldsymbol{x}_i, \widetilde{\boldsymbol{\varepsilon}}(\boldsymbol{x}_i)\right)\right\}_{i=1}^N \in \left(\mathcal{X} \times \mathbb{R}^m\right)^N,\ \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\Big) \subseteq \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}\Big(\left\{\left(\boldsymbol{x}_i, \widetilde{\boldsymbol{\varepsilon}}(\boldsymbol{x}_i)\right)\right\}_{i=1}^N\Big).$

The proof is given in C.

The Proposition 2 states that any dominated design is dominated by at least one non-dominated design, which seems reasonable, as there has to be at least one dominant design in a given set. Lemma 1 introduces one of the main advantages of the Bounding-Box strategy by assessing the robustness of the boxed dominance compared to the classical one. Following a boxed approach, each truly efficient design is retained as boxed efficient.

A novelty of this paper compared to Fusi's Bounding-Box approach in [35] is the possibility to impose a precision threshold for the convergence of the objective functions. In [35], *exact* estimations for non-dominated designs are computed (*i.e.* the associated error is considered equal to zero), while here, non-dominated boxes are only converged to a given user-defined threshold. As seen above, this boxed Pareto front features a robust behavior by ensuring not to discard any truly efficient design.

Theorem 1. *For any initial set* $\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},0} = \{\boldsymbol{x}_i\}_{i=1}^{N} \in \mathcal{X}^N$*, approximation sequence* $\left(\widetilde{\boldsymbol{f}}^k\right)_{k\in\mathbb{N}_+}$ *and conservative error set sequence* $\left(\widetilde{\boldsymbol{\varepsilon}}^k\right)_{k\in\mathbb{N}_+}$*, the robustness is verified* $\forall k \in \mathbb{N}_+$*, i.e.:*

$$\mathcal{X}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}}\Big(\{\boldsymbol{x}_i\}_{i=1}^{N}\Big) \subseteq \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}. \tag{6}$$

The proof is given in D.

This first theorem illustrates how the framework may be used in practice. To further reduce computational cost, only non-dominated boxes are converged to a non-zero width, still retaining every efficient design. This low-cost Pareto-front approximation allows a fast analysis and can be supplemented by refined approximations in post-processing if needed.

Note that the convergence of the discrete approximated front extension $\widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big)$ toward the real continuous front $\mathcal{P}_c$ still has to be proven. This would imply the convergence of the preimage, because of the hypothesis of continuity related to the Assumption 3.

Theorem 2. *For any initial set* $\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},0} \in \mathcal{X}^N$*, approximation sequence* $\left(\widetilde{\boldsymbol{f}}^k\right)_{k\in\mathbb{N}_+}$ *and conservative error set sequence* $\left(\widetilde{\boldsymbol{\varepsilon}}^k\right)_{k\in\mathbb{N}_+}$*, the following convergence is verified:*

$$\lim_{(N,k)\to(+\infty,+\infty)} d_H\Big(\widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big),\ \mathcal{P}_c\Big) = 0, \tag{7}$$

with the classical Hausdorff distance denoted as d_H*.*

The proof is given in E.

3.1.3 Some comments on the Bounding-Box strategy

In summary, it has been observed that:

- The convergence of the tightest boxes is not required here as the concept of boxes is robustly extended to Pareto optima. The boxed Pareto front contains all truly efficient design and can be refined to the desired level of accuracy. This allows more parsimony through intermediate results to guide the user.
- The convergence analysis has been carried out under several assumptions on the computability of conservative errors and the good behavior of both the optimizer and the objective functions. Considering the continuous front built on a discrete approximation, the set of interest (the recursive non-dominated boxes) shows good converging properties toward the true Pareto front.

Note that the notations introduced in the previous paragraphs are well designed to be used in the effective algorithms.

One of the main conclusions of [35] was a significant drop in the performance of the Bounding-Box strategy throughout the execution, as the designs converge toward the Pareto optima. This observation has been the starting point of the additional feature of the proposed framework, which is presented in the next section.

3.2 Surrogate-assisting model

3.2.1 Principles and mathematical proof

In [35], boxes are rarely dominated while being poorly refined, except during the first optimization iterations, because of the convergence toward $\mathcal{X}_{\mathcal{P}}^{\boldsymbol{f}}$. Fusi's strategy behaves during the

last iterations of the algorithm as a classical optimization with tight refinement on every design (classical approach from Definition 10).

The targeted strategy here is to bypass the full computation and refinement of the objective functions by generating a surrogate model as a substitute for costly evaluations (when the error is low enough). Intuitively, the surrogate model is more and more used throughout the execution as its accuracy improves each time some objective functions are computed. During the last iterations of the optimization, the surrogate model should be massively used, since it should already be very accurate in the optimal area. Therefore, this approach aims to allow a faster estimation and densification of the Pareto front.

The surrogate model built on $\boldsymbol{f}$ is noted as $\widetilde{\boldsymbol{f}}_d$, with an associated conservative error $\widetilde{\boldsymbol{\varepsilon}}_d$. Moreover, as these approximations depend on the amount of information known about $\boldsymbol{f}$, they are noted as $\widetilde{\boldsymbol{f}}_d^l$ and $\widetilde{\boldsymbol{\varepsilon}}_d^l$, with $l \in \mathbb{N}$ describing this amount of information, for example the number of training points used to build the metamodel. The following is then assumed:

Assumption 4. Given functions $\boldsymbol{f}$, both the surrogate model built on $\boldsymbol{f}$ and their conservative error converge as follows:

$$\lim_{l\to\infty} \widetilde{\boldsymbol{f}}_d^l = \boldsymbol{f},$$
$$\lim_{l\to\infty} \widetilde{\boldsymbol{\varepsilon}}_d^l = \boldsymbol{0}.$$

Remarks

- The designs $\boldsymbol{x}_i$ are added at a certain level $l_{in} = l_{in}(\boldsymbol{x}_i)$ depending on the design. To lighten the notation, it is assumed that $\widetilde{\boldsymbol{f}}_d^{l_{in}}(\boldsymbol{x}_i) = \widetilde{\boldsymbol{f}}_d^{l_{in}(\boldsymbol{x}_i)}(\boldsymbol{x}_i)$, $\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) = \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}(\boldsymbol{x}_i)}(\boldsymbol{x}_i)$.
- One may note that Assumption [4] implies that the surrogate model is able to represent the underlying objective functions $\boldsymbol{f}$. This is not always the case as several metamodeling strategies rely on a regularization term (to impose some smoothness) or are based on a finite number of features chosen *a priori*, such as quadratic regression. In such cases, the framework can be applied as well but depending on the value of $\boldsymbol{s}_1$, the surrogate model may not be used, which would ultimately give a behavior similar to [17].

For each level l, the surrogate model $\widetilde{\boldsymbol{f}}_d^l$ is used for the designs satisfying $\forall j \in \mathcal{I}_1^m, \widetilde{\varepsilon}_{d_j}^l \leq s_{1_j}$ with $s_{1_j} \in \mathbb{R}_+$ a threshold fixed by the user. This criterion is noted as $\widetilde{\boldsymbol{\varepsilon}}_d^l \leq \boldsymbol{s}_1$ in the following.

Definition 11. The objective functions used in the following are a combination of given $\boldsymbol{f}_{opt}$ and of the surrogate-based approximations:

$$\boldsymbol{f}_{OBJ} = \boldsymbol{f}_{opt}.\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}} > \boldsymbol{s}_1} + \widetilde{\boldsymbol{f}}_d^{l_{in}}.\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}} \leq \boldsymbol{s}_1}.$$

Note that f_{opt} represents the original (not surrogate-assisted) objective functions.

Proposition 3. *Given the threshold $\boldsymbol{s}_1$, the Pareto optima of the non-boxed real problem are included in the following boxed Pareto optima:*

$$\forall \left\{\boldsymbol{x}_i\right\}_{i=1}^N \in \mathcal{X}^N,\ \mathcal{X}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}_{opt}}\left(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\right) \subseteq \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\boldsymbol{f}_{OBJ}}\left(\left\{\left(\boldsymbol{x}_i, \boldsymbol{r}(\boldsymbol{x}_i)\right)\right\}_{i=1}^N\right)$$

with $\boldsymbol{s}_1 = \boldsymbol{0}$ implying equality and $\boldsymbol{r}$ defined from one of the following definitions:

$$\textit{Rough}: \boldsymbol{r}(\boldsymbol{x}_i) = \boldsymbol{s}_1;$$
$$\textit{Refined}: \boldsymbol{r}(\boldsymbol{x}_i) = \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i);$$
$$\textit{Accurate}: \boldsymbol{r}(\boldsymbol{x}_i) = \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i).\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) \leq \boldsymbol{s}_1}(\boldsymbol{x}_i).$$

The proof is provided in F.

However, the surrogate model cannot be directly built on the objective functions since they are affected by some errors. More precisely, following the Bounding-Box approach, an efficient design $\boldsymbol{x} \in \mathcal{X}$ will have its objective values refined up to a threshold $\boldsymbol{s}_2$ and a dominated one will only be refined up to dominance. Let k_{min} (note that k_{min} is a simplified notation of $k_{min}(\boldsymbol{x})$) be the smallest integer so that the Bounding-Box is either converged to $\boldsymbol{s}_2$ or dominated.

Then, $\boldsymbol{f}_{OBJ}$ can be defined with $\boldsymbol{f}_{opt} = \widetilde{\boldsymbol{f}}^{k_{min}}$, which provides the coupled objective functions that will be used in the following sections:

$$\boldsymbol{f}_{OBJ} = \widetilde{\boldsymbol{f}}^{k_{min}}.\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}} > \boldsymbol{s}_1} + \widetilde{\boldsymbol{f}}_d^{l_{in}}.\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}} \leq \boldsymbol{s}_1}. \tag{8}$$

Remarks

- Note that $\widetilde{\boldsymbol{f}}^{k_{min}}$ and $\widetilde{\boldsymbol{\varepsilon}}^{k_{min}}$ are extended here to the entire domain $\mathcal{X}$, while their value have only been computed on some of the training points $\boldsymbol{x}_i$. Hence, There is a tricky choice to make in Theorem 3 on how to extend $\widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x})$, or to compute a conservative approximation $\widetilde{\boldsymbol{s}}_2(\boldsymbol{x})$, so that $\forall \boldsymbol{x}$,

$$\left|\widetilde{\boldsymbol{f}}^{k_{min}}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x})\right| \leq \widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}) \leq \widetilde{\boldsymbol{s}}_2(\boldsymbol{x}).$$

- From a numerical point of view, because of the noise on the training points $\left(\boldsymbol{x}_i, \widetilde{\boldsymbol{f}}^{k_{min}}(\boldsymbol{x}_i)\right)$ on which the surrogate model is built, a regression seems to be more adequate *a priori* than an interpolation.

Theorem 3. *Given two user-defined thresholds $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, the non-boxed discrete Pareto optima are included in the following boxed Pareto optima:*

$$\forall \left\{\boldsymbol{x}_i\right\}_{i=1}^N \in \mathcal{X}^N,\ \mathcal{X}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}}\left(\left\{\boldsymbol{x}_i\right\}_{i=1}^N\right) \subseteq \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\boldsymbol{f}_{OBJ}}\left(\left\{\left(\boldsymbol{x}_i, \boldsymbol{r}(\boldsymbol{x}_i)\right)\right\}_{i=1}^N\right),$$

where equality holds when $\boldsymbol{s}_1 = \boldsymbol{s}_2 = \boldsymbol{0}$ and $\widetilde{\boldsymbol{s}}_2(\boldsymbol{x}) = \boldsymbol{s}_2$ for non-dominated designs, and where $\boldsymbol{r}$ can represent a rough or accurate boxing, based on the threshold or the errors, as follows:

$$\begin{aligned} \textit{Rough}: \boldsymbol{r}(\boldsymbol{x}_i) &= \boldsymbol{s}_1 + \widetilde{\boldsymbol{s}}_2(\boldsymbol{x}_i); \\ \textit{Accurate}: \boldsymbol{r}(\boldsymbol{x}_i) &= \widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i) + \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i).\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) \leq \boldsymbol{s}_1}(\boldsymbol{x}_i). \end{aligned}$$

As observed before, $\widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i)$ is only known when the whole computation is performed, a trade-off box width is proposed, as follows:

$$\textit{Trade-off}: \ \boldsymbol{r}(\boldsymbol{x}_i) = \widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i).\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) > \boldsymbol{s}_1}(\boldsymbol{x}_i) + \left(\widetilde{\boldsymbol{s}}_2(\boldsymbol{x}_i) + \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i)\right).\mathbb{1}_{\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) \leq \boldsymbol{s}_1}(\boldsymbol{x}_i).$$

The proof is given in G.

