Structural and Multidisciplinary Optimization (2019) 59:351–372
https://doi.org/10.1007/s00158-018-2069-1

RESEARCH PAPER

Multi-objective optimization for design under uncertainty problems through surrogate modeling in augmented input space

J. Zhang[1] · A. A. Taflanidis[1,2]

Received: 14 September 2017 / Revised: 29 July 2018 / Accepted: 9 August 2018 / Published online: 18 September 2018

Abstract

Multi-objective design under uncertainty problems that adopt probabilistic quantities as performance objectives and consider their estimation through stochastic simulation are examined in this paper, focusing on development of a surrogate modeling framework to reduce computational burden for the numerical optimization. The surrogate model is formulated to approximate the system response with respect to both the design variables and the uncertain model parameters, so that it can simultaneously support both the uncertainty propagation and the identification of the Pareto optimal solutions. Kriging is chosen as the metamodel, and its probabilistic nature (its ability to offer a local estimate of the prediction error) is leveraged within different aspects of the framework. To reduce the number of simulations for the expensive system model, an iterative approach is established with adaptive characteristics for controlling the metamodel accuracy. At each iteration, a new metamodel is developed utilizing all available training points. A new Pareto front is then identified utilizing this surrogate model and is compared, for assessing stopping criteria, to the front that was identified in the previous iteration. This comparison utilizes explicitly the potential error associated with the metamodel predictions. If stopping criteria are not achieved, a set of refinement experiments (new training points) is identified and process proceeds to the next iteration. A hybrid design of experiments is considered for this refinement, with a dual goal of global coverage and local exploitation of regions of interest, separately identified for the design variables and the uncertain model parameters.

Keywords Design under uncertainty · Multi-objective design · Augmented input space · Adaptive metamodels · Kriging · Iterative optimization

1 Introduction

Design optimization plays an important role in analysis of engineering systems, aiming to identify the optimal design configuration for some chosen performance objectives and constraints. For engineering applications that involve conflicting and incommensurable objectives, this design approach is frequently formulated as a multi-objective problem (Marler and Arora 2004), and its solution identifies a set of Pareto optimal design configurations representing different trade-offs for these objectives. Such optimization problems pose always greater computational challenges than single-objective ones, with a requirement to thoroughly explore the design space in order to comprehensively assess performance trade-offs. Their complexity increases when modeling uncertainties are also addressed in the problem formulation. Such uncertainties, stemming from the incomplete knowledge about the examined system and its environment (Helton et al. 2004; Beck and Taflanidis 2013), impact performance predictions and therefore the design process. A rational framework to explicitly consider them in this process is a probability logic approach (Beck and Taflanidis 2013), employed by assigning probability distributions to the uncertain model characteristics. The design objectives are then expressed through the expectation of some chosen performance measure (Beck and Santana Gomes 2012; Papadimitriou and Papadimitriou 2016). Multi-objective design problems within this context are commonly formulated as applications of robust design optimization (RDO) (Lagaros and Papadrakakis 2007; Ren and Rahman 2013; Medina and Taflanidis 2015), with the conflicting

Responsible Editor: Junji Kato.

✉ A. A. Taflanidis
a.taflanidis@nd.edu

[1] Department of Civil & Environmental Engineering & Earth Sciences, University of Notre Dame, Notre Dame, USA

[2] Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, USA

objectives related to the mean and variance of the (same) system response. More general approaches, though, exist that examine broader performance quantifications, adopting objectives that relate to different response outputs and to advanced probabilistic descriptions for performance (Wang and Stengel 2000; Verros et al. 2005; Gidaris et al. 2016; Liang and Mahadevan 2017). This paper discusses such multi-objective design under uncertainty problems that employ general probabilistic quantities as design objectives (no specialized structure assumed), and focuses on applications that involve complex numerical models. For such applications, stochastic (i.e., Monte Carlo) simulation is the only applicable generalized method for calculating the probabilistic performances. Unfortunately, this approach has an associated high computational cost, requiring a large number of calls to the deterministic, computationally intensive simulation model.

A framework relying on surrogate modeling to approximate the system response is discussed here to alleviate this burden. Within the aforementioned multi-objective design under uncertainty setting, traditional approaches for incorporating metamodels can be distinguished in two general categories. In the first category, metamodels are formulated in the design optimization level, by approximating the objective function (estimated utilizing the exact time-consuming system model) in the design space (Ruiz et al. 2016; Leotardi et al. 2016). In the second category, metamodels are exploited at the uncertainty quantification level, by approximating the system performance in the random variable space for each design configuration the optimizer examines (Poles and Lovison 2009). Approaches that combine these two formulations also exist (Coelho et al. 2011), combining separate metamodels for an outer loop (optimization in the design space) and an inner loop (uncertainty quantification in the random variables space). Here, an alternative approach is adopted that considers the formulation of a single metamodel in the so-called augmented input space of the probabilistic design problem, composed of both the design variables and the random variables. The metamodel then simultaneously supports both the uncertainty propagation and the design optimization. Such a formulation has been examined primarily for single-objective applications, in a variety of settings with respect to performance objective definition. Approaches based on Kriging metamodeling have been extensively investigated (Dubourg et al. 2011; Bichon et al. 2013; Moustapha et al. 2016) for problems where the probability of response exceeding certain threshold or response's specific quantile is of interest, as often encountered in reliability-based design optimization (RBDO). Applications dealing directly with the expected value of the response also exist, relying on Kriging (Janusevskis and Le Riche 2013) or stochastic collocation (Kouri et al. 2014). In Zhang et al. (2016), and using again Kriging as metamodel, a unified implementation was discussed (accommodating different statistical measures) through the introduction of a performance function related to the system response. Recently (Dubreuil et al. 2018), a metamodeling implementation combining polynomial chaos expansion and Kriging was examined for optimization of the extreme value of the response, investigating the performance under any possible system realization considering the uncertainty in the model description. All these single-objective applications have incorporated some approach to adaptively control the metamodel accuracy, focusing primarily on critical, near-optimum regions. In contrast, for multi-objective problems, the augmented input space formulation has been adopted simply as a direct implementation of the aforementioned ideas (Gidaris et al. 2016), with no effort to adaptively control the metamodel accuracy.

