An adaptive strategy on the error of the objective functions for uncertainty-based derivative-free optimization

Francesca Fusi, Pietro Marco Congedo

HAL Id: hal-01251604
https://inria.hal.science/hal-01251604

Submitted on 28 May 2016

An adaptive strategy on the error of the objective functions for uncertainty-based derivative-free optimization

F. Fusi[a], P.M. Congedo[b]

[a]*Department of Aerospace Science and Technology, Politecnico di Milano, 20156 Milan, Italy*
[b]*Team CARDAMOM, INRIA Bordeaux Sud-Ouest, Talence, 33405 Cedex, France*

Abstract

In this work, a strategy is developed to deal with the error affecting the objective functions in uncertainty-based optimization. We refer to the problems where the objective functions are the statistics of a quantity of interest computed by an uncertainty quantification technique that propagates some uncertainties of the input variables through the system under consideration. In real problems, the statistics are computed by a numerical method and therefore they are affected by a certain level of error, depending on the chosen accuracy. The errors on the objective function can be interpreted with the abstraction of a bounding box around the nominal estimation in the objective functions space. In addition, in some cases the uncertainty quantification methods providing the objective functions also supply the possibility of adaptive refinement to reduce the error bounding box. The novel method relies on the exchange of information between the outer loop based on the optimization algorithm and the inner uncertainty quantification loop. In particular, in the inner uncertainty quantification loop, a control is performed to decide whether a refinement of the bounding box for the current design is appropriate or not. In single-objective problems, the current bounding box is compared to the current optimal design. In multi-objective problems, the decision is based on the comparison of the error bounding box of the current design and the current Pareto front. With this strategy, fewer computations are made for clearly dominated solutions and an accurate estimate of the objective function is provided for the interesting, non-dominated solutions. The results presented in this work prove that the proposed method improves the efficiency of the global loop, while preserving the accuracy of the final Pareto front.

Keywords:
Uncertainty-based optimization, Robust optimization, Adaptive error bounding boxes

1. Introduction

In many engineering applications, the design process has been greatly improved by the employment of optimization methods. Optimization can explore the design space to define

---

a new, different optimal design or improve an existing one with a local modification. Any optimization metod requires a model of the system under consideration, e.g. a system of governing equations. Several modelling assumptions may be necessary at this stage to define a model of the system that may be representative of the real-world system, with moderate complexity. For instance, it may be necessary to neglect some physical phenomena or to approximate them with a simple model. In addition, it may be difficult to assign a deterministic value to the variables included in the model of the system, because they are known within an uncertainty band. For example, the environmental conditions in which the system will operate or the geometric tolerance of the final product fall into this latter case. A comprehensive discussion on the numerous sources of uncertainty that are encountered in engineering design is given in Ref. [1].

It is clear that the uncertainty on the system model itself or the system variables may impact the result of the optimization. To take into account the effect of uncertainty at the very beginning of the design problem, uncertainty-based design optimization has been introduced in the last decades [2, 3]. An example of such an approach is the robust design presented by Taguchi in Ref. [4]. In robust design, the optimization seeks the design with improved mean value and minimal variance with respect to uncertain operating conditions. Another application of uncertainty-based optimization methods is reliability-based design, where a design is computed to minimize the probability of system failure.

1.1. Methods for uncertainty-based optimization

In general, a method for uncertainty-based optimization relies on the coupling of an optimization algorithm and an Uncertainty Quantification (UQ) method. The former is a method which solves a single- or multi-objective optimization problem. The UQ method propagates the effect of the uncertainty on the input variables into the performance, by computing the statistics of the performance. A representation of the coupling of an optimization algorithm and a UQ method is given in Fig. 1. The outer loop is performed by the optimization algorithm which generates a new design $\mathbf{x}_i$ based on the history of the optimization procedure. The new design is passed on to the inner loop in which a UQ method computes an approximation of the statistics $\mathbf{f}(\mathbf{x}_i)$ of the performance of interest. The statistics are the objective functions of the optimization loop, which are passed on to the optimization
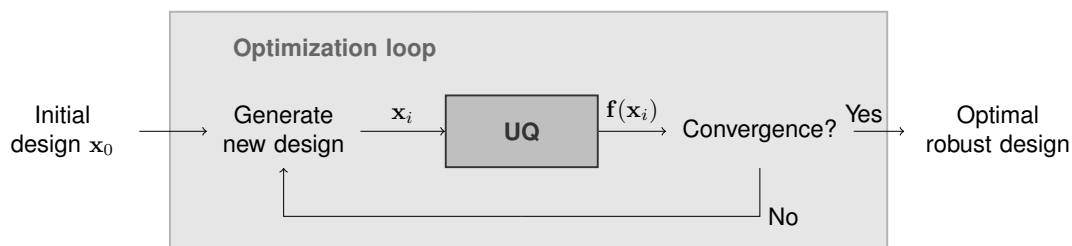

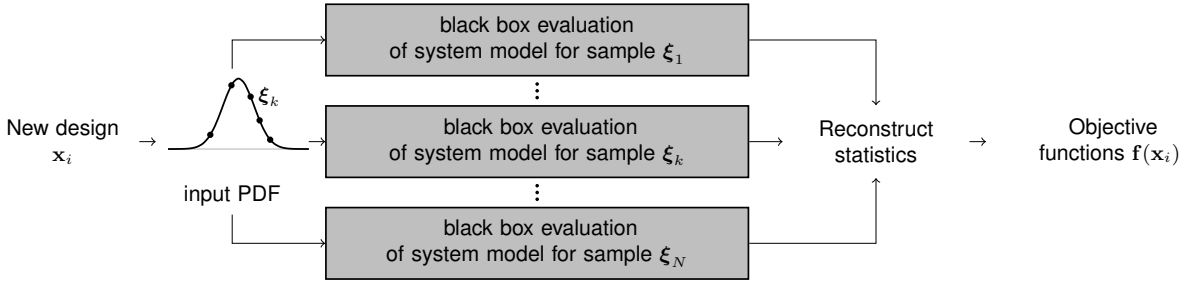Figure 1: Block diagram of the uncertainty-based optimization loop.

Figure 2: Block diagram of non-intrusive UQ method.

algorithm, which in turn checks if convergence is reached. If the convergence criterion is not met, the loop goes on with a new design.

Classical approaches for robust design optimization, called in this paper *decoupled* approaches, are based on two different and independent methods for solving the optimization problem and the reconstruction of the statistics, respectively. Decoupled approaches are widely employed in the uncertainty-based optimization literature [5, 6] and they are very expensive. In fact, performing a deterministic optimization generally requires several computations of the performance, since many designs has to be tested in the loop. This means that many computations for solving the equations describing the system under consideration are required. In the case of uncertainty-based optimization, the cost is even higher. When considering non-intrusive methods for UQ, the system of equations is considered as a blackbox, and the equations are solved several times for different samples of the input uncertain variables (see Fig. 2). Then, in order to reconstruct the statistics for a given design, many samples of the performance are required which correspond to different realizations of the input uncertain parameters. The cost of solving the equations and therefore the optimization problem can be very high in most engineering applications. For instance, let us consider an optimization problem in aerodynamics. A typical goal of aerodynamic optimization is the maximization of the aerodynamic efficiency, i.e. the ratio of aerodynamic lift over drag, by means of a modification of the wing shape. In this case, the mathematical model describing the aerodynamic system is represented by the governing equations of fluid dynamics. The numerical solution of this equation is achieved by Computational Fluid Dynamics (CFD) tools that require significant CPU time. Thus, a single function evaluation is very expensive from a computational point of view and in this case, decoupled approaches are not very attractive.

A possible way to alleviate the cost of uncertainty-based optimization is to create a stronger coupling between the inner loop and the outer loop. For instance, the two loops could share information, and the accuracy in the stochastic space may be increased as the algorithm gets closer to the optimum. Only few attempts to develop such an idea are given in the literature. For instance, in Ref. [7] a method has been presented, which improves the coupling between a Monte Carlo simulation and the stochastic annealing algorithm. To this

end, the number of samples used to compute the statistics of the performance in the Monte Carlo simulation are increased as the annealing temperature decreases. This strategy has been extended to genetic algorithms in Ref. [8] to deal with multi-objective problems. In this case, the number of samples increases after a certain number of generations. Another example is the hierarchical stochastic optimization presented in Ref. [9], where the resolution in the stochastic space is increased over the iterations. A more recent work [10] illustrates the Simplex^2 method, a method which efficiently couples the Nelder-Mead algorithm for the optimization loop and the Simplex Stochastic Collocation [11, 12] for the uncertainty quantification part. This method adresses the problem of single-objective optimization.

1.2. Objective of the novel strategy

In such a context, the present work presents a novel strategy to improve the efficiency of uncertainty-based optimization. The method is based on an adaptive strategy in the UQ loop to reduce the number of samples in the stochastic space for those designs that are not interesting for optimization purposes. The main idea is to use the error bounds on the estimates of the objective functions provided by the UQ method and to define a bounding box in the objective space around the objective function estimate. A comparison between the bounding box and the current Pareto front (or current best design in the case of single objective optimization) during the optimization loop determines whether a refinement is needed in the stochastic space. This method applies to both single- and multi-objective optimization problems. Unlike the methods presented in the literature, the adaptation is not set a priori, but instead it depends on the convergence history of the optimization procedure and the performance of each new design. The strategy is here developed with either a Non-dominated Sorting Genetic Algorithm or the BiMADS algorithm for the optimization loop, and the Simplex Stochastic Colocation for the UQ method. However, the strategy may apply to different algorithms in both the optimization and the stochastic part, as long as it is possible to define an adaptive strategy with the available UQ method.

The concept of a bounding box around a design vector has been discussed in several papers to take into account the uncertainty affecting the objectives in deterministic optimization. For instance in Ref. [13] the bounding box is defined to consider the confidence interval of the objective function in deterministic contrained optimization. In Refs. [14, 15], the authors replace the Pareto dominance with a probability of Pareto dominance and modify the operators of a genetic algorithm accordingly. Similarly, evidence theory is used in Ref. [16]. To the authors' knowledge all these attempts have never tackled the coupling of an optimizer with a UQ method and they have never been used in conjuction with an adaptive strategy to accelerate the optimization loop.

The paper is outlined as follows. Section 2 presents a generic uncertainty-based optimization problem and the nested loops used to solve it. The novel strategy is described in general terms in Section 3. The method is implemented for the optimization algorithms and the UQ method explained in Section 4. Section 5 is devoted to a description of the estimates of the error used in the definition of the error bounding box. In the second part of this section, the estimates of the error are computed for some analytic test cases. The results of the proposed strategy are discussed in Section 6 for two algebraic test cases. Finally,

the proposed strategy is applied in Sec. 7 to an engineering test case, i.e. the aerodynamic optimization of an airfoil with uncertain operating conditions.

2. Uncertainty-based optimization problem

The goal of optimization in engineering systems is to find the design that improves at most a desired performance. The quantity of interest u typically depends on the solution ϕ of a mathematical model describing the system under consideration. For this general description, let us consider a model for a time- and space-dependent system, which could be represented by the following equation

$$\mathcal{L}(\mathbf{x}, t, \mathbf{y}; \phi(\mathbf{x}, t, \mathbf{y})) = 0, \tag{1}$$

with appropriate initial and boundary conditions. To give a rigorous mathematical description, the operator $\mathcal{L}$ is defined on the domain $D \times T \times \mathcal{X}$, where $\mathbf{y} \in D$ and $t \in T$ are the spatial and temporal coordinates with $D \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, and $T \subset \mathbb{R}$. The vector of the design variables $\mathbf{x}$ is defined on the domain $\mathcal{X} \subset \mathbb{R}^n$, where n is the number of design variables. The solution of Eq. (1) is $\phi(\mathbf{x}, t, \mathbf{y})$ and the performance is a function of this solution $u = u(\phi(\mathbf{x}, t, \mathbf{y}))$.

Now, let us consider that some variables in the system model are affected by uncertainty. To highlight the dependency on the uncertain variables, equation (1) is written as the following

$$\mathcal{L}(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi}(\boldsymbol{\omega}); \phi(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi}(\boldsymbol{\omega}))) = 0. \tag{2}$$

The new element in this equation is the vector $\boldsymbol{\xi}$ collecting the uncertain variables, which are modelled as a set of n_ξ second-order random parameters: $\boldsymbol{\xi}(\boldsymbol{\omega}) = \{\xi_1(\omega_1), \ldots, \xi_{n_\xi}(\omega_{n_\xi})\} \in \Xi$, with $\Xi \subset \mathbb{R}^{n_\xi}$. The symbol $\boldsymbol{\omega} = \{\omega_1, \ldots, \omega_{n_\xi}\} \in \Omega \subset \mathbb{R}^{n_\xi}$ denotes events in the complete probability space $(\Omega, \mathcal{F}, P)$ with $\mathcal{F} \subset 2^\Omega$, the σ–algebra of subsets of Ω and P a probability measure. The random variables $\boldsymbol{\omega}$ are by definition standard uniformly distributed as $\mathcal{U}(0, 1)$. Random parameters $\boldsymbol{\xi}(\boldsymbol{\omega})$ can have any arbitrary probability density $p_\xi(\boldsymbol{\xi}(\boldsymbol{\omega}))$. The argument $\boldsymbol{\omega}$ is dropped from here on to simplify the notation. In the case presented in Eq. (2), the solution ϕ and consequently the performance u also depends on the uncertain variables $\boldsymbol{\xi}$: $u = u(\phi(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi}((\omega))))$.

When uncertainty is considered, the goal of the optimization problem is typically to control the variability of the performance u with respect to the uncertainty in the system variables $\boldsymbol{\xi}$. For instance, in Taguchi robust design [4] the goal is to find a design which provides a satisfactory mean performance, and minimizes its variability at the same time. In this way, the system will operate adequately even in off-design conditions. For this case, the objectives of the optimization are two low-order statistical moments of the performance of interest u: the mean value and the variance. In general, a problem of uncertainty-based optimization is stated in mathematical terms as the following

$$
\begin{aligned}
\text{minimize:} \quad & \mathbf{f}\left(u\left(\mathbf{x}, \boldsymbol{\xi}\right)\right) \\
\text{subject to:} \quad & \mathbf{g}\left(u\left(\mathbf{x}, \boldsymbol{\xi}\right)\right) \geq \mathbf{0} \\
\text{by changing:} \quad & \mathbf{x} \in \mathcal{X},
\end{aligned}
\tag{3}
$$

where the m objective functions are collected in vector $\mathbf{f} \subset \mathbb{R}^m$, and $\mathbf{g} \subset \mathbb{R}^q$ are the q constraints of the optimization problem. Please note that the arguments $\mathbf{y}$ and t are dropped in the preceding equation to highlight only the dependency on the design and uncertain variables.

As previously mentioned, the objective vector $\mathbf{f}$ is composed of one or more statistics of the performance of interest u; for instance, in robust design

$$\mathbf{f} = \{\mu_u^1(\mathbf{x}), \mu_u^2(\mathbf{x})\}, \tag{4}$$

where the statistical moment $\mu_u^\alpha(\mathbf{x})$ of order α of the output of interest u is given by

$$\mu_u^\alpha(\mathbf{x}) = \int_\Xi u(\mathbf{x}, \boldsymbol{\xi})^\alpha p_\xi(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}. \tag{5}$$

Please note that the objective functions $\mathbf{f}$ do not depend on the uncertain variables.

The statistics are a function only of the design variables and they have to be computed at each iteration in the optimization procedure. Beside the numerical model of the system, two are the most important ingredients for uncertainty-based design optimization: (i) an optimization algorithm to solve the optimization stated in Eq. (3) and (ii) an uncertainty quantification technique that computes the statistics of the output of interest (Eq. (5)) starting from the input uncertainty. The latter operation has to be performed for each design vector in the optimization loop. With reference to Fig. 1, an uncertainty-based optimization problem can be regarded as two nested loops. The outer loop, based on the optimization algorithm, solves Eq. (3), while the inner UQ loop computes Eq. (5). The next section presents the novel strategy to couple the UQ method and the optimization loop.