3.2.2 Coupling with the Bounding-Box approach

Note that the surrogate model is iteratively built on the objective functions throughout the optimization process. Hence, using a one-iteration parallel sampling method on the design space for the optimization would not benefit from the assisting surrogate model. On the contrary, a genetic algorithm or any classical iterative optimization process would allow the refinement of the surrogate model at each generation or iteration of the optimization.

A sketch about the coupling is provided in Algorithm 1. Steps *(a)* to *(c)* are then detailed in the following sections.

Algorithm 1 Algorithm overview

```
while Optimization running do
    Read new designs
    if Assisting surrogate model precise enough for some designs then
        (a) Use surrogate to approximate objective functions
    else
        (b) Compute and (c) refine the boxed Pareto front to a given threshold
        Update the surrogate model
    end if
end while
```

3.2.3 Some comments

The surrogate model construction described in the previous section is well designed to supplement the Bounding-Box approach. We consider that the mathematical convergence of the coupled strategy can be easily demonstrated since the only difference is that the limit $k \to +\infty$ becomes $(\boldsymbol{s}_1, \boldsymbol{s}_2) \to (\mathbf{0}, \mathbf{0})$.

The drawback of the coupling with the surrogate is an increase of the box sizes associated with the designs whose objective values are computed with the surrogate. However, it allows a significant decrease of the global computational cost, especially at the end of the execution, when the surrogate is considered converged in the optimal area.

This strategy is easily implementable. The variables $k_{min}(\boldsymbol{x})$ and $l_{in}(\boldsymbol{x})$ are fixed once computed, $\boldsymbol{f}_{OBJ}(\boldsymbol{x})$ and $\boldsymbol{r}(\boldsymbol{x})$ only have to be stored after being computed.

In the following, a numerical algorithm is proposed and applied to some analytical test-cases in order to validate the framework, and get a first overview of the potential gains in industrial applications.

4 Algorithm of the framework

This section illustrates the numerical algorithm associated with the proposed framework.

As seen in the previous sections, two user-defined thresholds $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ have to be fixed. The value $\boldsymbol{s}_1$ is compared to the surrogate error to determine whether the surrogate can be used or not. The other threshold, $\boldsymbol{s}_2$, is compared to the approximation error of the objective functions, providing the accuracy at which boxes are no further refined. Here, these thresholds are defined as a percentage of the objective functions ranges, which are updated throughout the optimization.

For illustrating the algorithm, three design sets $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{D}}_{new}$ and $\boldsymbol{\mathcal{D}}'$ are introduced. The set $\boldsymbol{\mathcal{D}}$ is initialized at the beginning of the algorithm. It contains all the non-dominated designs and is updated at each refinement of the objective functions. It corresponds to the current set from Definition 10. The set $\boldsymbol{\mathcal{D}}_{new}$ represents the new designs at each optimization iteration. It can contain several designs (*e.g.* with a genetic algorithm's generation) or only one design (for sequential optimization). Finally, the set $\boldsymbol{\mathcal{D}}'$ contains the non-dominated non-interpolated designs of $\boldsymbol{\mathcal{D}}_{new}$, that may get refined. Practically, at each optimization iteration, when a box is not approximated with the surrogate, it is assigned to $\boldsymbol{\mathcal{D}}'$ as long as it is non-dominated. As

for $\boldsymbol{\mathcal{D}}$, this set is updated every time an approximation is refined, but $\boldsymbol{\mathcal{D}}'$ is reinitialized at each optimization iteration.

The algorithm follows the developments of Section 3 and is provided in Alg. 2. The main steps are the following: i) in the subroutine `compareDesignMM`, if the error of the surrogate is low enough, the associated objective values are returned. A first approximation is computed for the other designs and added to $\boldsymbol{\mathcal{D}}'$; ii) the inner while loop refines each design of $\boldsymbol{\mathcal{D}}'$ to the $\boldsymbol{s}_2$ threshold one by one within the `iterateRefinement` subroutine and checks for new dominations at each iteration to discard boxes from $\boldsymbol{\mathcal{D}}'$; iii) finally, the surrogate is updated and new designs can be analyzed.

Algorithm 2 Detailed pseudo-algorithm of the framework

Set $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$
Initialize $\boldsymbol{\mathcal{D}}$ empty
while Optimization running **do**
  Read new designs $\boldsymbol{\mathcal{D}}_{new} = \left\{\boldsymbol{x}_i\right\}_{i=1}^{N}$
  $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}} \cup \boldsymbol{\mathcal{D}}_{new}$
  Initialize $\boldsymbol{\mathcal{D}}'$ empty
  Call `compareDesignMM`($\boldsymbol{\mathcal{D}}_{new}$, $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\widetilde{\boldsymbol{f}}_d$, $\widetilde{\boldsymbol{\varepsilon}}_d$, $\widetilde{\boldsymbol{f}}^0$, $\widetilde{\boldsymbol{\varepsilon}}^0$, $\boldsymbol{f}_{OBJ}$, $\boldsymbol{r}$, $\boldsymbol{\mathcal{D}}'$) (Alg. 3) ▷ *(a,b)*
  Compute $\boldsymbol{\mathcal{D}} = \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}_{OBJ},0} = \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\boldsymbol{f}_{OBJ}}\left(\left(\boldsymbol{\mathcal{D}}, \boldsymbol{r}(\boldsymbol{\mathcal{D}})\right)\right)$
  $\boldsymbol{\mathcal{D}}' = \boldsymbol{\mathcal{D}} \cap \boldsymbol{\mathcal{D}}'$
  $k = 0$
  while $\max\left(\widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x})\right) > \boldsymbol{s}_2,\ \boldsymbol{x} \in \boldsymbol{\mathcal{D}}'$ **do**
    $k = k + 1$
    Select $\boldsymbol{x}_r \equiv \boldsymbol{x}$ to refine in $\boldsymbol{\mathcal{D}}'$
    Call `iterateRefinement`($\boldsymbol{\mathcal{D}}'$, $\boldsymbol{x}_r$, $\widetilde{\boldsymbol{f}}^{k-1}$, $\widetilde{\boldsymbol{\varepsilon}}^{k-1}$, $\widetilde{\boldsymbol{f}}^k$, $\widetilde{\boldsymbol{\varepsilon}}^k$) (Alg. 4) ▷ *(c)*
    for each $\boldsymbol{x} \in \boldsymbol{\mathcal{D}}'$ **do**
      $\boldsymbol{f}_{OBJ}(\boldsymbol{x}) = \widetilde{\boldsymbol{f}}^k(\boldsymbol{x})$
      $\boldsymbol{r}(\boldsymbol{x}) = \widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x})$
    end for
    Compute $\boldsymbol{\mathcal{D}} = \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}_{OBJ},k} = \mathcal{X}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}^{\boldsymbol{f}_{OBJ}}\left(\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}_{OBJ},k-1}, \boldsymbol{r}\left(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f}_{OBJ},k-1}\right)\right)\right)$
    $\boldsymbol{\mathcal{D}}' = \boldsymbol{\mathcal{D}} \cap \boldsymbol{\mathcal{D}}'$
  end while
  Update $\widetilde{\boldsymbol{f}}_d$ and $\widetilde{\boldsymbol{\varepsilon}}_d$
end while
Return $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{B}}_{\boldsymbol{f}_{OBJ}}\left(\boldsymbol{\mathcal{D}}, \boldsymbol{r}(\boldsymbol{\mathcal{D}})\right)$

Remark In Algorithm 2, $\left\{\left(\mathcal{D}_i, \boldsymbol{r}(\mathcal{D}_i)\right)\right\}_{i=1}^{Card(\boldsymbol{\mathcal{D}})}$ is lightened as $\left(\boldsymbol{\mathcal{D}}, \boldsymbol{r}(\boldsymbol{\mathcal{D}})\right)$, and $\boldsymbol{\mathcal{B}}_{\boldsymbol{f}_{OBJ}}\left(\boldsymbol{\mathcal{D}}, \boldsymbol{r}(\boldsymbol{\mathcal{D}})\right)$ is the set of boxes $\mathcal{B}_{\boldsymbol{f}_{OBJ}}\left(\mathcal{D}_i, \boldsymbol{r}(\mathcal{D}_i)\right)$.

The two subroutines `compareDesignMM` and `iterateRefinement` are provided in the following, to complete Algorithm 2. `compareDesignMM` (in Algorithm 3) takes as inputs the set of new designs, the thresholds and the actual surrogate. It returns the first approximations of the objective functions, the coupled objective functions $\boldsymbol{f}_{OBJ}$, the boxes size $\boldsymbol{r}$ and the set of non-dominated non-interpolated new designs. For each new design, the error of the surrogate model is compared to the threshold $\boldsymbol{s}_1$. If the error is small enough, the coupled objective functions are computed with the surrogate. Otherwise, first approximations are computed and the design is added to $\boldsymbol{\mathcal{D}}'$.

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

for each $\boldsymbol{x} \in \boldsymbol{\mathcal{D}}_{new}$ **do**
  if $\widetilde{\varepsilon}_d(\boldsymbol{x}) \leq \boldsymbol{s}_1$ **then** ▷ *(a)*
    $\boldsymbol{f}_{OBJ}(\boldsymbol{x}) = \widetilde{\boldsymbol{f}}_d(\boldsymbol{x})$
    $\boldsymbol{r}(\boldsymbol{x}) = \widetilde{\boldsymbol{s}}_2(\boldsymbol{x}) + \widetilde{\boldsymbol{\varepsilon}}_d(\boldsymbol{x})$
  else ▷ *(b)*
    Compute $\widetilde{\boldsymbol{f}}^0(\boldsymbol{x})$ and $\widetilde{\boldsymbol{\varepsilon}}^0(\boldsymbol{x})$
    $\boldsymbol{f}_{OBJ}(\boldsymbol{x}) = \widetilde{\boldsymbol{f}}^0(\boldsymbol{x})$
    $\boldsymbol{r}(\boldsymbol{x}) = \widetilde{\boldsymbol{\varepsilon}}^0(\boldsymbol{x})$
    $\boldsymbol{\mathcal{D}}' = \boldsymbol{\mathcal{D}}' \cup \boldsymbol{x}$
  end if
end for
Return $\widetilde{\boldsymbol{f}}^0$, $\widetilde{\boldsymbol{\varepsilon}}^0$, $\boldsymbol{f}_{OBJ}$, $\boldsymbol{r}$ and $\boldsymbol{\mathcal{D}}'$

Then, `iterateRefinement` (in Algorithm 4) computes the approximation $\widetilde{\boldsymbol{f}}^k$ and its associated error for each design, which can be refined. The inputs are the set of non-dominated non-interpolated new designs and the current approximations. The outputs are the updated approximations. Note that the boxes are refined one at a time to sequentially check for domination.