This work establishes a complete framework for adaptive implementation of surrogate modeling in the augmented input space for multi-objective design under uncertainty problems, and formally establishes the MODU-AIM (multi-objective design under uncertainty with augmented input metamodels) algorithm. Kriging (Sacks et al. 1989; Kleijnen 2009) is adopted as metamodel since it has been proven highly efficient for approximating complex functions, while simultaneously being able to provide a local approximation to the metamodel prediction error (Jin et al. 2003). The main novel contribution of this work is the development of an iterative optimization scheme that further reduces the number of calls to the time-consuming simulation model. The underlying principle of leveraging an iterative formulation to reduce computational burden is the same as the approach advocated in Zhang et al. (2016), though the implementation here for multi-objective problems is [and needs to be (Yang et al. 2002)] fundamentally different. A global metamodel covering the entire design domain is proposed and iteratively refined using information for the metamodel accuracy and the updated (at each iteration) Pareto front. The design of experiments (DoE) for this refinement and accuracy/convergence criteria for the iterative process are based on global comparisons, as opposed to the local, trust-region-based approach developed in Zhang et al. (2016). The probabilistic nature of the Kriging metamodel is explicitly leveraged in multiple aspects of the proposed framework: in the modification of the probabilistic objectives, in the DoE, and in the stopping criteria for assessing convergence.

In the next section, the multi-objective design under uncertainty problem is formulated, and in Section 3, the modification of the problem through the metamodel implementation is discussed. In Section 4, the iterative optimization scheme is presented, and in Section 5, the overall framework is reviewed, before discussing the illustrative applications in Section 6.

2 Problem formulation

Consider an engineering system with design vector $\mathbf{x} \in X \subset \Re^{n_x}$, where X is the admissible design space, and uncertain parameters (i.e., random variables) $\theta \in \Theta \subset \Re^{n_\theta}$ with probability density function (PDF) $p(\theta)$ (facilitating ultimately the uncertainty quantification). Let $\mathbf{z}(\mathbf{x}, \theta) \subset \Re^{n_z}$ denote the *response vector* of the system model (with z_m representing its mth component), assumed here to be obtained through a time-consuming call to a deterministic numerical function (simulator), and let $h_i(\mathbf{x}, \theta) : \Re^{n_x \times n_\theta} \rightarrow \Re$ denote the *performance measure* used for quantification of the ith design objective. The n_h-dimensional performance measure vector, including all such measures, will be denoted $\mathbf{h}$. The notation $\mathbf{h}[\mathbf{z}|\, \mathbf{x},\theta]$ will be also used herein to denote the performance measure vector; this notation indicates explicitly that this measure is a function of the response vector $\mathbf{z}$ but assumes also knowledge of design variables $\mathbf{x}$ and model parameters θ.

The ith performance function (i.e., design objective) is defined as

$$H_i(\mathbf{x}) = \mathbb{E}_{p(\theta)}[h_i(\mathbf{x}, \theta)] = \int_\Theta h_i(\mathbf{x}, \theta) p(\theta) d\theta \tag{1}$$

where $\mathbb{E}_{p(\theta)}[.]$ denotes the expectation under probability model $p(\theta)$. The probabilistic *performance function vector* $\mathbf{H}(\mathbf{x})$ is defined as the vector with components $H_i(\mathbf{x})$. The multi-objective design under uncertainty problem corresponds to the identification of the feasible design vector that minimizes $\mathbf{H}(\mathbf{x})$:

$$\mathbf{x}^* = \underset{\mathbf{x} \in X}{\operatorname{argmin}}\ \mathbf{H}(\mathbf{x}) \tag{2}$$

Note that in this representation of the design problem, any deterministic constraints are assumed to be incorporated in the definition of admissible design space X.