3. Uncertainty-based optimization with adaptive error bounding boxes

This section presents an overview of the concepts and methods used in the optimization loop of the uncertainty-based optimization framework. As presented in Section 2, a general uncertainty-based optimization problem (Eq. (3)) is inherently multi-objective: the mean value is used in conjunction with the variance in robust design, or with the probability of the occurrence of failure events in the application to reliability-based design. Therefore, a discussion on the methods to tackle multi-objective problems is here presented.

In the first part, the concept of Pareto dominance is discussed. This dominance relation has been developed to define the concept of "optimality" in a multi-dimensional objective space. Also, the discussion serves as a basis to the development of the proposed strategy. In fact, the Pareto dominance relation is adapted in the second part of the section for the application to the proposed adaptive method. It is clear that the Pareto dominance criterion is a general criterion that can be applied to the single-objective case as well.

3.1. Pareto dominance relation

In many applications, conflicting criteria drive the design process. In this case the optimization problem has a multi-objective formulation. The solutions of multi-objective problems ($m > 1$) is a set of solutions (unless of course the objectives are concurrent). These

solutions represent a trade-off between the objectives considered in the optimization and they are selected according to the Pareto dominance relation.

Definition 1. *For two vectors* $\mathbf{a}, \mathbf{b} \in \mathcal{X}$ *with* $\mathcal{X}$ *the design variables domain*

$$\mathbf{a} \succ \mathbf{b} \text{ (}\mathbf{a} \text{ dominates } \mathbf{b}\text{) iff } \beta_j f_j(\mathbf{a}) \leq \beta_j f_j(\mathbf{b}) \quad \forall j = 1, ..., m \text{ and}$$
$$\exists j \text{ such that } \beta_j f_j(\mathbf{a}) < \beta_j f_j(\mathbf{b})$$
$$\mathbf{a} \sim \mathbf{b} \text{ (}\mathbf{a} \text{ is indifferent } \mathbf{b}\text{) iff } \mathbf{a} \text{ does not dominate } \mathbf{b} \text{ and } \mathbf{b} \text{ does not dominate } \mathbf{a}$$

Coefficient β_j *indicates the goal of the optimization of function* f_j:

$$\beta_j = \begin{cases} 1 & \text{if } f_j \text{ is to be minimized} \\ -1 & \text{if } f_j \text{ is to be maximized} \end{cases}$$

Definition 2. *A decision vector* $\mathbf{x} \in \mathcal{X}$ *is said to be* non-dominated *regarding a set* $\mathcal{A} \subseteq \mathcal{X}$ *if and only if*

$$\nexists \mathbf{a} \in \mathcal{A} : \mathbf{a} \succ \mathbf{x}$$

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

The solution vectors that are non-dominated regarding $\mathcal{X}$ are the Pareto optimal solutions, and they belong to the set denoted as $\mathcal{X}_\mathcal{P}$. The image of the set $\mathcal{X}_\mathcal{P}$ is called Pareto front and it defined as $\mathcal{P} = \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_\mathcal{P}\}$.

The Pareto dominance is used in multi-objective optimization problems to rank the design vectors and to define the optimal solutions. Please note that single-objective optimization ($m = 1$) can also be treated with this criterion. In this case, the third option in Definition 1 cannot be satisfied, since only one function is considered: either vector $\mathbf{a}$ dominates vector $\mathbf{b}$ or viceversa.

3.2. Pareto dominance with error bounding boxes

Let us assume that each estimate of the objective functions $f_j, j = 1, ..., m$ is affected by an error $\varepsilon_{f_j}, j = 1, ..., m$. The error is associated to the approximate method used in the UQ loop to reconstruct the statistics of the performance u, i.e. the objective functions. Because of the error in the estimation of the statistics, each objective is a function f'_j that could take any value in the interval

$$f'_j(\mathbf{x}) \in [\underline{f}_j, ..., \bar{f}_j] \quad \forall i = 1, ..., m, \tag{6}$$

where the lower bound $\underline{f}_j$ and upper bound $\bar{f}_j$ are obtained respectively by subtracting and adding the error ε_{f_j} to the function estimate f_j. For instance, assuming the first objective f_1 is the mean value μ, the lower bound $\underline{f}_1$ and and the upper bound $\bar{f}_1$ read

$$\underline{f}_1 = \mu - \varepsilon_\mu \qquad \bar{f}_1 = \mu + \varepsilon_\mu,$$

where ε_μ is the error affecting the mean value. Inside the adaptive UQ loop, the error ε_{f_j} decreases with increasing number of samples. Typically, a tolerance τ is used, and when the

error ε_{f_j} is lower than the tolerance the loop breaks. Thus, for a given design vector $\mathbf{a}$ the amplitude of the errors is a function of the tolerance τ applied in the adaptive uncertainty quantification method. This holds true also for the objective function f'_j. Thus, both the objective function f'_j and the associated error ε_{f_j} depends not only on the design variables $\mathbf{x}$, but also on the tolerance value τ:

$$f'_j = f'_j(\mathbf{x}, \tau) \qquad \varepsilon_{f_j} = \varepsilon_{f_j}(\mathbf{x}, \tau) \qquad \forall i = 1, ..., m. \tag{7}$$

Once the error is lower than the limiting tolerance $\hat{\tau}$, the objective function passed on to the optimization algorithm is set equal to the estimate f_j given by the UQ method.

Typically, the same value of the tolerance is set for each design in the optimization loop, and the optimization and the uncertainty quantification are completely decoupled. Instead, in the proposed strategy, the value of the tolerance is updated inside the inner UQ loop, according to a comparison between the current Pareto front and the m-dimensional hypercube composed of the upper and lower bounds of each objective function. The hypercube is referred to as *error bounding box* and it is a function of the design vector and the tolerance:

$$\mathcal{B}(\mathbf{x}, \tau) = [\underline{f_1}, \bar{f_1}] \times ... \times [\underline{f_m}, \bar{f_m}]. \tag{8}$$

The error bounding box has also been used in Refs. [17, 13] to consider a confidence interval around the objective function. To compare the box and the Pareto front, a modified definition of the Pareto dominance criterion is used, which takes into account the error associated to the objective function of the current design. Let us consider a single iteration inside the optimization loop: the comparison is carried out between the current design $\mathbf{a}$ and one of the non-dominated solution vector $\mathbf{b}$ belonging to the current Pareto set.

Definition 3. *Consider two vectors $\mathbf{a}, \mathbf{b} \in \mathcal{X}$. The objective functions $\mathbf{f}(\mathbf{a}, \tau)$ of the first vector depend on the current tolerance τ (and thereby current refinement). The objective functions $\mathbf{f}(\mathbf{b})$ of the second vector have been computed in a previous step up to the limiting tolerance $\hat{\tau}$ and do not depend on τ. It is possible to state that:*

$$\mathbf{a} \succ \mathbf{b} \ (\mathbf{a} \ \text{dominates} \ \mathbf{b}) \ \text{iff} \ \beta_j \left[f_j(\mathbf{a}, \tau) - \beta_j \varepsilon_{f_j}(\mathbf{a}, \tau) \right] \leq \beta_j f_j(\mathbf{b}) \quad \forall j = 1, ..., m \ \text{and}$$
$$\exists j \ \text{such that} \ \beta_j \left[f_j(\mathbf{a}, \tau) - \beta_j \varepsilon_{f_j}(\mathbf{a}, \tau) \right] < \beta_j f_j(\mathbf{b})$$
$$\mathbf{a} \sim \mathbf{b} \ (\mathbf{a} \ \text{is indifferent} \ \mathbf{b}) \ \text{iff} \ \mathbf{a} \ \text{does not dominate} \ \mathbf{b} \ \text{and} \ \mathbf{b} \ \text{does not dominate} \ \mathbf{a}$$

where coefficient β_j indicates the goal of the optimization of function f_j:

$$\beta_j = \begin{cases} 1 & \text{if } f_j \text{ is to be minimized} \\ -1 & \text{if } f_j \text{ is to be maximized} \end{cases}$$

With the use of the coefficient β_j, the definition considers the lower bound of the i-th function for a problem of minimization of f_j, and analogously the upper bound for maximization problem. Let us consider the example presented in Fig. 3 where a minimization of two objectives $\mathbf{f} = \{f_1, f_2\}$ is performed. In this case, to compare the grey shaded box with
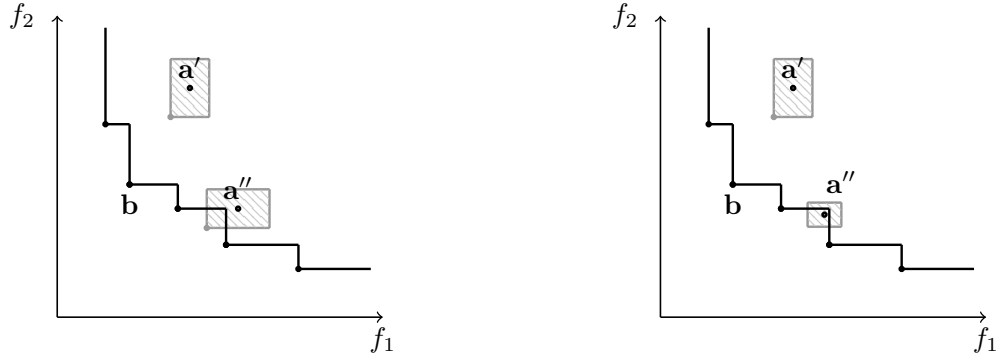
(a) Pareto dominance comparison: vector $\mathbf{a}'$ is clearly dominated, while the objectives of vector $\mathbf{a}''$ has to be computed with further refinement.

(b) Boxes after refinement of design $\mathbf{a}''$: the refinement proves that vector $\mathbf{a}''$ is non-dominated.

Figure 3: Scheme of Pareto dominance with error bounding boxes for a minimization problem.

the current front in the objective space, the point corresponding to the lower bounds ($f_1 - \varepsilon_{f_1}, f_2 - \varepsilon_{f_2}$) is taken (the point is marked with a grey circle in the figure). Design $\mathbf{a}'$ is clearly dominated, whereas the box associated with design $\mathbf{a}''$ crosses the front. In the latter case, neglecting the error on the objective functions would result in excluding the design from the non-dominated set. However, a refinement in the stochastic space could improve the estimate of the objective functions and it could prove that the design is actually non-dominated (Fig. 3b).

In practice, if an adaptive UQ loop is considered, an estimate of the actual error is provided for each objective function: $\tilde{\varepsilon}_{f_j}$. The bounding box is built with the estimate of the error, rather than the error itself, and it is important that the estimate is conservative.

Definition 4. *In a minimization problem of m objectives affected by an error ε_{f_j}, $j = 1, ..., m$, it is possible to state with certainty that*

$$\mathbf{b} \succ \mathbf{a} \ (\mathbf{b} \ \text{dominates} \ \mathbf{a})$$

if and only if

$$f_j(\mathbf{a}, \tau) - f_j(\mathbf{b}, \tau) > \varepsilon_{f_j}(\mathbf{a}, \tau) + \varepsilon_{f_j}(\mathbf{b}, \tau) \quad \forall j = 1, ..., m. \tag{9}$$

When the objective functions are not affected by error, the previous definiton corresponds to the inequality of the Pareto dominance relation (i.e. Definition 1).

Theorem 1. *Given a conservative estimate $\tilde{\varepsilon}_{f_j}$ of the error ε_{f_j} affecting the m objectives of a minimization problem, if*

$$f_j(\mathbf{a}, \tau) - f_j(\mathbf{b}, \tau) > \tilde{\varepsilon}_{f_j}(\mathbf{a}, \tau) + \tilde{\varepsilon}_{f_j}(\mathbf{b}, \tau) \quad \forall j = 1, ..., m \tag{10}$$

then with certainty

$$\mathbf{b} \succ \mathbf{a} \ (\mathbf{b} \text{ dominates } \mathbf{a}).$$

Proof.
If the estimate is conservative, then

$$\tilde{\varepsilon}_{f_j} > \varepsilon_{f_j} \quad \forall j = 1, ..., m \tag{11}$$

thus

$$\begin{aligned}
\varepsilon_{f_j}(\mathbf{a}, \tau) + \varepsilon_{f_j}(\mathbf{b}, \tau) &< \tilde{\varepsilon}_{f_j}(\mathbf{a}, \tau) + \tilde{\varepsilon}_{f_j}(\mathbf{b}, \tau) \\
&< f_j(\mathbf{a}, \tau) - f_j(\mathbf{b}, \tau) \qquad \forall j = 1, ..., m.
\end{aligned} \tag{12}$$

The latter relation is Definition 4. Thus, the dominance criterion with a conservative estimate of the error guarantees the detection the truly non-dominated solutions.

3.3. Algorithm

Leveraging the idea of the bounding box, an adaptive refinement on the error affecting the objective function is developed. The novel strategy proposes to modify only the inner UQ loop (Fig. 1), which provides the objective function $\mathbf{f}$ for a given design.

The operations of the adaptive strategy are presented in Algorithm 1. For the i-th design, the current design $\mathbf{x}_i$ and the set $\mathcal{Y}_i$ of the objective functions of the old design vectors are passed to the function. The set is used to compute the Pareto front according to Definition 1. The tolerance τ is initialized to a coarse, moderately high value $\bar{\tau}$. Then, the objective function is evaluated up to the current tolerance with the UQ method of choice. With the adaptive UQ method, the error typically deacreases by increasing the number of samples n_s, so the number of samples is increased if the error is higher than the tolerance. When $\tilde{\varepsilon}_{f_j} < \tau \ \forall j = 1, ..., m$, the error bounding box is compared with the current Pareto front: if the box is non-dominated, the tolerance is reduced by an order of magnitude and another iteration is performed. If the box is dominated according to Definition 3, the while loop is broken and the next individual is computed. The loop stops in any cases when the error is lower than the tight tolerance $\underline{\tau}$.

The algorithm is formulated in such a way that the accuracy of the objective estimates is variable throughout the design space. In fact, the limiting tolerance $\hat{\tau}$ is different for each design. Design vectors far from the Pareto front are computed with the coarse tolerance $\hat{\tau} = \bar{\tau}$, the Pareto solutions are computed with the tolerance $\hat{\tau} = \underline{\tau}$ providing the highest accuracy (and highest number of samples), and the intermediate solutions are computed with an intermediate tolerance on a case-by-case basis in the loop described in Algorithm 1.

10

Algorithm 1 Scheme of error bounding box refinement

1: **procedure** EVALUATE($\mathbf{x}_i$, $\mathcal{Y}_i$)
2:     compute Pareto front from $\mathcal{Y}_i$ with Definition 1
3:     $\tau = \bar{\tau}$
4:     $\tilde{\varepsilon} = BIG$
5:     **while** ($\tau \geq \underline{\tau}$) $\wedge$ ($\tilde{\varepsilon} > \tau$) **do**
6:         evaluate $\mathbf{f}(\mathbf{x}_i)$ and $\tilde{\varepsilon}(\mathbf{x}_i)$ with current τ
7:         **if** $\tilde{\varepsilon} < \tau$ **then**
8:             compare current box $\mathcal{B}(\mathbf{x}_i)$ with front according to Definition 3
9:             **if** $\mathcal{B}(\mathbf{x}_i)$ is non-dominated **then**
10:                 $\tau\ = \tau/10$
11:                 $n_s = n_s + 1$
12:                 continue
13:             **else**
14:                 exit while
15:             **end if**
16:         **else**
17:             $n_s = n_s + 1$
18:         **end if**
19:     **end while**
20:     **return** $\mathbf{f}(\mathbf{x}_i)$
21: **end procedure**

As a final remark, the refinement proposed in this strategy is based on a comparison of the current design with the Pareto front. Thus, it is strictly associated to the performance of each design and to the importance of such a design in the optimization history. This is a key difference with respect to other approaches in the literature. For instance, the number of samples increases as the annealing temperature decreases in the stochastic annealing [7], or as the number of generations increases in stochastic genetic algorithm [8]. However in these cases, there is not a real connection with the values of the performance: due to the poor accuracy at the beginning of the optimization history, a good design may be discarded and, viceversa, a poor design may be included in the non-dominated set. In the proposed strategy, the non-dominated solutions are always computed with the highest accuracy (corresponding to the lowest tolerance $\underline{\tau}$). This requires a higher computational effort also for those individuals that may temporarily appear as Pareto optimal during the optimization process, and will be later dominated. However, the increase in the cost for these individuals is a necessary price to pay to guarantee that the optimization would end up to the accurate Pareto front. In future works, this condition may be relaxed to further increase the computational efficiency.

