ETHzürich

MULTI-OBJECTIVE ROBUST OPTIMIZATION USING ADAPTIVE SURROGATE MODELS FOR PROBLEMS WITH MIXED CONTINUOUS-CATEGORICAL PARAMETERS

M. Moustapha, A. Galimshina, G. Habert and B. Sudret

Data Sheet

Multi-objective robust optimization using adaptive surrogate models for problems with mixed continuous-categorical parameters

M. Moustapha[1], A. Galimshina[2], G. Habert[2], and B. Sudret[1]

[1] *Chair of Risk, Safety and Uncertainty Quantification,*

ETH Zurich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland

[1] *Chair of Sustainable Construction,*

ETH Zurich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland

Abstract

Explicitly accounting for uncertainties is paramount to the safety of engineering structures. Optimization which is often carried out at the early stage of the structural design offers an ideal framework for this task. When the uncertainties are mainly affecting the objective function, robust design optimization is traditionally considered. This work further assumes the existence of multiple and competing objective functions that need to be dealt with simultaneously. The optimization problem is formulated by considering quantiles of the objective functions which allows for the combination of both optimality and robustness in a single metric. By introducing the concept of common random numbers, the resulting nested optimization problem may be solved using a general-purpose solver, herein the non-dominated sorting genetic algorithm (NSGA-II). The computational cost of such an approach is however a serious hurdle to its application in real-world problems. We therefore propose a surrogate-assisted approach using Kriging as an inexpensive approximation of the associated computational model. The proposed approach consists of sequentially carrying out NSGA-II while using an adaptively built Kriging model to estimate the quantiles. Finally, the methodology is adapted to account for mixed categorical-continuous parameters as the applications involve the selection of qualitative design parameters as well. The methodology is first applied to two analytical examples showing its efficiency. The third application relates to the selection of optimal renovation scenarios of a building considering both its life cycle cost and environmental impact. It shows that when it comes to renovation, the heating system replacement should be the priority.

1   Introduction

Engineering systems are nowadays systematically designed for optimal efficiency in a resource-scarce environment. The optimization process is most often carried out using computational models predicting the behavior of the system for any given configuration. While these models are becoming more and more accurate thanks to algorithmic advances and the availability of low-cost computing power, the parameters required to describe the system are not always accurately known. This is due to the inherent variability of the system parameters or simply to the designer's lack of knowledge. Regardless of their nature, *uncertainties* need to be accounted for in the design process as they affect the prediction made by the computational model and henceforth the design solution. Within the probabilistic framework, various techniques have been developed for optimization under uncertainties (Schuëller and Jensen, 2008; Beck et al., 2015; Lelievre et al., 2016). The optimization problem of interest in this paper comprises two components often treated separately in the literature, namely *robust design optimization* and *multi-objective optimization*. The first one, which directly deals with uncertainties, entails finding an optimal solution which is at the same time not sensitive to perturbations in its vicinity. This technique has been widely studied in the literature and state-of-the-art reviews can be found in Zang et al. (2005); Beyer and Sendhoff (2007), among others. Following the pioneering sigma-to-noise ratio laid by Taguchi and Phadke (1989), the most widely used metric of robustness combines the performance first- and second-order moments (Doltsinis and Kang, 2004; Beck et al., 2015). This naturally leads to a multi-objective optimization problem. More recently, *quantiles* of the performance function have been considered (Pujol et al., 2009; Razaaly et al., 2020; Sabater et al., 2021). Owing to its simplicity and straightforward account of uncertainties through a unique scalar, this approach is considered in the present work.

The second component, *i.e.*, multi-objective optimization, relates to the presence of multiple performance functions. Very often the objectives described by these functions are conflicting and a compromise needs to be found. The associated methods may be classified into *a priori* and *a posteriori* approaches, according to when the decision maker sets preferences among the different objectives. In the former, the designer reformulates the multi-objective problem into one or a series of single objectives by aggregation (Miettinen, 1999; De Weck, 2014; Ehrgott, 2005; Liang and Mahadevan, 2017). In contrast, in *a posteriori* methods no preference is set before the optimization is carried out. Trade-off solutions are found first and only then are the decision makers given possible design solutions to choose from (Marler and Arora, 2014).

Evolutionary algorithms are another class of methods that have emerged and have shown

to be powerful and particularly suited for multi-objective optimization, as they possess several desirable characteristics (Zitzler et al., 2004; Emmerich and Deutz, 2018). These are basically metaheuristics that evolve a set of candidate designs by various mechanisms so as to converge to an approximation of the so-called *Pareto front*, *i.e.*, a series of optimal design solutions describing a trade-off between competing objectives. Examples of such methods include the vector evaluated genetic algorithm (Schaffer, 1985), the multi-objective genetic algorithm (Fonseca and Flemming, 1993), the strength Pareto evolutionary algorithm 2 (SPEA2) (Zitzler and Thiele, 1999) the Pareto archived evolution strategy (Knowles and Corne, 2000) or the non-dominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2002). The latter is considered in this work as it is efficient and easy to implement.

A commonly known drawback of evolutionary algorithms, and in particular of NSGA-II, is their high computational cost. This is even more problematic when the objective function stems from the propagation of uncertainties through a possibly expensive-to-evaluate computational model. Surrogate modelling is a proven solution to address these cost issues and has been widely used in both uncertainty quantification and design optimization fields. A surrogate model is an inexpensive approximation of a model that aims at replacing it in analyses requiring its repeated evaluation. One of the most popular methods, namely Kriging a.k.a. Gaussian process (GP) modelling (Sacks et al., 1989; Santner et al., 2003; Rasmussen and Williams, 2006), is considered in this work. Kriging has been extensively used for both multi-objective and robust optimization problems, as the brief literature review in Section 4.2 shows.

Finally, the developments in this paper were motivated by an application related to the optimal selection of renovation scenarios for building renovation under uncertainties (Galimshina et al., 2020, 2021). The formulated optimization problem is not only multi-objective and robust but also contains a subset of parameters which are categorical. *Categorical variables* are characterized by the fact that they are composed of a discrete and finite set of values which do not have any intrinsic ordering. There are also often referred to as *qualitative* or *nominal*. In the case of building renovation, a typical example is the heating system, which may be selected from technical solutions such as oil, gas, heat pump, etc. To address this aspect, we consider slight adaptations of both NSGA-II and Kriging which allow us to use general implementation of both methods without resorting to new developments.

Based on these premises, the goal of this paper is to develop a general purpose multi-objective algorithm for robust optimization problems using an adaptive Kriging model and capable of handling mixed categorical-continuous variables. To achieve this, we propose formulating the multi-objective and robust optimization problem using quantiles of the performance functions, which are computed by Monte Carlo simulation. The resulting optimization problem is solved using NSGA-II and the computational cost is reduced by the use of an adaptively built Gaussian process model. Focus is put on the exploration/exploitation balance of this adaptive scheme, which allows us to solve complex problems with a relatively

small computational cost.

The paper is organized as follows. In Section 2, the quantile-based formulation of the optimization problem is presented considering separately the robust and multi-objective components. Section 3 presents a nested level solution scheme which couples optimization and quantile estimation. Section 4 introduces Kriging to the framework following a short literature review. Section 5 presents the adaptations of the proposed methodology to handle mixed categorical-continuous variables. Finally, Section 6 presents two applications: an analytical example and a real case study which deals with the optimal renovation of a building under uncertainties using life cycle assessment.

2 Problem formulation

In this paper, we are interested in solving multi-objective *and* robust optimization problems. Even though robust design often involves minimizing two conflicting objectives (*e.g.*, a measure of performance and a measure of dispersion), these two classes of problems are most often treated separately in the literature. As briefly mentioned in the introduction, the state-of-the-art in robust optimization comprises various formulations. In this work, we are interested in minimizing conservative quantiles of the objective function given the uncertainties in the input. Mathematically, the following problem is solved:

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d} \in \mathbb{D}} Q_\alpha \left(\mathfrak{c}; \boldsymbol{X}\left(\boldsymbol{d}\right), \boldsymbol{Z}\right) \text{subject to:} \quad f_j\left(\boldsymbol{d}\right) \le 0 \quad \{j = 1, \ldots, c\}, \tag{1}$$

where Q_α, which represents the α-quantile of the cost function $\mathfrak{c}$ and reads

$$Q_\alpha \left(\mathfrak{c}; \boldsymbol{X}\left(\boldsymbol{d}\right), \boldsymbol{Z}\right) \equiv \inf \left\{q \in \mathbb{R} : \text{Prob}\left[\mathfrak{c}\left(\boldsymbol{X}\left(\boldsymbol{d}\right), \boldsymbol{Z}\right) \le q\right] \ge \alpha\right\}. \tag{2}$$

This quantile is minimized with respect to the design parameters $\boldsymbol{d} \in \mathbb{D} \subset R^{M_d}$, where $\mathbb{D}$ denotes the design space which generally defines lower and upper bounds of the design variables. A set of c constraint functions $\{f_j, j = 1, \ldots, c\}$ are additionally considered and are assumed to be simple and easy-to-evaluate analytical functions further restricting the design space. No other hard constraints are considered and to simplify the following developments, we will denote the subset of the design space that is feasible by $\mathbb{S} = \{\boldsymbol{d} \in \mathbb{D} : f_j\left(\boldsymbol{d}\right) \le 0, j = 1, \ldots, c\}$. In contrast, the cost function is assumed to result from the evaluation of a complex, and possibly expensive-to-evaluate, computational model $\mathcal{M}$. This model takes as inputs the uncertain random parameters of the analysis, which are here split into two groups: the random variables $\boldsymbol{X} \sim f_{\boldsymbol{X}|\boldsymbol{d}}$ represent the variability associated to the design parameters $\boldsymbol{d}$ whereas $\boldsymbol{Z} \sim f_{\boldsymbol{Z}}$ are the so-called *environmental variables* which affect the system response without necessarily being controlled by the designer. Manufacturing tolerances (design) and loading (environmental) are typical examples of the two categories of uncertain parameters in the context of structural engineering design.

As in the applications considered in this work, engineering systems sometimes require the simultaneous optimization of multiple quantities of interest. Considering a set of m

cost functions denoted by $\{\mathcal{M}_k, \, k = 1, \, \dots, m\}$, the optimization problem of interest now becomes

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d} \in \mathbb{S}} \left\{ Q_{\alpha_1}\left(\mathcal{M}_1; \boldsymbol{X}\left(\boldsymbol{d}\right), \boldsymbol{Z}\right), \, \dots, Q_{\alpha_m}\left(\mathcal{M}_m; \boldsymbol{X}\left(\boldsymbol{d}\right), \boldsymbol{Z}\right) \right\}, \tag{3}$$

where $\{\alpha_1, \, \dots, \alpha_m\} \in [0, \, 1]^m$ are the levels of the quantiles as introduced in Eq. (2). It is assumed here without loss of generality that the m objective functions are derived from the same computational model $\mathcal{M}$.

3 Problem solution

3.1 Nested levels solution scheme

The direct solution of the optimization problem stated in Eq. (3) classically involves two nested levels. In the outer level, the design space is explored whereas in the inner one the quantiles corresponding to a given design choice are computed. Even though variance-reduction simulation techniques can be used for the computation of quantiles (Glynn, 1996; Dong and Nakayama, 2017), we consider here crude Monte Carlo simulation (MCS). Due to the fact that relatively large values of α are considered for robust design optimization, MCS may indeed provide accurate estimates of the quantiles with relatively few sample points. More specifically, we consider $\alpha_1 = \alpha_2 = ... = \alpha_m = \alpha = 0.90$ throughout this paper.

The first step in the inner level is to draw a Monte Carlo sample set for a given design choice $\boldsymbol{d}^{(i)}$:

$$\mathcal{C}\left(\boldsymbol{d}^{(i)}\right) = \left\{ \left(\boldsymbol{x}^{(j)}, \boldsymbol{z}^{(j)}\right), j = 1, \, \dots, N \right\}, \tag{4}$$

where $\boldsymbol{x}^{(j)}$ and $\boldsymbol{z}^{(j)}$ are realizations of $X \sim f_{\boldsymbol{X}|\boldsymbol{d}^{(i)}}$ and $Z \sim f_{\boldsymbol{Z}}$, respectively. The computational model is then evaluated on each of these points and henceforth the corresponding cost function values are obtained. After ordering the latter, the quantiles corresponding to each output are eventually empirically estimated and denoted by $q_k\left(\boldsymbol{d}^{(i)}\right)$.