For the multi-objective problem in (2), a design configuration $\mathbf{x}$ is defined as *dominating* (*strongly*) any other $\mathbf{x}'$ if all components in its performance function vector are lower. If objectives are competing, there is no single design configuration that dominates all others. In such applications, design optimization of (2) is transformed to identification of the *Pareto optimal* solutions, corresponding to designs that are not dominated by any other feasible ones. The set of all such configurations is denoted as the *Pareto set* $\mathbf{X}_p$ with its elements denoted as $\mathbf{x}_p$. The *Pareto front* is the performance function space representation of the Pareto set $\mathbf{H}_p = \{\mathbf{H}(\mathbf{x})|\, \mathbf{x} \in \mathbf{X}_p\}$. It is generally impractical to find all Pareto optimums so the optimization strategies usually aim at finding a subset of them that represents $\mathbf{H}_p$ well and can provide the decision maker with a comprehensive picture of trade-offs (Zitzler et al. 2000).

For identifying the Pareto set, the probabilistic objectives need to be calculated, and as discussed in Section 1, this is established here through stochastic simulation. Using N_i samples from some importance sampling (IS) density $q_i(\theta)$, introduced for improving computational efficiency by concentrating the stochastic simulation effort in domains of importance for the uncertain model parameters (Robert and Casella 2004), the integral in (1) is estimated as

$$\hat{H}_i(\mathbf{x}|\{\theta\}_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} h_i(\mathbf{x}, \theta_i^j) \frac{p(\theta_i^j)}{q_i(\theta_i^j)} \tag{3}$$

where $\{\theta\}_i = \{\theta_i^j, j = 1, \ldots, N_i\}$ is the sample set from $q_i(\theta)$ and the notation $\hat{H}_i(\mathbf{x}|\{\theta\}_i)$ stresses that this estimate is dependent on the sample set utilized. Note that the IS density $q_i(\theta)$ and number of samples N_i can be different for each objective. Should be pointed out that selection of an efficient IS density can be challenging (Medina and Taflanidis 2014), especially in the context of multi-objective optimization where the domains of importance for θ might be drastically different across the different Pareto optimal solutions. Identifying and properly combining information from these domains is not straightforward. In the context of the proposed Kriging-aided optimization framework, this will be further discussed in Section 5.1.

By combining all component estimates $\hat{H}_i(\mathbf{x}|\{\theta\}_i)$ obtained through (3), the estimate for the probabilistic objective vector $\hat{\mathbf{H}}(\mathbf{x}|\{\theta\})$ is obtained, and the multi-objective design under uncertainty problem can be then solved by substituting $\mathbf{H}\,(\mathbf{x})$ with that estimate. The resultant optimization problem is computationally challenging to solve, primarily due to the high computational cost associated with each performance function vector evaluation and the existence of an unavoidable stochastic simulation estimation error. To address these challenges, a metamodeling-aided approach in augmented input space is formulated in this paper.

3 Kriging-aided multi-objective optimization in augmented input space

3.1 Kriging-aided optimization

The metamodel output is chosen to correspond to the response vector $\mathbf{z}$, rather than the performance measure vector $\mathbf{h}$. The advantages of this choice are discussed in detail in Zhang et al. (2016). With respect to the metamodel input, this is chosen to encompass both the design and random variables, leading to definition of augmented input vector as $\mathbf{y} = [\mathbf{x};\ \theta]$. Modifications in this definition for applications that knowledge of all components of vectors $\mathbf{x}$ and θ is not necessary for evaluating $\mathbf{z}$ are discussed in Zhang et al. (2016). As discussed in Section 1, Kriging is adopted as metamodel. The fundamental hypothesis in Kriging is that any component in the response vector z_m is approximated as a realization of a Gaussian process (Sacks et al. 1989; Kleijnen 2009) with some chosen correlation function $R(\mathbf{y},\mathbf{y}')$. For forming the

Kriging metamodel, the output $\{z_m(\mathbf{y}^t),\ t = 1, \ldots, n\}$ is observed at n distinct locations for the input $\{\mathbf{y}^t,\ t = 1, \ldots, n\}$, called *training* (or *support*) *points* or *experiments*. The selection of these locations is called design of experiments (DoE). Ultimately, Kriging provides the following distribution for the Gaussian process approximating z_m based on observations $\mathbf{Y} = [\mathbf{y}^1 \ldots \mathbf{y}^n]^T$ (Sacks et al. 1989)

$$M_m(\mathbf{y}|\mathbf{Y}) \sim N\left(\mu_m(\mathbf{y}), \sigma_m^2(\mathbf{y})\right) \tag{4}$$

where N (a,b) stands for Gaussian distribution with mean a and variance b, whereas $\mu_m(\mathbf{y})$ and $\sigma_m^2(\mathbf{y})$ correspond to the predictive mean and predictive variance, respectively, for the output z_m. Once the Kriging metamodel is established, all estimates for the predictive mean and variance are provided with negligible computational burden (Lophaven et al. 2002; Jia and Taflanidis 2013). For the entire vector $\mathbf{z}$, the approximation is established by combining the different components of the response vector and will be denoted as $M_\mathbf{z}(\mathbf{y}|\ \mathbf{Y})$. The essential parts of this approximation are the predictive mean, $\boldsymbol{\mu}_\mathbf{z}(\mathbf{y})$, and variance, $\boldsymbol{\sigma}_\mathbf{z}^2(\mathbf{y})$, vectors assembled through the components of $\mu_m(\mathbf{y})$ and $\sigma_m^2(\mathbf{y})$, respectively. Further details for the metamodel formulation in this context (augmented input space) may be found in Zhang et al. (2016). For applications whose response vector $\mathbf{z}$ is high-dimensional, developing a separate metamodel for each response component might be impractical. In such instances, alternative approaches should be preferred, for example, developing a single metamodel for the entire vector $\mathbf{z}$ as detailed in Zhang et al. (2016) [single metamodel directly predicts entire $\boldsymbol{\mu}_\mathbf{z}(\mathbf{y})$ and $\boldsymbol{\sigma}_\mathbf{z}^2(\mathbf{y})$ vectors] or combining dimension reduction techniques (Jia and Taflanidis 2013). Adoption of such approaches avoids any constraints related to the n_z value.