3.4. Convergence analysis

In this section, a proof of convergence is presented to prove that with a bounding box approach an optimization algorithm able to converge to the correct solution will converge also in the presence of an error up to the desired tolerance $\hat{\tau}$, if the error estimates are conservative.

From the property of the adaptive UQ method, the estimate of the objective function of a design vector $\mathbf{a}$ satisfies the following

$$\lim_{\tau \to 0} \varepsilon_{f_j}(\mathbf{a}, \tau) = 0 \tag{13}$$

Regarding the order of convergence, the UQ method used in this work is the Simplex Stochastic Collocation (described in Sec. 4.2.1), whose convergence is linear for first-order piecewise polynomials in a one-dimensional stochastic space, and superlinear if a p-refinement is used. In the proposed strategy, a very low tolerance $\underline{\tau}$ is used for the solutions belonging to the Pareto front $\mathcal{P}$. In other words, by construction of the method, if $\mathbf{b} \in \mathcal{P}$, then $\varepsilon_{f_j}(\mathbf{b}, \tau) \to \underline{\tau}$.

Definition 5. *Let us consider the i-th iteration of the proposed optimization method, for a minimization problem of m objectives affected by an error ε_{f_j}, $j = 1, ..., m$. Considering the i-th design vector $\mathbf{x}_i$, the estimate of the error satisfies*

$$\lim_{i \to \infty} \tilde{\varepsilon}_{f_j}(\mathbf{x}_i) = \underline{\tau} \qquad \forall j = 1, ..., m \tag{14}$$

Let us suppose to have an optimization method to solve the minimization problem of function f, whose solution is f^*. The quantity $f(\mathbf{x}_i)$ is the solution at the i-th iteration, which corresponds to the statistical moment of a given quantity for the design $\mathbf{x}_i$. Note that this quantity is considered as the exact statistical moment associated to the design $\mathbf{x}_i$. Instead, the approximated function computed with a tolerance and affected by an error is denoted with $\widetilde{f}_{j,i}(\mathbf{x}_i)$.

Definition 6. *A conservative estimate $\tilde{\varepsilon}_{f_j}$ of the error ε_{f_j} that affects the estimation $\widetilde{f}_j(\mathbf{x}_i)$ of the function $f_j(\mathbf{x}_i)$ is defined as follows*

$$\left\| f_j(\mathbf{x}_i) - \widetilde{f}_j(\mathbf{x}_i) \right\| \triangleq \varepsilon_{f_j} < \tilde{\varepsilon}_{f_j} \qquad \forall j = 1, ..., m \tag{15}$$

Note that this conservative error estimation can be applied on the error of the mean, of the variance, or of a linear combination of some statistics if this is the objective function.

Theorem 2. *Let us suppose that the method converges to the solution of the optimization problem, i.e.*

$$\lim_{i \to +\infty} \left\| f_{j,i}(\mathbf{x}_i) - f_j^* \right\| = 0 \qquad \forall j = 1, ..., m \tag{16}$$

or

$$\lim_{i \to +\infty} \left\| \delta_{j,i} \right\| = 0, \qquad \forall j = 1, ..., m \tag{17}$$

Then, by considering the adaptive bounding-box approach, the convergence of the optimization method is preserved up to the tight tolerance $\underline{\tau}$ used to compute the objective functions, provided that a conservative error estimation is computed, i.e.

$$\lim_{i \to +\infty} \left\| \widetilde{f}_{j,i}(\mathbf{x}_i) - f_j^* \right\| = \underline{\tau} \tag{18}$$

Proof.

The proof stems directly from the definition of the conservative error estimation. In particular, for the j-th objective function $\left\| \widetilde{f}_{j,i} - f_j^* \right\|$ can be written as follows:

$$\begin{aligned}
\left\| \widetilde{f}_{j,i} - f_j^* \right\| &= \left\| \widetilde{f}_{j,i} - f_j^* - f_j + f_j \right\| \\
&= \left\| (\widetilde{f}_{j,i} - f_j) + (f_j - f_j^*) \right\|
\end{aligned} \tag{19}$$

Using the triangle inequality, $\|a + b\| \le \|a\| + \|b\|$, we have

$$\begin{aligned}
\left\| \widetilde{f}_{j,i} - f_j^* \right\| &= \left\| (\widetilde{f}_{j,i} - f_j) + (f_j - f_j^*) \right\| \\
&\le \left\| \widetilde{f}_{j,i} - f_j \right\| + \left\| f_j - f_j^* \right\| \\
&\le \varepsilon_{f_j} + \delta_{j,i}.
\end{aligned} \tag{20}$$

By recalling the definition of conservative error estimation, the inequality also gives that

$$\left\| \widetilde{f}_{j,i} - f_j^* \right\| < \tilde{\varepsilon}_{f_j,i} + \delta_{j,i}. \tag{21}$$

By assumption $\delta_{j,i} \to 0$ as $i \to +\infty$. Since by construction of the algorithm, the conservative estimate $\tilde{\varepsilon}_{f_j,i}$ goes to the tight tolerance $\underline{\tau}$ (recall Definition 5), taking the limit yields

$$\lim_{i \to +\infty} \tilde{\varepsilon}_{f_j,i} + \delta_{j,i} = \underline{\tau}. \tag{22}$$

The previous result shows that the optimal solutions are known up to the tight tolerance $\underline{\tau}$. If a decoupled strategy and a loose tolerance $\bar{\tau}$ are used, the difference from the computed optimal solutions and the exact solutions may be significant. Instead, if a very tight tolerance is used, the discrepancy may be negligible. In addition, in the limit of $\tau \to 0$, Eq. (22) reads

$$\lim_{i \to +\infty} \lim_{\tau \to 0} \tilde{\varepsilon}_{f_j,i} + \delta_{j,i} = 0. \tag{23}$$

This means that if the tolerance in the estimation of the objective function is equal to zero, the computed optimal solutions converge to the exact solutions.

Finally, the order of convergence of the proposed strategy strictly depends on the convergence of the UQ method (the limit $\tau \to 0$) and the order of the optimization algorithm (the limit $i \to +\infty$).

4. Methods for the optimization and UQ loops

The method proposed in this paper and presented in the preceding section is flexible with respect to the optimization algorithms and UQ methods. This section presents the specific optimization algorithms and the UQ method used in this work to assess the proposed strategy.

4.1. Derivative-free optimization algorithms

In this work, the attention is focused on problems in which the functional dependency of the objective function cannot be exploited, either because it is absent or it is unreliable. In this kind of problems, derivative-free optimization methods can be used to solve the optimization problem. These methods are very versatile, using the model of the system as a "black box". This is advantageous because it does not require an intrusive modification of the model. For instance, in the application to fluid dynamics, it is not necessary to modify the CFD solver. It also permits to easily change the objective function of the optimization. On the other hand, they may be very expensive from a computational point of view, because of their rate of convergence. Examples of such approaches are the Nelder-Mead method and its modified versions, the pattern search methods, or genetic algorithms. In this paper, the Non-dominated Sorting Genetic Algorithm and the BiMads algorithm are chosen, because they are also suited to tackle the multi-objective case. These methods are described in the following.

4.1.1. Genetic Algorithm

Evolutionary algorithms have been employed to tackle many engineering optimization problems. Applications to aerodynamic optimization problems can be found for instance in Refs. [6, 18, 19]. Main advantages of such approaches are the possibility to tackle multi-objective problems without scalarization and to exhaustively explore the design space. On the other hand, they are very expensive from a computational point of view because they require many computations of the objective function, even in region of the domain where designs with poorer performance are present.

Genetic algorithms uses operators inspired by natural evolution to get to the optimal solution. In such a framework, each design vector is considered as an individual, and the design variables in the vector represent the chromosomes. A group of individuals is called a generation. Each individual is assigned a fitness function, that is a measure of the probability that this individual will be a parent of the individuals of the next generation. The fitness in the optimization algorithms is associated to the objective function: the better the objective function, the higher the fitness value. In the single objective case, the fitness function is a measure of the objective function. In the multi-objective case, the fitness function is assigned based on the position of the individual in the objective space with respect to the current Pareto front. With the fitness value, the selection step is performed to determine the parents of the new generation; mutation and crossover operators are then used to mantain genetic diversity and to obtain different children from a single set of parents.
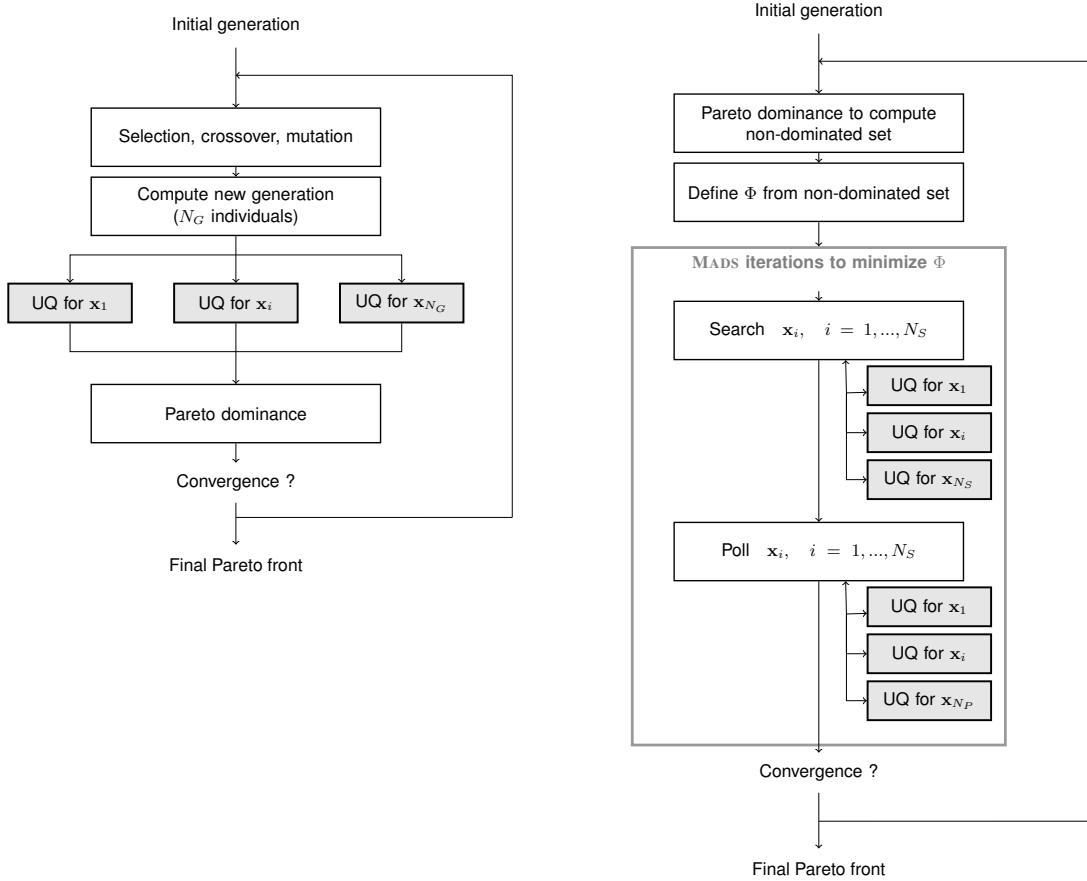
(a) Genetic algorithm. (b) BiMads algorithm.

Figure 4: Scheme of optimization algorithms used in the work.

The genetic algorithm performs a loop over the generations (see Fig. 4a). The starting point is an initial generation of individuals that are typically randomly chosen. For each iteration, a new generation of individuals is computed using selection, crossover and mutation operators. Then, the objective functions are evaluated for each individual, and each design vector is assigned a fitness function. From the fitness of these individuals, a new generation is computed and the loop goes on for a sufficient number of generations.

In this work, the Non-Dominated Sorting Genetic Algorithm is used [20]. Typical values for the crossover and mutation probabilities $pc = 0.9$, $pm = 0.1$ are chosen and the sharing parameter is set using a formula based on the number of design variables.

4.1.2. BiMads

Another derivative-free optimization method is employed, the BiMads algorithm [21]. The method applies to bi-objective optimization problems and it is based on the Mads (Mesh Adaptive Direct Search) algorithm [22]. The Mads algorithm is a direct search algorithm

15

for single-objective optimization. The underlying idea is to search an improved solution by evaluating the objective function for all the points of a mesh defined around the current design vector in the design variables space. The procedure consists on a two-step iteration based on a "search" and a "poll" step. The search step evaluates the objective function in the mesh points around the current design vector; if a better design vector is not found, a poll step is performed, which consists in a local exploration of the design variables space. If an improved solution is found, by either the search or the poll step, this one is set as the new design vector around which a new search mesh is built. Otherwise, the mesh is refined and the iteration is repeated. The optimization loop stops either when a fixed number of function evaluations is reached or when the mesh size reaches a given tolerance.

For bi-objective problems, an approximation of the Pareto front is constructed using the BiMads method, which launches successive runs of Mads. In the initialization of the method, two separate single-objective optimization problems (one for each objective) are performed by means of Mads. From the evaluations of the objective functions computed in the initialization, an initial list of non-dominated solutions is defined. The following loop is a repetition of three steps (see Fig. 4b). The first is the computation of a reference point in the objective space from the list of non-dominated solutions. This reference point is used to define a scalarization function Φ of the objective functions (for instance, the function is a measure of the distance from the reference point in the objective space). Function Φ is defined in such a way that it attains lower values in the non-dominated region of the objective space. Secondly, a Mads run is performed to minimize this scalarized function. Then, a new list of non-dominated solutions is determined from the evaluations obtained in the preceding step. Finally, the repetition of the three step is terminated according to a fixed number of iterations. A version of this algorithm exists for optimization problem with more than two objective functions; it is not employed here because only bi-objective cases are considered.

The BiMads method is very efficient for low-dimensional problems, but its rate of convergence degrades as the number of design variables increases. Similarly to genetic algorithms, the BiMads method treats the evaluation of the function for each design vector as a black-box. Therefore, it is very suited to coupling with the UQ inner loop. Concerning the implementation, the NOMAD software [23, 24] is coupled with the UQ method, presented in the next section.

4.2. Adaptive UQ methods

In this section, a discussion is presented about the method for computing the statistics, i.e. solving Eq. (5). As presented in Section 2, the objective of uncertainty propagation in a probabilistic framework is to find the probability distribution of $u(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi})$ and its statistical moments $\mu_u^\alpha(\mathbf{x}, t, \mathbf{y})$. Note that the statistical moments depend on the vector of design variables $\mathbf{x}$, and they have to be computed for a given set of design vectors. If the i-th iteration of the outer optimization loop is considered, the design variables are the design vector of the i-th design: $\mathbf{x} = \mathbf{x}_i$.

When using non-intrusive methods, the function u has to be computed for a set of samples $\boldsymbol{\xi}_k$. To some extent this set represents a numerical grid in the space of the uncertain
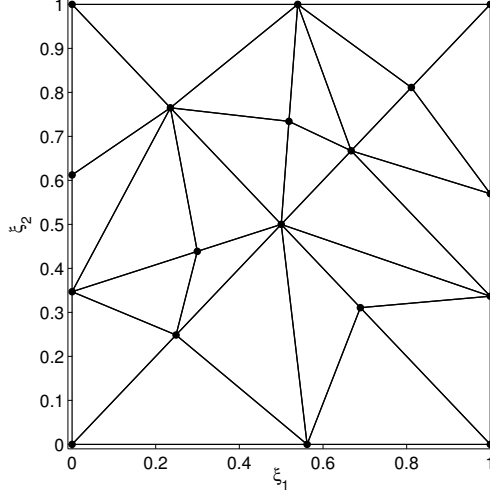
Figure 5: Simplex elements (triangles in two-dimensional stochastic space).

variables. An interesting approach is that of refining the grid with a step-by-step approach according to the quality of the current computation. The advantage of an adaptive strategy is using only as many samples as required for a sufficiently accurate computation of the statistics. In this work, the Simplex Stochastic Collocation (SSC) is chosen. SSC is a multi-element uncertainty propagation method which relies on adaptive refinement of the sampling in the stochastic space [11]. Other examples of methods in which an adaptive refinement could be used are Monte Carlo simulation methods, kriging-based or Polynomial-based approaches.