By their very nature, quantiles are random variables and plugging Monte Carlo estimates in Eq. (1) or (3) would lead to a stochastic optimization problem. Solving such a problem may be cumbersome, and even more so in the presence of multiple objectives. To avoid dealing with the resulting issues, the concept of *common random numbers* is considered (Spall, 2003). This essentially translates into using the same stream of random numbers within iterations of the optimization problem. More specifically, the random variables $\left\{ \boldsymbol{x}\left(\boldsymbol{d}^{(1)}\right), \boldsymbol{x}\left(\boldsymbol{d}^{(2)}\right), \boldsymbol{x}\left(\boldsymbol{d}^{(3)}\right), ... \right\}$ generated at each iteration of the optimization procedure share the same seed. For the environmental variables, the same set of realizations of $\left\{ \boldsymbol{z}^{(j)}, j = 1, \, \dots, N \right\}$ is used throughout the optimization. Using this trick, the mapping $\boldsymbol{d}^{(i)} \mapsto q_k\left(\boldsymbol{d}^{(i)}\right)$ becomes deterministic. As such, Eq. (3) can be simplified into

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d} \in \mathbb{S}} \left\{ q_1\left(\boldsymbol{d}\right), \, \dots, q_m\left(\boldsymbol{d}\right) \right\}. \tag{5}$$

This problem can then be solved using any classical multi-objective optimization solver. Such problems are however ill-defined in a sense, as there is generally no single solution that minimizes all the objective functions simultaneously. Therefore, the traditional approach is to seek for a set of solutions representing a compromise. To define such solutions, a widely-used concept is that of *Pareto dominance* (Miettinen, 1999; Ehrgott, 2005). Given two solutions $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{D}$, $\boldsymbol{a}$ dominates $\boldsymbol{b}$ (denoted by $\boldsymbol{a} \prec \boldsymbol{b}$) if and only if the following two conditions hold:

$$\begin{cases} \forall k \in \{1, \, \ldots \, , m\}, & q_k\left(\boldsymbol{a}\right) \leq q_k\left(\boldsymbol{b}\right), \\ \exists k \in \{1, \, \ldots \, , m\}, & q_k\left(\boldsymbol{a}\right) < q_k\left(\boldsymbol{b}\right). \end{cases}$$

The goal of the search is then to find solutions which are not dominated by any other feasible point. A set of such solutions is said to be *Pareto optimal* and defined as follows:

$$\mathcal{D}^* = \left\{\boldsymbol{d} \in \mathbb{S} : \nexists \, \boldsymbol{d}' \in \mathbb{S}, \, \boldsymbol{d}' \prec \boldsymbol{d}\right\}. \tag{6}$$

The *Pareto front* $\mathcal{F} = \{\boldsymbol{q}\left(\boldsymbol{d}\right) \in \mathbb{R}^m : \boldsymbol{d} \in \mathcal{D}^*\}$ is the image of $\mathcal{D}^*$ in the objective space.

Typically, a discrete approximation of the Pareto front is sought. As mentioned in the introduction, we will consider in this paper an elitist evolutionary algorithm, namely the *non-dominated sorting genetic algorithm II* (NSGA-II) (Deb et al., 2002). This algorithm is one of the most popular in the field of multi-objective optimization and is briefly described in Appendix A.

NSGA-II is a powerful algorithm for multi-objective optimization. However, as all evolutionary algorithms, its effectiveness comes at the expense of a high computational cost, in the order of hundreds to thousands of evaluations of the fitness function. On top of that, each evaluation of the fitness, *i.e.*, the quantiles for a given design, requires thousands of evaluations of the computational model. The overall computational cost is then prohibitive, especially when the computational model is expensive (*e.g.*, results from a finite element analysis). To alleviate this burden, we consider surrogate models, more precisely Kriging, as described in the next section.

4 Use of Kriging surrogates

4.1 Basics of Kriging

Several types of surrogate models have been used to reduce the computational cost of multi-objective evolutionary algorithms (Díaz-Manríquez et al., 2016). These surrogates are often embedded in an active learning scheme where they are iteratively updated so as to be especially accurate in areas of interest. Gaussian process modelling a.k.a. Kriging naturally lends itself to such approaches as it features a built-in error measure that can be used to sequentially improve its own accuracy.

Kriging (a.k.a. Gaussian process modelling) is a stochastic method that considers the model to approximate as a realization of a Gaussian process indexed by $\boldsymbol{w} \in \mathbb{R}^M$ and defined

by (Sacks et al., 1989; Santner et al., 2003; Rasmussen and Williams, 2006)

$$\widehat{\mathcal{M}}(\boldsymbol{w}) = \boldsymbol{\beta}^T \boldsymbol{f} + Z(\boldsymbol{w}), \tag{7}$$

where $\boldsymbol{\beta}^T \boldsymbol{f} = \sum_{j=1}^{p} \beta_j f_j(\boldsymbol{w})$ is a deterministic trend described here in a polynomial form (universal Kriging) and $Z(\boldsymbol{w})$ is a zero-mean stationary Gaussian process. The latter is completely defined by its auto-covariance function $\mathrm{Cov}\left[Z(\boldsymbol{w}), Z(\boldsymbol{w}')\right] = \sigma^2 R(\boldsymbol{w}, \boldsymbol{w}'; \boldsymbol{\theta})$, where σ^2 is the process variance, R is an auto-correlation function and $\boldsymbol{\theta}$ are hyperparameters to calibrate. The auto-correlation function, which is also referred to as *kernel*, encodes various assumptions about the process of interest such as its degree of regularity or smoothness. In this paper, we consider the Gaussian auto-correlation function, which is defined in its anisotropic form as

$$R(\boldsymbol{w}, \boldsymbol{w}'; \boldsymbol{\theta}) = \prod_{i=1}^{M} \exp\left[-\frac{1}{2}\left(\frac{w_i - w_i'}{\theta_i^2}\right)^2\right]. \tag{8}$$

The calibration of the Kriging model consists in training it on an experimental design (ED)
$\mathcal{D} = \left\{\left(\boldsymbol{w}^{(1)}, \mathcal{M}\left(\boldsymbol{w}^{(1)}\right)\right), \ldots, \left(\boldsymbol{w}^{(n_0)}, \mathcal{M}\left(\boldsymbol{w}^{(n_0)}\right)\right)\right\}$ and hence finding estimates of the three sets of hyperparameters $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2\}$. This can be achieved using methods such as maximum likelihood estimation or cross-validation (Santner et al., 2003; Bachoc, 2013; Lataniotis et al., 2018). Once the parameters are found, it is assumed that for any new point $\boldsymbol{w}$, $\widehat{\mathcal{M}}(\boldsymbol{w})$ follows a Gaussian distribution $\mathcal{N}\left(\mu_{\widehat{\mathcal{M}}}, \sigma^2_{\widehat{\mathcal{M}}}\right)$ defined by its mean and variance as follows:

$$\begin{aligned}
\mu_{\widehat{\mathcal{M}}}(\boldsymbol{w}) &= \boldsymbol{f}^T(\boldsymbol{w})\,\widehat{\boldsymbol{\beta}} + r(\boldsymbol{w})\,\boldsymbol{R}^{-1}\left(\mathcal{Y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right), \\
\sigma^2_{\widehat{\mathcal{M}}}(\boldsymbol{w}) &= \widehat{\sigma}^2\left(1 - \boldsymbol{r}(\boldsymbol{w})^T \boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{w}) + \boldsymbol{u}(\boldsymbol{w})^T\left(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{u}(\boldsymbol{w})\right),
\end{aligned} \tag{9}$$

where $\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{R}^{-1}\mathcal{Y}$ is the generalized least-square estimate of the regression coefficients $\boldsymbol{\beta}$, $\widehat{\sigma}^2 = \frac{1}{N}\left(\mathcal{Y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right)^T \boldsymbol{R}^{-1}\left(\mathcal{Y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right)$ is the estimate of the process variance with $\boldsymbol{F} = \left\{f_j\left(\boldsymbol{w}^{(i)}\right), j = 1, \ldots, p, i = 1, \ldots, n_0\right\}$ being the Vandermonde matrix, $\boldsymbol{R}$ the correlation matrix with $R_{ij} = R\left(\boldsymbol{w}^{(i)}, \boldsymbol{w}^{(j)}; \boldsymbol{\theta}\right)$ and $\mathcal{Y} = \left\{\mathcal{Y}^{(i)} = \mathcal{M}\left(\boldsymbol{w}^{(i)}\right), i = 1, \ldots, n_0\right\}$. Furthermore, $\boldsymbol{r}(\boldsymbol{w})$ is a vector gathering the correlation between the current point $\boldsymbol{w}$ and the experimental design points and finally, $\boldsymbol{u}(\boldsymbol{w}) = \boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{w}) - \boldsymbol{f}(\boldsymbol{w})$ has been introduced for convenience.

The prediction for any point $\boldsymbol{w}$ is then given by the mean $\mu_{\widehat{\mathcal{M}}}(\boldsymbol{w})$ in Eq. (9). On top of that, Kriging also provides a prediction variance $\sigma^2_{\widehat{\mathcal{M}}}(\boldsymbol{w})$ which is a built-in error measure that can be used to compute confidence intervals about the prediction. This variance is the main ingredient that has favored the development of adaptive methods such as Bayesian global optimization, as shown in the next section.

4.2 Kriging for multi-objective and robust optimization

Robust optimization

In robust optimization, one aims at finding an optimal solution which shows little sensi-

tivity to the various uncertainties in the system. This is generally achieved by minimizing a measure of robustness, *e.g.* the variance and/or mean or a quantile of the cost function (See Eq. (1)). The computation of the robustness measure coupled to the optimization incurs a prohibitive computational cost to the analysis. Hence, surrogate-assisted techniques, which are a means to reduce this cost, have received a lot of attention lately. However, due to the fact that the robustness measure that is minimized is only a by-product of the performance function which is usually approximated by a Kriging model, Bayesian optimization techniques are seldom considered. Instead, most of the literature focuses on non-adaptive schemes. In an early contribution, Chatterjee et al. (2019) performed a benchmark using several surrogate models for robust optimization. Lee and Park (2006) proposed approximating the logarithm of the variance of the performance function, which is then minimized using a simulated annealing algorithm. Some contributions however addressed the problem while considering active learning in a nested loop scheme, *e.g.*, Razaaly et al. (2020) where a quantile or the mean of the response is minimized. In this work, the authors built local surrogate models for each design and used them to estimate the quantiles. Similarly, Ribaud et al. (2020) proposed to minimize the Taylor series expansions of the performance function's expectation and variance. They considered various configurations for which they used either the expected improvement (or its multi-points version) of the quantities of interest. Finally, Sabater et al. (2021) developed a method based on Bayesian quantile regression to which they added an infill sampling criterion for the minimization of the quantile.

Multi-objective optimization

The naive and direct approach to leverage the optimization CPU cost with Kriging is to calibrate the latter using the experimental design $\mathcal{D}$ and then to use it in lieu of the original model throughout the optimization process. This approach is however expensive as it would require an accurate Kriging model in the entire design space. A more elaborate technique has been developed under the framework of *efficient global optimization* (EGO, Jones et al. (1998)) where the experimental design is sequentially enriched to find the minimizer of the approximated function. This enrichment is made by maximizing the *expected improvement* (EI), which is a merit function indicating how likely a point is to improve the currently observed minimum given the Kriging prediction and variance. The EGO framework has shown to be very efficient and new merit functions have been proposed to include other considerations such as a better control of the exploration/exploitation balance and the inclusion of constraints (Schonlau et al., 1998; Bichon et al., 2008). The first extension for multi-objective optimization was proposed by Knowles (2005) through the ParEGO algorithm. In ParEGO, the expected improvement is applied on the scalarized objective function derived using the augmented Tchebycheff approach. By varying weights in the scalarization, a Pareto front can be found. Zhang et al. (2010) proposed a similar approach using both the Tchebycheff and weighted sum decomposition techniques with the possibility of adding multiple well-spread

points simultaneously. However, such approaches inherit the pitfalls of the underlying decomposition methods. Keane (2006) proposed new extensions of EI that allows one to directly find a Pareto front without transforming the multi-objective problem into a mono-objective one. Another adaptation of a mono-objective criterion has been proposed by Svenson and Santner (2010) with their expected maximin improvement function. This is based on the Pareto dominance concept and involves a multi-dimensional integration problem which is solved by Monte Carlo simulation, except for the case when $m = 2$ for which an analytical solution is provided. Apart from such adaptations, some researchers have proposed new improvement functions directly built for a discrete approximation of the Pareto front. This includes the contribution of Shu et al. (2021) where a new acquisition function based on a modified hypervolume improvement and modified overall spread is proposed. Moving away from the EGO paradigm, Picheny (2015) proposed a stepwise uncertainty reduction (SUR) method for multi-objective optimization. Alternatively, the hypervolume measure of a set, which is the volume of the subspace dominated by the set up to a reference point, has been used to derive infill criteria. Namely, Emmerich et al. (2006) proposed a hypervolume-based improvement criterion. Originally based on Monte Carlo simulation, Emmerich et al. (2011) proposed a procedure to compute the criterion numerically. This method, however, does not scale well with the number of objective functions or samples in the approximated Pareto set. Efforts have been put in reducing this computational cost. This includes the contributions of Couckuyt et al. (2014); Hupkens et al. (2015); Yang et al. (2019). They all involve solving numerically the underlying integration problems by partitioning the space into slices using various methods. Finally, another attempt at decreasing the cost has been made by Gaudrie et al. (2020) where the authors proposed to focus the search of the Pareto front to a subspace according to some design preference. By doing so, the size of the discrete Pareto set can be reduced and the CPU cost of estimating the expected hypervolume improvement altogether.

In this work, we rather use a two-level nested approach as described in Section 4.3. We do not consider a direct Bayesian global optimization approach for two reasons. First, as shown above, such techniques in the context of multi-objective optimization are computationally intensive. Second, and most of all, the functions to optimize are not the direct responses of the model that would be approximated by a surrogate because of the robustness component of the problem. More specifically, in Bayesian optimization, the cost function $\mathfrak{c}$ is approximated by the GP model and minimized at the same time. In contrast, we consider here as objective function the quantile $Q_\alpha \left(\mathfrak{c}; \boldsymbol{X} \left(\boldsymbol{d} \right), \boldsymbol{Z} \right)$. Using this quantile directly in Bayesian optimization would require approximating the mapping $\boldsymbol{d} \mapsto Q_\alpha \left(\mathfrak{c}; \boldsymbol{X} \left(\boldsymbol{d} \right), \boldsymbol{Z} \right)$. However, computing the quantile for a given value of $\boldsymbol{d}$ requires running a Monte Carlo simulation where the underlying expensive computational model $\mathcal{M}$ is repeatedly evaluated. The overall computational cost would therefore be prohibitive. For this reason, we resort to a nested two-level approach where the approximation of the computational model and the optimization problems are decoupled.