The optimization problem can be now solved by utilizing the metamodel approximation for $\mathbf{z}(\mathbf{y})$ when estimating the probabilistic performances, with metamodel error (and ultimately its probabilistic nature) explicitly incorporated in the formulation (Zhang et al. 2016). This leads to the following *predictive performance measure* h_i^{krig} that approximates h_i

$$\begin{aligned} h_i^{krig}(\mathbf{x}, \boldsymbol{\theta}) &= \mathbb{E}_M[h_i[\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}]|\mathbf{Y}] \\ &= \int_{\Re^{n_z}} h_i[\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}]\phi\left(\frac{\mathbf{z}-\boldsymbol{\mu}_\mathbf{z}(\mathbf{y})}{\boldsymbol{\sigma}_\mathbf{z}(\mathbf{y})}\right)d\mathbf{z} \\ &= h_i^{krig}[\boldsymbol{\mu}_\mathbf{z}(\mathbf{y}), \boldsymbol{\sigma}_\mathbf{z}(\mathbf{y})|\mathbf{x}, \boldsymbol{\theta}] \end{aligned} \tag{5}$$

where $\mathbb{E}_M[.|\mathbf{Y}]$ denotes conditional expectation under $M_\mathbf{z}(\mathbf{y}|\ \mathbf{Y})$ and ϕ corresponds to the standard Gaussian PDF. The integration in (5) is equivalent to propagation of the error associated with the Kriging metamodel and typically can be analytically performed (Zhang et al. 2016). It is assumed herein that $h_i^{krig}(\mathbf{x}, \boldsymbol{\theta}) = h_i^{krig}[\boldsymbol{\mu}_\mathbf{z}(\mathbf{y}), \boldsymbol{\sigma}_\mathbf{z}(\mathbf{y})|\mathbf{x}, \boldsymbol{\theta}]$ is an easy to compute measure and that it is continuous and differentiable with respect to $\boldsymbol{\mu}_\mathbf{z}(\mathbf{y})$ and $\boldsymbol{\sigma}_\mathbf{z}(\mathbf{y})$.

This approach facilitates then the following approximation for H_i, denoted as the *predictive performance function* H_i^{krig}:

$$H_i^{krig}(\mathbf{x}) = \mathbb{E}_{p(\boldsymbol{\theta})}\left[h_i^{krig}(\mathbf{x}, \boldsymbol{\theta})\right] = \int_\Theta h_i^{krig}(\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} \tag{6}$$

which may be estimated though stochastic simulation as:

$$\hat{H}_i^{krig}(\mathbf{x}|\{\boldsymbol{\theta}\}_i) = \frac{1}{N_i}\sum_{j=1}^{N_i}\left(h_i^{krig}(\mathbf{x}, \boldsymbol{\theta}_i^j)\frac{p(\boldsymbol{\theta}_i^j)}{q_i(\boldsymbol{\theta}_i^j)}\right) \tag{7}$$

where, recall, $\{\boldsymbol{\theta}\}_i = \{\boldsymbol{\theta}_i^j, j = 1, \ldots, N_i\}$ is the sample set from $q_i(\boldsymbol{\theta})$. Selection of N and $q_i(\boldsymbol{\theta})$ to establish a target accuracy for the estimate in (7) within the optimization framework will be addressed in Section 5. Evaluation of the gradient of the predictive performance function is discussed in Appendix 1.

3.2 Solution through epsilon-constraint method

For performing the multi-objective optimization, the epsilon-constraint method is adopted because of its ability to identify a Pareto front with a desired, prescribed density, even if the latter has nonconvex regions (Haimes et al. 1971). Herein, we focus on the application to bi-objective problem ($n_h = 2$), though the presented framework can be extended to multi-objective problems with $n_h > 2$, albeit with considerable increase in the computational complexity when combined with the epsilon-constraint approach (Mavrotas 2009). Without loss of generality assuming H_1 as *optimization function* (and H_2 *constraint function*), the epsilon-constraint method converts the multi-objective optimization problem to a set of single-objective constraint optimization problems with different constraint bounds ε^r:

$$\begin{aligned} &\mathbf{x}_p^r = \underset{\mathbf{x}\in X}{\operatorname{argmin}}\ H_1(\mathbf{x}) \\ &\text{such that } H_2(\mathbf{x}) \le \varepsilon^r \end{aligned} \tag{8}$$

where the superscript r is utilized to describe the rth such constraint. Systematic variation of ε^r facilitates identification of the Pareto front.