4.2.1. Simplex Stochastic Collocation

In local UQ methods such as the multi-element stochastic collocation approach [25, 26] the weighted integrals in Eq. (5) defined over the parameter space Ξ are computed as a summation of integrals over n_e disjoint subdomains $\Xi = \bigcup_{j=1}^{n_e} \Xi_j$

$$\mu_u^\alpha(\mathbf{x}, t, \mathbf{y}) \approx \sum_{j=1}^{n_e} \int_{\Xi_j} u(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi})^\alpha \, p_\xi(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} \; + \varepsilon, \tag{24}$$

where ε is the error involved in the approximation. The elements in SSC are simplexes as shown in Fig. 5 for a two-dimensional space.

In the SSC approach [11, 12], the integrals in the simplex elements Ξ_j are computed by approximating the response surface $u(\boldsymbol{\xi})$ by an interpolation $w(\boldsymbol{\xi})$ of n_s samples $\mathbf{v} = \{v_1, \ldots, v_{n_s}\}$. Each sample v_k is computed with the solution $\phi_k(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi}_k)$ obtained by solving Eq. (2) for realization $\boldsymbol{\xi}_k$ of the random parameter vector $\boldsymbol{\xi}$

$$\mathcal{L}(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi}_k; \phi_k(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi}_k)) = 0 \tag{25}$$

for $k = 1, \ldots, n_s$. The interpolation of the samples $w(\boldsymbol{\xi})$ consists of a piecewise polynomial function

$$w(\boldsymbol{\xi}) = w_j(\boldsymbol{\xi}) \qquad \text{for } \boldsymbol{\xi} \in \Xi_j, \tag{26}$$

17

with $w_j(\boldsymbol{\xi})$ a polynomial interpolation of degree p of the samples $\mathbf{v}_j = \{v_{k_{j,1}}, \ldots, v_{k_{j,n_{se}}}\}$ at the sampling points $\{\boldsymbol{\xi}_{k_{j,1}}, \ldots, \boldsymbol{\xi}_{k_{j,n_{se}}}\}$ in element Ξ_j, where $k_{j,l} \in \{1, \ldots, n_s\}$ for $j = 1, \ldots, n_e$ and $l = 1, \ldots, n_{se}$, with n_{se} the number of samples in the simplex.

The main features of the SSC method [11, 12] are: (i) the possibility to compute an error estimate, (ii) the adaptive h-refinement and (iii) p-refinement capabilities, (iv) treatment of non-hypercube parameter spaces, and (v) treatment of discontinuities in the stochastic space. Properties (i), (ii), (iii) are discussed here, because they are the key elements of the method proposed in the paper. Please refer to Refs. [11, 12] for the other features. With regard to the first property, the error ε of the interpolation is estimated using hierarchical surpluses considering in each element Ξ_j, the difference between interpolated and exact value. In this way, both local and global error in each simplex can be computed, thus identifying the simplex element, where the error is largest.

Concerning h-refinement, SSC has two components: simplex selection (based on the previous error estimator), and simplex splitting, designed to avoid clustering. In the first step, the simplex element where the error ε is largest is selected for refinement. Then, a region of this element is selected for node insertion and the new sample $\boldsymbol{\xi}_{j,k}$ is inserted (see Fig. 6a). A threshold τ is assigned to determine whether to stop the refinement: when $\varepsilon < \tau$, the adaptive refinement is stopped.
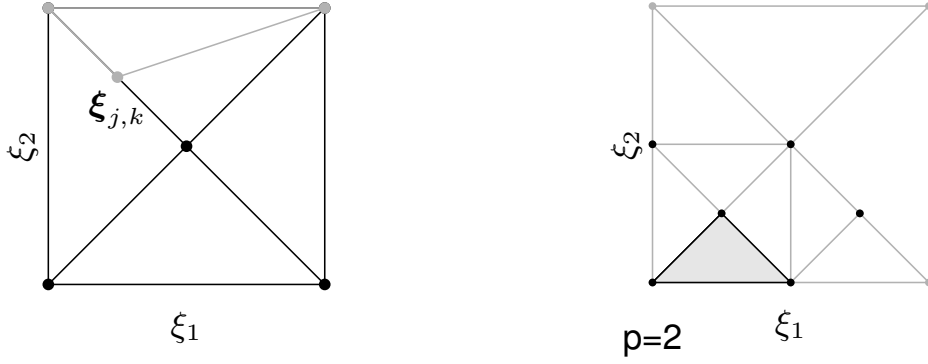
When using SSC, the error convergence of first degree SSC decreases for an increasing n_ξ. To cure this issue, a p-refinement criterion for higher degree SSC has been conceived. In practice, the idea is to use higher degree polynomial interpolation stencil based on Newton-Cotes quadrature. Figure 6b shows for a two-dimensional stochastic space how more than three samples can be used to build a higher degree polynomial interpolation by enlarging the stencil. The choice of the optimal polynomial degree p is selected based on the number of uncertainties n_ξ and the required order of convergence. A super-linear convergence can be obtained for smooth responses (see [11] for more details).

The inner loop of the SSC is constituted by the following operations:

- initial discretization (2^{n_ξ} vertexes of the hypercube enclosing the probability space Ξ and one sampling point in the interior);

- $n_{s_{\text{init}}}$ initial samples v_k computed by solving $n_{s_{\text{init}}}$ deterministic problems (Eq. (25)) for the parameter values corresponding to the initial sampling points $\boldsymbol{\xi}_k$ located in Ξ only;

- Polynomial approximation (Eq. (26)) and p-refinement;

- Error ε estimate and h-refinement according to the threshold τ.

Finally, the probability distribution function and the statistical moments μ_u^α of $u(\boldsymbol{\xi})$ given by Eq. (24) are then approximated by the probability distribution and the moments μ_w^α of $w(\boldsymbol{\xi})$

$$\mu_u^\alpha(\mathbf{x}, t, \mathbf{y}) \approx \mu_w^\alpha(\mathbf{x}, t, \mathbf{y}) = \sum_{j=1}^{n_e} \int_{\Xi_j} w_j(\mathbf{x}, t, \mathbf{y}, \boldsymbol{\xi})^\alpha \, p_\xi(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \tag{27}$$

18

(a) h-refinement.

(b) p-refinement: samples higlighted with black circles are used in the interpolation for $p = 1$ (left) and $p = 2$ (right).

Figure 6: Scheme of the h- and p-refinement of the SSC in two-dimensional stochastic space.

in which the multi-dimensional integrals are evaluated using a weighted Monte Carlo integration of the response surface approximation $w(\boldsymbol{\xi})$ with $n_{mc} \gg n_{\mathrm{s}}$ integration points.

5. Estimation of error bounds in the uncertainty quantification method

As presented in the preceding section, there is a straightforward, quantitative way to relate the cost of the uncertainty propagation procedure and the accuracy of the result: the lower the tolerance τ on the error between function w and function v, the higher the cost and the better the estimates. On the other hand, a rough, cheap estimate on the statistics of the output v can be obtained with a coarser tolerance and fewer samples. To control the accuracy of the result, a measure of the error of the estimate is required. This measure of the error is the quantity that should be compared to the desired tolerance.

Different errors are computed in the SSC method, which are quantitative estimates of the error of the statistics (i.e. mean value and variance) due to the polynomial approximation w. In the following, three groups of errors are presented: Monte Carlo errors, estimated errors and conservative estimated errors. The first set serves as a reference for the assessment of the other estimates. Note that the errors on mean and variance are considered here, but analogous estimates can be derived also for higher order statistics.

The first set is composed of the *Monte Carlo (MC) errors* , which are based only on the value of function, and are obtained by means of random Monte Carlo sampling. The mean MC error ε_μ reads

$$\varepsilon_\mu = ||\mu_{mc} - \mu_w||, \tag{28}$$

where n_{mc} are the samples of the MC integration, and the mean μ_{mc} based on the exact

19

function evaluation and the mean based on the approximation μ_w are computed as follows:

$$\mu_{mc} = \frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} v(\boldsymbol{\xi}_i)$$

$$\mu_w = \frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} w(\boldsymbol{\xi}_i). \tag{29}$$

Similarly, the variance MC error is the following

$$\varepsilon_{\sigma^2} = ||\sigma_{mc}^2 - \sigma_w^2||$$

$$\sigma_{mc}^2 = \left(\frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} u(\boldsymbol{\xi}_i) \right)^2 - \mu_{mc}^2$$

$$\sigma_w^2 = \left(\frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} w(\boldsymbol{\xi}_i) \right)^2 - \mu_w^2. \tag{30}$$

These errors are useful to compare the function w and v with the same sampling technique, by ruling out the error associated with the MC method. As a result, these errors represent the accuracy of the polynomial approximation w. However they rely on the computation of the exact function in many random samples, thus they are not employed in practical cases where it is desirable to compute the exact function only for a small number of samples.

Therefore, the SSC provides a set of *estimated errors* that are based on the hierarchical surplus and the order of the polynomial employed in each element of the grid. In this case, the estimation of the error only relies on the (small) set of samples obtained in each vertex of the SSC grid. Let us define the hierarchical surplus for the new k-th sampling point $\boldsymbol{\xi}_{j,k}$ belonging to the j-th element (see Fig. 6a), as follows

$$\varepsilon_j = w(\boldsymbol{\xi}_{j,k}) - v(\boldsymbol{\xi}_{j,k})$$

where w is the interpolation before refinement and v is the exact function evaluation. Based on the hierarchical surplus, we define the estimated mean error $\hat{\varepsilon}_\mu$ as follows

$$\hat{\varepsilon}_\mu = \sum_{j=1}^{n_e} \Omega_j \tilde{\varepsilon}_j$$

$$\tilde{\varepsilon}_j = \frac{\varepsilon_j}{2^{\frac{p_j+1}{n_\xi}}}, \tag{31}$$

where Ω_j is the probability contained in j-th simplex, p_j is the order of the polynomial approximation in the j-th simplex, and n_ξ is the number of the uncertain variables. The estimated variance error $\hat{\varepsilon}_{\sigma^2}$ then reads

$$\hat{\varepsilon}_{\sigma^2} = \left|\left| \sigma_w^2 - \left[\frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} (w(\boldsymbol{\xi}_i) - \tilde{\varepsilon}_{j,i})^2 - \left(\frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} (w(\boldsymbol{\xi}_i) - \tilde{\varepsilon}_{j,i}) \right)^2 \right] \right|\right|, \tag{32}$$

20

where $\tilde{\varepsilon}_{j,i}$ is the corrected hierarchical surplus of the element which point i belongs to. This estimate is found by assuming that one could approximate the function $v(\boldsymbol{\xi}_i)$ by $w(\boldsymbol{\xi}_i) - \varepsilon_{j,i}$; this choice is conservative since typically the vertex based on which the surplus is computed is the one with the greatest error in each simplex. This error estimate works quite well for smooth functions when the interpolation reaches its theoretical convergence rate. However, in some cases it may be useful to adopt a more conservative estimate which is only based on the hierarchical surplus.

The *conservative estimated errors* are defined with a very similar structure with respect to the estimated errors, but they employ directly the hierarchical surplus ε_j, without using extrapolation based on h- and p-refinement. The conservative estimated mean error $\tilde{\varepsilon}_\mu$ is defined as

$$\tilde{\varepsilon}_\mu = \sum_{j=1}^{n_e} \Omega_j \varepsilon_j, \tag{33}$$

whereas the conservative estimated variance error $\hat{\varepsilon}_{\sigma^2}$ is

$$\tilde{\varepsilon}_{\sigma^2} = \left\| \sigma_w^2 - \left[\frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} (w(\boldsymbol{\xi}_i) - \varepsilon_{j,i})^2 - \left(\frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} (w(\boldsymbol{\xi}_i) - \varepsilon_{j,i}) \right)^2 \right] \right\|, \tag{34}$$

where $\varepsilon_{j,i}$ is the hierarchical surplus of the element which point i belongs to. To assess these estimates a comparison is presented for some analytical cases in the following subsection.

### 5.1. Analytic testcases for error estimation

To begin with, a one-dimensional sine function is considered:

$$f(\xi) = 3 + \sin(2\pi\xi_1) \tag{35}$$

where the uncertain variable ξ_1 is uniformly distributed in the domain $[0, 1]$. In Fig. 7a the behavior of the errors and estimators is presented, which confirms the effectiveness of the estimates. In fact, the estimated error and the conservative estimated errors are larger than the MC error, proving that they are conservative estimates of the actual error. In other words, the exact value of the statistics lies in the error bar defined around the SSC estimate and whose width can be described either by the estimated or the conservative error. In the case of the sine function, the error on the variance provides more information on the accuracy of the response approximation rather than the mean value, which is equal to the constant term in Eq. (35) and it is therefore affected by a small error even in the first iteration.