4.3 Proposed approach

Motivation

In the previous section, we have very briefly reviewed the literature for multi-objective and robust optimization. The two topics were treated separately because, to the authors' best knowledge, very little to no research has been done for the combined problem. It is important to stress here that by "multi-objective and robust optimization" we disregard robust optimization methods that are formulated by minimizing the mean and the variance of a single performance function using a multi-objective optimization scheme. Such methods are often coined multi-objective robust optimization in the literature but are not the object of the present paper. We instead consider problems which are defined by multiple and conflicting performance functions, regardless of the robustness measure considered.

There have been some recent works combining multi-objective robust optimization as described here with Gaussian process modelling. Most notably, Zhang and Taflanidis (2019) proposed a framework for the solution of multi-objective problems under uncertainties using an adaptive Kriging model built in an augmented space. The latter is introduced as a means to combine the design and random variables space and hence to allow for the construction of a unique surrogate model that could simultaneously support optimization and uncertainty propagation (Kharmanda et al., 2002; Au, 2005; Taflanidis and Beck, 2008). Earlier works considering the augmented random space for optimization include Dubourg et al. (2011); Taflanidis and Medina (2014); Moustapha et al. (2016).

In this work, we also consider building a surrogate model in an augmented space, however using a two-level approach instead of direct Bayesian optimization as proposed in Zhang and Taflanidis (2019). The latter proposed to use the ε-constraint method to solve the multi-objective problem and devised a hybrid enrichment scheme enabled by the fact that their robustness measure is the expectation of the objective function. This is a more restrictive definition as the variance or dispersion within the performance function is not accounted for. Ribaud et al. (2020) estimated the expected improvement for the variance of the objective function by Monte Carlo simulation, which can be computationally expensive.

In this paper, we consider a nested level approach, hence allowing us to rely on a surrogate of the objective function itself rather than that of the robustness measure, herein the quantile. The latter is then estimated using Monte Carlo simulation with the obtained surrogate and the resulting optimization problem is solved using an evolutionary algorithm, namely *NSGA-II*. Coupling the optimization and the surrogate model construction in the augmented space, an enrichment scheme is devised as described in Step 6 of the algorithm described in the next section.

Worflow of the proposed method

The workflow of the algorithm is detailed in Figure 1 and summarized in the following:

1. **Initialization:** The various parameters of the algorithm are initialized. This includes:
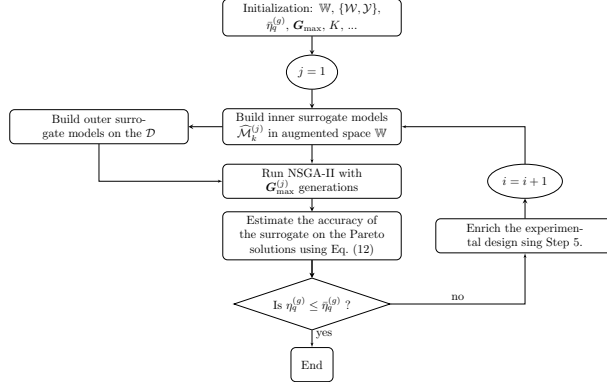
Figure 1: Flowchart of the proposed approach.

- the augmented random space, which is defined as in Moustapha and Sudret (2019), *i.e.*

$$\mathbb{W} = \mathbb{X} \times \mathbb{Z}, \tag{10}$$

where $\mathbb{Z}$ is the support of the random variables $\boldsymbol{Z} \sim f_{\boldsymbol{Z}}$ and $\mathbb{X}$ is the design space extended to account for the variability in the extreme values of the design variables. This *confidence space*, which is defined considering the cumulative distribution of the random variables at the lower and upper design bounds, respectively denoted by $F_{X_i|d_i^-}$ and $F_{X_i|d_i^+}$, reads

$$\mathbb{X} = \prod_{i=1}^{M_d} \left[x_i^-,\, x_i^+ \right], \tag{11}$$

where $x_i^- = F_{X_i|d_i^-}\left(\alpha_{d_i^-}\right)$ and $x_i^+ = F_{X_i|d_i^+}\left(1 - \alpha_{d_i^+}\right)$ are bounds on the design random variable space with respect to confidence levels of $\alpha_{d_i^-}$ and $\alpha_{d_i^+}$. Note that if there were no uncertainties on the design parameters, we would simply use $\mathbb{X} = \mathbb{D}$;

- the initial experimental design $\mathcal{D} = \left\{ (\mathcal{W}, \mathcal{Y}) = \left(\boldsymbol{w}^{(i)}, \boldsymbol{\mathcal{Y}}^{(i)} \right) \right\}$, where the inputs $\boldsymbol{w}^{(i)} \in \mathbb{W} \subset \mathbb{R}^M$ are sampled using Latin hypercube sampling (LHS, McKay et al. (1979)) and $\boldsymbol{\mathcal{Y}}^{(i)} = \mathcal{M}\left(\boldsymbol{w}^{(i)}\right) \in \mathbb{R}^m$, $i = \{1, \dots, n_0\}$, which each of the m components of $\boldsymbol{\mathcal{Y}}^{(i)}$ corresponding to one of the objective functions;

- the NSGA-II related parameters and convergence threshold such as the maximum number of generations $G_{\max}^{(j)}$, and

- the enrichment parameters, such as the number of enrichment points K per iteration.

2. **Surrogate model construction:** m Kriging models, denoted by $\widehat{\mathcal{M}}_k^{(j)}$, are built in the augmented space $\mathbb{W}$ using $\mathcal{D}$, as described in Section 4.1. This corresponds to building a separate Kriging model for each of the m performance functions.

3. **Optimization:** The NSGA-II algorithm is then run to solve the problem in Eq. (5) where the quantiles are computed using the surrogate model $\widehat{\mathcal{M}}^{(j)}$ in lieu of the original

11

model. Apart from the actual convergence criteria of NSGA-II, a maximum number of generations $G_{\max}^{(j)}$ is set. Its value is chosen arbitrarily low at the first iterations. The idea is that at first, emphasis is put more on the exploration of the design space. Since NSGA-II starts with a space-filling LHS over the entire design space, the first generations are still exploring. By stopping the algorithm then, we can enrich the ED by checking the accuracy of the quantiles estimated by the early sets of design samples. This allows us to direct part of the computational budget for enrichment in areas pointing towards the Pareto front without skipping some intermediate areas of interest.

4. **Accuracy estimation:** The local accuracy of the quantiles corresponding to the points in the current Pareto front $\mathcal{F}^{(j)}$ is estimated by considering the variance of each of the m Kriging models. More specifically, the relative quantile error for each objective in every point of the Pareto set is estimated as follows:

$$\eta_{q_k}^{(i)} = \frac{q_{\alpha_k}^+\left(\boldsymbol{d}^{(i)}\right) - q_{\alpha_k}^-\left(\boldsymbol{d}^{(i)}\right)}{q_{\alpha_k}\left(\boldsymbol{d}^{(i)}\right)}, \qquad \left\{\boldsymbol{d}^{(i)} \in \mathcal{F}^{(j)}, i = 1, \dots, \mathrm{Card}\left(\mathcal{F}^{(j)}\right)\right\}, \quad (12)$$

where $q_{\alpha_k}^{\pm}$ are upper and lower bounds of the k-th quantile estimated using the predictors $\boldsymbol{\mu}_{\widehat{\mathcal{M}}_k^{(j)}} \pm 1.96\,\boldsymbol{\sigma}_{\widehat{\mathcal{M}}_k^{(j)}}$. These are not actual bounds *per se* but a "kind of" 95%-confidence interval indicating how locally accurate are the Kriging models. Note that when the quantiles are close to 0, it is possible to replace the denominator in Eq. (12) by another normalizing quantity, such as the variance of the model responses at the first iteration.

5. **Convergence check:** The convergence criterion is checked for all the points of the Pareto set with outliers filtered out to accelerate convergence. Samples with values larger than $\eta_{q_k}^{90} + 1.5\,(\eta_{q_k}^{90} - \eta_{q_k}^{10})$ are considered to be outliers, where $\eta_{q_k}^{90}$ and $\eta_{q_k}^{10}$ are respectively the 90-th and 10-th percentile of the convergence criterion for the k-th objective. A usual definition of outliers is based on the interquartile range (McGil et al., 1978) but we consider here a more conservative definition by rather using the interdecile range. Convergence is assumed if the relative quantile errors $\eta_{q_k}^{(i)}$ for all the points of the Pareto front (potential outliers excluded), as computed by Eq. (12), are below a threshold $\bar{\eta}_q^{(j)}$. The latter can also be set adaptively so as to be loose in the initial exploratory cycles of the algorithm. The algorithm then goes to Step 8 if convergence is achieved or proceeds with the next step otherwise.

6. **Enrichment:** K points are added per iteration. The enrichment is carried out in two steps. The idea is to add multiple points to the ED within a single iteration by splitting them into two sets. While the first set consists of the points that directly maximize the learning function, the second set is spread out evenly among the best candidates for enrichment. In practice, they are obtained as follows:

(a) First set: For each objective that does not satisfy the convergence criterion, the

12

point in the Pareto set with the largest error is identified and denoted by $\boldsymbol{d}_k^{\max}$. The corresponding enrichment sample is then chosen as the one which maximizes the local Kriging variance, $i.e.$,:

$$\boldsymbol{w}_{\text{next}} = \arg \max_{\boldsymbol{w} \in \mathcal{C}_{q_k}} \sigma_{\widehat{\mathcal{M}}_k^{(j)}}(\boldsymbol{w}), \tag{13}$$

where $\mathcal{C}_{q_k} = \left\{ \left(\boldsymbol{x}^{(i)} \left(\boldsymbol{d}_k^{\max} \right), \boldsymbol{z}^{(i)} \right), i = 1, \ldots, N \right\}$ is the sample set used to compute the quantiles for the design point $\boldsymbol{d}_k^{\max}$. The number of added points is denoted by K_1 and corresponds to the number of objectives for which the convergence criterion is not respected.

(b) Second set: The remaining $K_2 = K - K_1$ points are identified by first selecting all the design solutions in the Pareto front that produce errors larger than the threshold $\bar{\eta}_q^{(j)}$. This set is then reduced into K_2 evenly distributed points using K-means clustering. For each of these points, the corresponding enrichment sample is chosen as the one maximizing the Kriging variance among the samples used to compute the corresponding quantiles, similarly to Eq. (13).

7. **Experimental design update:** The original computational model is then evaluated on the K enrichment points identified in the previous step, hence leading to K new pairs of samples $\mathcal{D}_{\text{next}}^{(j)} = \left\{ \left(\boldsymbol{w}_{\text{next}}^{(1)}, \mathcal{M} \left(\boldsymbol{w}_{\text{next}}^{(1)} \right) \right), \ldots, \left(\boldsymbol{w}_{\text{next}}^{(K)}, \mathcal{M} \left(\boldsymbol{w}_{\text{next}}^{(K)} \right) \right) \right\}$. These samples are then added to the current experimental design, $i.e.$, $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{next}}^{(j)}$ and the algorithm returns to Step 2.

8. **Termination:** The outliers identified at Step 5, if any, are removed and the remaining points in the Pareto front and set are returned.

To accelerate the whole procedure, we propose to optionally build at each iteration, and for each objective, an outer surrogate model to be used by the optimizer. The corresponding experimental design reads:

$$\mathcal{D}_{\text{out}}^{(j)} = \left\{ \left(\boldsymbol{d}^{(i)}, \widehat{q}_{\alpha_k} \left(\boldsymbol{d}^{(i)} \right) \right), i = 1, \ldots, n_{\text{out}}^{(j)} \right\} \tag{14}$$

where $\widehat{q}_{\alpha_k}$ is the quantile estimated using the current surrogate model $\widehat{\mathcal{M}}_k^{(j)}$. Since this surrogate is built using another surrogate model, it is not necessary to use active learning at this stage. Instead, we simply draw a unique and large space-filling experimental design of size $n_{\text{out}}^{(j)}$. To increase the accuracy around the Pareto front in the outer surrogate, the experimental design inputs are updated after each cycle considering two different aspects. First, the samples in the Pareto set of the previous cycle are added to the space-filling design. Second, the accuracy of the outer surrogate w.r.t. the inner one in estimating the quantiles in the Pareto front is checked after each cycle. The related error is monitored and calculated as follows:

$$\eta_{q_k,\text{out}}^{(j)} = \max_{i \in \left\{ 1, \ldots, \text{Card} \left(\mathcal{F}^{(j)} \right) \right\}} \frac{\left| \widehat{q}_{\alpha_k} \left(\boldsymbol{d}^{(i)} \right) - \mu_{\widehat{q}_k} \left(\boldsymbol{d}^{(i)} \right) \right|}{\widehat{q}_{\alpha_k} \left(\boldsymbol{d}^{(i)} \right)}, \tag{15}$$

13

where $\mu_{\widehat{q}_k}$ is the quantile predicted by the outer surrogate model. If this error is larger than a threshold $\bar{\eta}_{q,\mathrm{out}}$ the size of the ED for the outer surrogate is increased before the next cycle.

5   The case of categorical variables

The methodology presented in the previous section assumes that all variables are continuous. However, in some cases, and particularly in the applications considered in this paper, some of the design variables may be categorical. Categorical variables are defined in a discrete and finite space and have the particularity that they are qualitative and cannot be meaningfully ordered. As such, the Euclidean distance metric does not apply to such variables.