Utilizing the metamodel-aided approximations for the performance functions (7), we establish the following single-objective constraint optimization problem:

$$\begin{aligned} &\mathbf{x}_p^r = \underset{\mathbf{x}\in X}{\operatorname{argmin}}\ \hat{H}_1^{krig}(\mathbf{x}|\{\boldsymbol{\theta}\}_1) \\ &\text{such that } \hat{H}_2^{krig}(\mathbf{x}|\{\boldsymbol{\theta}\}_2) \le \varepsilon^r \end{aligned} \tag{9}$$

Exploiting the numerical efficiency of the Kriging metamodels and the ability to obtain gradient information (as discussed in Appendix 1), the approximate optimization problem of (9) can be efficiently and accurately solved by any appropriate optimizer, even for a large number of constraints (different values of ε^r).

4 Iterative optimization scheme

The important question is how to obtain a predictive Pareto set through the use of the Kriging metamodel [i.e., optimization of (9)] that approximates the actual Pareto set [i.e., optimization of (8)] well, and perform this with small computational effort (i.e., number of calls to exact numerical models). With no a priori knowledge of regions of importance in Y (especially Pareto optimal regions), a globally accurate metamodel has to be developed. This could be, though, overly conservative and create an excessive computational burden. An iterative approach is established for this purpose, adaptively controlling the metamodel accuracy and identifying regions of importance for efficiently solving the multi-objective problem at hand.

4.1 Fundamentals for the iterative optimization

At each iteration, the metamodel is constructed from the current set of experiments, and the design configurations belonging to the predictive Pareto set are identified and compared to the set identified in the previous iteration. If convergence is not reached yet, the set of experiment is enriched with refinement experiments, a new metamodel is developed, and the optimization proceeds to the next iteration. In this manner, refinement experiments can be selected based on information from previous iterations.

To formalize this approach, let $\mathbf{X}_p^{(k-1)} = \left\{\mathbf{x}_p^{r(k-1)}; r = 1, \ldots, n_p^{(k-1)}\right\}$ be the predictive Pareto set available at the beginning of the kth iteration (this corresponds to the set identified at the end of the k-1st iteration), further denoted as *current* Pareto set. A superscript in parenthesis $^{(k)}$ will be used herein to denote the iteration. The total number of training points for developing the metamodel is $n^{(k)}$ whereas for all stochastic simulations performed within the kth iteration the same sample set $\{\theta\}_i^{(k)}$ is utilized for each of the performance functions. This approach corresponds to the concept of exterior sampling and creates a consistent estimation error in the comparisons within the current iteration (Spall 2003). This set changes, though, across iterations (interior sampling). All evaluations of the performance are established utilizing the current metamodel, leading to approximation $h_i^{krig(k)}(\mathbf{x}, \theta)$ and estimate for the objective $\hat{H}_i^{krig(k)}\left(\mathbf{x}|\{\theta\}_i^{(k)}\right)$ [obtained by (7)]. Using this estimate, the Pareto front can be identified using any appropriate algorithm, potentially using members of the current Pareto set $\mathbf{X}_p^{(k-1)}$ as initial guesses (if chosen algorithm can leverage such information).

At each iteration, the developed framework includes three key tasks: identifying the new (representative subset of the) Pareto set, checking for convergence, and enriching the set of experiments. Details of these tasks are discussed sequentially next.

4.2 Pareto set generation

The Pareto set generation is facilitated through the proper selection of the constraint ε^r. This selection is performed with a dual goal: (i) restrict optimization of (9) in feasible regions; (ii) adequately represent the entire Pareto front by populating evenly its different regions. Figure 1 demonstrates some of these concepts in the context of the second illustrative example considered later.

The first goal is satisfied by identifying first the *anchor points*, i.e., the points representing the minimum for individual objectives. This is established through the unconstrained single-objective optimization:

$$\mathbf{x}_i^{an(k)} = \text{argmin}_{\mathbf{x}\in X}\, \hat{H}_i^{krig}\left(\mathbf{x}|\{\theta\}_i^{(k)}\right);\quad i = 1, 2 \tag{10}$$

These optimizations define the boundaries of current Pareto front and also establish the range for feasible constraints as

$$\varepsilon^r \in \left[\hat{H}_2^{krig}\left(\mathbf{x}_2^{an(k)}|\{\theta\}_2^{(k)}\right), \hat{H}_2^{krig}\left(\mathbf{x}_1^{an(k)}|\{\theta\}_2^{(k)}\right)\right].$$

For the second goal, the remaining solutions (*interior points*) in the Pareto set are generated as following. The performance functions for the design configurations in $\mathbf{X}_p^{(k-1)}$ are first updated using the current metamodel and new sample set $\{\theta\}_i^{(k)}$ (diamonds in Fig.1). Design configurations whose corresponding values for $\hat{H}_2^{krig}$ belong in the feasible constraint range $\left[\hat{H}_2^{krig}\left(\mathbf{x}_2^{an(k)}|\{\theta\}_2^{(k)}\right), \hat{H}_2^{krig}\left(\mathbf{x}_1^{an(k)}|\{\theta\}_2^{(k)}\right)\right]$ are retained and utilized as values ε^r to update the Pareto set in their vicinity. If $\underset{\sim}{\mathbf{X}}_p^{(k-1)} = \left\{\underset{\sim}{\mathbf{x}}_p^{r(k-1)}; r = 1, \ldots, \underset{\sim}{n}_p^{(k-1)}\right\}$ denotes the set of such retained configurations, then this update corresponds to solution of optimization problem (squares in Fig. 1),