Then, the method is applied to a more complex function, the one-dimensional Michalewicz function, that is a very challenging testcase in global optimization problems
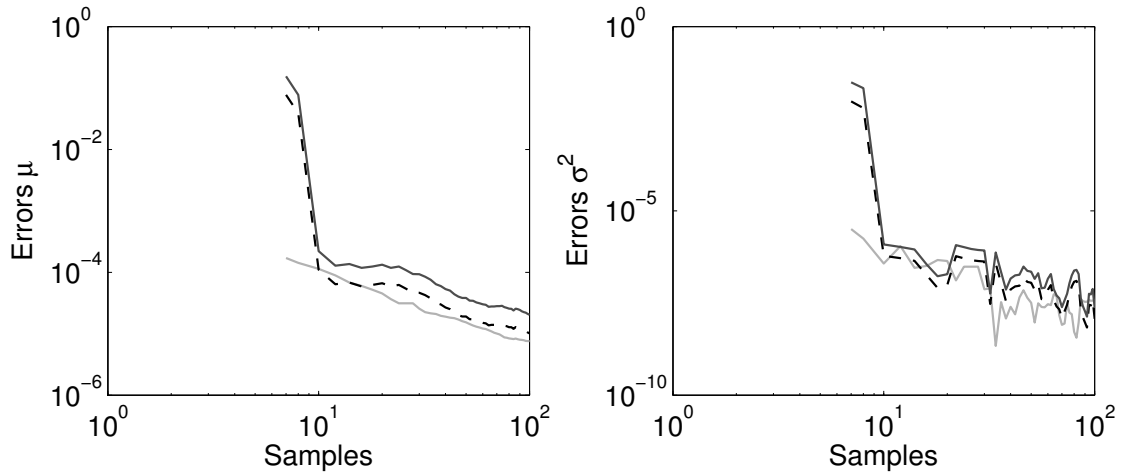
$$f(\xi) = -\sin(\xi_1) \ \sin^{20}\left(\frac{\xi_1^2}{\pi} \right). \tag{36}$$
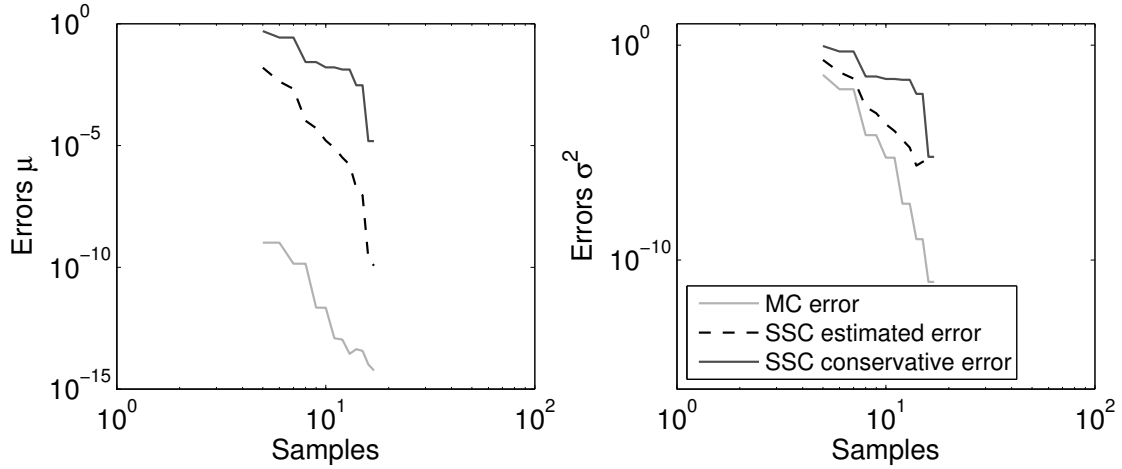
(a) One-dimensional sine function.
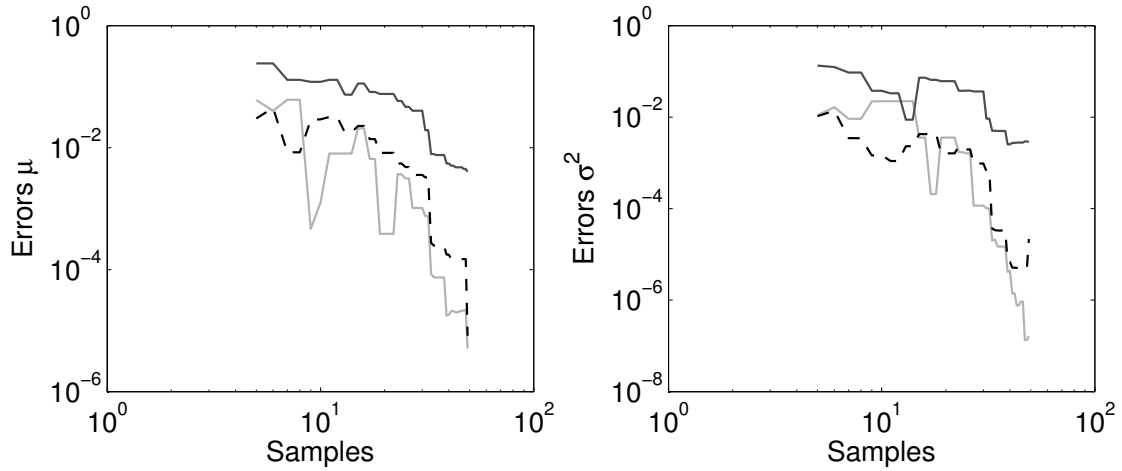
(b) One-dimensional Michalewicz function.

(c) Two-dimensional inverse tangent function.

Figure 7: Monte Carlo, estimated and conservative errors for three analytic functions.

The uncertain variable ξ_1 is again uniformly distributed in the domain $[0,1]$. Figure 7b shows the result of the application of SSC and proves that in this case the conservative error is a safer estimate of the error.

Finally, a two-dimensional function is considered to explore multi-dimensional problems. The inverse tangent presented in Ref. [11] is analysed, which reads

$$f(\xi) = \text{atan} \left(\boldsymbol{\xi} \cdot \boldsymbol{\xi}^* + \xi_1^{*2} \right), \tag{37}$$

with $\boldsymbol{\xi}^* \in [0,1]^2$ is a two-dimensional vector of arbitrary values. In this case, the errors of the mean value have a smoother behavior with respect to the errors of the variance (see Fig. 7c). However, even in the latter case the convergence trend is clear and it appears that, despite the oscillations, the estimated error and the conservative error are consistent with the MC error.

As a final remark, these estimates are associated only to the error due to the polynomial approximation w. Although the method developed in this work considers a generic error on the estimate of the objective functions, regardless of the sources of that error, it is worth noting that these estimates are obtained by means of MC sampling, which is affected by an error itself. However, this error would decrease anyway when increasing the number of MC sampling points. Adding a bias to the estimates would be a possibile strategy to include this error as well, without affecting the proposed optimization method.

6. Uncertainty-based optimization of algebraic test cases

This section presents some results obtained with the proposed strategy for two algebraic test cases, where there is a simple algebraic relationship between the performance u and the design and uncertain variables. In the comparison with the standard non-adaptive method, the proposed strategy is referred to as *coupled* strategy, and the classical approach is called *decoupled*.

6.1. Single-objective algebraic problem

In this section a single-objective ($m = 1$) optimization is considered, with two uncertain variables ($n_\xi = 2$) and two design variables ($n = 2$). The optimization problem reads

$$
\begin{aligned}
\text{minimize:} \quad & f(\mathbf{x}) = \mu(u) + \sigma(u) \\
\text{by changing:} \quad & \mathbf{x} \in \mathcal{X} = [0,6] \times [0,6] \\
\text{with:} \quad & \xi \sim \mathcal{U}(0,1)
\end{aligned}
\tag{38}
$$

where the function representing the performance is the following

$$u(\mathbf{x}, \xi) = \left[e^{-\sum_{i=1}^n (x_i/\kappa)^{2\gamma}} - 2e^{-\sum_{i=1}^n \xi_i (x_i - \pi)^2} \right] \cdot \prod_{i=1}^n \cos^2 x_i \quad \kappa = 15, \ \gamma = 5. \tag{39}$$

This function was presented in Ref. [27] and it is a modification to the classical test case for deterministic optimization. In this case, the uncertain variables are present in the exponential
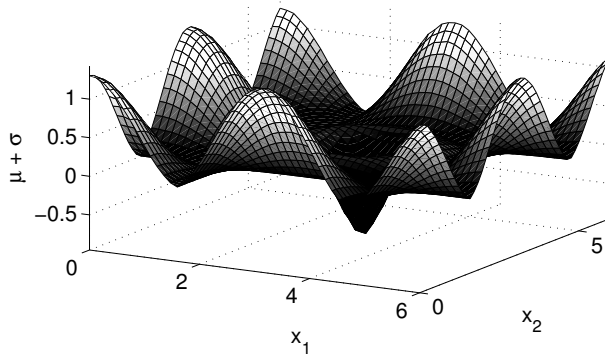
Figure 8: Objective of the optimization based on Yang function.

Table 1: Comparison of results and cost between decoupled and coupled approaches for problem in Eq. (38)

Strategy	Percentual error on $f(\mathbf{x}^*)$	Distance from $\mathbf{x}^*$	N. function evaluations
Decoupled (1e-3)	1.0e-6	6e-4	24956
Decoupled (1e-1)	1.2e-5	2e-3	16820
Coupled	1.7e-7	4e-4	18875

terms and they are as many as the design variables. The deterministic function has one global minimum $u(\mathbf{x}^*) = -1$ in $\mathbf{x}^* = \{-\pi, ..., -\pi\}$. When uncertain variables are considered the function changes its shape, but the global minimum remains the same for any realization of the uncertain variables. The mean value of u evaluated in the minimum is equal to -1 and the variance is equal to 0. Thus the solution of the robust case in Eq. (38) is equal to the deterministic global minimum $f(\mathbf{x}^*) = -1$ in $\mathbf{x}^* = \{-\pi, ..., -\pi\}$. The solution is close to the mid-point of the design variables domain, as represented in Fig. 8.

The solution of the optimization problem in Eq. (38) is obtained by means of the genetic algorithm described in Section. 4.1.1 and the SSC method (Section. 4.2.1). The coupling is obtained in three different ways: (i) with a decoupled approach in which the tolerance is set to 1e-3, (ii) a decoupled approach in which the tolerance is set to 1e-1, (ii) the coupled strategy. In the latter case, the upper value $\bar{\tau}$ of the tolerance is set to 1e-1 and the lower value $\underline{\tau}$ is 1e-3. The three approaches reach the minimum value of the function in the correct minimum point, in spite of a slightly higher error for the decoupled approach with the loose tolerance (see Tab. 1). The computational cost of the decoupled strategy with tight tolerance is reduced by 24% with the coupled approach. The cost of the coupled method is comparable with that of the decoupled approach with loose tolerance, and the estimate of the minimum is very accurate. However, in this case a very good estimate of the optimal design is obtained even with a loose decoupled strategy. In the next section, a test case is presented which shows the real benefit of the coupled approach.
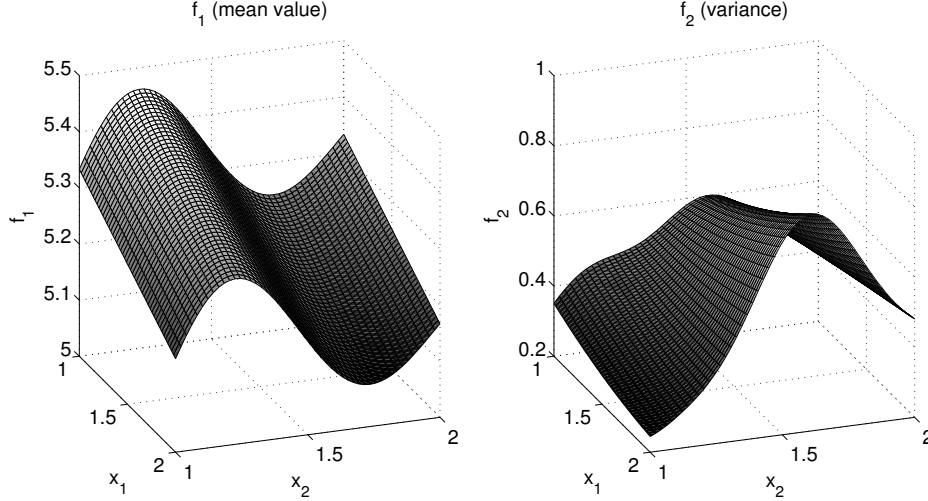
Figure 9: Objective function for problem (40).

6.2. Multi-objective algebraic problem

In this section a bi-objective ($m = 2$) optimization is considered, with one uncertain variable ($n_\xi = 1$) and two design variables ($n = 2$). The optimization problem reads

$$\begin{aligned}
\text{minimize:} \quad & \mathbf{f}(\mathbf{x}) = \{f_1, f_2\} = \{\mu(u), \sigma^2(u)\} \\
\text{by changing:} \quad & \mathbf{x} \in \mathcal{X} = [1, 2] \times [1, 2] \\
\text{with:} \quad & \xi \sim \mathcal{U}(0, 1)
\end{aligned} \tag{40}$$

where the function representing the performance is the following

$$u(\mathbf{x}, \xi) = \xi - x_1 \xi^5 + \cos(2\pi x_2 \xi) + \kappa, \qquad \kappa = 5. \tag{41}$$

The mean value and variance of function u are represented in the design space in Fig. 9. For this test case, the tight tolerance $\underline{\tau}$ is set to 1e-4 and the loose tolerance $\bar{\tau}$ is equal to 1e-1. The set of results presented in the first subsection are obtained by coupling the SSC method with the genetic algorithm. In the second subsection the BIMADS method is used for the optimization loop in order to highlight the flexibility of the proposed method.

6.2.1. Coupled method with NSGA

Figure 10a presents the Pareto fronts of the analytic function and the fronts obtained by means of the decoupled method with tight tolerance, the decoupled method with loose tolerance and the coupled strategy. The sets of solution obtained with the tight tolerance and the coupled approach compare well, while the loose tolerance results in a different branch of the front. Because of this, the front obtained with loose tolerance has been re-computed with the analytic function and plotted in the objective space for comparson. The distance to the analytic Pareto front is presented in Fig. 10b and it is computed by taking the mean of the distance of each point of the Pareto front to the closest point of the analytical Pareto
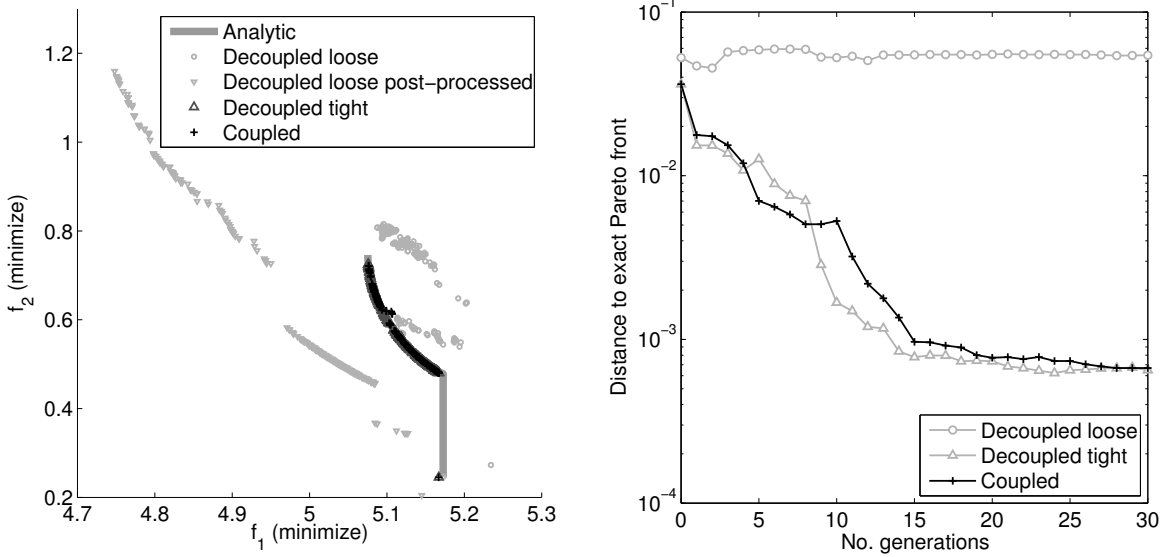
(a) Pareto optimal fronts: the loose tolerance result has been computed after the optimization with the analytical function for comparison.

(b) Distance to the Pareto front of the analytical function.