However, NSGA-II, Kriging and K-means clustering rely on such a metric since it is used in the definition of the cross-over and mutation operators, and in the evaluation of the kernel function. We will thus consider adaptations of these methods to handle the mixed categorical-continuous problems treated in this paper. The main idea of these adaptations is to remain within the general implementation of these methods, so as to use existing tools without further developments.

To highlight the nature of the variables, we introduce the notations $\boldsymbol{w} = (\boldsymbol{w}_{\mathrm{con}}, \boldsymbol{w}_{\mathrm{cat}})$ where $\boldsymbol{w}_{\mathrm{con}} \in \mathbb{R}^{M_{\mathrm{con}}}$ is a vector gathering the continuous variables while $\boldsymbol{w}_{\mathrm{cat}} \in \mathbb{R}^{M_{\mathrm{cat}}}$ gathers the categorical parameters. Each component w_{cat_j} can take one of b_j values, called levels, and denoted by $\{\ell_1, \dots, \ell_{b_j}\}$. Hence, there is a total of $b = \prod_{j=1}^{M_{\mathrm{cat}}} b_j$ categories.

NSGA-II with mixed continuous-categorical variables

Prompted by their reliance on heuristics, a large variety of both cross-over and mutation operators have been proposed in the literature throughout the years. The ones developed for NSGA-II, *i.e.*, the simulated binary crossover (SBX) and polynomial mutation, are dedicated to continuous variables only (See Appendix A for more details). Various adaptations have been proposed but they may only be used for discrete variables as they involve numerical rounding. For operators dedicated to categorical variables, we look into those developed for binary genetic algorithm (Umbarkar and Sheth, 2015). Since GA allows for treating each variable separately, we first split the continuous and categorical variables into two groups. For the continuous variables, the original operators for NSGA-II are used without any modification. For the categorical variables, we consider two operators typically used in binary genetic algorithms, namely the one-point crossover operator and a simple random mutation.

In practice, let us consider two parents whose categorical components are denoted by $\boldsymbol{w}_{\mathrm{cat}}^{(1)}$ and $\boldsymbol{w}_{\mathrm{cat}}^{(2)}$. One-point cross-over is achieved by randomly choosing an integer p such that $1 \leq p < M_{\mathrm{cat}}$ which will serve as a cross-over point where the parents are split and

swapped to create two offsprings, *i.e.*,

$$\begin{cases} \boldsymbol{w}_{\text{cat}}^{\text{offspring},(1)} = \left\{ \ell_1^{(1)}, \, \dots \, , \ell_p^{(1)}, \ell_{p+1}^{(2)}, \, \dots \, , \ell_{M_{\text{cat}}}^{(2)} \right\}, \\ \boldsymbol{w}_{\text{cat}}^{\text{offspring},(2)} = \left\{ \ell_1^{(2)}, \, \dots \, , \ell_p^{(2)}, \ell_{p+1}^{(1)}, \, \dots \, , \ell_{M_{\text{cat}}}^{(1)} \right\}. \end{cases} \tag{16}$$

As for the mutation, the components to be mutated are simply replaced by another level of the same variables drawn with equi-probability. Let us assume that the component $w_{\text{cat},j}^{\text{offspring},(1)} = \ell_q$ of an offspring has been selected for mutation. It is then replaced by drawing uniformly from the set $\{\ell_1, \, \dots \, , \ell_{M_{\text{cat}}}\} \setminus \ell_q$, where each level has a probability of $1/(M_{\text{cat}} - 1)$ to be selected. It should be noted at this point that mutation is only performed with a probability of $1/M$, meaning that on average only one variable (both categorical and continuous) is mutated per iteration.

Kriging with mixed continuous-categorical variables

One of the most important features of Kriging is the auto-correlation or kernel function, which encodes assumptions about the centered Gaussian process $Z(\boldsymbol{w})$. Most Kriging applications rely on continuous variables and there is a rich list of possible kernel functions for such variables. Examples include the polynomial, Gaussian, exponential or Matérn kernels (Santner et al., 2003). Such kernels are built exclusively for quantitative variables and need some tedious adaptations or pre-processing such as one-hot encoding when it comes to qualitative variables. An alternative yet cumbersome technique is to build multiple models, one associated to each category. At an intermediate level, Tran et al. (2019) proposed using a Gaussian mixture based on clusters corresponding to combination of categorical variables. A more convenient way of handling mixed categorical-continuous variable problems is through the definition of adapted kernels. This is facilitated by an important property which allows one to build valid kernels by combining other valid kernels through operators such as product, sum or ANOVA (Roustant et al., 2020). Considering the product operator and splitting the input variables into their continuous and categorical components, a kernel can be obtained as follows:

$$k(\boldsymbol{w}, \boldsymbol{w}') = k_{\text{con}}(\boldsymbol{w}_{\text{con}}, \boldsymbol{w}'_{\text{con}}) \cdot k_{\text{cat}}(\boldsymbol{w}_{\text{cat}}, \boldsymbol{w}'_{\text{cat}}), \tag{17}$$

where k_{con} and k_{cat} are kernels defined on the space of continuous and categorical variables, respectively.

Thanks to this property, it is possible to build a different kernel for each variable separately. For the continuous variables, the traditional kernels can be used without any modification. As for the categorical variables, a few approaches have been proposed in the literature. One of the earliest contributions is that of Qian et al. (2008) which however involved a tedious calibration procedure. Zhou et al. (2011) proposed an enhancement based on hypersphere decomposition with a much more simplified calibration procedure. However, they use the most generic parametrization approach which does not scale well with the overall number of categories b.

More parsimonious representations have been proposed, *e.g.,* the so-called *compound symmetry* which assumes the same correlation among different levels of the same categorical variable. In this work, we consider this approach combined with a simple dissimilarity measure for each categorical variable, *i.e.,*

$$S_{w_k w'_k} = \begin{cases} 0 & \text{if} \quad w_k = w'_k, \\ 1 & \text{if} \quad w_k \neq w'_k; \end{cases} \tag{18}$$

The corresponding general form of the compound symmetry kernel for one categorical variable is (Pelamatti et al., 2020):

$$k\left(w_{\text{cat}_k}, w'_{\text{cat}_k}\right) = \begin{cases} 1 & \text{if} \quad w_{\text{cat}_k} = w'_{\text{cat}_k}, \\ c & \text{if} \quad w_{\text{cat}_k} \neq w'_{\text{cat}_k}, \end{cases} \tag{19}$$

where $0 < c < 1$. In this work, the same compound symmetry kernel is built but embedded within usual stationary kernels such as the Gaussian, exponential or Matérn correlation functions. Considering the Gaussian kernel for instance, the corresponding uni-dimensional kernel reads:

$$k\left(w_{\text{cat}_k}, w'_{\text{cat}_k}\right) = \exp\left(-\frac{1}{2}\left(\frac{S_{w_{\text{cat}_k} w'_{\text{cat}_k}}}{\theta_k}\right)^2\right). \tag{20}$$

Combining all dimensions, the following kernel is eventually obtained:

$$k\left(\boldsymbol{w}, \boldsymbol{w}'\right) = \exp\left(-\frac{1}{2}\sum_{k=1}^{M_{\text{con}}}\left(\frac{w_{\text{con}} - w'_{\text{con}}}{\theta_{\text{con}_k}}\right)^2 - \frac{1}{2}\sum_{k=1}^{M_{\text{cat}}}\left(\frac{S_{w_{\text{cat}_k} w'_{\text{cat}_k}}}{\theta_{\text{cat}_k}}\right)^2\right). \tag{21}$$

This formulation allows us to build and calibrate the Kriging model using the same tools as for the continuous case, since the hyperparameters θ_{cat_k} are defined similarly to θ_{cont_k} and both can be calibrated in the same set-up.

Finally it should be noted, as also mentioned in Roustant et al. (2020), that despite using different constructions, this is nearly identical to the one proposed by Halstrup (2016), where the Gower distance, a distance in the mixed variable space, is embedded within the Matérn auto-correlation function. The only difference is that the Euclidean distance is used for the continuous part, instead of a measure based on the range of the variables.

This kernel is flexible enough and can be implemented easily within a generic software such as UQLab (Marelli and Sudret, 2014). However, as shown in Pelamatti et al. (2020), such kernels may be limited when there are a high number of levels or categories. In fact, a kind of stationary assumption is embedded in this construction as the kernel only depends on the difference between two levels, regardless of their actual values. More advanced techniques such as group kernels (Roustant et al., 2020) or latent variables Gaussian process (Zhang et al., 2020; Wang et al., 2021) have been proposed in the literature but they are not considered in this work.

Let us eventually note that the Gower distance measure is also used to compute distances in the K-means clustering algorithm in the enrichment scheme. In our implementation, the cluster centers are updated by calculating, in each cluster, the mean for the continuous variables and the mode for the categorical values.

6 Applications

In this section, we will consider three applications examples. The first two ones are analytical and will serve to illustrate and validate the proposed algorithm. The last one is the engineering application related to optimal building renovation strategies.

For each NSGA-II cycle, we consider a population size of $L = 100$ and a maximum number of iterations $G_{\max} = 100$. The probability of mutation and crossover are respectively set to 0.5 and 0.9.

To assess the robustness of the proposed methodology, the analysis is repeated 10 times for the first two examples. The results are summarized using boxplots, where the central mark indicates the median, the bottom and top edges of the bars indicate the 25th and 75th percentiles, respectively, and the outliers are plotted using small circles. The reference solution is obtained by solving this problem using NSGA-II and without surrogates. The accuracy is assessed by evaluating the closeness of the obtained Pareto front with the reference one. The hypervolume, which is the volume of the space dominated by the Pareto front, is chosen as basis for comparison. It is often computed up to a given reference point as illustrated in Figure 2. In this work, the reference point is considered as the Nadir point, *i.e.*,

$$R = (R_1, R_2) = \left(\max_{\boldsymbol{d} \in \mathcal{D}^*_{\mathrm{ref}}} \mathfrak{c}_1 \left(\boldsymbol{d} \right), \max_{\boldsymbol{d} \in \mathcal{D}^*_{\mathrm{ref}}} \mathfrak{c}_2 \left(\boldsymbol{d} \right) \right), \tag{22}$$

where $\mathcal{D}^*_{\mathrm{ref}}$ is the Pareto set corresponding to the reference Pareto front $\mathcal{F}_{\mathrm{ref}}$.
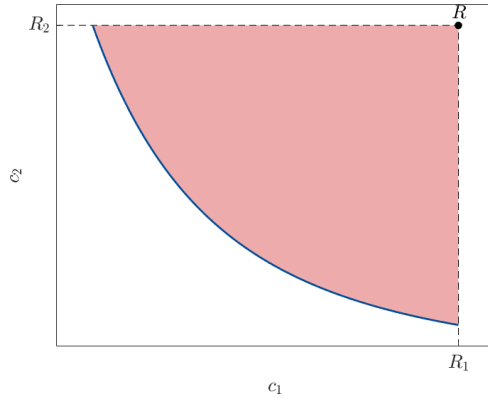

Figure 2: Illustration of the hypervolume as the red shaded area. The reference Pareto front $\mathcal{F}_{\mathrm{ref}}$ is repesented by the blue line while the Nadir point R is shown by the black dot.

The hypervolume (here area) up to the reference point R is approximated using the trapezoidal rule for integration. By denoting the one obtained from the i-th repetition by $A^{(i)}$, the resulting relative error measure reads

$$\Delta^{(i)}_{\mathrm{HV}} = \frac{\mid A^{(i)} - A_{\mathrm{ref}} \mid}{A_{\mathrm{ref}}}, \tag{23}$$

where A_{ref} is the hypervolume estimated using the reference solution.