$$\begin{aligned}&\mathbf{x}_{in}^{r(k)} = \underset{\mathbf{x}\in X}{\text{argmin}}\ \hat{H}_1^{krig(k)}\left(\mathbf{x}|\{\theta\}_1^{(k)}\right) \text{ s.t.}\\ &\hat{H}_2^{krig}\left(\mathbf{x}|\{\theta\}_2^{(k)}\right) \le \varepsilon^r = \hat{H}_2^{krig(k)}\left(\underset{\sim}{\mathbf{x}}_p^{r(k-1)}|\{\theta\}_2^{(k)}\right);\\ &r = 1, \ldots, \underset{\sim}{n}_p^{(k-1)}\end{aligned} \tag{11}$$

In cases the set $\underset{\sim}{\mathbf{X}}_p^{(k-1)}$ is unavailable (i.e., in the first iteration) or in the extreme cases that none of its design configurations gets retained ($\underset{\sim}{n}_p^{(k-1)} = 0$), the constraint problem in

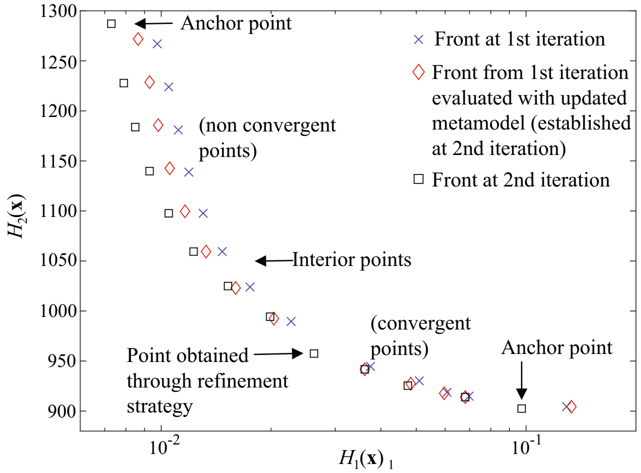


Fig. 1 Evolution of Pareto front in the $k = 2$nd iteration for a sample run of the optimization algorithm (results are for the car suspension design problem discussed later)

(11) can be solved with a pre-defined number (n_{dp}) of equi-spaced constraints across the boundaries of ε^r instead. Combining all interior points, obtained through (11) with the two anchor points obtained through (10), the candidate Pareto set is obtained. Even distribution of the corresponding Pareto front is enforced by an adaptive refinement (Kim and De Weck 2006), removing solutions in over-populated regions of the front, and identifying new solutions in under-populated regions (see also Fig. 1).

4.3 Stopping criteria

As stopping criteria, the discrepancy between the Pareto sets identified in consecutive iterations is utilized, using both convergence and coverage concepts (Deb 2001) to evaluate discrepancy. In early iterations, it is expected that the improvement of the metamodel accuracy will lead to identification of design configurations clearly dominating some precedent ones. As the optimization progresses, and the identified approximate Pareto set approaches the actual Pareto set, it is expected that design configurations identified in subsequent iterations should be non-dominant over each other, and also extend over similar regions in the objective function space. Since it is computationally prohibitive to evaluate the actual performance of the identified solutions, a probabilistic assessment of the performance is adopted (Teich 2001), utilizing the metamodel for the comparison, but explicitly considering its probabilistic nature (error).

To accommodate this, define the *probability of superiority* for the ith objective function $\mathbb{P}_i(\mathbf{x}\succ\mathbf{x}')$ as the probability that the ith performance objective for $\mathbf{x}$ will be lower than the one for $\mathbf{x}'$ considering the uncertainty incorporated through the probabilistic nature of the metamodel. To calculate this probability, each performance function $\{H_i, i = 1, 2\}$ is approximated through the conditional realization $\{H_i^{cr}, i = 1, 2\}$ considering the predictive distribution for the metamodel output $M_{\mathbf{z}}(\mathbf{y}|\,\mathbf{Y})$. The conditional realizations can be given by an expression similar to (6) by substituting $h_i^{krig}(.)$ with $h_i[\mathbf{z}|\,.]$ and using realizations of response $\mathbf{z}$ from distribution $M_{\mathbf{z}}(\mathbf{y}|\,\mathbf{Y})$. The probability $\mathbb{P}_i(\mathbf{x}\succ\mathbf{x}')$ can be finally expressed as:

$$\mathbb{P}_i\left(\mathbf{x}\succ\mathbf{x}'\right) = \int_{\mathfrak{R}^2}\mathbb{I}\left[H_i^{cr}(\mathbf{x})-H_i^{cr}\left(\mathbf{x}'\right)\right]p\left(H_i^{cr}(\mathbf{x}),H_i^{cr}\left(\mathbf{x}'\right)\right)dH_i^{cr}(\mathbf{x})dH_i^{cr}\left(\mathbf{x}'\right) \tag{12}$$