Figure 10: Results of multi-objective algebraic problem (Eq. (40)) with: (i) decoupled strategy with tight tolerance (1e-4), (ii) decoupled strategy with loose tolerance (1e-1) and (iii) coupled strategy.

front, as suggested in Ref. [28]. In mathematical terms, the distance d_{A-B} of the Pareto set $\mathcal{P}_A$ to the Pareto set $\mathcal{P}_B$ reads

$$d_{A-B} = \frac{1}{|\mathcal{P}_A|} \sum_{i=0}^{|\mathcal{P}_A|} \min_{\mathbf{f} \in \mathcal{P}_B} ||\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_{\mathcal{P}_B})||, \qquad (42)$$

where symbol $|\cdot|$ indicates the cardinality (i.e. number of elements) of the set, symbol $||\cdot||$ is the Euclidean distance, $\mathbf{f}$ and $\mathbf{x}$ are respectively the objective functions and the design variables of the solutions in the Pareto set. The comparison of the difference computed at each generation proves that the cheap optimization obtained with the loose tolerance results in an inaccurate front, while the coupled approach yields a front consistent with the accurate result. In addition, the approximation $w(\xi)$ obtained with the SSC method is presented in Fig. 11 for a representative design of the inaccurate Pareto front. In particular, the design $\tilde{\mathbf{x}}$ is selected, which is trade-off design vector in the Pareto front obtained with the loose decoupled approach. The SSC with the tight tolerance provides the best fit of the analytical function, whereas a good approximation is obtained with the coupled method. Instead, the number of samples used with the loose tolerance is not sufficient to correctly capture the behavior of the function. Thus, for this test case a tight tolerance is crucial to obtain the correct solutions of the optimization problem.
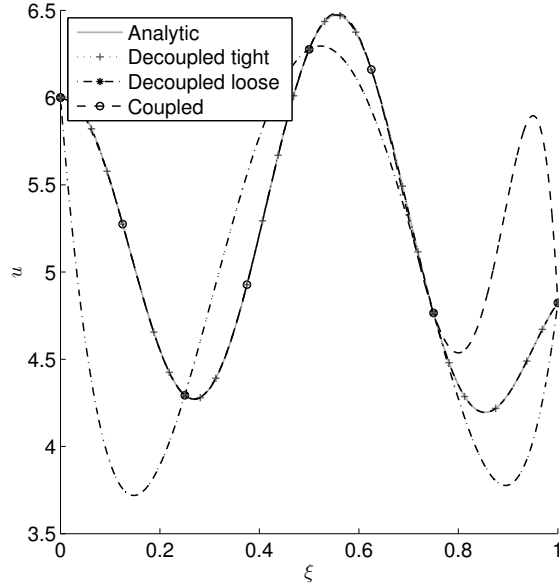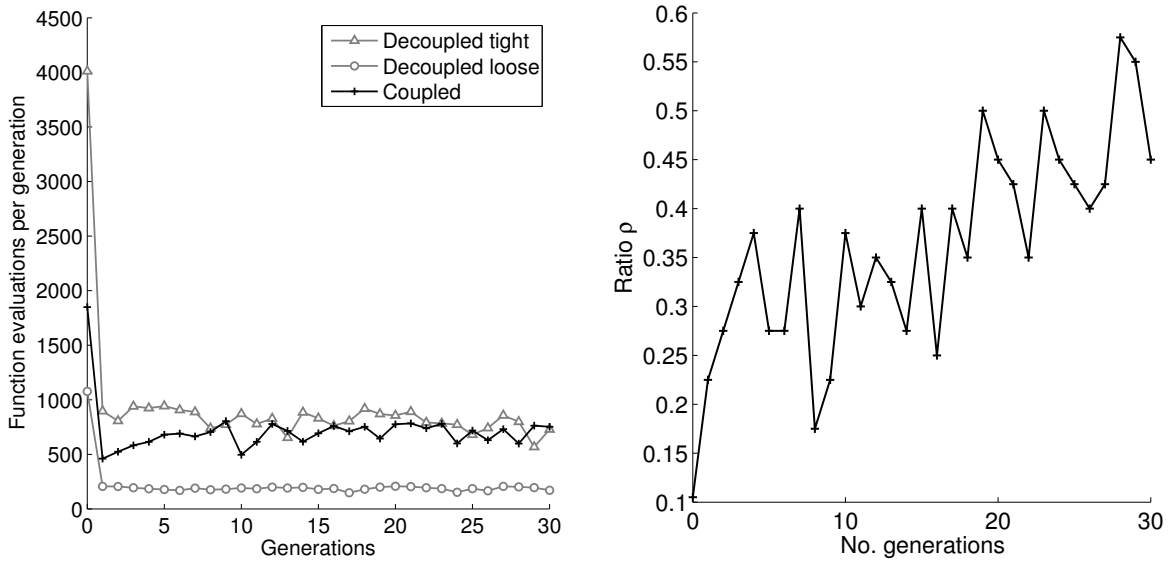
26

Figure 11: Approximations of the function u for the design vector $\tilde{\mathbf{x}}$ in the Pareto set obtained with the decoupled loose ($\bar{\tau} = 1\mathrm{e}\text{-}1$) method.

(a) Number of function evaluations.

(b) Ratio of potentially non-dominated solutions to generation size.

Figure 12: Cost of the multi-objective algebraic optimization (Eq. (40)) with NSGA.

With regard to the computational cost of the coupled method, the number of function evaluations for each generation is shown in Fig. 12a and reported in Tab. 2. In the very first generation, the coupled approach has a significant reduction of the cost with respect to

Table 2: Comparison of total number of function evaluations in the NSGA for the multi-objective algebraic optimization.

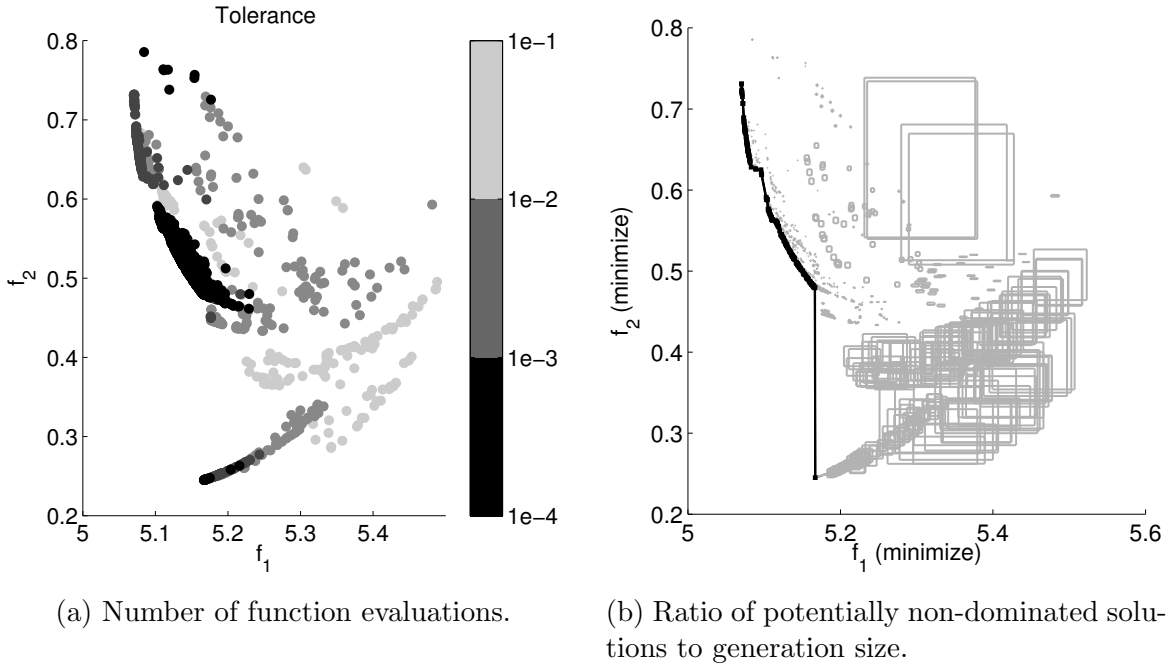| Strategy	Number of function evaluations		
	Initial generation	Evolution	Global
Decoupled (1e-1)	1077	5609	6686
Decoupled (1e-4)	4011	24482	28493
Coupled	1849	20368	22217

(a) Number of function evaluations.

(b) Ratio of potentially non-dominated solutions to generation size.

Figure 13: Tolerance and bounding boxes of the coupled approach for the multi-objective algebraic test case (Eq. (40)) with NSGA.

the tight decoupled approach: the reduction is close to 54%, thus the number of evaluations has been reduced to less than half the reference value. With increasing number of generations, the cost of the coupled approach increases and it almost converges to the cost of the tight decoupled method. In fact, with increasing number of generations during an evolution, each generation typically becomes more and more rich of individuals close to the current Pareto front, in other words design vectors with good objectives. For these "good" individuals the coupled approach has to compute the objective functions with a tighter tolerance, and the cost becomes higher. This is also shown in Fig. 12b, where the ratio of potentially non-dominated solution over the generation size is presented. A solution is defined as potentially non-dominated if the objective vector is non-dominated with respect to the individuals computed up to that point. In mathematical terms, given the number of individuals in a

generation N_G the ratio ρ is expressed as follows

$$\rho = \frac{\sum_{i=1}^{N_G} c_{k,i}}{N_G} \tag{43}$$

where the coefficient $c_{k,i}$ indicates whether the design vector $\mathbf{x}_{i,l}$ occurring at the l-th iteration in the current generation and i-th iteration in the global population is non-dominated regarding the population at the i-th iteration $\mathcal{X}_i$

$$c_{i,l} = \begin{cases} 1 & \text{if } \mathbf{x}_{i,l} \text{ is non-dominated regarding } \mathcal{X}_i \\ 0 & \text{otherwise.} \end{cases}$$

It appears that when the ratio exceeds 0.3, the cost of the coupled approach becomes similar to the cost of the tight decoupled approach. Nevertheless, the cost remains lower and the global cost is reduced by 22 % (see Tab. 2). Please note that the ratio ρ is a lower bound of the cost of the decoupled strategy, because it is computed in a post-processing stage with the non-dominated solutions based on the final estimate. Therefore, it does not include the refinement required for dominated solutions close to the Pareto front.

Figures 13a presents the tolerance τ used for each design vector in the objective space. During the optimization, the algorithm uses the tighter tolerance $\underline{\tau} = 1e\text{-}4$ only in the region close to the actual Pareto front. The loose tolerance $\bar{\tau} = 1e\text{-}1$ is used for higher values of the mean value f_1. As a matter of fact, the resulting error bounding boxes are wider in that area, as presented in Fig. 13b.

Discussion on error estimates. During the post-processing stage of the optimization, the error with increasing number of samples has been computed for a group of design vectors. For instance, in Fig. 14 the MC errors, estimated and conservative errors are plotted for the
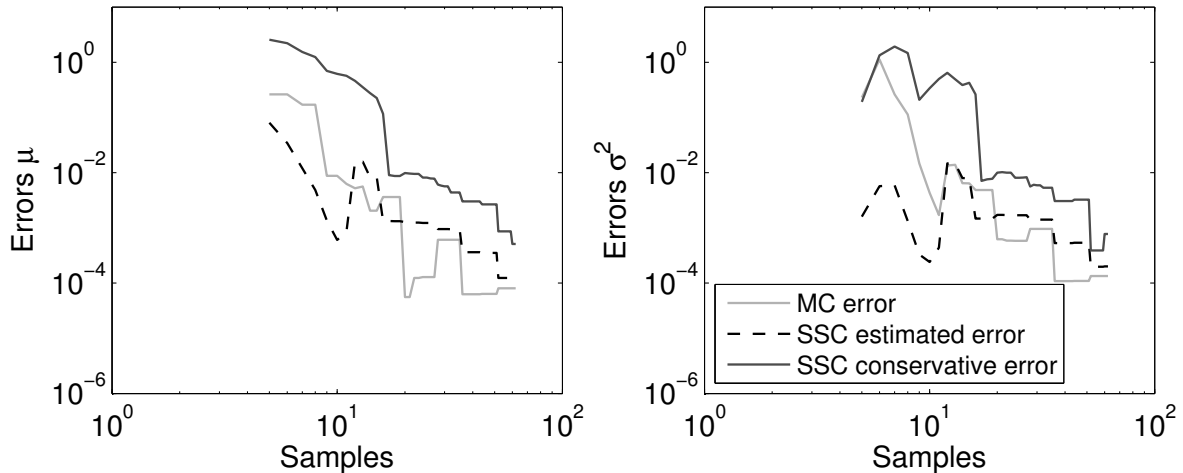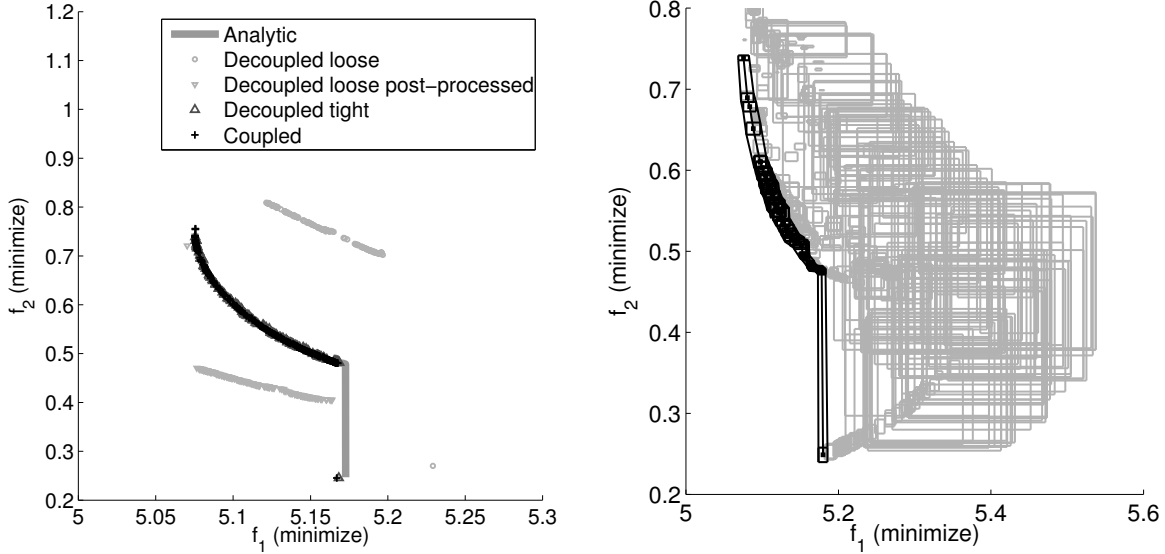

Figure 14: Errors affecting the mean value and the variance of function u for a representative design vector of the decoupled loose Pareto front.

(a) Pareto optimal fronts: the loose tolerance result has been computed after the optimization with the analytical function for comparison.

(b) Error bounding boxes in the coupled method.

Figure 15: Results of multi-objective algebraic problem (Eq. (40)) with conservative estimates.

design $\tilde{\mathbf{x}}$, a trade-off design belonging to the Pareto front obtained with the decoupled loose method. The figure shows that in the first iterations of the SSC method the estimated errors is not conservative with respect to the MC error (which represents the reference value, as explained in Sec. 5).

In light of this result, the robust optimization has been performed again considering the conservative estimated errors affecting the objective functions. The resulting Pareto fronts are presented in Fig. 15a, which shows a situation similar to the one presented in the preceding section. The coupled approach results in a front overlapping the result of the decoupled method with tight tolerance. Once again, the loose tolerance does not provide accurate results. The error bounding boxes computed in the coupled approach are larger in this case, due to the higher order of magnitude of the conservative estimated errors. The computational cost of the coupled approach in this case is higher and it approaches the result of the tight tolerance (see Fig. 16a). This is due to two main reasons. First, the ratio of potentially non-dominated solutions over generation size ρ is greater in this case as presented in Fig. 16b. Second, since the conservative estimates overestimate the actual errors, the bounding box is greater than the box that could be defined with the actual errors (see the boxes in Fig. 15b). As a result, it is more likely for the bounding box to dominate the Pareto front, thereby requiring a higher number of samples even for the design vectors that are far from the Pareto front.
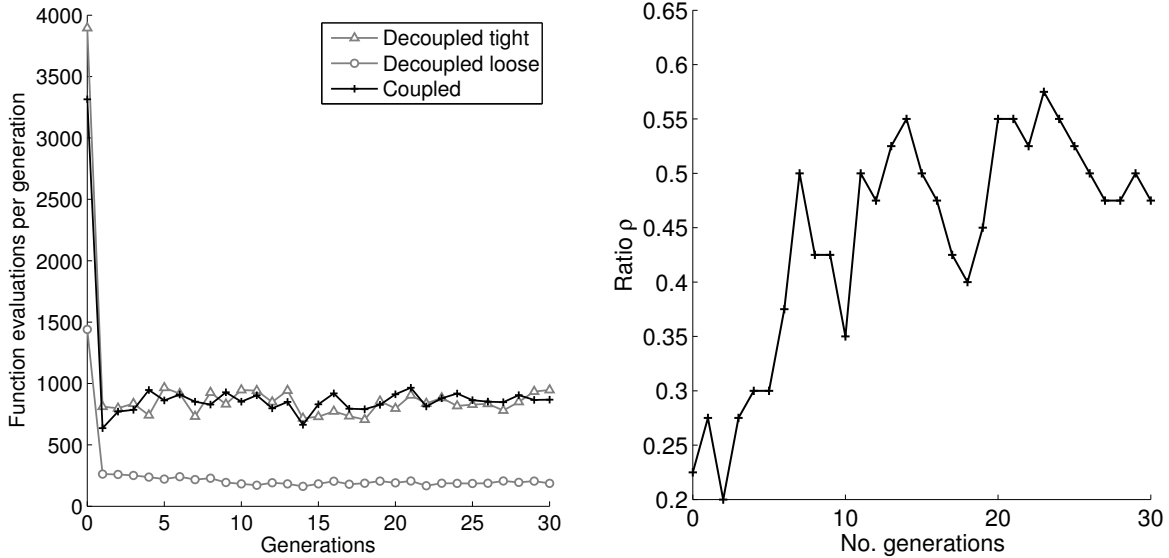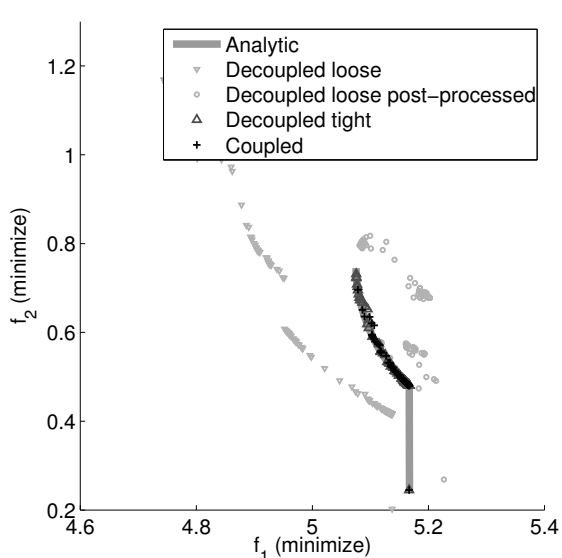
(a) Number of function evaluations.