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

for each $\boldsymbol{x} \in \boldsymbol{\mathcal{D}}'$ **do**
  if $\boldsymbol{x} = \boldsymbol{x}_r$ **then**
    Refine and compute $\widetilde{\boldsymbol{f}}^k(\boldsymbol{x})$ and $\widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x})$ from $\widetilde{\boldsymbol{f}}^{k-1}(\boldsymbol{x})$ and $\widetilde{\boldsymbol{\varepsilon}}^{k-1}(\boldsymbol{x})$ ▷ *(c)*
  else
    $\widetilde{\boldsymbol{f}}^k(\boldsymbol{x}) = \widetilde{\boldsymbol{f}}^{k-1}(\boldsymbol{x})$
    $\widetilde{\boldsymbol{\varepsilon}}^k(\boldsymbol{x}) = \widetilde{\boldsymbol{\varepsilon}}^{k-1}(\boldsymbol{x})$
  end if
end for
Return $\widetilde{\boldsymbol{f}}^k$ and $\widetilde{\boldsymbol{\varepsilon}}^k$

5 Test-cases and analysis

This section is devoted to the application of the framework on several test-cases and to the performance analysis. The following choices have been made:

- The optimization problems deal with analytical functions, and a noise is arbitrarily added to the objective functions depending on the number of evaluations computed, so that the approximation can be refined. The number of evaluations is used as an indicator to compare several strategies. In practice, the error $\widetilde{\varepsilon}_i$ is computed in the following way for each objective function f_i:

$$\begin{aligned}
\varepsilon_{noise_i} &= \frac{\max(f_i) - \min(f_i)}{4 + 6X_1} \quad \text{with } X_1 \sim \mathcal{U}[0,1], \\
\varepsilon_{bias_i} &= \varepsilon_{noise_i} X_2 \quad \text{with } X_2 \sim \mathcal{U}[-1,1], \\
\widetilde{f}_i &= f_i + \varepsilon_{bias_i}, \\
\widetilde{\varepsilon}_i &= 2.\varepsilon_{noise_i}.
\end{aligned}$$

Then, this approximation is refined by choosing a refinement factor $r_f > 1$, as follows:

$$\widetilde{f}_i \leftarrow f_i + \frac{\widetilde{f}_i - f_i}{r_f},$$
$$\widetilde{\varepsilon}_i \leftarrow \frac{\widetilde{\varepsilon}_i}{r_f}.$$

- The optimization is performed by a simple sequential Monte-Carlo sampling, meaning that at each iteration, the set of new designs is constituted by ten random designs. The effect of converging toward the optimal area is studied in test-cases 2 and 3.
- The surrogate model is a noisy evaluation of the objective functions. In particular, this noise is considered proportional to the distance, noted as $d_{\min}$, to the closest training point relatively to the parameter interval width. This distance is also proportional to the range covered by the objective functions. For each i, it holds that:

$$\begin{aligned}
\widetilde{\varepsilon}_i &= \frac{d_{\min}}{\frac{1}{N_x}\sum_j \big(\max(x_j) - \min(x_j)\big)} . \frac{\max(f_i) - \min(f_i)}{2}, \\
\varepsilon_{noise_i} &= \frac{\widetilde{\varepsilon}_i}{2} X_1 \quad \text{with } X_1 \sim \mathcal{U}[0,1], \\
\varepsilon_{bias_i} &= \varepsilon_{noise_i} X_2 \quad \text{with } X_2 \sim \mathcal{U}[-1,1], \\
\widetilde{f}_i &= f_i + \varepsilon_{bias_i}.
\end{aligned}$$

- Note also that the extension $\widetilde{\boldsymbol{s}}_2$ of the conservative error to interpolated boxes is simply taken equal to the threshold $\boldsymbol{s}_2$, which implies that only non-dominated interpolated boxes are ensured conservative.

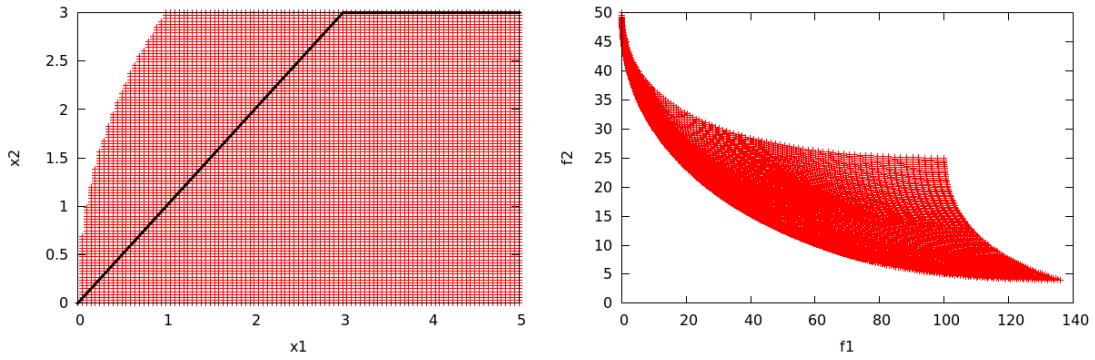
In the following, the proposed framework (SABBa) is compared to three different approaches: i) the *classical approach* (denoted as *Class*), *i.e.* non-recursive Bounding-Box approach defined in Def. 10, where each box is refined to the threshold $\boldsymbol{s}_2$; ii) the recursive approach, *i.e.* Fusi's Bounding-Box strategy [35], where only non-dominated boxes are refined to the threshold $\boldsymbol{s}_2$; iii) a surrogate-assisted classical approach (denoted as *Class-acc*), where the surrogate is built on the converged boxes.

5.1 Test-case 1: The BNH problem

The first test-case deals with a bi-objective constrained problem, proposed by Binh and Korn (1997):

$$\begin{aligned}
\text{minimize:} \quad & \boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} 4x_1^2 + 4x_2^2 \\ (x_1-5)^2 + (x_2-5)^2 \end{pmatrix} \\
\text{subject to:} \quad & (x_1-5)^2 + x_2^2 \leq 25 \\
& (x_1-8)^2 + (x_2+3)^2 \geq 7.7 \\
\text{by changing:} \quad & (x_1, x_2) \in [0,5] \times [0,3]
\end{aligned} \tag{9}$$

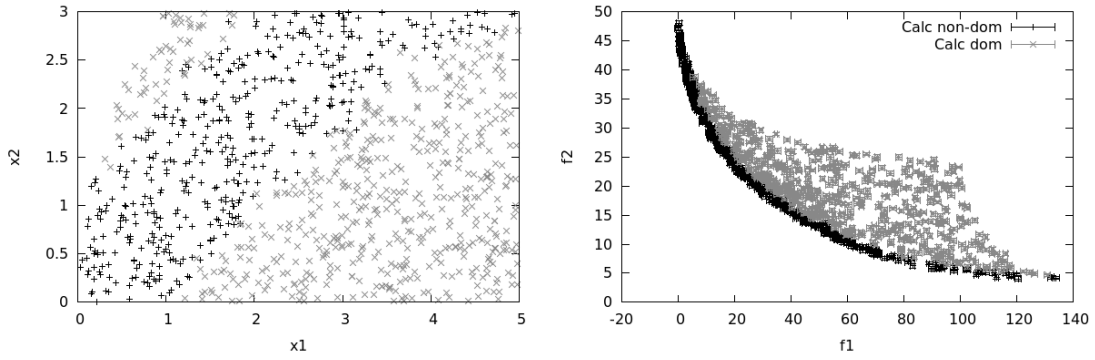
The acceptable design area and its counterpart in the objective space are represented in Fig. 4. Note that the Pareto optimal set in the design space is represented by the thick black curve. The new designs are chosen randomly in $[0,5] \times [0,3]$ under the two stated geometric constraints (the second being always inactive here).



(a) (b)

Figure 4: Test-case 1, a) acceptable design and Pareto optima (thick line), b) image in the objective space.

In Figures 5 and 6, the results in terms of optimal designs and objectives, for the classical and the recursive approaches are provided, respectively. This optimization is performed with 100 optimization iterations and $\boldsymbol{s}_2 = 1\%$. One can see that even if $\boldsymbol{s}_2$ is small, the found optimal area is quite large due to the low gradient around the optima. Note that designs are represented as points in the design space and as boxes in the objective space. Efficient boxes and designs are depicted in black and the others are drawn in grey.

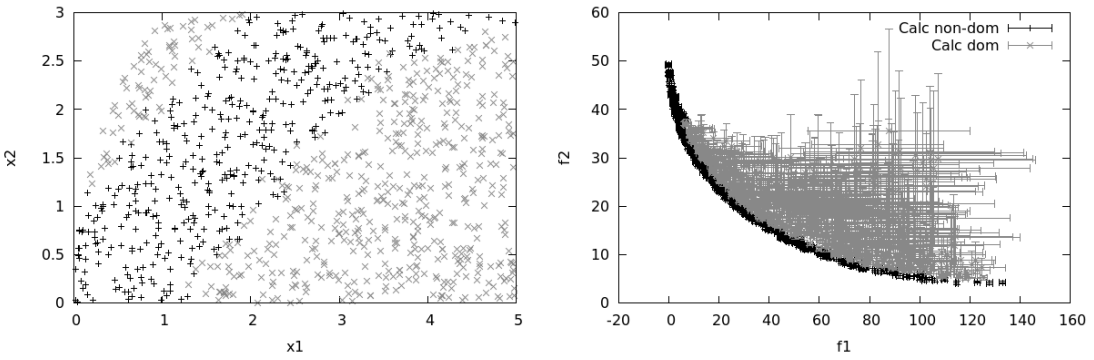


(a) (b)

Figure 5: Test-case 1, *Class approach*, non-dominated designs in black and dominated ones in grey: a) design space, b) objective space.

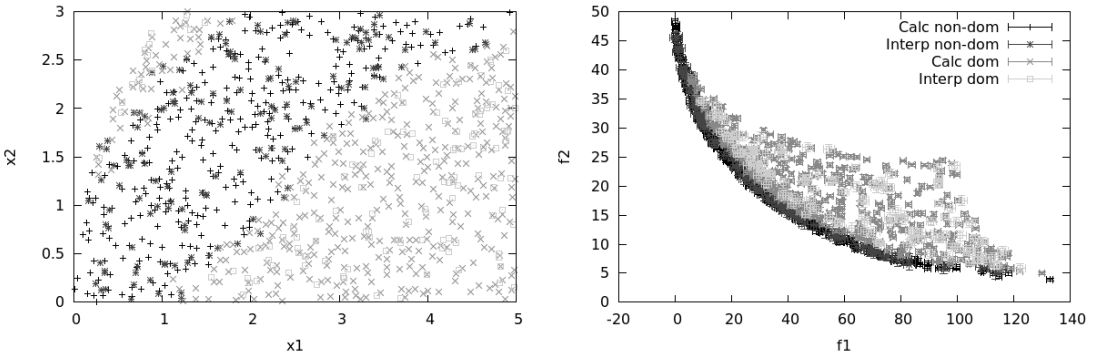
As expected, the Pareto front of these two strategies are very similar and both provided the same Pareto-optimal design area. However, in the objective space, a significant portion of the boxes are not converged to $\boldsymbol{s}_2$ in the recursive approach, thus reducing the number of function evaluations, and the computational cost of the optimization.

In Figures 7 and 8, the effect of the assisting surrogate model is illustrated for the classical approach and the proposed framework, respectively. The thresholds $\boldsymbol{s}_2 = 1\%$ and $\boldsymbol{s}_1 = 1\%$ are used.



(a) (b)

Figure 6: Test-case 1, *Fusi approach*, non-dominated designs in black and dominated one in grey: a) design space, b) objective space.



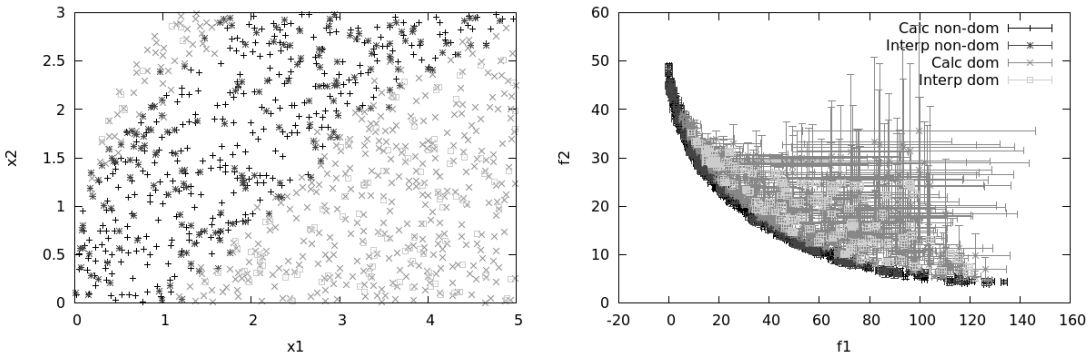
(a) (b)

Figure 7: Test-case 1, *Class-acc approach*, non-dominated interpolated designs added in dark grey, dominated interpolated designs in light grey. a) design space, b) objective space.