To estimate the part of the error due to the outer surrogate model, we compute the same error again, but this time using the original model and the Pareto set. Denoting the

corresponding hypervolume by A', the following relative error is estimated:

$$\Delta_{\mathrm{HV}}^{\prime(i)} = \frac{\mid A'^{(i)} - A_{\mathrm{ref}} \mid}{A_{\mathrm{ref}}}. \tag{24}$$

6.1 Example 1: 7-dimensional analytical problem

The first example considered here is an analytical problem built for illustration and validation purposes. The original problem, referred to as BNH, is popular in benchmarks for multi-objective optimization and was introduced in Binh and Korn (1997). It is a two-dimensional problem which is however deterministic and only contains continuous variables. We therefore add two categorical variables and three random variables so as to build a multi-objective and robust optimization problem. The original problem, which reads

$$\boldsymbol{d}^* = \arg \min_{\boldsymbol{d} \in [0,\,5] \times [0,\,3]} \left\{ \tilde{\mathfrak{c}}_1 = 4\left(d_1^2 + d_2^2\right) \, ; \, \tilde{\mathfrak{c}}_2 = (d_1 - 5)^2 + (d_2 - 5)^2 \right\}$$
$$\text{subjet to:} \left\{ (d_1 - 5)^2 + d_1^2 - 25 \le 0 \, ; \, -(d_1 - 8)^2 - (d_2 + 3)^2 + 7.7 \le 0 \right\}, \tag{25}$$

is modified using the following two steps:

- First, the two objective functions are modified to include two categorical variables d_3 and d_4, which can take each three possible levels:

$$\begin{cases} \bar{\mathfrak{c}}_1 = \tilde{\mathfrak{c}}_1 + 5 & \bar{\mathfrak{c}}_2 = \tilde{\mathfrak{c}}_2 + 5 & \text{if} \quad d_3 = 1, \\ \bar{\mathfrak{c}}_1 = \tilde{\mathfrak{c}}_1 - 2 & \bar{\mathfrak{c}}_2 = \tilde{\mathfrak{c}}_2 - 2 & \text{if} \quad d_3 = 2, \\ \bar{\mathfrak{c}}_1 = \tilde{\mathfrak{c}}_1 & \bar{\mathfrak{c}}_2 = \tilde{\mathfrak{c}}_2 & \text{if} \quad d_3 = 3, \end{cases} \tag{26}$$

$$\begin{cases} \hat{\mathfrak{c}}_1 = 2\,\bar{\mathfrak{c}}_1 & \hat{\mathfrak{c}}_2 = 2\,\bar{\mathfrak{c}}_2 & \text{if} \quad d_4 = 1, \\ \hat{\mathfrak{c}}_1 = 0.8\,\bar{\mathfrak{c}}_1 & \hat{\mathfrak{c}}_2 = 0.95\,\bar{\mathfrak{c}}_2 & \text{if} \quad d_4 = 2, \\ \hat{\mathfrak{c}}_1 = 0.95\,\bar{\mathfrak{c}}_1 & \hat{\mathfrak{c}}_2 = 0.8\,\bar{\mathfrak{c}}_2 & \text{if} \quad d_4 = 3, \end{cases} \tag{27}$$

- Then the random variables are added as follows:

$$\begin{cases} \mathfrak{c}_1 = (\hat{\mathfrak{c}}_1 + z_5^2)\, z_7, \\ \mathfrak{c}_2 = (\hat{\mathfrak{c}}_2 + z_6^2)\, z_7, \end{cases} \tag{28}$$

where $Z_5 \sim \mathrm{Lognormal}(5, 0.5^2)$, $Z_6 \sim \mathrm{Lognormal}(4, 0.4^2)$ and $Z_7 \sim \mathrm{Gumbel}(1, 0.2^2)$. In this notation, the two parameters are the mean and variance of Z_5, Z_6 and Z_7, respectively.

The final optimization problem therefore reads:

$$\boldsymbol{d}^* = \arg \min_{\boldsymbol{d} \in [0,\,5] \times [0,\,3]} \left\{ Q_\alpha\left(\mathfrak{c}_1; \boldsymbol{X}\left(\boldsymbol{d}\right), \boldsymbol{Z}\right), Q_\alpha\left(\mathfrak{c}_2; \boldsymbol{X}\left(\boldsymbol{d}\right), \boldsymbol{Z}\right) \right\}$$
$$\text{subject to:} \left\{ (d_1 - 5)^2 + d_1^2 - 25 \le 0 \, ; \, -(d_1 - 8)^2 - (d_2 + 3)^2 + 7.7 \le 0 \right\} \tag{29}$$

For each of the 10 repetitions, the analysis is started with an initial experimental design of size $n_0 = 3M = 21$. We consider five values of the stopping criterion, namely $\bar{\eta}_q = \{0.1, 0.05, 0.03, 0.01, 0.001\}$ (See Eq. (12)). Figure 3 shows the relative error on the hypervolumes (as in Eq. (23) and (24)) for each of these cases. As expected, the accuracy

of the results increases as the convergence criterion is made tighter. This is however true only up to a given point. With $\bar{\eta}_q = 10^{-3}$, the criterion Δ'_{HV}, which is based on the Pareto set keeps decreasing while it increases noticeably when considering Δ_{HV} which is directly computed using the estimated Pareto front. This discrepancy can be attributed to the outer surogate model that becomes less accurate, probably due to overfitting, as the number of samples sizes is increased. It should be reminded here that the outer surrogate is built using a unique experimental design whose size may increase together with the iterations. Finally, we note that the number of evaluations of the original model increases rapidly together with the threshold for convergence, as shown in Figure 4.

(a) Relative error w.r.t. surrogate model   (b) Relative error w.r.t. original model
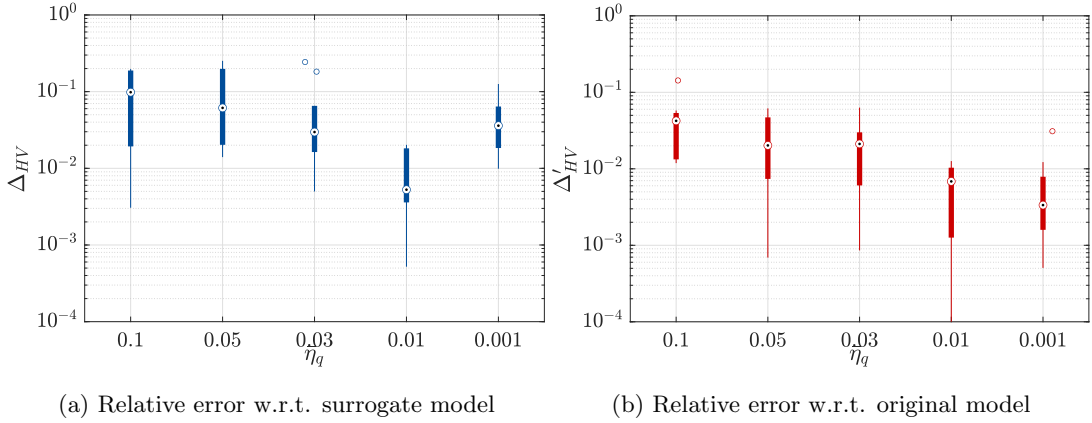
Figure 3: Example 1: Relative errors w.r.t. the reference hypervolume for various thresholds of the stopping criterion.
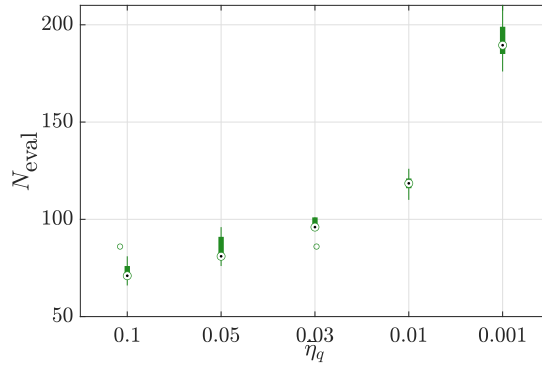

Figure 4: Example 1: Number of model evaluations for various thresholds of the stopping criterion.

To further explore these results, we consider the run with the median accuracy at the threshold of $\bar{\eta}_q = 0.03$. Figure 5 shows the convergence of the selected analysis with the boxplots representing the relative error on the quantile for each point of the Pareto fronts. The green lines correspond to the worst quantile relative error after excluding outliers. The vertical dashed lines together with the triangular markers show where the algorithm would

stop for various thresholds of the convergence criterion. After 20 iterations, the rate of convergence decreases considerably, which explains the large added number of model evaluations required to reach the target of $\bar{\eta}_q = 10^{-3}$.
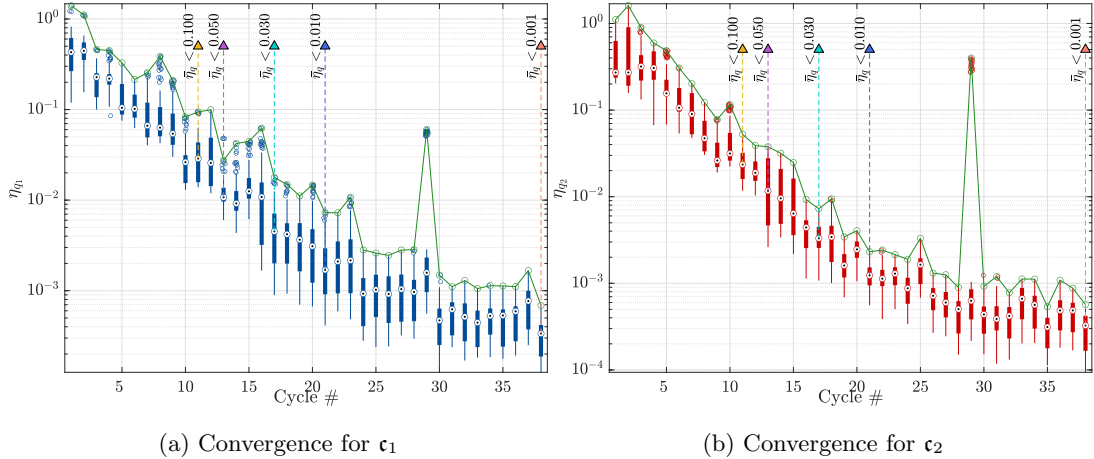

(a) Convergence for $\mathfrak{c}_1$

(b) Convergence for $\mathfrak{c}_2$

Figure 5: Example 1: Relative error of the 90% quantiles of the costs $\mathfrak{c}_1$ and $\mathfrak{c}_2$ for the entire Pareto front at the end of each cycle. The upper convergence limit is shown by the continuous line.

Figure 6 shows the corresponding Pareto front (Figure 6a) and set (Figure 6b) together with the reference ones. As expected, given the way the categorical variables were introduced, there are two combinations of the categorical variables in the Pareto set: $(d_3, d_4) = (2, 2)$ or $(2, 3)$. The two fronts cover the same volume and are spread in a similar way. In the input space, we can also see that the solutions cover roughly the same area. For this highlighted example, convergence is achieved in 16 cycles with a total of 101 evaluations of the original model. This contrasts with the total of $5 \cdot 10^7$ model evaluations (100 iterations of NSGA-II with 100 samples per generation and $5,000$ samples for the estimation of the quantile for each design) required for the reference solution using a brute force approach.

6.2 Example 2: Analytical problem with discontinuous Pareto front

This second analytical example is adapted from Manson et al. (2021). The Pareto front for this example is concave and presents two discontinuities. Further it only features design variables (*i.e.*, there is no environmental variables), some of which are random. This allows us to showcase the versatility of the proposed algorithm. The two deterministic cost functions read

$$
\mathfrak{c}_1 = \begin{cases} 1 - \exp\left(-\sum_{i=1}^{2}\left(d_i - 1/\sqrt{2}\right)^2\right) & \text{if } d_3 = 1, \\ 1.25 - \exp\left(-\sum_{i=1}^{2}\left(d_i - 1/\sqrt{2}\right)^2\right) & \text{if } d_3 = 2, \end{cases}
$$
$$
\mathfrak{c}_2 = \begin{cases} 1 - \exp\left(-\sum_{i=1}^{2}\left(d_i + 1/\sqrt{2}\right)^2\right) & \text{if } d_3 = 1, \\ 0.75 - \exp\left(-\sum_{i=1}^{2}\left(d_i + 1/\sqrt{2}\right)^2\right) & \text{if } d_3 = 2. \end{cases}
\tag{30}
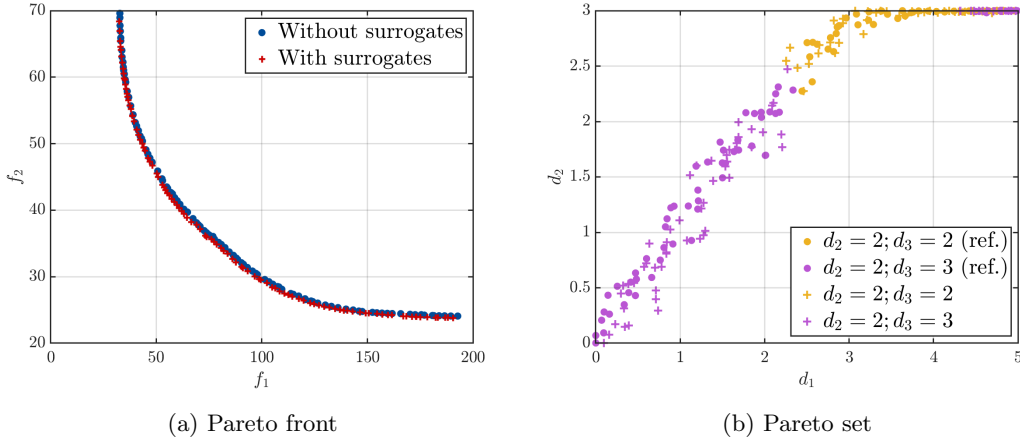$$

20

(a) Pareto front (b) Pareto set

Figure 6: Example 1: Comparison of the Pareto fronts and sets for the results with median relative hypervolume error and the reference solution.

We then add some random variables which we associate to the design variables d_1 and d_2. Both are assumed normal, *i.e.* $X_i \sim \mathcal{N}\left(d_i, 0.1^2\right)$, $i = \{1, 2\}$.