(b) Ratio of potentially non-dominated solutions to generation size.

Figure 16: Cost of the multi-objective algebraic optimization (Eq. (40)) with NSGA and conservative estimates.

The results obtained with the estimated and conservative estimated errors highlight two important aspects. First, even if the error estimates in the first iterations of the SSC are not strictly conservative, the proposed strategy is able to attain the accurate Pareto front as presented in the preceding section. In fact, for the non-dominated solutions the method always uses the lowest tolerance $\underline{\tau}$, overcoming the problems that may arise with a non-conservative error estimates in the first iterations. Thus, the method is robust and it is able to assess whether a solution that may appear non-dominated due to a poor accuracy of the objective function is actually dominated. Nevertheless, conservative estimates are important in the opposite situation, that is to avoid discarding potentially non-dominated solutions. Secondly, not only it is crucial to have an error estimate that is conservative, but also the estimate should be consistent with the actual error. A too conservative estimate may jeopardize the efficiency of the adaptive strategy.

6.2.2. Coupled method with BIMADS

The multi-objective robust optimization problem presented in Eq. (40) is tackled also with the BIMADS algorithm coupled to the SSC method. The results are presented in Fig. 17a, where the front obtained with the decoupled and coupled approaches are compared to the result of the analytic function. Similarly to the case presented in the preceding subsection, the distance computed with respect to the analytic result in Fig. 17b proves that the decoupled method with tight tolerance and the coupled method reaches the analytical result. Furthermore, the convergence is reached in almost five iterations of the algorithm.
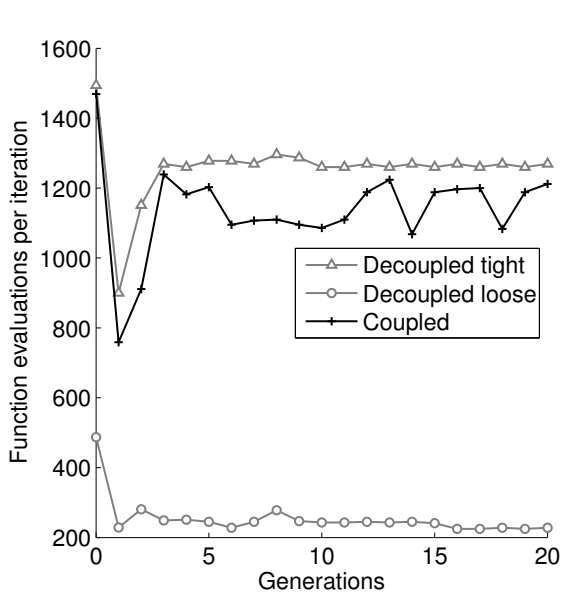
(a) Pareto optimal fronts: the loose tolerance result has been computed after the optimization with the analytical function for comparison.
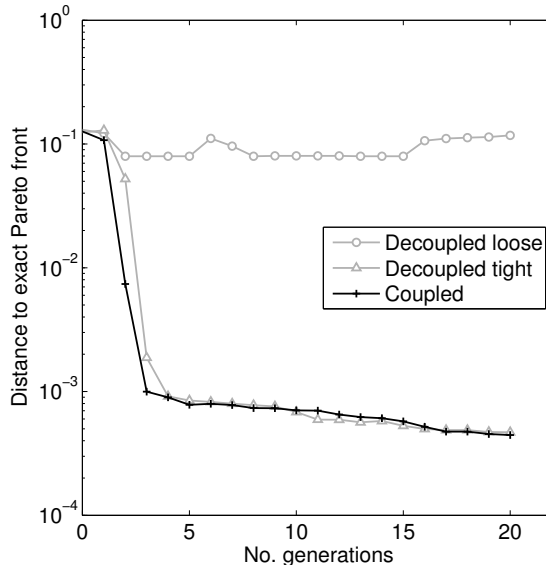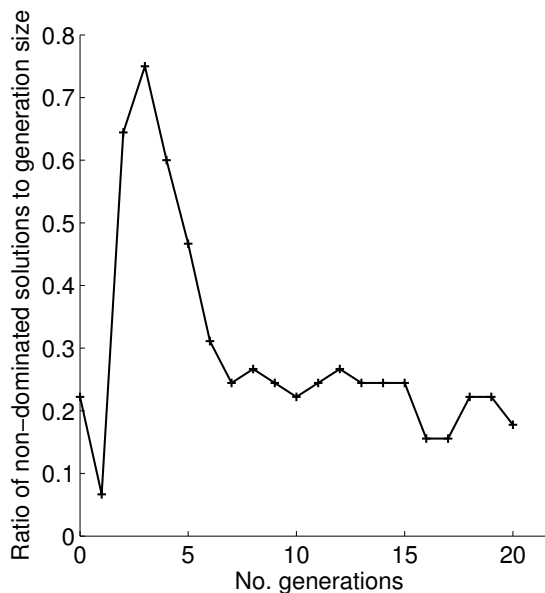
(b) Distance to Pareto front of the analytical function.

Figure 17: Results of the multi-objective algebraic test case (Eq. (40)) with BiMADS.

(a) Number of function evaluations.

(b) Ratio of potentially non-dominated solutions to generation size.

Figure 18: Cost of the multi-objective optimization with BiMADS.

Table 3: Comparison of total number of function evaluations in the BɪMADS optimization loop.

Strategy	Decoupled (1e-1)	Decoupled (1e-4)	Coupled
Number of function evaluations	5331	26387	23894

The cost of the decoupled approach with tight tolerance is reduced by 30% with the coupled approach in the first iteration (see Fig. 18a). In the following iterations the coupled approach becomes more expensive and the cost gets closer to the cost of the tight decoupled method, although it remains lower. The reduction of the global cost is about 10% in this case. In fact, in the first iterations many design vectors close to the Pareto front are computed: as reported in Fig. 18b over 70% of the design vectors are potentially non-dominated solutions. In additon, during the remaining iterations, the design vectors tested are very close to the Pareto front, so a tighter tolerance is required in those cases. On average, the coupled approach in the last iterations uses 25 evaluations per design vector, whereas the decoupled tight approach uses 28 function evaluations.

7. Uncertainty-based optimization of an engineering testcase

In this section an engineering test case is presented. The optimization problem is stated in Sec. 7.1 and the results of the coupled strategy are discussed in Sec. 7.2.

7.1. Robust optimization problem

The robust optimization method proposed in the paper is applied to the optimal selection of the airfoil for a helicopter rotor blade in hover. Instead of considering a deterministic, representative operating condition, it is assumed that the freestream condition of the airfoil section is affected by some level of uncertainty. In this paper we consider sources of uncertainty in the physical modelling employed at the design stage that could impact the local pitch angle θ and the rotor induced velocity V_i (see Fig. 19) at which the airfoil of the blade operates. A complete description of the problem is presented in Ref. [29]. The uncertainty on the pitch angle and induced velocity affects the angle of attack, Mach number and Reynolds number of the flow encountered by the airfoil.

Figure 19: Scheme of operating conditions at a blade section: the uncertain variables are the pitch angle θ and the induced velocity V_i. The velocity component U_T is the velocity tangential to the rotor disk.

The objective of the aerodynamic optimization is to maximize the lift-to-drag ratio E, which is a measure of the aerodynamic efficiency of the airfoil. Also, a robust optimal airfoil shape is sought. Robustness means that the design parameters $\mathbf{x}$ defining the shape of the airfoil allows simultaneous improvement in the overall performance and minimization of the sensitivity with respect to changes in the system operative conditions. In other words, the optimal design must maximize the mean value μ_E and minimize the variance σ_E^2 of the objective function, computed by taking into account the uncertainties of the operating conditions. The objective function E is a function of the output of the aerodynamic system, that is the solution of the differential equation modelling the aerodynamics:

$$\mathcal{L}\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}; \phi(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi})\right) = 0, \tag{44}$$

where $\mathcal{L}$ is a nonlinear spatial differential operator describing the problem. The problem is well posed by considering a constraint on the geometry of the airfoil to avoid more than one inflection point on the camber line, and an aerodynamic constraint on the pitching moment. The latter constraint is applied when the moment coefficient about the quarter-chord location exceeds 0.03 to prevent excessive torque on the blade structure. In mathematical terms, the resulting optimization problem can be stated as:

$$
\begin{aligned}
\text{maximize:} \quad & f_1 = \mu\left(E\left(\phi\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}\right)\right)\right) \\
\text{and minimize:} \quad & f_2 = \sigma^2\left(E\left(\phi\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}\right)\right)\right) \\
\text{subject to:} \quad & \mathcal{L}\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}; \phi(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi})\right) = 0 \\
& \mathbf{g}\left(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}; \phi(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi})\right) \leq \mathbf{0} \\
\text{by changing:} \quad & \mathbf{x}
\end{aligned}
\tag{45}
$$

with uncertain input parameters $\boldsymbol{\xi} = \{\theta, V_i\}$ and under the constraints previously described and collected in vector $\mathbf{g}$.

The uncertain variables are described with a probabilistic approach as uniformly distributed variables. In particular, they are assigned a nominal value and an uncertainty band around it. The nominal conditions considered in the test case are summarised in Tab. 4, and they refer to the 50% of the radius of the operating conditions of the Bo105 rotor [30]. The uncertainty band is set to 15% for the induced velocity and ± 2.5 deg for the pitch angle [29].

The design variables are represented by the coefficients of a parameterization used to discretize the shape of the airfoil in a set of finite design variables. In particular, the Class/Shape function Transformation [31] is employed, which is based on a truncated series of Bernstein polynomials. The number of design variables are set to 17 after a convergence analysis [29].

The aerodynamic model is based on the coupling of a potential model for the outer region and a boundary layer method close to the airfoil. The numerical solver is XFOIL [32], an aerodynamic code which couples panel and integral boundary layer methods developed for the analysis of subsonic, isolated airfoils. This model is well suited for the simulation of flows around an airfoil at moderate values of the angle of attack and Mach number, and it permits a rapid evaluation of the aerodynamic coefficients.

Table 4: Nominal operating conditions of the robust aerodynamic optimization.

$\theta[\deg]$	$V_i[\mathrm{m/s}]$	$\alpha[\deg]$	M	Re
12.1	11.7	5.9	0.32	1.6e6

(a) Pareto front of decoupled tight approach. (b) Pareto front of coupled method.
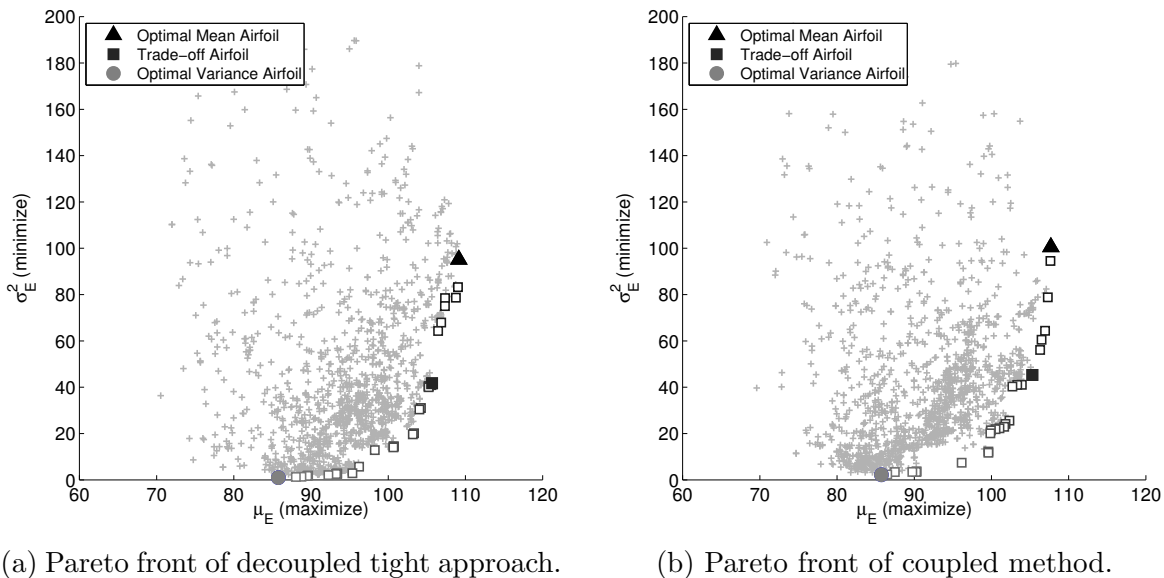
Figure 20: Pareto fronts of decoupled and coupled methods.

7.2. Results

The robust aerodynamic optimization has been performed by a decoupled approach with tight tolerance ($\underline{\tau} = $ 5e-1) and a coupled approach with $\underline{\tau} = $ 5e-1 and $\bar{\tau} = 5$. The tolerance values are increased with respect to the algebraic cases due to the higher order of magnitude of the objective functions. In addition, the decoupled approach with loose tolerance is not considered here, because it is deemed that in real-world applications due to the substantial cost of the robust optimization procedure it may not be possible to assess the effect of the tolerance. If a single run has to be performed, it is best to choose a tight tolerance that guarantees good results. For the optimization loop the NSGA has been used, because a high number of design variables are employed.

The Pareto fronts obtained with the decoupled and coupled methods are presented in Fig. 20a and Fig. 20b, respectively. In the figures, the solution with higher mean value, higher variance and a trade-off solution are highlighted. From the figures it is clear that the coupled method compares well to the decoupled approach. To give a rigorous, quantitative measure of this comparison, the Pareto front obtained with the coupled approach has been post-processed with the decoupled method. The resulting front is presented in Fig. 21 with the result of the decoupled method. The coupled front almost overlaps the decoupled results; the distance between the fronts computed according to Eq. (42) is equal to 2.25. The solutions belonging to the front improves the mean value of the efficiency of the reference
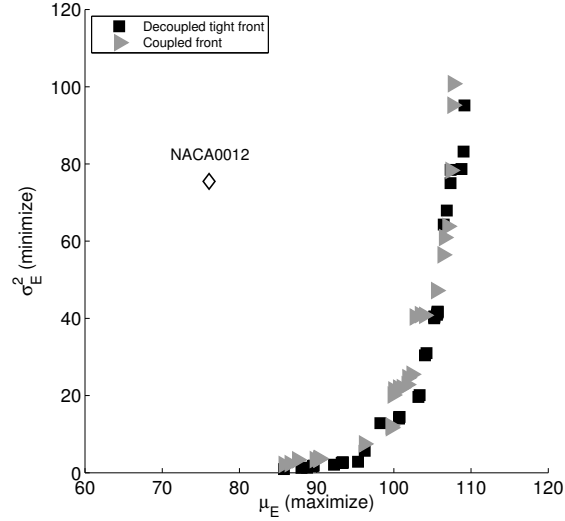
35

Figure 21: Comparison of Pareto fronts (front obtained with coupled approach has been post-processed with tight tolerance for comparison).