It can be observed in Figures 7 and 8 that enough information was gathered to allow the surrogate model to converge and bypass some function evaluations. Surrogate-evaluated designs and boxes are drawn in dark grey when non-dominated and light grey otherwise. One may note that the non-dominated area and the Pareto front are similar to the fully-computed classical approach shown in Figure 5.

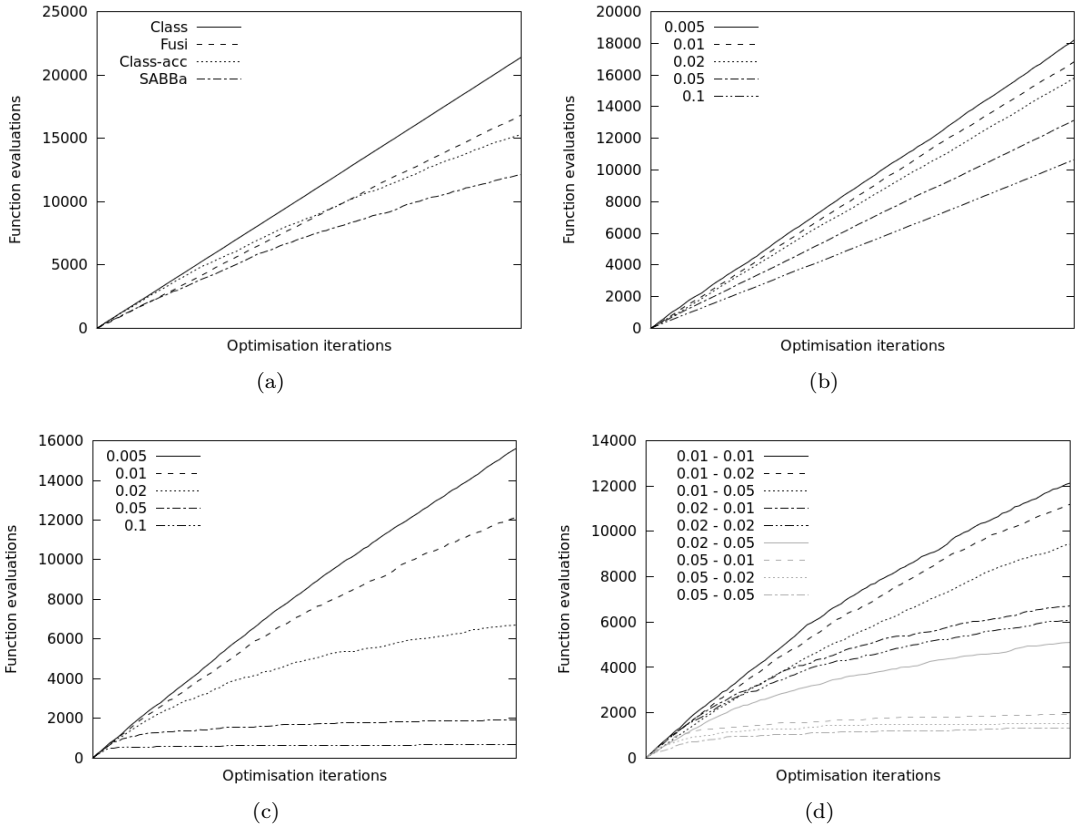
The cost reduction is verified by comparing the number of function evaluations needed for each strategy. Note that this number is only a basic indicator to compare several strategies. These results are reported in Figure 9:

- As expected, the recursive (denoted as Fusi [35]) strategy has a beneficial impact on the number of function evaluations. More precisely, it improves the slope of the cost curve by a ratio that can be interpreted as the mean computational cost in the recursive strategy with regards to the full convergence cost. Nearly half of the designs are non-dominated in this test-case. This high number of fully converged boxes implies limited computational gains.



(a) (b)

Figure 8: Test-case 1, *SABBa approach*, non-dominated interpolated designs added in dark grey, dominated interpolated designs in light grey: a) design space, b) objective space.



(a) (b)

(c) (d)

Figure 9: Test-case 1: a) number of function evaluations for each method of interest ($\boldsymbol{s}_1 = \boldsymbol{s}_2 = 0.01$), b) Fusi's approach varying $\boldsymbol{s}_2$, c) SABBa framework varying $\boldsymbol{s}_1$ with $\boldsymbol{s}_2 = 0.01$; d) varying both $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$.

- As suggested previously, the coupling with the surrogate model appears to be more and more effective throughout the optimization. This is due to the surrogate model needing several optimization iterations before reaching a low approximation error. Therefore, during the first iterations, the slope of the cost curve is the same as the non-assisted strategies.

In order to stress the influence of the tuning parameters, further comparisons are performed by varying $\boldsymbol{s}_2$ in Fusi's approach, and then $\boldsymbol{s}_1$ and both $\boldsymbol{s}_2$ and $\boldsymbol{s}_1$ in SABBa. Associated results in terms of number of function evaluations are reported in Figure 9.

As can be observed in Figure 9b, $\boldsymbol{s}_2$ has a significant impact on the computational cost. This can be explained by the high number of fully converged boxes. As a consequence, varying the convergence threshold influences all these boxes and has an important effect on the global cost. However, this threshold has a tremendous impact on the accuracy of the Pareto optimal set in the design space. This is qualitatively represented in Fig. 10 where the outputs with finest and coarsest threshold from Fig. 9b are compared.

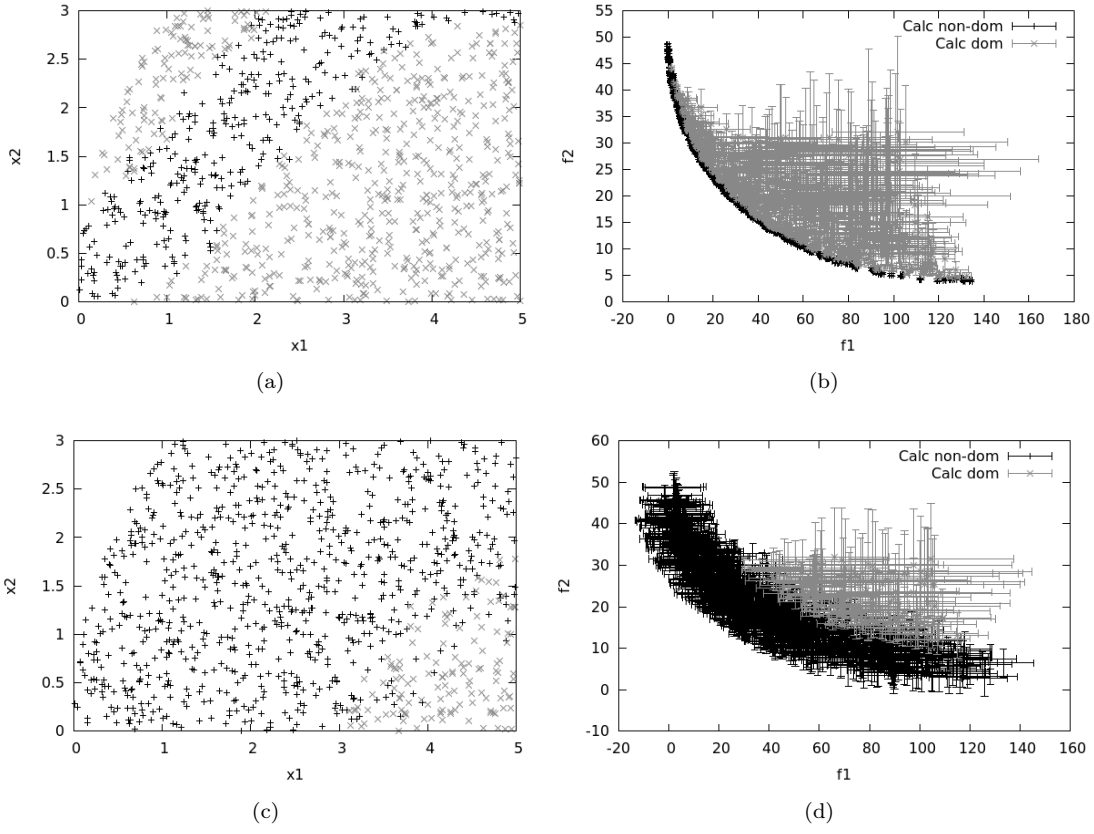


Figure 10: Impact of $\boldsymbol{s}_2$: Outputs with $\boldsymbol{s}_2 = 0.005$ in a) design space and b) objective space; Outputs with $\boldsymbol{s}_2 = 0.1$ in c) design space and d) objective space.

Finally, Figures 9c and 9d illustrate how the surrogate model can allow a momentous cost reduction. This however requires trust in the surrogate, with a non-negligible probability of underestimating the error, *i.e.* when high local variations are not yet captured. Globally, the framework seems effective as a cost-reduction strategy but requires some tuning of the thresholds

in order to provide the best results.

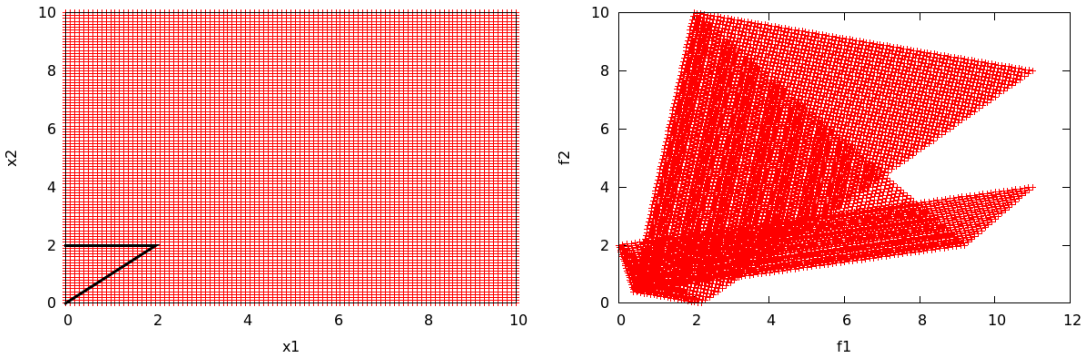
The next test-case deals with an optimization problem with a smaller Pareto-optimal area with respect to the entire space of variation.

5.2 Test-case 2: The Triangle problem

This problem is a bi-objective unconstrained optimization:

$$\begin{aligned} \text{minimize:} \quad & \boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} \frac{x_1+x_2}{10} + |x_1 - x_2| \\ \frac{x_1}{5} + |x_2 - 2| \end{pmatrix} \\ \text{by changing:} \quad & (x_1, x_2) \in [0, 10]^2 \end{aligned} \tag{10}$$

The design space with the optimal set (represented with a thick black curve) and the objective space counterpart are represented in Fig. 11.



(a) (b)

Figure 11: Test-case 2: a) acceptable design and Pareto optima (thick line), b) image in the objective space.