The initial experimental design is set to $n_0 = 3\,M = 9$. For this example, the variations in accuracy due to tighter convergence criteria are not so noticeable, as can be seen in Figure 7. In fact, the resulting Pareto front is already accurate enough with $\bar{\eta}_q = 0.1$. It should be noted that some of the variability in estimating the relative error is due to the approximations inherent to computing the hypervolume using the trapezoidal rule for integration and the limited set of points it relies upon. This rapid convergence may also

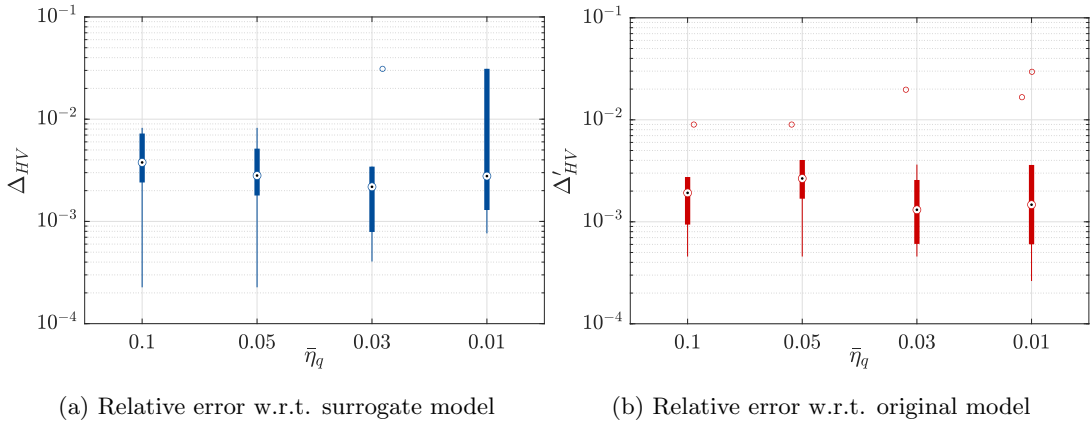(a) Relative error w.r.t. surrogate model (b) Relative error w.r.t. original model

Figure 7: Example 2: Relative errors w.r.t. the reference hypervolume for various thresholds of the stopping criterion.

be seen on the small increase in number of model evaluations and cycles to convergence as shown in Figures 8 and 9.

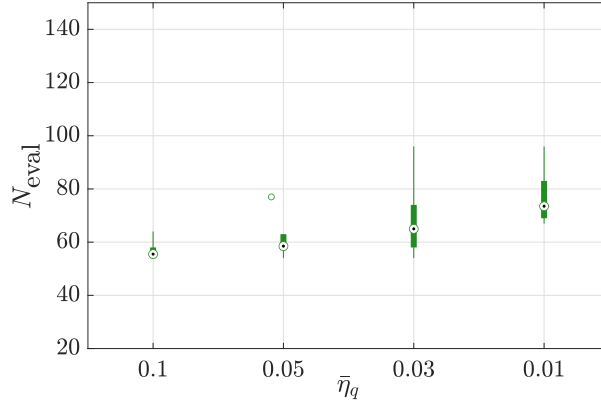Finally, we show in Figure 10 the Pareto front and sets obtained for the median solution at

21

Figure 8: Example 2: Number of model evaluations for various thresholds of the stopping criterion.
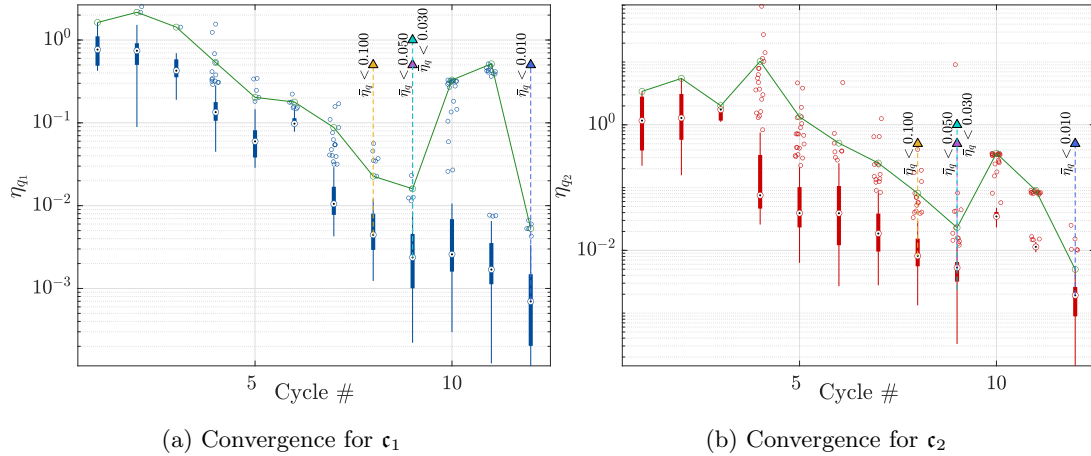

(a) Convergence for $\mathfrak{c}_1$

(b) Convergence for $\mathfrak{c}_2$

Figure 9: Example 2: Relative error of the quantiles of the costs $\mathfrak{c}_1$ and $\mathfrak{c}_2$ for the entire Pareto front at the end of each cycle. The upper convergence limit is shown by the continuous line.

the threshold $\bar{\eta}_q = 0.03$. The two Pareto fronts coincide in the objective space, showing good convergence of the algorithm. Similarly, the Pareto sets of the reference and approximated solutions overlap in the input space.

6.3 Example 3: Application to building renovation

This third example deals with building renovation, which is actually the application that has motivated this work. Because buildings are responsible for 40% of energy consumption and 36% of greenhouse gas emissions from energy in Europe, the European union has recently pledged to renovate 35 million buildings in the next 10 years (European Commission, 2020). Building renovation is indeed an important lever since current buildings are not energy-efficient but yet are expected for the most part to still stand by 2050.

Renovation thus needs to take into account the entire life cycle of the building, which may span over decades. This implies accounting for various uncertainties, be it in socio-

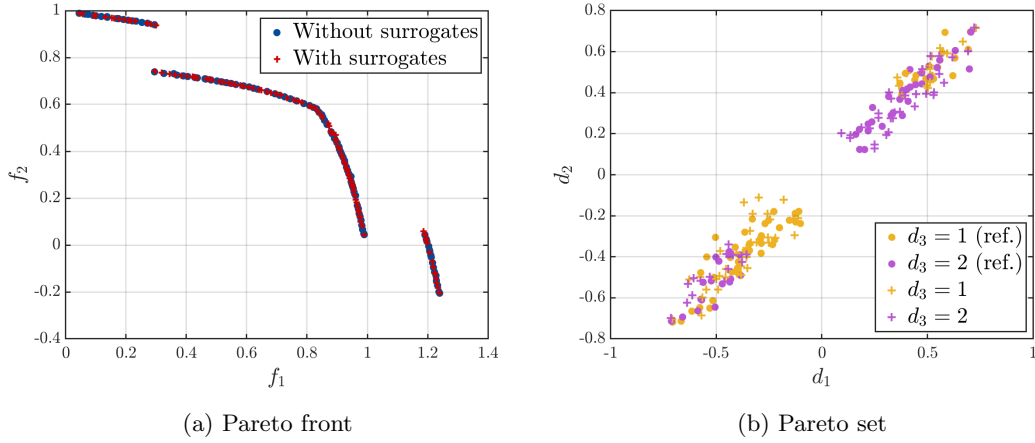(a) Pareto front　　　　　　　　　　(b) Pareto set

Figure 10: Example 2: Comparison of the Pareto fronts and sets for the results with median relative hypervolume error and the reference solution.

economic and environmental conditions or in the variability of the parameters of the selected renovation strategies. This can be achieved using life cycle analysis where two quantities of interest are often considered: the *life cycle cost* (*LCC*) and the *life cycle environmental impact* (*LCEI*). The former includes various costs such as the cost of production of new materials, the related cost of replacement or repair, the labor cost, etc. The latter refers to the overall greenhouse gas emissions over the entire life cycle of the building posterior to the renovation.

The stakeholders need to weigh these two quantities to decide which are the optimal renovation strategies for a given building while accounting for the various sources of uncertainty. To this aim, robust multi-objective optimization may be used as a reliable way of exploring the extremely large design space (*i.e.*, the combination of all possible choices available to the stakeholders). Using $\mathfrak{c}_1 = LCC$ and $\mathfrak{c}_2 = LCEI$, the problem may be formulated as in Eq. (5) and the proposed methodology may be used to solve it.

As an application, we consider in this paper, a building situated in Western Switzerland and constructed in 1911 (See Figure 11). The LCA is carried out using a model developed in Galimshina et al. (2020). The computational model is implemented in Python and a single run lasts a few seconds. The model contains more than a hundred parameters. However, using expert knowledge and global sensitivity analysis, screening allowed us to reduce the input to 23 parameters, among which 6 are design parameters and 13 are environmental variables (Galimshina et al., 2020). The design parameters include 4 categorical variables as shown in Table 1, which lead to 3, 600 levels. This includes the 6 types of heating system: oil, gas, heat pump, wood pellets, electricity and district heating. Various types of walls and windows with different characteristics that are selected from a publicly available catalog are also considered. The remaining two design parameters are the insulation thickness associated to the selected walls and slabs. The environmental variables are all random and

23

their distributions are shown in Table 2. They are split into three groups which pertain to the occupancy usage, economic parameters and renovation components variability.

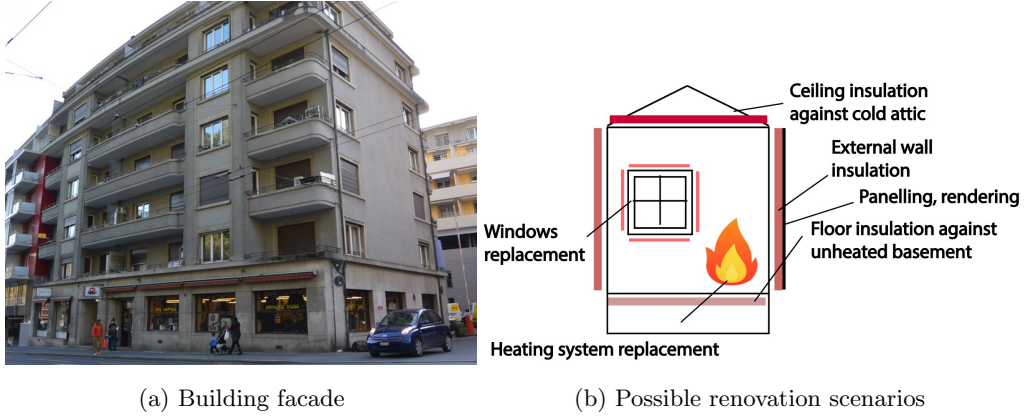(a) Building facade   (b) Possible renovation scenarios

Figure 11: Building considered from renovation together with a few possible renovation scenarios (adapted from Galimshina et al. (2020)).

Table 1: Design parameters selected for the building renovation application. Curly brackets $\{\cdot\}$ correspond to categorical variables. Square brackets $[\cdot]$ define the interval of interest for continuous variables.

Parameter	Range
Heating system	$\{1, 2, , \dots, 6\}$
Exterior wall type	$\{1, 2, \dots, 10\}$
Exterior wall insulation thickness (m)	$[0.1, 0.5]$
Windows type	$\{1, 2, \dots, 10\}$
Slab against unheated type	$\{1, 2, \dots, 6\}$
Slab against unheated insulation thickness (m)	$[0.1, 0.5]$

The analysis is initialized with an experimental design of size $n_0 = 3M = 69$ points drawn using an optimized Latin hypercube sampling scheme. The threshold for convergence is set to $\bar{\eta}_q = 0.03$, which resulted in 57 cycles of the algorithm as shown in Figure 12. This corresponds to a total of 271 model evaluations. The Pareto front is shown in Figure 13. It features some discontinuities which are due to some noticeable changes in the properties of the categorical variables. The main driver to decrease both LCC and $LCEI$ is the heating system. The upper part of the Pareto front corresponds to a heat pump which leads to small values of LCC and larger values of $LCEI$. This is in contrast with the wood pellets which correspond to the lower part of the Pareto front. For this example, we eventually select one solution, which is in the upper part and is highlighted by the red diamond in Figure 13. This choice reflects a preference on the cost with respect to the environmental impact. Table 3

Table 2: Distribution of the environmental variables for the building renovation application.

Parameter	Distribution	Parameter 1[a]	Parameter 2[b]
Room temperature (degree C)	Uniform	20	23
Airflow (m^3/h,m^2)	Uniform	0.7	1
Inflation rate (%)	Uniform	0.005	0.02
Discount rate (%)	Uniform	0.035	0.045
Uncertainty on exterior wall cost (%)	Uniform	-20	20
Uncertainty on exterior wall GWP[c](%)	Uniform	-20	20
Uncertainty on windows cost (%)	Uniform	-20	20
Uncertainty on windows GWP (%)	Uniform	-20	20
Uncertainty on slab against unheated cost (%)	Uniform	-20	20
Uncertainty on slab against unheated GWP (%)	Uniform	-20	20
Uncertainty on slab against unheated RSL[d](%)	Uniform	-20	20
Heating efficiency loss (%)	Uniform	15	25
Exterior wall conductivity	Gumbel	0.69	0.18
Exterior wall RSL (years)	Lognormal	32	2
Windows RSL (years)	Lognormal	32	2

[a] Corresponds to the lower bound for uniform distributions and to the mean for non uniform distributions.

[b] Corresponds to the upper bound for uniform distributions and to the standard deviation for non uniform distributions.

[c] GWP stands for global warming potential.

[d] RSL stands for reference service life.

shows the detailed values of this selected renovation strategy.

Table 3: Selected renovation strategy from the Pareto front for the building renovation.