(a) Optimal mean airfoils.

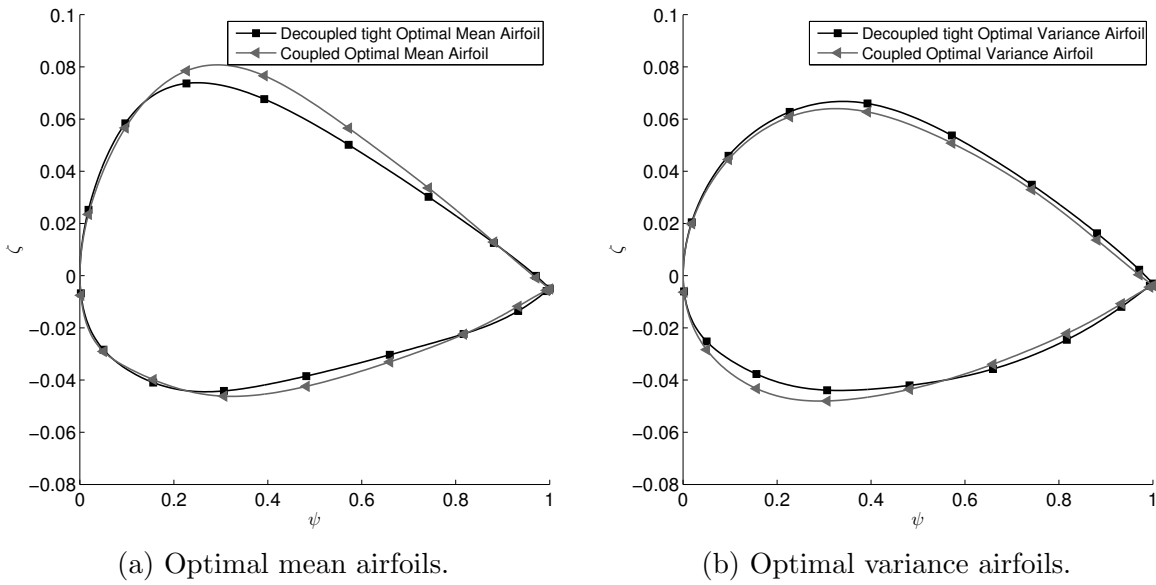(b) Optimal variance airfoils.

Figure 22: Optimal airfoils of decoupled and coupled methods.

airfoil, i.e. the NACA 0012, and a subset has also reduced variance. To compare the optimal shapes, the airfoils with higher mean value and higher variance are presented in Fig. 22a and Fig. 22b. Despite minor differences, the airfoils of the fronts obtained with the two methods share similar trends.

With regard to computational cost, the coupled method greatly reduces the cost of the decoupled approach (see Tab. 5): in the first generation the cost is reduced by 58% and the gain on the global cost is equal to 40%. The great reduction is due to the fact that the

(a) Number of function evaluations.

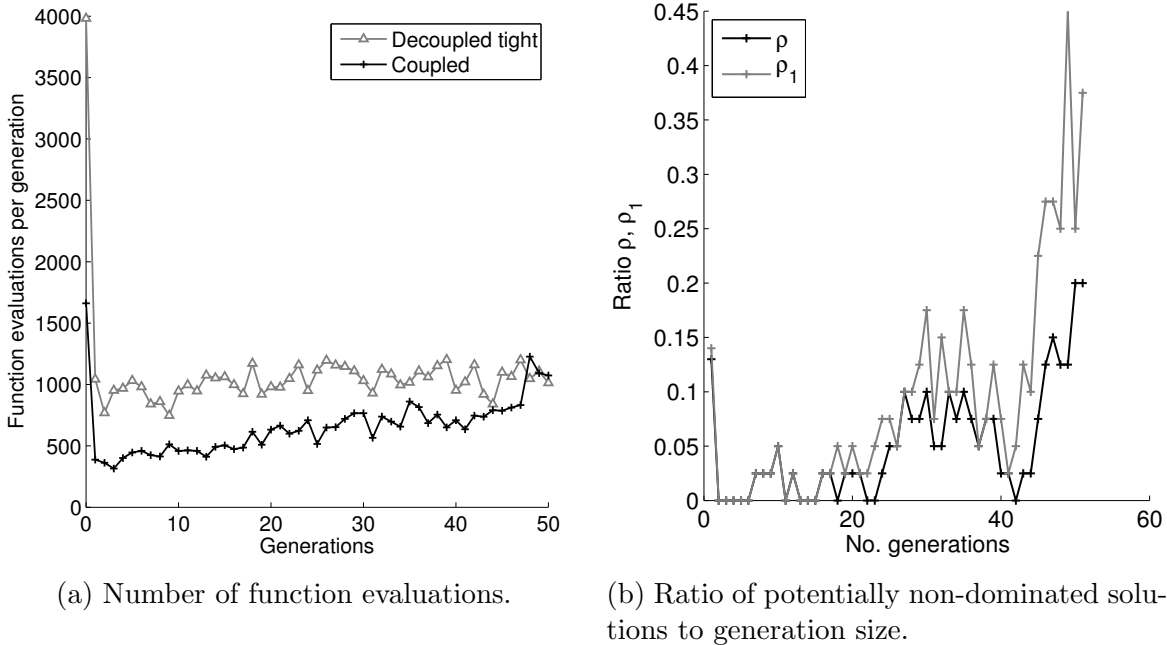(b) Ratio of potentially non-dominated solutions to generation size.

Figure 23: Cost of the decoupled and coupled methods for the robust aerodynamic optimization.

Table 5: Comparison of total number of function evaluations for the robust aerodynamic optimization.

Strategy	Number of function evaluations		
	Initial generation	Evolution	Global
Decoupled tight	3982	51258	55240
Coupled	1661	31760	33421

optimization problem is high-dimensional and nonlinear. Thus, the optimization algorithm samples many design vectors in the design space with poor objectives, especially in the first iterations. In fact, the ratio ρ is lower than 0.2 (see Fig. 23b) with a growing trend after 20 generations. This means that the generation becomes rich of non-dominated individuals only after a few generations of the algorithm. Please recall that the ratio ρ is a lower bound of the actual cost, since it does not take into account the refinement needed for dominated solutions close to the Pareto front (as explained in Sec. 6.2.1). To highlight this aspect, Figure 23b presents the modified ratio ρ_1, computed taken into account not only the non-dominated solutions, but also the solutions within a distance to the current Pareto front of $\underline{\tau}$ in each direction of the objective space. The second ratio has a trend closer to the one of the actual cost of the coupled strategy. To conclude, the coupled approach gives an accurate estimate of the Pareto front with a greatly reduced computational effort.

Conclusion

In this paper, a novel strategy is presented to merge the optimization algorithm and the UQ method in an uncertainty-based optimization framework. The strategy relies on an adaptive refinement in the stochastic space only for the design with good performance. A bounding box based on the estimates of the error affecting the objective function is defined around the objectives vector. A comparison between the error bounding box and the current Pareto front determines whether a further refinement is necessary. The method only requires minor modifications to the UQ method, and it applies to derivative-free optimization.

The novel strategy has been applied to single- and multi-objective optimization problems, using two optimization algorithms, i.e. a genetic algorithm (NSGA) and an adaptive direct search method (BiMADS). The results proves that the method reduces the computational cost with respect to the classical decoupled approach, especially in the first iterations of the optimization algorithm. The accuracy of the actual Pareto front is preserved by the novel strategy. The algebraic test cases have also shown that the great benefit of the method holds for those problems where changing the tolerance on the statistics affects the optimization result. The method has proved robust to non-conservative estimates of the error, although a conservative estimate is always recommended.

The novel approach has been applied to an engineering test case, i.e. the robust aerodynamic optimization of an airfoil with application to helicopter rotor blades. In this case, the reduction of the computational cost is even higher owing to the higher dimensionality and nonlinearity of the optimization problem, which requires several iterations to converge to the optimal solution. Finally, the strategy proposed in the paper is very flexible and it could be easily extended to different optimization algorithms and adaptive UQ methods.

Future works will be oriented to formulate more intrusive approaches for taking into account the variability associated to the bounding boxes when applying the optimization operators.

Bibliography

[1] C. J. Roy, W. L. Oberkampf, A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing, Computer Methods in Applied Mechanics and Engineering 200 (25-28) (2011) 2131–2144. doi:10.1016/j.cma.2011.03.016.
URL http://linkinghub.elsevier.com/retrieve/pii/S0045782511001290

[2] H.-G. Beyer, B. Sendhoff, Robust optimization A comprehensive survey, Computer Methods in Applied Mechanics and Engineering 196 (33-34) (2007) 3190–3218. doi:10.1016/j.cma.2007.03.003.
URL http://linkinghub.elsevier.com/retrieve/pii/S0045782507001259

[3] G. Schuëller, H. Jensen, Computational methods in optimization considering uncertainties An overview, Computer Methods in Applied Mechanics and Engineering 198 (1)

(2008) 2–13. doi:10.1016/j.cma.2008.05.004.
URL http://linkinghub.elsevier.com/retrieve/pii/S0045782508002028

[4] G. Taguchi, M. Phadke, Quality engineering through design optimization, in: K. Dehnad (Ed.), Quality Control, Robust Design, and the Taguchi Method, Springer US, 1989, pp. 77–96. doi:10.1007/978-1-4684-1472-1.

[5] D. Lucor, C. Enaux, H. Jourden, P. Sagaut, Stochastic design optimization: Application to reacting flows, Computer Methods in Applied Mechanics and Engineering 196 (2007) 5047–5062.

[6] P. Congedo, C. Corre, J.-M. Martinez, Shape optimization of an airfoil in a BZT flow with multiple-source uncertainties, Computer Methods in Applied Mechanics and Engineering 200 (1-4) (2011) 216–232. doi:10.1016/j.cma.2010.08.006.
URL http://linkinghub.elsevier.com/retrieve/pii/S0045782510002392

[7] L. Painton, U. Diwekar, Stochastic annealing for synthesis under uncertainty, European Journal of Operational Research 83 (3) (1995) 489 – 502. doi:http://dx.doi.org/10.1016/0377-2217(94)00245-8.

[8] U. M. Diwekar, W. Xu, Improved genetic algorithms for deterministic optimization and optimization under uncertainty. part i. algorithms development, Industrial & Engineering Chemistry Research 44 (18) (2005) 7132–7137. doi:10.1021/ie049127d.

[9] S. Sankaran, C. Audet, A. L. Marsden, A method for stochastic constrained optimization using derivative-free surrogate pattern search and collocation, Journal of Computational Physics 229 (12) (2010) 4664 – 4682. doi:http://dx.doi.org/10.1016/j.jcp.2010.03.005.
URL http://www.sciencedirect.com/science/article/pii/S0021999110001154

[10] P. Congedo, J. Witteveen, G. Iaccarino, A simplex-based numerical framework for simple and efficient robust design optimization, Computational Optimization and Applications 56 (1) (2013) 231–251. doi:10.1007/s10589-013-9569-0.

[11] J. A. S. Witteveen, G. Iaccarino, Refinement criteria for simplex stochastic collocation with local extremum diminishing robustness, SIAM Journal on Scientific Computing 34 (3) (2012) A1522–A1543. doi:10.1137/100817498.

[12] J. A. S. Witteveen, G. Iaccarino, Simplex stochastic collocation with random sampling and extrapolation for nonhypercube probability spaces, SIAM Journal on Scientific Computing 34 (2) (2012) A814–A838. doi:10.1137/100817504.

[13] M. Mlakar, T. Tušar, B. Filipič, Comparing solutions under uncertainty in multiobjective optimization, Mathematical Problems in Engineering 2014 (2014) 1–10. doi:10.1155/2014/817964.

[14] J. Teich, Pareto-front exploration with uncertain objectives, in: E. Zitzler, L. Thiele, K. Deb, C. Coello Coello, D. Corne (Eds.), Evolutionary Multi-Criterion Optimization, Vol. 1993 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 314–328. doi:10.1007/3-540-44719-9_22.

[15] C. Haubelt, J. Teich, Accelerating design space exploration using pareto-front arithmetics, in: Proceedings of the 2003 Asia and South Pacific Design Automation Conference, ASP-DAC '03, ACM, New York, NY, USA, 2003, pp. 525–531. doi:10.1145/1119772.1119882.

[16] P. Limbourg, Multi-objective optimization of problems with epistemic uncertainty, in: C. Coello Coello, A. Hernndez Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization, Vol. 3410 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 413–427. doi:10.1007/978-3-540-31880-4_29.

[17] M. Hendriks, M. Geilen, T. Basten, Pareto analysis with uncertainty, in: Proceedings of the 2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing, EUC '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 189–196. doi:10.1109/EUC.2011.54.

[18] D. S. Lee, J. Periaux, E. Onate, L. F. Gonzalez, N. Qin, Active Transonic Aerofoil Design Optimization Using Robust Multiobjective Evolutionary Algorithms, Journal of Aircraft 48 (3) (2011) 1084–1094.

[19] R. A. E. Mäkinen, J. Periaux, J. Toivanen, Shape design optimization in 2d aerodynamics using genetic algorithms on parallel computers, in: Parallel Computational Fluid Dynamics, Elsevier Science, 1995, pp. 395–402.

[20] N. Srinivas, K. Deb, Multiobjective function optimization using nondominated sorting genetic algorithms, Evolutionary Computation 2 (3) (1994) 221–248. doi:10.1.1.26.8615.

[21] C. Audet, G. Savard, W. Zghal, Multiobjective optimization through a series of single–objective formulations, SIAM Journal on optimization 19 (1) (2008) 188–210.

[22] C. Audet, J. Dennis, Jr., Mesh adaptive direct search algorithms for constrained optimization, SIAM Journal on Optimization 17 (1) (2006) 188–217. doi:doi:10.1137/040603371.

[23] S. Le Digabel, Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm, ACM Transactions on Mathematical Software 37 (4) (2011) 1–15.

[24] M. Abramson, C. Audet, G. Couture, J. Dennis, Jr., S. Le Digabel, C. Tribes, The NOMAD project, Software available at `https://www.gerad.ca/nomad/`.

[25] J. Foo, G. E. Karniadakis, Multi-element probabilistic collocation method in high dimensions, Journal of Computational Physics 229 (5) (2010) 1536 – 1557.

doi:http://dx.doi.org/10.1016/j.jcp.2009.10.043.
URL http://www.sciencedirect.com/science/article/pii/S0021999109006044

[26] X. Ma, N. Zabaras, An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations, Journal of Computational Physics 229 (10) (2010) 3884 – 3915. doi:http://dx.doi.org/10.1016/j.jcp.2010.01.033.
URL http://www.sciencedirect.com/science/article/pii/S0021999110000562

[27] X. S. Yang, S. Deb, Engineering optimization by cuckoo search, International Journal of Mathematical Modelling and Numerical Optimization 4 (2010) 330–343.

[28] H. Shen, E. McBean, Y. Wang, Sensor placement under nodal demand uncertainty for water distribution systems, in: R. M. Clark, S. Hakim (Eds.), Securing Water and Wastewater Systems, Vol. 2 of Protecting Critical Infrastructure, Springer International Publishing, 2014, pp. 123–133. doi:10.1007/978-3-319-01092-2-5.

[29] F. Fusi, A. Guardone, G. Quaranta, P. M. Congedo, Multi-fidelity physics-based method for robust optimization with application to a hovering rotor airfoil, AIAA Journal (accepted).

[30] B. G. van der Wall, 2nd HHC Aeroacoustic Rotor Test (HART II) - Part I: Test Documentation., Braunschweig, Germany, 2003.

[31] B. M. Kulfan, J. E. Bussoletti, "Fundamental" Parametric Geometry Representations for Aircraft Component Shapes, in: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA, 2006. doi:10.2514/6.2006-6948.

[32] M. Drela, Xfoil: An analysis and design system for low reynolds number airfoils., Conference on Low Reynolds Number Airfoil Aerodynamics, University of Notre Dame 54 (1989) 1–12.