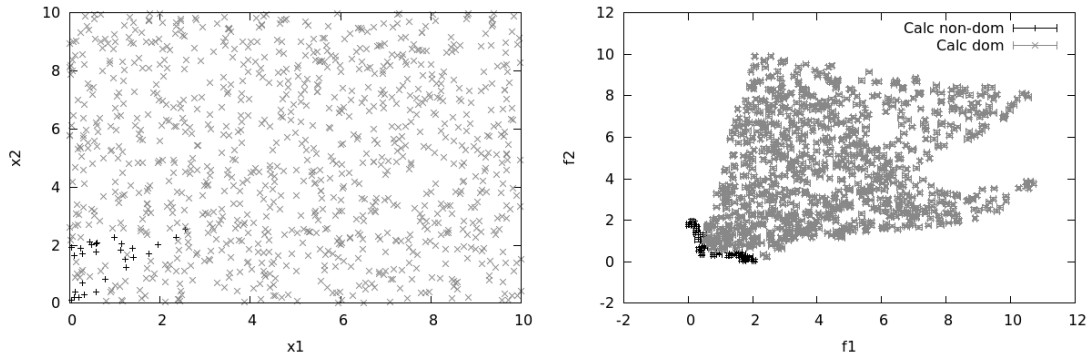
This small optimal area is well fit to study the resilience of the proposed SABBa framework to the convergence of the search domain. In particular, two different variants are proposed: a) at each iteration, 10 random new designs in $\mathcal{X} = [0, 10] \times [0, 10]$ are taken into account; b) 10 designs are considered, in a domain $\mathcal{X}^k$ converging toward $[0, 2] \times [0, 2]$ throughout the iteration, with $\mathcal{X}^0 = \mathcal{X}$.

These two variants are proposed here for assessing the behavior of the proposed framework over a drawback raised in [35], *i.e.* the decreasing rate of convergence of the Bounding-Box approach during the optimization.

5.2.1 Test-case 2 a): No convergence of the designs

In this variant, new designs are chosen randomly within the $[0, 10]^2$ design space. Of course, this study gives poor Pareto front refinement and is highly inefficient, as it can be seen in Fig. 12.

In the following, the effect of converging the search domain toward the optimal area is presented and discussed.



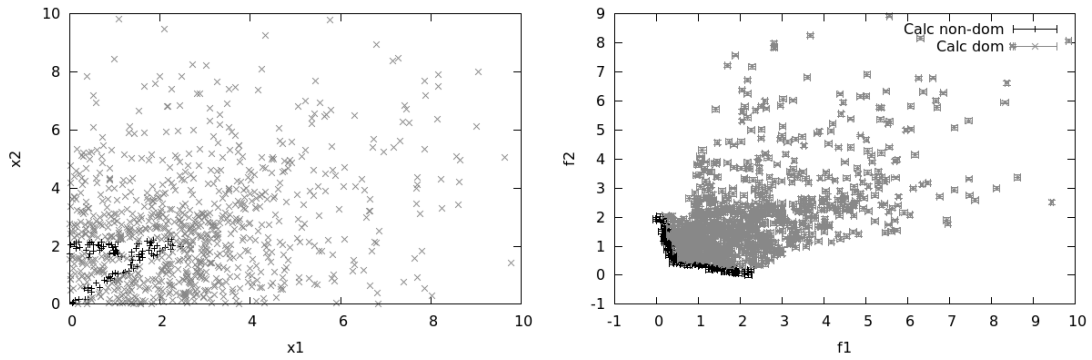
(a) (b)

Figure 12: Test-case 2: *Classical approach*, non-converging domain for new designs: a) design space, b) objective space.

5.2.2 Test-case 2 b): Convergence of the designs

For this case, new designs are randomly chosen in a domain converging toward the optimal area. In practice, with j the optimization iteration, the new designs are chosen as follows, for i in $\{1, 2\}$,

$$x_i = X\Big(2 + 8\exp\Big(-\frac{j}{45}\Big)\Big), \quad \text{with } X \sim \mathcal{U}[0, 1].$$



(a) (b)

Figure 13: Test-case 2: *Classical approach*, converging domain for new designs: a) design space, b) objective space.

The main effect of the convergence toward the optimal area is a better refinement of the Pareto optima. This can be seen in Figs. 12 and 13, where the Pareto-optimal designs and Pareto front are more precisely captured.

A quantitative comparison is performed on the number of function evaluations in Fig. 14. As concluded in [35], it can be seen that the convergence toward the optimal area induces a decreasing impact of the recursive strategy. In practice, one can see the slope of the cost

curve slowly relapsing to the one of the classical (solid) curve in Fig. 14b. This is due to the increasing ratio of non-dominated designs among the converging search domain. SABBa relies on the assisting surrogate model to become quickly accurate on the optimal area, which is highly sampled, in order to ultimately bypass all function evaluations in this area. Such a behavior can indeed be seen in Fig. 14b where both surrogate-assited strategies (*Class-acc* and *SABBa*) benefit from the convergence to reach their plateau faster. Despite the cost increase of the Fusi strategy alone, SABBa allows fewer evaluations and a very reduced final slope.

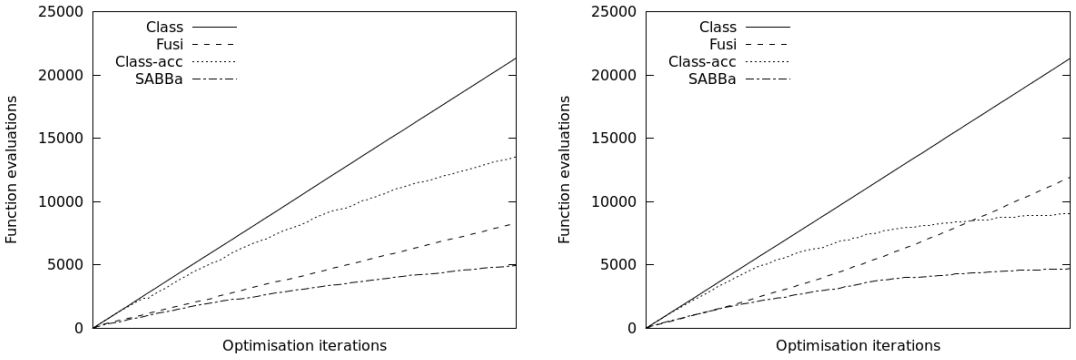


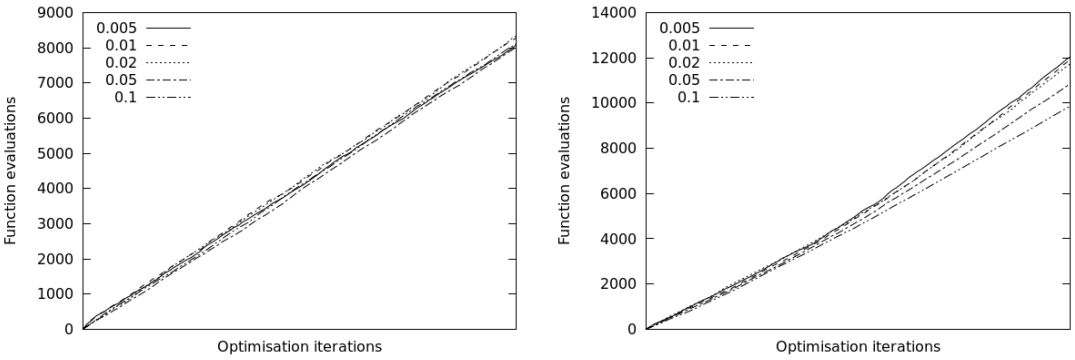
Figure 14: Test-case 2: number of function evaluations depending on the strategy for a) the non-converging case, b) the converging case.

Remark Fig. 14a can also be compared with Fig. 9a, where the gain induced by the recursive strategy was quite limited. Indeed, one can see in Fig. 12a that very few designs are non-dominated, which allows significant gains from the recursive strategy. By converging toward the optimal area, Figs. 13a has more and more non-dominated designs throughout the iterations, which explains the decreasing gains from the recursive approach alone. This can also be apprehended through the impact of $\boldsymbol{s}_2$ on the cost for both the converging and non-converging optimizations. As seen in Fig. 15a, the number of non-dominated designs is low enough for the $\boldsymbol{s}_2$ threshold to have a negligible impact on the computational cost. On the contrary, the convergence toward the optimal area increases the number of fully converged designs and therefore increases the impact of $\boldsymbol{s}_2$ throughout the iterations, which explains the separation of the curves in Fig. 15b.

5.3 Test-case 3: The Kursawe problem

The Kursawe bi-objective optimization is performed here, in order to study the framework behavior on a more complex Pareto front shape and a 3D design domain. The problem is the following:

$$\begin{aligned} \text{minimize:} \quad & \boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} \sum_{i=1}^{2} \left[-10 \exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2} \right) \right] \\ \sum_{i=1}^{3} \left[|x_i|^{0.8} + 5\sin(x_i^3) \right] \end{pmatrix} \\ \text{by changing:} \quad & (x_1, x_2, x_3) \in [-5, 5]^3 \end{aligned} \tag{11}$$



(a) (b)

Figure 15: Test-case 2: number of function evaluations (*Fusi* strategy) depending on $\boldsymbol{s}_2$ for a) the non-converging case, b) the converging case.

This problem is known to have a complex Pareto front with several discontinuities, associated with a small area of the design space. In the following, the convergence toward the optimal zone is forced, similarly to test-case 2 b). As concluded in the previous test-case, the recursive strategy is expected to show a decreasing impact throughout the iterations.

The image by $\boldsymbol{f}$ of the whole design space and the associated Pareto front are given in Figs. 16a and 16b. One can see that the front is highly discontinuous and that the range of the objective functions should allow for a high influence of the recursive strategy through poorly refined dominated boxes. The Pareto-optimal designs are represented in Fig. 16c in the $[-5, 5]^3$ domain and in Fig. 16d in a close-up view. These designs form complex disjoint sets and the approximations will be qualitatively compared to Fig. 16c.

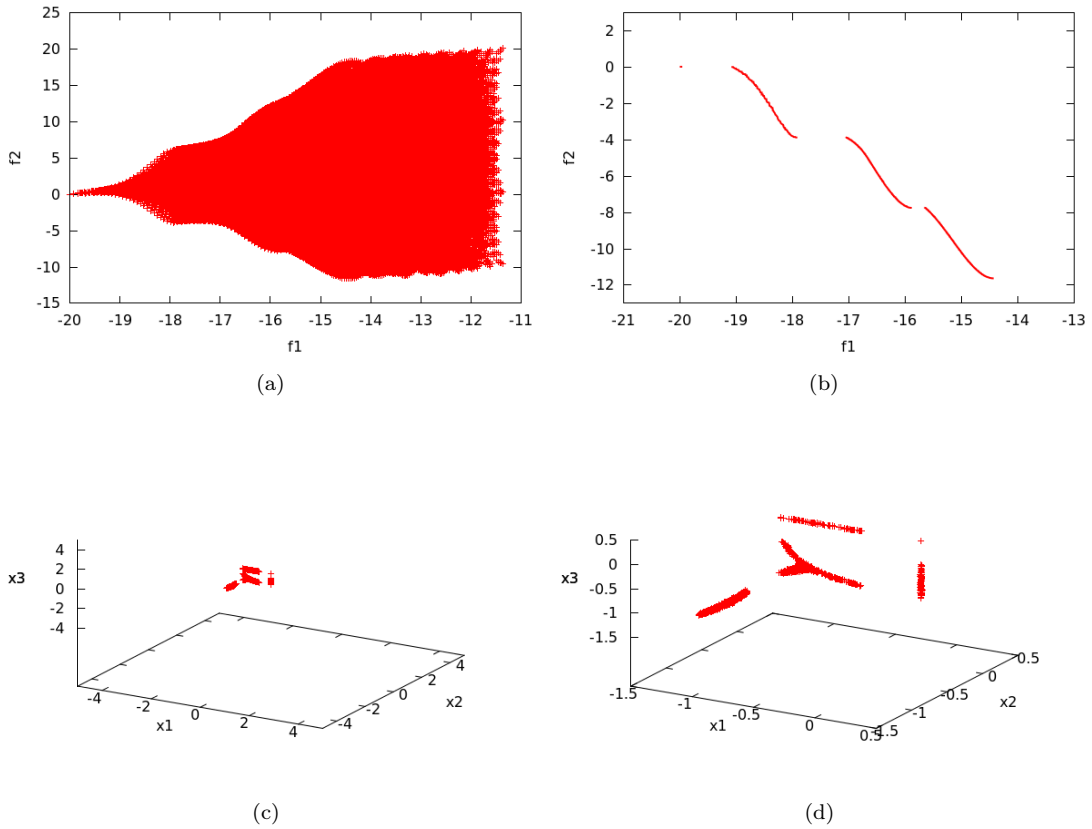
The SABBa framework is applied in the following with three levels of thresholds. First, one can see in Fig. 17 the outputs obtained with SABBa using a fine threshold, which gives results highly comparable to Fig. 16, but with only 28510 function evaluations over 48913 for the classical approach (**gain = 42%**).