Parameter	Value
Heating system	Heat pump
Exterior wall type	#4
Exterior wall insulation thickness (m)	0.4778
Windows type	#9
Slab against unheated type	#5
Slab against unheated insulation thickness (m)	0.4070

Finally, to assess the accuracy of this analysis, we sample a validation set

(a) Convergence for $\mathfrak{c}_1 = LCC$ (b) Convergence for $\mathfrak{c}_2 = LCEI$
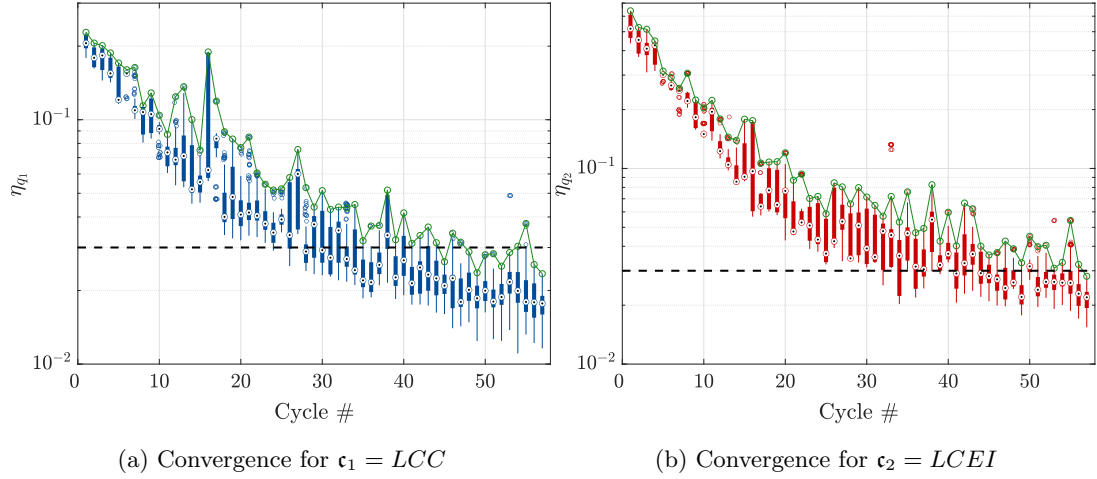
Figure 12: Example 3: Relative quantile error of the entire Pareto front at the end of each cycle.
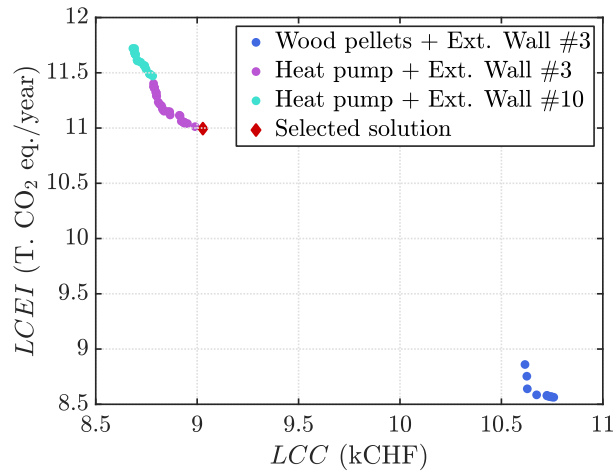
Figure 13: Example 3: Pareto front and selected solution.

$\mathcal{C}_{\mathrm{val}} = \left\{ \left(\boldsymbol{d}^*, \boldsymbol{z}^{(i)} \right), i = 1, \ldots, 500 \right\}$, where $\boldsymbol{d}^*$ is the selected solution. Figure 14 compares the prediction by the final surrogate model and the original response on this validation set. As expected, the surrogate model is able to correctly approximate the original model around the chosen solution. This is confirmed by comparing the normalized mean-square error (NMSE) and the 90%-quantile shown in Table 4. Even though the Monte Carlo sample set size is reduced, the estimated quantiles allow us to have an idea of how accurate the surrogate models are. More specifically the relative errors of the quantiles of LCC and $LCEI$ due to the introduction of the surrogates are approximately 0.2% and 0.4%, respectively. The NMSE for $LCEI$ is slightly larger compared to that of LCC, which is consistent with the convergence history in Figure 12. This value could be reduced by selecting a smaller value of $\bar{\eta}_q$. This would however lead to a larger computational cost and analysis run time.

26

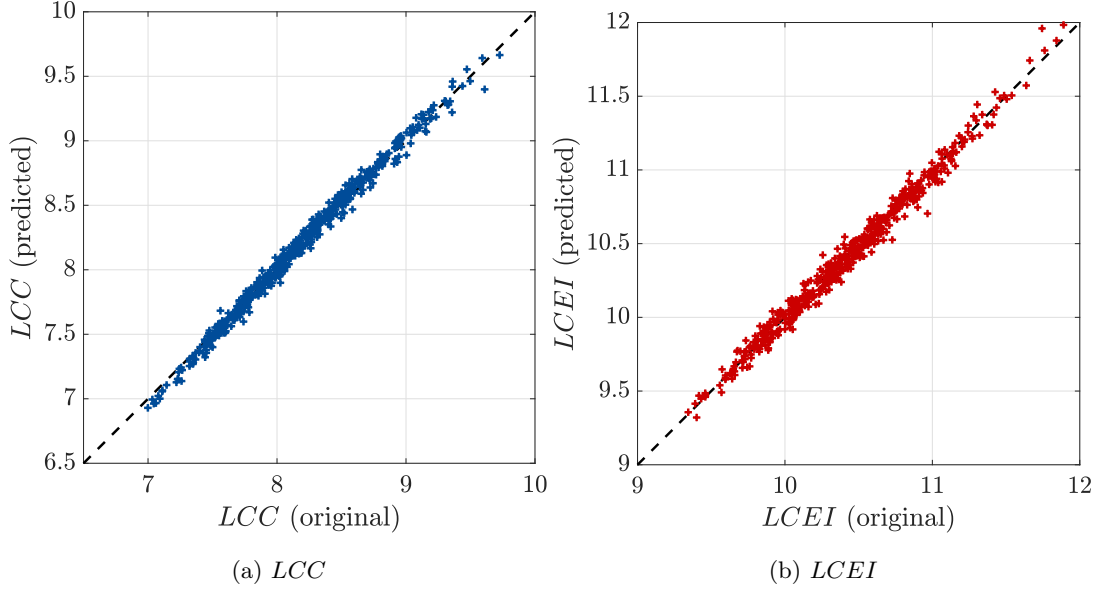|	(a) *LCC*	(b) *LCEI*

Figure 14: Example 3: Original *vs.* predicted responses for a subset of the Monte Carlo set used to compute the quantiles at the selected solution.

Table 4: Validation of the selected solution.

	LCC (kCHF)		*LCEI* (T. CO$_2$ eq. / a)	
Parameter	Original	Surrogate	Original	Surrogate
Quantile	8.9386	8.9530	11.0599	11.0183
NMSE		0.0095		0.0146

7   Conclusion

In this paper, we proposed a methodology for the solution of multi-objective robust optimization problems involving categorical and continuous variables. The problem was formulated using quantiles as a measure of robustness, the level of which can be set to control the desired degree of robustness of the solution. A nested solution scheme was devised, where optimization is carried out in the outer loop while uncertainty propagation is performed in the inner loop. This however results in a stochastic optimization problem which may be cumbersome to solve. The concept of common random numbers was introduced to approximate this stochastic problem by a deterministic one, which is much easier to solve. This allows us then to use a general-purpose multi-objective solver, namely the non-dominated sorting genetic algorithm II (NSGA-II).

To reduce the computational burden of this nested solution scheme, Kriging was introduced in the proposed framework using an adaptive sampling scheme. This is enabled by checking the accuracy of the quantile estimates in areas of interest, namely the area around the Pareto front. Finally, the proposed approach was adapted to account for mixed

categorical-continuous parameters. Two validation examples were built analytically while considering all the characteristics of the problem at hand.

The methodology was then applied to the optimization of renovation scenarios for a building considering uncertainties in its entire life cycle post-renovation. This has shown that a driving parameter for such a renovation is the replacement of the heating system by either wood pellets or heat pump. This result was validated by running the original model around the selected solution, hence showing that the final surrogate model was accurate. This methodology was eventually applied in a detailed analysis presented in Galimshina et al. (2021).

8 Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

9 Replication of results

The results presented in this paper were obtained using an implementation of the proposed methodology within UQLab (www.uqlab.com) (Marelli and Sudret, 2014), a MATLAB-based framework for uncertainty quantification. The codes can be made available upon request.

References

Au, S.-K. (2005). Reliability-based design sensitivity by efficient simulation. *Comput. Struct. 83*(14), 1048–1061.

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecifications. *Comput. Stat. Data Anal. 66*, 55–69.

Beck, A. T., W. J. S. Gomes, R. H. Lopez, and L. F. F. Miguel (2015). A comparison between robust and risk-based optimization under uncertainty. *Struct. Multidisc. Optim. 52*, 479–492.

Beyer, H.-G. and B. Sendhoff (2007). Robust optimization - A comprehensive survey. *Comput. Methods Appl. Mech. Engrg. 196*, 3190–3218.

Bichon, B. J., M. S. Eldred, L. Swiler, S. Mahadevan, and J. McFarland (2008). Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal 46*(10), 2459–2468.

Binh, T. T. and U. Korn (1997). MOBES: A multiobjective evolution strategy for constrained

optimization problems. In J. D. Schaffer (Ed.), *Proc. 3rd international conference on genetic algorithms (MENDEL97)*.

Chatterjee, T., S. Chakraborty, and R. Chowdhury (2019). A critical review of surrogate assisted robust design optimization. *Arch. Comput. Method. E. 26*(1), 245–274.

Couckuyt, I., D. Deschrijver, and T. Dhaene (2014). Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *J. Glob. Optim. 60*, 575–594.

De Weck, O. L. (2014). Multiobjective optimization: History and premise. In *Proc. 3rd China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Kanazawa, Japan, Oct. 30th- Nov. 12th*.

Deb, K. and S. Agrawal (1999). A niched-penalty approach for constraint handling in genetic algorithms. In A. Dobnikar, N. Steele, and R. F. Pearson, D. W. andAlbrecht (Eds.), *Proc. Artificial neural nets and Genetic Algorithms, Portoroz, Slovenia, 1999*.

Deb, K., A. Pratap, S. Agrawal, and T. Meyarivan (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput. 6*, 182–197.

Díaz-Manríquez, A., G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal (2016). A review of surrogate assisted multiobjective evolutionary algorithms. *Comput. Intel. Neurosci. 2016*, 1–14.

Doltsinis, I. and Z. Kang (2004). Robust design of structures using optimization methods. *Comput. Method. Appl. M. 193*, 2221–2237.

Dong, H. and M. N. Nakayama (2017). Quantile estimation using conditional Monte Carlo and Latin Hypercube Sampling. In W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page (Eds.), *Proc. 2017 Winter Simulation Conference (WSC), December 3-7, 2017, Las Vegas, NV, USA*.

Dubourg, V., B. Sudret, and J.-M. Bourinet (2011). Reliability-based design optimization using Kriging and subset simulation. *Struct. Multidisc. Optim. 44*(5), 673–690.

Ehrgott, M. (2005). *Multicriteria optimization*. Springer Verlag - Berlin Heidelberg.

Emmerich, M. T. M. and A. H. Deutz (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Nat. Comput. 17*, 369–395.

Emmerich, M. T. M., A. H. Deutz, and J. W. Klinkerberg (2011). Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Proc. IEEE Congress on Evolutionary Computation, CE C2011, New Orleans, LA, USA, 5-8 June, 2011*. IEEE.

Emmerich, M. T. M., K. C. Giannakoglou, and B. Naijoks (2006). Single- and multiobjective evolutionary optimization assisted by Gaussian random fields metamodels. *IEEE Transactions on Evolutionary Computation 10*(4), 421–439.

European Commission (2020). A renovation wave for Europe - greening our buildings, creating jobs, improving lives. Commnication from the Commission of the European Parliament, the Council, the European Economic and Social Committee and the Committee of the regions.

Fonseca, C. and C. Flemming (1993). Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In S. Forrest (Ed.), *Proc. 5th Int. Conf. on Genetic Algorithms, July 17-21, 1993, Urbana-Champaign, Illinois, USA*.

Galimshina, A., M. Moustapha, A. Hollberg, P. Padey, S. Lasvaux, B. Sudret, and G. Habert (2020). Statistical method to identify robust building renovation choices for environmental and economic performance. *Build. Environ. 183*, 107143.

Galimshina, A., M. Moustapha, A. Hollberg, P. Padey, S. Lasvaux, B. Sudret, and G. Habert (2021). What is the optimal robust environmental and cost-effective solution for building renovation? not the usual one. *Energy and Buildings 251*(111329), 111329.

Gaudrie, D., R. Le Riche, V. Picheny, B. Enaux, and V. Herbert (2020). Targeting solutions in Bayesian multi-objective optimization: sequential and batch versions. *Ann. Math. Artif. Intel. 88*, 187–212.

Glynn, P. W. (1996). Importance sampling for Monte Carlo estimation of quantiles. In W. de Gruyter (Ed.), *Proc. 2nd Internation workshop on mathematical methods in stochastic simulation and experimental design, St. Petersburg, Russia, June 18-22, 1996*.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley Longman Publishing Co., Inc.

Halstrup, M. (2016). *Black-box optimization of mixed discrete-continuous optimization problems.* Ph. D. thesis, Technische Universität Dortmund.

Hupkens, I., A. H. Deutz, and M. Emmerich (2015). Faster exact algorithms for computing expected hypervolume improvement. In A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. (Eds.), *Proc. Evolutionary Multi-criterion Optimization (EMO 2015), Guimarães, Portugal, March 29th - April 1st, 2015*.

Jones, D. R., M. Schonlau, and W. J. Welch (1998). Efficient global optimization of expensive black-box functions. *J. Global Optim. 13*(4), 455–492.

Keane, A. J. (2006). Statistical improvement criteria for use in multiobjective design optimization. *AIAA Journal 44*, 879–891.

Kharmanda, G., A. Mohamed, and M. Lemaire (2002). Efficient reliability-based design optimization using a hybrid space with application to finite element analysis. *Struct. Multidisc. Optim. 24* (3), 233–245.