where the indicator function $\mathbb{I}[.]$ is one if the condition within the brackets is satisfied, else it is zero. Evaluation of this two-dimensional integral requires generation of samples for $H_i^{cr}(\mathbf{x})$ and $H_i^{cr}(\mathbf{x}')$. This is established through the approach discussed in Appendix 2 which yields N_{cr} samples for $\left[\hat{H}_i^{cr}(\mathbf{x})\right]^t$ and $\left[\hat{H}_i^{cr}(\mathbf{x}')\right]^t; t = 1, \ldots, N_{cr}$. The probability $\mathbb{P}_i(\mathbf{x}\succ\mathbf{x}')$ is then estimated as

$$\hat{\mathbb{P}}_i\left(\mathbf{x}\succ\mathbf{x}'\right)\approx\frac{1}{N_{cr}}\sum_{t=1}^{N_{cr}}\mathbb{I}\left\{\left[\hat{H}_i^{cr}(\mathbf{x})\right]^t < \left[\hat{H}_i^{cr}\left(\mathbf{x}'\right)\right]^t\right\} \tag{13}$$

Expression $|\mathbb{P}_i(\mathbf{x}\succ\mathbf{x}') - 0.5|$ quantifies proximity of $\mathbf{x}$ and $\mathbf{x}'$ with respect to the ith objective, with a value close to 0 meaning that $\mathbf{x}$ does not exhibit superiority or inferiority to $\mathbf{x}'$. The maximum value for this expression across both objectives is an indicator of closeness in the objective space. This observation leads to definition of the *convergence indicator*, which compares a new point to all members of a set (here, the previous Pareto set $\mathbf{X}_p^{(k-1)}$ is utilized)

$$I_{conv}\left(\mathbf{x}|\mathbf{X}_p^{(k-1)}\right) = \min_{r=1,2,\ldots,n_p^{(k-1)}}\left(\max_{i=1,2}\left|\mathbb{P}_i\left(\mathbf{x}\succ\mathbf{x}_p^{r(k-1)}\right)-0.5\right|\right) \tag{14}$$

A small value for I_{conv} means that $\mathbf{x}$ is close to some member of $\mathbf{X}_p^{(k-1)}$. A large value indicates a significant difference between $\mathbf{x}$ and all members of $\mathbf{X}_p^{(k-1)}$ (concept also illustrated in Fig. 1). As the process approaches convergence, precedent Pareto set $\mathbf{X}_p^{(k-1)}$ should be near Pareto optimal even in current iteration. This means, equivalently, that I_{conv} should not take high values for any point in $\mathbf{X}_p^{(k)}$ which leads to following *convergence stopping criterion*:

$$\forall\mathbf{x}\in\mathbf{X}_p^{(k)}: \quad I_{conv}\left(\mathbf{x}|\mathbf{X}_p^{(k-1)}\right)\leq\delta_{conv} \tag{15}$$

where δ_{conv} is a predetermined threshold.

For evaluating coverage of the Pareto front, a secondary criterion is needed that compares performance to the anchor points of the previous front. The *coverage indicator* is introduced for this reason, defined as

$$I_{cov}\left(\mathbf{x}|\left\{\mathbf{x}_{an,i}^{(k-1)}\right\}_{i=1,2}\right)=\max_{i=1,2}\hat{\mathbb{P}}_i\left(\mathbf{x}\succ\mathbf{x}_{an,i}^{(k-1)}\right) \tag{16}$$

A large value for I_{cov} indicates that point $\mathbf{x}$ clearly dominates anchor point across the objective it corresponded to a minimum for, so coverage of the Pareto front has changed. This leads to the second *coverage stopping criterion*:

$$\forall\mathbf{x}\in\mathbf{X}_p^{(k)}:\quad I_{cov}\left(\mathbf{x}|\left\{\mathbf{x}_{an,i}^{(k-1)}\right\}_{i=1,2}\right)\leq\delta_{anch} \tag{17}$$

where δ_{anch} is a predetermined threshold.

The optimization progress is terminated when both convergence and coverage stopping criteria are satisfied. Parameters δ_{conv} and δ_{anch} affect the convergence rate of the iterative scheme and their selection represents a trade-off between quality of solutions and algorithmic efficiency. As a reference, consider the ideal case that current Pareto front is identical to the precedent one, leading to $\mathbb{P}_i\left(\mathbf{x}\succ\mathbf{x}'\right)=0.5$ for all objectives. Hence, δ_{anch} should be selected around 0.5–0.7 (probability up to 20% of exceeding previous minimums) and δ_{conv} also around 0–0.2 (probability of superiority is bounded between 30 and 70%). Parametric studies on the illustrative examples considered later showed little impact on the quality of the identified solutions of these selections (as long as they are chosen in the recommended range).

4.4 Hybrid design of experiments for refinement of metamodel

If convergence has not been achieved at the end of the kth iteration, the current set of experiments needs to be enriched to inform the development of a more accurate (refined) metamodel. In this work, we employ a hybrid DoE that combines both *space-filling DoE* and *adaptive DoE* to achieve: (Ro.i) satisfactory global accuracy and (Ro.ii) sufficient local accuracy for regions of importance for the augmented input under the multi-objective optimization context. New experiments n_{a1} and n_{a2} are obtained from each strategy for a total of $n_a=n_{a1}+n_{a2}$ additional experiments. This leads to a total of $n^{(k+1)}=n^{(k)}+n_a$ training points for the metamodel in the next iteration.