The framework is also used with larger thresholds in order to further reduce the computational burden. It can be qualitatively seen that the optimal designs are less accurately captured in Fig. 18a and the associated Pareto front is depicted with a higher imprecision in Fig. 18b. However, this study is performed with only 8294 evaluations (**gain = 83%**).

Finally, very coarse user-defined thresholds are used. As it can be seen in Fig. 19a, non-dominated designs are still located in the right area, however local optimal shape is not captured. This can however be used to get a first approximation of the optimal set as the computational cost is here reduced to 3546 evaluations (**gain = 93%**).

The cost graphs are given in Fig. 20. As expected, Figs. 20a and 20b show a reduction of the impact of the recursive strategy alone and an increasing sensitivity to $\boldsymbol{s}_2$ throughout the optimization, similarly to Figs. 14 and 15.

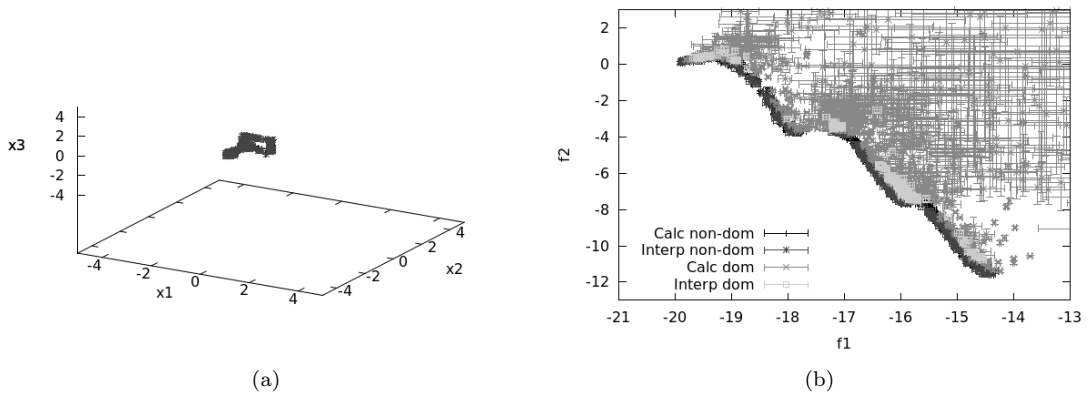
The plateau is reached in this test-case before the end of the optimization process, and Fig. 20c illustrates the momentous influence of the threshold $\boldsymbol{s}_1$ on the plateau value. The interaction between $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ is observed in Fig. 20d. One can see that, because of the low impact of $\boldsymbol{s}_2$ on the recursive approaches during the first optimization iterations, the influence of this threshold on the overall SABBa framework is mostly dependent on $\boldsymbol{s}_1$. In practice, with a coarse $\boldsymbol{s}_1$, function evaluations will be bypassed before converging toward the optimal area, which explains



(a) (b)

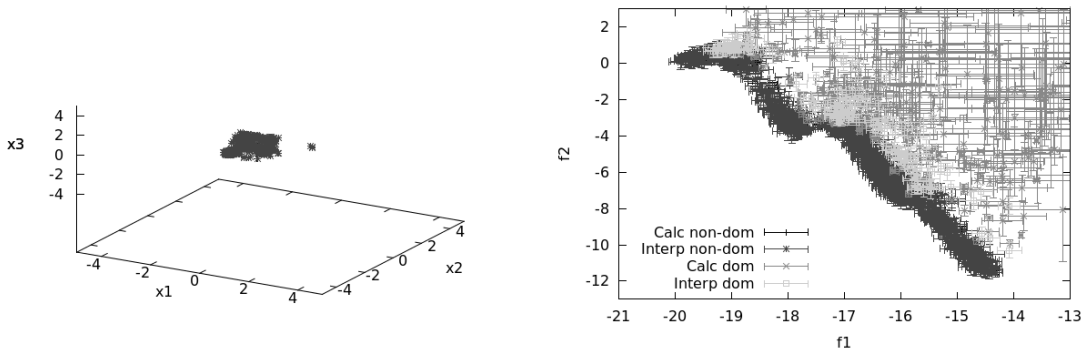
(c) (d)

Figure 16: Test-case 3: a) image in the objective space, b) associated Pareto front, c) Pareto optima in the design space, d) in a close-up view.



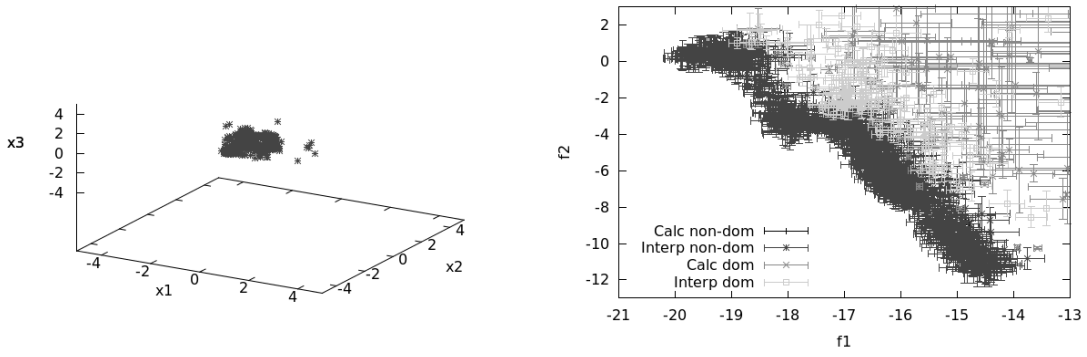
(a) (b)

Figure 17: Test-case 3: SABBa ouputs with fine thresholds in a) the design space, b) the objective space. Computational cost saving: 42%.



(a) (b)

Figure 18: Test-case 3: SABBa ouputs with moderate thresholds in a) the design space, b) the objective space. Computational cost saving: 83%.



(a) (b)

Figure 19: Test-case 3: SABBa ouputs with coarse thresholds in a) the design space, b) the objective space. Computational cost saving: 93%.

why the last three curves of Fig. 20d are very similar, as $\boldsymbol{s}_2$ has a negligible impact. On the contrary, with a fine $\boldsymbol{s}_1$, the surrogate-assisting strategy permits to avoid true computations only after a significant number of iterations. In this case, many non-dominated boxes are computed and $\boldsymbol{s}_2$ has a non-negligible impact on the global cost.

The SABBa framework yields here a significant decrease in the computational cost associated with the optimization process. The user-defined thresholds should however be carefully chosen as they directly impact both outputs accuracy and global computational cost.

6 Conclusive remarks

In this paper, a novel framework has been introduced to efficiently deal with multi-objective optimization problem with approximated objective functions. This kind of problem could notably arise in uncertainty-based optimization, when robustness measures are chosen as objective functions. The proposed strategy couples the recursive Bounding-Box approach for error-based

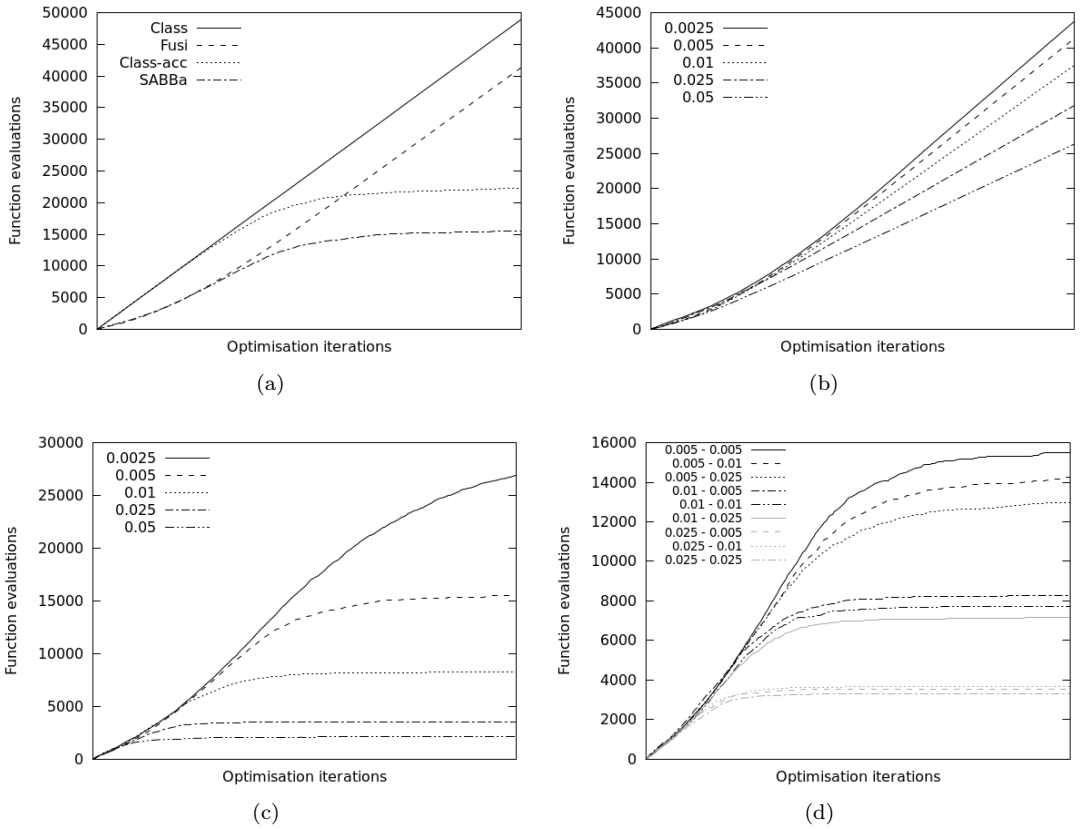


Figure 20: Test-case 3: a) number of function evaluations for each method of interest ($\boldsymbol{s}_1 = \boldsymbol{s}_2 = 0.01$), b) Fusi's approach varying $\boldsymbol{s}_2$, c) SABBa framework varying $\boldsymbol{s}_1$ with $\boldsymbol{s}_2 = 0.01$; d) varying both $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$.

objective functions developed in [35] with a surrogate-assisting strategy, based on a model refined on the fly, that aims at bypassing function evaluations. The intuitive idea is to use the Bounding-Box approach to quickly converge toward the optimal area through a poor refinement of non-efficient designs and to simultaneously build an assisting surrogate model, which refinement is driven by the optimization process toward the optimal area, to bypass the costly black-box evaluations when the surrogate error is estimated low enough. First, we have provided a mathematical proof of the robustness of the framework and of its convergence under some assumptions. Then, a numerical validation has been carried out, in order to validate the convergence, quantify the computational cost reduction and make explicit the main behaviors of the strategy.

Some innovative concepts, such as the boxed Pareto dominance or the Pareto-optimal set sequence, have been formulated in the first sections. Some theorems have also been introduced and proven in order to provide mathematical evidence about the convergence of the strategy toward the real Pareto-optimal set, the robustness of the imprecise approach (with non-zero thresholds) and the coupling with the surrogate-assisting strategy.

From a numerical point of view, the algorithm associated with the framework has been formulated and applied to several analytical test-cases.

The framework has been tested and analyzed on three different test-cases. It appears to yield a consistent cost-reduction compared to the Fusi [35] approach, which already performed better than the classical nested optimization problem. Accuracy of the outputs have been qualitatively analyzed with regards to the associated cost reduction. Finally, the influence of the user-defined thresholds have been investigated, in order to give the keys to tune them efficiently.