Knowles, J. (2005). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation 10*, 50–66.

Knowles, J. D. and D. W. Corne (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation 8*, 149–172.

Lataniotis, C., S. Marelli, and B. Sudret (2018). The Gaussian process modeling module in UQLab. *Soft Computing in Civil Engineering 2*(3), 91–116.

Lee, K.-H. and G.-J. Park (2006). A global robust optimization using Kriging based approximation model. *JSME Int. J. 49*, 779–788.

Lelievre, N., P. Beaurepaire, C. Mattrand, N. Gayton, and A. Otsmane (2016). On the consideration of uncertainty in design: optimization - reliability - robustness. *Struct. Multidisc. Optim. 54*, 1423–1437.

Liang, C. and S. Mahadevan (2017). Pareto surface construction for multi-objective optimization under uncertainty. *Struct. Multidisc. Optim. 55*, 1865–1882.

Manson, J. A., T. W. Chamberlain, and R. A. Bourne (2021). MVMOO: Mixed variable multi-objective optimisation. *J. Global Optim. 80*, 865 – 886.

Marelli, S. and B. Sudret (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, pp. 2554–2563.

Marler, R. T. and J. S. Arora (2014). Survey of multi-objective optimization methods for engineering. *Struct. Multidisc. Optim. 26*, 369–395.

McGil, R., J. W. Tukey, and W. A. Larsen (1978). Variations of boxplots. *The American Statsitician 32*, 12–16.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 2*, 239–245.

Miettinen, K. M. (1999). *Nonlinear multiobjective optimization*. Springer Science+ Business media, LLC.

Moustapha, M. and B. Sudret (2019). Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework. *Struct. Multidisc. Optim. 60*, 2157–2176.

Moustapha, M., B. Sudret, J.-M. Bourinet, and B. Guillaume (2016). Quantile-based optimization under uncertainties using adaptive Kriging surrogate models. *Struct. Multidisc. Optim. 54* (6), 1403–1421.

Pelamatti, J., L. Brevault, M. Balestdent, E.-G. Talbi, and Y. Guerin (2020). Bayesian optimization of variable-size design space problems. *Optim. Eng. 22*, 387–447.

Picheny, V. (2015). Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Stat. Comput. 25*, 1265–1289.

Pujol, G., R. Le Riche, X. Bay, and O. Roustant (2009). Minimisation de quantiles - application en mécanique. In *9e colloque national en calcul des structures, CSMA, 25-29 Mai 2009, Giens, France.* Association Calcul des Structures et Modélisation - Association Français de Mécanique.

Qian, P. Z. G., H. Wu, and J. C. F. Wu (2008). Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics 50*, 383–396.

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian processes for machine learning* (Internet ed.). Adaptive computation and machine learning. Cambridge, Massachusetts: MIT Press.

Razaaly, N., G. Persico, G. Gori, and P. M. Congedo (2020). Quantile-based robust optimization of a supersonic nozzle for organic rankine cycle turbines. *App. Math. Model. 82*, 802–824.

Ribaud, M., C. Blanchet-Scalliet, C. Herbert, and F. Gillot (2020). Robust optimization: A Kriging-based multi-objective optimization approach. *Reliab. Eng. Sys. Saf. 200*, 106913.

Roustant, O., E. Padonou, Y. Deville, A. Clément, G. Perrin, J. Giorla, and H. Wynn (2020). Group kernels for Gaussian process metamodels with categorical inputs. *SIAM/ASA J. Uncertain. 8*, 775–806.

Sabater, C., O. Le Maitre, P. M. Congedo, and S. Görtz (2021). A Bayesian approach for quantile optimization problems with high-dimensional uncertainty sources. *Comput. Methods Appl. Mech. Engrg. 376*, 113632.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). Design and analysis of computer experiments. *Stat. Sci. 4*, 409–435.

Santner, T. J., B. J. Williams, and W. I. Notz (2003). *The Design and Analysis of Computer Experiments.* Springer, New York.

Schaffer, J. D. (1985). Multi-objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette (Ed.), *Proc. 1st Int. Conf. on Genetic Algorithms and their applications, July 24-26, 1985, Pittsburg, PA, USA.*

Schonlau, M., W. J. Welch, and D. R. Jones (1998). Global versus local search in constrained optimization of computer codes. *Institute of Mathematical Statistics Lectures Notes - Monograph series 34,* 11–25.

Schuëller, G. I. and H. A. Jensen (2008). Computational methods in optimization considering uncertainties - An overview. *Comput. Methods Appl. Mech. Engrg. 198,* 2–13.

Shu, L., P. Jiang, X. Xao, and Y. Wang (2021). A new multi-objective Bayesian optimization formulation with the acquisition function for convergence and diversity. *J. Mech. Design 142,* 091703:1–10.

Spall, J. C. (2003). *Introduction to stochastic search and optimization: Estimation, simulation and control.* John Wiley & Sons.

Svenson, J. D. and T. J. Santner (2010). Multiobjective optimization of expensive black-box functions via expected maximin improvement. *Optimization and engineering 64,* 17–32.

Taflanidis, A. A. and J. L. Beck (2008). Stochastic subset optimization for optimal reliability problems. *Prob. Eng. Mech 23,* 324–338.

Taflanidis, A. J. and A. C. Medina (2014). Adaptive Kriging for simulation-based design under uncertainty: Development of metamodels in augmeted input space and adaptive tuning of their characteristics. In *Proc. 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications, August 28-30, 2014, Vienna, Austria.*

Taguchi, G. and M. Phadke (1989). Quality engineering through design optimization. In K. Dehnad (Ed.), *Quality Control, Robust Design, and the Taguchi Method,* pp. 77–96. Springer.

Tran, A., M. Tran, and Y. Wang (2019). Constrained mixed-integer Gaussian mixture Bayesian optimization and its applications in designing fractal and auxetic metamaterials. *Struct. Multidisc. Opt. 59,* 2131–2154.

Umbarkar, A. J. and P. D. Sheth (2015). Crossover operators in genetic algorithms: a review. *ICTAT J. Soft Comput. 6,* 1083–1092.

Wang, L., S. Tao, P. Zhu, and W. Chen (2021). Data-driven topology optimization with multiclass microstructures using latent variable Gaussian process. *J. Mech. Design 143,* 031708:1–13.

Yang, K., M. Emmerich, A. Deutz, and T. Bäck (2019). Efficient computation of expected hypervolume improvement using box decomposition algorithms. *J. Glob. Optim. 75,* 3–34.

Zang, C., M. I. Friswell, and J. E. Mottershead (2005). A review of robust optimal design and its applications in dynamics. *Comput. Struct. 83*, 315–326.

Zhang, J. and A. A. Taflanidis (2019). Multi-objective optimization for design under uncertainty problems through surrogate modeling in augmented input space. *Struct. Multidisc. Optim. 59*, 351–372.

Zhang, Q., W. Liu, E. Tsang, and B. Virginas (2010). Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation 14*, 456–474.

Zhang, Y., S. Tao, W. Chen, and D. W. Apley (2020). A latent variable approach to gaussian process modeling with qualitative and quantitative factors. *Technometrics 62*(3), 291–302.

Zhou, Q., P. Z. G. Qian, and S. Zhou (2011). A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics 53*, 266–273.

Zitzler, E., M. Laumanns, and S. Bleuler (2004). *A tutorial on evolutionary multiobjective optimization*, Volume 535, pp. 3–37. Springer Verlag.

Zitzler, E. and L. Thiele (1999). Multiobjective evolutionary algorithms: A comparative study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation 3*, 257–271.

Zitzler, E. and L. Thiele (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput. 8*, 173–195.

A   Non-dominated sorting genetic algorithm II

Generally speaking, genetic algorithms (Goldberg, 1989) are a sub-class of evolutionary methods where a population of individuals is evolved so as to produce better and better solutions until convergence. Genetic algorithms attempt to mimic natural evolution theory where individuals are combined through so-called *cross-over* and *mutation* operators, the idea being that the fittest individuals would pass their genes to the next generation. The non-dominated sorting genetic algorithm II (NSGA-II) belongs to a class of algorithms developed specifically to handle multi-objective optimization problems. The latter introduce a new approach for assigning fitness according to a rank derived from a non-dominated sorting. NSGA-II is an elitist and fast non-dominated sorting algorithm introduced by Deb et al. (2002) as an enhancement of the original NSGA (Deb and Agrawal, 1999). Even though NSGA has shown to be a popular evolutionary method, it features a computationally complex and slow sorting algorithm and, most of all, lacks elitism. Elitism, a property by which the fittest solutions in a given generation are copied to the next one so as not to lose the best genes,

has indeed proved to accelerate convergence (Zitzler and Thiele, 2000). The general workflow of NSGA-II is illustrated in Figure 15. The following section details the important steps of NSGA-II as described in Algorithm 1:
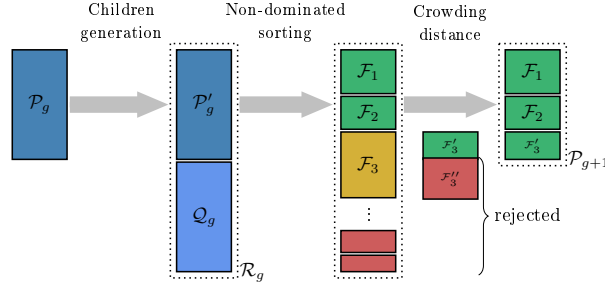

Figure 15: Flowchart of NSGA-II (adapted from Deb et al. (2002)).

1. **Children generation:** Following the initialization where the population $\mathcal{P}_g$ is sampled, the first step in Algorithm 1 is binary tournament selection, which is used to pre-select good individuals for reproduction. The idea is to form a new population $\mathcal{P}'_g$ by selecting the best among two (or more) randomly selected individuals. The new population is then of the same size as the original but comprises repetitions. Individuals in this population constitute the basis for the generation of children using cross-over and mutation operators. The former consists in combining two parents to generate two offsprings. There exists a wide variety of cross-over operators (Umbarkar and Sheth, 2015) and NSGA-II uses the so-called *simulated binary cross-over* (SBX). Mutation is used to prevent premature convergence to a local minimum. Following a predefined probability, a new offspring is mutated, *i.e.*, some of its components are randomly perturbed, thus allowing for more exploration of the design space. The *polynomial mutation* operator has been developed for NSGA-II. The resulting set of offsprings is denoted by $\mathcal{Q}_g$.

2. **Non-dominated sorting:** Once the new population $\mathcal{R}_g$ is generated by assembling $\mathcal{P}'_g$ and $\mathcal{Q}_g$, it is subjected to a recursive non-dominated sorting. This means that the set of all non-dominated individuals are identified and assigned rank one. This set, denoted by $\mathcal{F}_1$, is then set aside and non-dominated sorting is again performed on $\mathcal{R}_g \setminus \mathcal{F}_1$ to find the front of rank two $\mathcal{F}_2$. The process is repeated until all individuals have been assigned a rank. The population for the next generation is then composed of the individuals of the first ranks whose cumulated cardinality is smaller than L.

3. **Crowding distance assignment:** To reach L offsprings, it is necessary to select only a subset of the population in the following front. To do so, the crowding distance, which measures the distance of a given solution to the closest individuals in its rank, is used. Diversity in the Pareto front is hence enforced by choosing for the same rank solutions with the largest crowding distance. Once the remaining solutions are chosen according to their crowding distance ranking, the new population $\mathcal{P}_{g+1}$ is defined.

Algorithm 1 Non-dominated sorting genetic algorithm II (Deb et al., 2002)

Initialization:

    Sample initial population $\mathcal{P}_0$ of size L $\triangleright$ *e.g.,* $L = 100$

    Set maximum number of iterations $G_{\max}$ $\triangleright$ *e.g.,* $G_{\max} = 100$

    Set $g = 0$; `NotConverged = true`,

1: **while** `NotConverged == true` **do**

2:     Perform tournament selection: $\mathcal{P}'_g \leftarrow \mathcal{P}_g$ $\triangleright$ $\mathcal{P}'_g$ is also of size L

3:     Generate offsprings Q_g $\triangleright$ using cross-over and mutation operators

4:     Set $\mathcal{R}_g = \mathcal{P}'_g \cup \mathcal{Q}_g$ $\triangleright$ Combine parents and offspring

5:     Perform non-dominated sorting $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, ...\}$

6:     Set $\mathcal{P}_{g+1} = \emptyset, \quad i = 1$

7:     **while** $\mathrm{Card}\left(\mathcal{P}_{g+1}\right) < L$ **do**

8:         $\mathcal{P}_{g+1} = \mathcal{P}_{g+1} \cup \mathcal{F}_i$

9:         $i \leftarrow i + 1$

10:     **end while**

11:     Sort $\mathcal{F}_i$ $\triangleright$ w.r.t. the individuals crowding distance

12:     Set $\mathcal{F}'_i = \texttt{sorted}(\mathcal{F}_i)\left[1 : L - \mathrm{Card}\left(\mathcal{P}_{g+1}\right)\right]$ $\triangleright$ Select the $L - \mathrm{Card}\left(\mathcal{P}_{g+1}\right)$ best elements in $\mathcal{F}_i$

13:     Set $\mathcal{P}_{g+1} = \mathcal{P}_{g+1} \cup \mathcal{F}'_i$

14:     $g \leftarrow g + 1$

15:     **if** Convergence is achieved or $g = G_{\max}$ **then**

16:         `NotConverged = false`

17:     **end if**

18: **end while**

    This entire process is repeated until convergence is reached.