Note however that this framework is based on several assumptions, such as the possibility to compute a conservative error approximation, the convergence of the optimization process and the interpolation of the surrogate-assisting model at infinite samples. Plus, the objective approximation error is represented through an uncorrelated bounded product of intervals. A high correlation could significantly lower the computational burden of computing dominating boxes, whereas the uncorrelated assumption could often overestimate the true variability. Note also that the assumption of bounded error is critical for the computation of conservative errors and therefore to ensure a proper convergence of the framework. A gaussian-distributed error, unbounded, can be treated within the proposed framework with a relaxation of the basic assumptions of the theorems presented here.

Finally, the behavior of the framework when dealing with a high number of parameters has not been treated here. One would expect the surrogate-assisting model to need more training data before convergence, thus slowing down the whole process. Note that this issue can be tackled with an appropriate choice of the surrogate model (*i.e.* Automatic Relevance Determination or additive kernel for gaussian processes). Furthermore, the choice of the threshold $\boldsymbol{s}_1$ could be of particular importance and may allow to balance the computational burden with respect to the accuracy of the Pareto-optimal area.

Future developments will be directed towards the application of the proposed framework to a problem of uncertainty-based multi-objective optimization in an engineering context and to the management of constraint functions.

A Proof 1

Proof. The proof here is trivial. By definition,

$$\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\big\{\boldsymbol{x}_i\big\}_{i=1}^N\Big) = \big\{\boldsymbol{x}_i,\ i \in \mathcal{I}_1^N \mid \text{non-domination condition}\big\} \subseteq \big\{\boldsymbol{x}_i,\ i \in \mathcal{I}_1^N\big\} \equiv \big\{\boldsymbol{x}_i\big\}_{i=1}^N.$$

The same can be said for the boxed Pareto optima. □ □

B Proof 2

Proof. For $\boldsymbol{y} \notin \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\big\{\boldsymbol{x}_i\big\}_{i=1}^N\Big)$, let us assume that $\nexists \boldsymbol{y}' \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\big\{\boldsymbol{x}_i\big\}_{i=1}^N\Big),\ \boldsymbol{y}' \succ \boldsymbol{y}$, then $\boldsymbol{y} \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\big\{\boldsymbol{x}_i\big\}_{i=1}^N\Big) \cup \boldsymbol{y}\Big) = \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\big\{\boldsymbol{x}_i\big\}_{i=1}^N\Big)$ from Equation 5, which proves the first implication by contradiction. The second implication is immediate from the definition of $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}$. □ □

C Proof 3

Proof. By using the explicit definition of the Pareto front as in Equation 3, the proof is immediate as Assumption 1 gives $\forall \boldsymbol{x} \in \mathcal{X}, \forall j \in \mathcal{I}_1^N,\ f_j(\boldsymbol{x}) \in \big[\widetilde{f}_j(\boldsymbol{x}) - \widetilde{\varepsilon}_j(\boldsymbol{x}),\ \widetilde{f}_j(\boldsymbol{x}) + \widetilde{\varepsilon}_j(\boldsymbol{x})\big]$. Hence,

$\forall \boldsymbol{x}_i \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\{\boldsymbol{x}_i\}_{i=1}^N\Big)$:

$$\forall k \in \mathcal{I}_1^N, \exists j \in \mathcal{I}_1^m,\ \pm f_j(\boldsymbol{x}_i) < \pm\, f_j(\boldsymbol{x}_k),$$
$$\pm\widetilde{f}_j(\boldsymbol{x}_i) - \widetilde{\varepsilon}_j(\boldsymbol{x}_i) < \pm\, \widetilde{f}_j(\boldsymbol{x}_k) + \widetilde{\varepsilon}_j(\boldsymbol{x}_k).$$

Therefore, $\boldsymbol{x}_i \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}\Big(\big\{\big(\boldsymbol{x}_i, \widetilde{\boldsymbol{\varepsilon}}(\boldsymbol{x}_i)\big)\big\}_{i=1}^N\Big)$, which ends the proof. □ □

D Proof 4

Proof. For proving this, mathematical induction can be used.
Let us assume that $\exists k \in \mathbb{N}_+$, so that $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\{\boldsymbol{x}_i\}_{i=1}^N\Big) \subseteq \widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}$. Then $\forall \boldsymbol{x} \in \widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}$:

- if $\boldsymbol{x} \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\{\boldsymbol{x}_i\}_{i=1}^N\Big)$, $\nexists \boldsymbol{y} \in \mathcal{X}$, so that $\boldsymbol{y} \succ \boldsymbol{x}$. Hence, $\nexists \boldsymbol{y} \in \widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}} \subseteq \mathcal{X}$, so that $\boldsymbol{y} \succ \boldsymbol{x}$, and therefore $\boldsymbol{x} \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}})$;
- if $\boldsymbol{x} \notin \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\{\boldsymbol{x}_i\}_{i=1}^N\Big)$, with Proposition 2, $\exists \boldsymbol{y} \in \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\{\boldsymbol{x}_i\}_{i=1}^N\Big) \subseteq \widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}$, so that $\boldsymbol{y} \succ \boldsymbol{x}$, therefore, $\boldsymbol{x} \notin \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}})$.

which means that:

$$\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\{\boldsymbol{x}_i\}_{i=1}^N\Big) = \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}). \tag{12}$$

Finally, Lemma 1 gives $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}) \subseteq \mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}_{\mathcal{B}}}\Big(\Big\{\Big(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_i}, \widetilde{\varepsilon}^k\Big(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_i}\Big)\Big)\Big\}_{i=1}^N\Big) = \widetilde{\mathcal{X}}^{\boldsymbol{f},k+1}_{\widetilde{\mathcal{P}}}$, which ends the inductive step of the proof, yielding $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\{\boldsymbol{x}_i\}_{i=1}^N\Big) \subseteq \widetilde{\mathcal{X}}^{\boldsymbol{f},k+1}_{\widetilde{\mathcal{P}}}$.
Of course, $\mathcal{X}^{\boldsymbol{f}}_{\widetilde{\mathcal{P}}}\Big(\{\boldsymbol{x}_i\}_{i=1}^N\Big) \subseteq \{\boldsymbol{x}_i\}_{i=1}^N = \widetilde{\mathcal{X}}^{\boldsymbol{f},0}_{\widetilde{\mathcal{P}}}$, therefore, the mathematical induction proves that the robustness inclusion is verified. □ □

E Proof 5

Proof. The triangle inequality gives :

$$d_H\Big(\widetilde{\mathcal{P}}_c(\widetilde{\boldsymbol{f}}^k(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}})),\ \mathcal{P}_c\Big) \leq d_H\Big(\widetilde{\mathcal{P}}_c(\widetilde{\boldsymbol{f}}^k(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}})), \widetilde{\mathcal{P}}_c(\boldsymbol{f}(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}))\Big) + d_H\Big(\widetilde{\mathcal{P}}_c(\boldsymbol{f}(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}})),\ \mathcal{P}_c\Big). \tag{13}$$

Assumption 1 implies that $\forall \boldsymbol{x} \in \widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}},\ d_\infty\big(\boldsymbol{f}(\boldsymbol{x}), \widetilde{\boldsymbol{f}}^k(\boldsymbol{x})\big) \leq \max\limits_j \widetilde{\boldsymbol{\varepsilon}}^k_j(\boldsymbol{x})$.
Now, let us suppose that $\exists \boldsymbol{a} \in \widetilde{\mathcal{P}}_c\big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}\big)\big)$, so that $d_\infty\Big(\boldsymbol{a}, \widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}})\big)\Big) > \max\limits_{(i,j)} \widetilde{\boldsymbol{\varepsilon}}^k_j(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_i})$.
This means that $\widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}})\big) \cap \big(\boldsymbol{a}, \widetilde{\boldsymbol{\varepsilon}}_{max}\big) = \emptyset$ with $\widetilde{\boldsymbol{\varepsilon}}_{max}$ being the m-dimensional vector where each component is equal to $\max\limits_{(i,j)} \widetilde{\boldsymbol{\varepsilon}}^k_j(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_i})$.
Therefore, from Def. 8, i) whether $\exists i \in \mathcal{I}_1^N$, then $\big(\widetilde{\boldsymbol{f}}^k(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_i}), 0\big) \underset{\mathcal{B}}{\succcurlyeq} \big(\boldsymbol{a}, \widetilde{\boldsymbol{\varepsilon}}_{max}\big)$ or ii) $\forall \boldsymbol{a}' \in \big(\boldsymbol{a}, \widetilde{\boldsymbol{\varepsilon}}_{max}\big), \nexists i \in \mathcal{I}_1^N$, so that $\widetilde{\boldsymbol{f}}^k(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_i}) \succ \boldsymbol{a}'$.

- In the first case i), the dominance can be formulated as follows: $\exists j \in \mathcal{I}_1^N$, so that $\big(\widetilde{\boldsymbol{f}}^k(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_j}), \widetilde{\boldsymbol{\varepsilon}}_{max}\big) \underset{\mathcal{B}}{\succcurlyeq} \big(\boldsymbol{a}, 0\big)$ and as $\boldsymbol{a} \in \widetilde{\mathcal{P}}_c\big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}}\big)\big), \nexists i \in \mathcal{I}_1^N, \boldsymbol{f}\big(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_i}\big) \succcurlyeq \boldsymbol{a}$. Therefore, it can be inferred that $\nexists i \in \mathcal{I}_1^N, \boldsymbol{f}\big(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_i}\big) \in \big(\widetilde{\boldsymbol{f}}^k(\widetilde{\mathcal{X}}^{\boldsymbol{f},k}_{\widetilde{\mathcal{P}}_j}), \widetilde{\boldsymbol{\varepsilon}}_{max}\big)$. However, this would

mean $\exists \boldsymbol{x} \in \widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k},\ d_\infty\big(\boldsymbol{f}(\boldsymbol{x}), \widetilde{\boldsymbol{f}}^k(\boldsymbol{x})\big) > \widetilde{\varepsilon}_{max_i} > \max_j \widetilde{\varepsilon}_j^k(\boldsymbol{x})$, which is contradictory with Assumption 1.

- The second case ii) implies that $\exists \boldsymbol{b} \in \widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big)$, so that $\big(\boldsymbol{a}, \widetilde{\varepsilon}_{max}\big) \underset{\mathcal{B}}{\succcurlyeq} \big(\boldsymbol{b}, 0\big)$. However, $\exists j \in \mathcal{I}_1^N$, so that $\boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_j}^{\boldsymbol{f},k}\big) \succ \boldsymbol{a}$. Therefore, it follows $\big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_j}^{\boldsymbol{f},k}\big), \widetilde{\varepsilon}_{max}\big) \underset{\mathcal{B}}{\succcurlyeq} \big(\boldsymbol{b}, 0\big)$. As $\boldsymbol{b} \in \widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big)$, from Def. 8, it follows that $\nexists i \in \mathcal{I}_1^N$, so that $\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_i}^{\boldsymbol{f},k}\big) \succcurlyeq \boldsymbol{b}$, hence, $\nexists i \in \mathcal{I}_1^N, \widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_i}^{\boldsymbol{f},k}\big) \in \big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_j}^{\boldsymbol{f},k}\big), \widetilde{\varepsilon}_{max}\big)$, which contradicts again Assumption 1.

Finally, it has been proven that $\forall \boldsymbol{a} \in \widetilde{\mathcal{P}}_c\big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big)$, it follows that $d_\infty\Big(\boldsymbol{a}, \widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big)\Big) \leq \max_{(i,j)} \widetilde{\varepsilon}_j^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_i}^{\boldsymbol{f},k}\big)$ by contradiction. This statement holds also when inverting the Pareto front continuous sets (real and approximated), and this can proved in the same way. As a consequence, the Hausdorff distance can be written as follows:

$$d_H\Big(\widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big), \widetilde{\mathcal{P}}_c\big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big)\Big) \leq \max_{(i,j)} \widetilde{\varepsilon}_j^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_i}^{\boldsymbol{f},k}\big).$$

Hence, Assumption 2 implies that:

$$\lim_{k\to\infty} d_H\Big(\widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big), \widetilde{\mathcal{P}}_c\big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big)\Big) = 0. \tag{14}$$

Let us focus now on the second part of the sum in Equation 13.

Of course, $\forall \boldsymbol{a} \in \widetilde{\mathcal{P}}_c\big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big), \exists \boldsymbol{a}' \in \mathcal{P}_c$, so that $\boldsymbol{a}' \succ \boldsymbol{a}$ (or $\boldsymbol{a}' = \boldsymbol{a}$). Moreover, $\forall \boldsymbol{b} \in \mathbb{R}^m$ such that $\exists \boldsymbol{a}' \in \mathcal{P},\ \boldsymbol{a}' \succcurlyeq \boldsymbol{b}$, Assumption 3 with Theorem 1 provides evidence that the recursive discrete efficient set converges toward the continuous real one and that this efficient set is included in $\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}$. In other words, $\forall k \in \mathbb{N}, \exists M \in \mathbb{N}^*, \exists l \in \mathcal{I}_1^M, s.t. \forall j \in \mathcal{I}_1^m, \big|a'_j - f_j\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_l}^{\boldsymbol{f},k}\big)\big| < \big|a'_j - b_j\big|$. Thus, $\forall k \in \mathbb{N}, \exists M \in \mathbb{N}^*, \exists l \in \mathcal{I}_1^M, \boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}_l}^{\boldsymbol{f},k}\big) \succcurlyeq \boldsymbol{b}$. Hence, $\widetilde{\mathcal{P}}_c$ is always dominated by $\mathcal{P}_c$ and any element dominated by $\mathcal{P}_c$ is dominated by $\widetilde{\mathcal{P}}_c$ after a sufficient number of iterations. From Definition 8, it can be deduced that:

$$\lim_{N\to\infty} d_H\Big(\widetilde{\mathcal{P}}_c\big(\boldsymbol{f}\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big),\ \mathcal{P}_c\Big) = 0. \tag{15}$$

Finally, by combining Equations 13, 14 and 15, it comes:

$$\lim_{(N,k)\to(+\infty,+\infty)} d_H\Big(\widetilde{\mathcal{P}}_c\big(\widetilde{\boldsymbol{f}}^k\big(\widetilde{\mathcal{X}}_{\widetilde{\mathcal{P}}}^{\boldsymbol{f},k}\big)\big),\ \mathcal{P}_c\Big) = 0,$$

which ends the proof. □ □

F Proof 6

Proof. The proof is similar to the one of Lemma 1 and relies on the fact that $\forall \boldsymbol{x}$,

$$\big|\boldsymbol{f}_{OBJ}(\boldsymbol{x}_i) - \boldsymbol{f}_{opt}(\boldsymbol{x}_i)\big| \leq \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i),$$

for the refined, and that

$$\Big(\widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) > \boldsymbol{s}_1\Big) \Rightarrow \Big(\boldsymbol{f}_{OBJ}(\boldsymbol{x}_i) = \boldsymbol{f}_{opt}(\boldsymbol{x}_i)\Big),$$

for the accurate. Thus, the rough is immediate,

$$\big|\boldsymbol{f}_{OBJ}(\boldsymbol{x}_i) - \boldsymbol{f}_{opt}(\boldsymbol{x}_i)\big| \leq \boldsymbol{s}_1.$$

All those inequalities come directly from Definition 11. □ □

G Proof 7

Proof. The proof is quite straightforward and comes from the following inequalities:
If $\widetilde{\varepsilon}_d^{l_{in}}(\boldsymbol{x}_i) > \boldsymbol{s}_1$,

$$\left|\boldsymbol{f}_{OBJ}(\boldsymbol{x}_i) - \boldsymbol{f}(\boldsymbol{x}_i)\right| = \left|\widetilde{\boldsymbol{f}}^{k_{min}}(\boldsymbol{x}_i) - \boldsymbol{f}(\boldsymbol{x}_i)\right| \leq \widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i) \leq \widetilde{\boldsymbol{s}}_2(\boldsymbol{x}_i).$$

Else,

$$\begin{aligned}\left|\boldsymbol{f}_{OBJ}(\boldsymbol{x}_i) - \boldsymbol{f}(\boldsymbol{x}_i)\right| &\leq \left|\widetilde{\boldsymbol{f}}_d^{l_{in}}(\boldsymbol{x}_i) - \widetilde{\boldsymbol{f}}^{k_{min}}(\boldsymbol{x}_i)\right| + \left|\widetilde{\boldsymbol{f}}^{k_{min}}(\boldsymbol{x}_i) - \boldsymbol{f}(\boldsymbol{x}_i)\right| \\ &\leq \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) + \widetilde{\boldsymbol{\varepsilon}}^{k_{min}}(\boldsymbol{x}_i) \\ &\leq \widetilde{\boldsymbol{\varepsilon}}_d^{l_{in}}(\boldsymbol{x}_i) + \widetilde{\boldsymbol{s}}_2(\boldsymbol{x}_i) \\ &\leq \boldsymbol{s}_1 + \widetilde{\boldsymbol{s}}_2(\boldsymbol{x}_i),\end{aligned}$$

which comes from Equation 8. □ □

References

[1] Yaochu Jin and J. Branke. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317, June 2005.

[2] Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization – a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33–34):3190 – 3218, 2007.

[3] Xiaoping Du, Agus Sudjianto, and Wei Chen. An integrated framework for optimization under uncertainty using inverse reliability strategy. *Journal of Mechanical Design*, 126(4):562–570, Aug 2004.

[4] Rajan Filomeno Coelho and Philippe Bouillard. Multi-objective reliability-based optimization with stochastic metamodels. *Evolutionary Computation*, 19(4):525–560, Jan 2011.

[5] Xiaoxia Lin, Stacy L. Janak, and Christodoulos A. Floudas. A new robust optimization approach for scheduling under uncertainty: i. bounded uncertainty. *Computers & Chemical Engineering*, 28(6–7):1069 – 1085, 2004. {FOCAPO} 2003 Special issue.

[6] Stacy L. Janak, Xiaoxia Lin, and Christodoulos A. Floudas. A new robust optimization approach for scheduling under uncertainty: ii. uncertainty with known probability distribution. *Computers & Chemical Engineering*, 31(3):171 – 195, 2007.

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

[8] Krzysztof Dolinski. First-order second-moment approximation in reliability of structural systems: Critical review and alternative approach. *Structural Safety*, 1(3):211 – 231, 1982–1983.

[9] Wei Chen, J. K. Allen, Kwok-Leung Tsui, and F. Mistree. A procedure for robust design: Minimizing variations caused by noise factors and control factors. *Journal of Mechanical Design*, 118(4):478–485, Dec 1996.

[10] Ioannis Doltsinis and Zhan Kang. Robust design of structures using optimization methods. *Computer Methods in Applied Mechanics and Engineering*, 193(23–26):2221 – 2237, 2004.

[11] Matthieu Basseur and Eckart Zitzler. *A Preliminary Study on Handling Uncertainty in Indicator-Based Multiobjective Optimization*, pages 727–739. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[12] Kay Chen Tan and Chi Keong Goh. *Handling Uncertainties in Evolutionary Multi-Objective Optimization*, pages 262–292. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[13] C. Barrico and C. H. Antunes. Robustness analysis in multi-objective optimization using a degree of robustness concept. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1887–1892, 2006.

[14] Mian Li, Shapour Azarm, and Vikrant Aute. A multi-objective genetic algorithm for robust design optimization. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO '05, pages 771–778, New York, NY, USA, 2005. ACM.

[15] Kalyanmoy Deb and Himanshu Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494, Nov 2006.

[16] H. Eskandari, C. D. Geiger, and R. Bird. Handling uncertainty in evolutionary multiobjective optimization: Spga. In *2007 IEEE Congress on Evolutionary Computation*, pages 4130–4137, Sept 2007.

[17] Evan J. Hughes. *Evolutionary Multi-objective Ranking with Uncertainty and Noise*, pages 329–343. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[18] J. E. Fieldsend and R. M. Everson. Multi-objective optimisation in the presence of uncertainty. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 243–250, Sept 2005.

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

[20] I. R. Meneghini, F. G. Guimarães, and A. Gaspar-Cunha. Competitive coevolutionary algorithm for robust multi-objective optimization: The worst case minimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 586–593, July 2016.

[21] David L. J. Alexander, David W. Bulger, James M. Calvin, H. Edwin. Romeijn, and Ryan L. Sherriff. Approximate implementations of pure random search in the presence of noise. *Journal of Global Optimization*, 31(4):601–612, Apr 2005.

[22] Walter J. Gutjahr and Georg Ch. Pflug. Simulated annealing for noisy cost functions. *Journal of Global Optimization*, 8(1):1–13, Jan 1996.

[23] Antanas Žilinskas. On similarities between two models of global optimization: statistical models and radial basis functions. *Journal of Global Optimization*, 48(1):173–182, Sep 2010.

[24] D. w. Gong, Na na Qin, and Xiao yan Sun. Evolutionary algorithms for multi-objective optimization problems with interval parameters. In *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pages 411–420, Sept 2010.

[25] P. Limbourg and D. E. S. Aponte. An optimization algorithm for imprecise multi-objective problem functions. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 459–466 Vol.1, Sept 2005.

[26] G. L. Soares, F. G. Guimaraes, C. A. Maia, J. A. Vasconcelos, and L. Jaulin. Interval robust multi-objective evolutionary algorithm. In *2009 IEEE Congress on Evolutionary Computation*, pages 1637–1643, May 2009.

[27] Xiaoping Du. Unified uncertainty analysis by the first order reliability method. *Journal of Mechanical Design*, 130(9):091401–091401–10, Aug 2008.

[28] Philipp Limbourg. *Multi-objective Optimization of Problems with Epistemic Uncertainty*, pages 413–427. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[29] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471 – 483, 2014.

[30] Jonas Ide and Anita Schöbel. Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts. *OR Spectrum*, 38(1):235–271, 2016.

[31] Rim Kalaı¨, Claude Lamboray, and Daniel Vanderpooten. Lexicographic alpha-robustness: An alternative to min–max criteria. *European Journal of Operational Research*, 220(3):722 – 728, 2012.

[32] K. Kuhn, A. Raith, M. Schmidt, and A. Schöbel. Bi-objective robust optimisation. *European Journal of Operational Research*, 252(2):418 – 431, 2016.

[33] Bernard Roy. Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200(3):629 – 638, 2010.

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

[35] 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.

[36] Virginia Torzcon and Michael W Trosset. Using approximations to accelerate engineering design optimization. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization: A Collection of Technical Papers, Part 2*, pages 738–748. American Institute of Aeronautics and Astronautics, 1998.

[37] Yaochu Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494, Oct 2002.

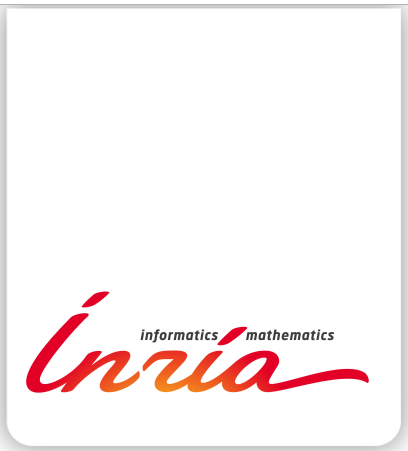
[38] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011.

[39] Michael Emmerich, Alexios Giotis, Mutlu Özdemir, Thomas Bäck, and Kyriakos Giannakoglou. *Metamodel—Assisted Evolution Strategies*, pages 361–370. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[40] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, Aug 2006.

[41] D. Buche, N. N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(2):183–194, May 2005.

[42] Malek Ben Salem, Olivier Roustant, Fabrice Gamboa, and Lionel Tomaso. Universal prediction distribution for surrogate models. 2015.



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