The space-filling DoE populates the entire domain for the augmented input $\mathbf{y}$ using Latin hypercube sampling (LHS). For $\mathbf{x}$ LHS from uniform distribution within the design domain, X is adopted, whereas for $\boldsymbol{\theta}$ LHS from distribution $p(\boldsymbol{\theta})$ is utilized. The space-filling DoE strategy guarantees that metamodel accuracy will be improved over the entire domain and, therefore, achieves (Ro.i), and reduces risk of missing optimum design configurations that could belong in yet unexplored regions with lower metamodel accuracy.

On the other hand, the adaptive DoE targets regions of importance for the augmented input. A sample-based DoE (Dubourg et al. 2011) is adopted here for this purpose that identifies a new set of experiments using a target density to represent domains of interests in Y, a utility function to incorporate metamodel accuracy in the DoE and clustering to eliminate close-proximity experiments. The target density is separately defined for the design variables $\mathbf{x}$ and the uncertain model parameters $\boldsymbol{\theta}$ since for each domain of interests have different attributes. For the former, a kernel-based density $\tilde{\pi}_d\left(\mathbf{x}|\mathbf{X}_d^{(k)}\right)$ around non-convergent regions of the Pareto front is chosen $\mathbf{X}_d^{(k)}=\left\{\mathbf{x}_d^{r(k)};r=1,\ldots,n_d\right\}$ whereas for the latter the density representing the IS density for each design configuration $\mathbf{x}$ is chosen, represented for computational efficiency as a kernel-based density conditional on the elements of $\mathbf{X}_d^{(k)}$, $\tilde{\pi}\left(\boldsymbol{\theta}|\mathbf{x}_d^{r(k)}\right)$. Details for these definitions are included in Appendix 3. The utility function incorporates in the DoE considerations about the metamodel accuracy by prioritizing candidate experiments with larger predicted error. The *linearized prediction variance* (Zhang et al. 2016) is chosen as utility function to facilitate this, given by

$$\mathrm{VAR}\left[h_i^{krig}(\mathbf{y})\right]=\sum_{m=1}^{n_z}\left(\left.\frac{\partial h_i[\mathbf{z}|\mathbf{x},\boldsymbol{\theta}]}{\partial z_m}\right|_{\boldsymbol{\mu}(\mathbf{y})}\cdot\sigma_m(\mathbf{y})\right)^2 \tag{18}$$

where $.|_{\mu_m(\mathbf{y})}$ denotes evaluation for $\mathbf{z}=\boldsymbol{\mu}(\mathbf{y})$.

The sample-based adaptive DoE strategy is composed of the following steps with objective to sample n_{a2} experiments

Step 1. Obtain a large number $n_c>>n_{a2}$ of candidate experiments $\mathbf{Y}=\{\mathbf{y}^j=[\mathbf{x}^j;\boldsymbol{\theta}^j],j=1,\ldots,n_c\}$ from the DoE target density $\pi_s^{(k)}(\mathbf{x},\boldsymbol{\theta})=\tilde{\pi}_d\left(\mathbf{x}|\mathbf{X}_d^{(k)}\right)\tilde{\pi}\left(\boldsymbol{\theta}|\mathbf{x}_d^{r(k)}\right)$.

Step 2. The linearized predictive variances of both performance measures $\mathrm{VAR}\left[h_i^{krig}(\mathbf{y})\right]$, $i=1,2$ are calculated for all n_c candidate experiments. As scalar accuracy metric, the following quantity is adopted

$$\overline{\mathrm{VAR}}(\mathbf{y})=\max_{i=1,2}\frac{\mathrm{VAR}\left[h_i^{krig}(\mathbf{y})\right]}{\frac{1}{n_c}\sum_{j=1}^{n_c}\left(\mathrm{VAR}\left[h_i^{krig}(\mathbf{y}^j)\right]\right)} \tag{19}$$

corresponding to the larger normalized variance, where normalization is established with respect to the sample mean. Candidate samples are then ranked based on this metric and only a certain portion $a_r n_c >> n_{a2}$ [where $a_r < 1$] are retained corresponding to higher $\overline{\mathrm{VAR}}(\mathbf{y})$ values.

Step 3. The retained experiments are finally clustered through K-means (Hartigan and Wong 1979) to the desired number of n_{a2} experiments.

One final question is how to achieve the balance between space-filling and adaptive DoE. In early iterations, greater focus should be put on space-filling DoE to avoid premature convergence due to poor metamodel accuracy in specific domains, with this focus shifted to local exploitation of regions of importance (i.e., adaptive DoE) as convergence is potentially established. This is achieved by choosing the ratio of experiments selected by the adaptive DoE, $n_{a2}^{(k)}/n_a$, as a monotonically decreasing function of the ratio of configurations in $\mathbf{X}_p^{(k)}$ being retained into the DoE set $\mathbf{X}_d^{(k)}$, i.e., $n_d^{(k)}/n_p^{(k)}$. A lower value indicates that a greater portion of the Pareto front has converged, inferring sufficient global coverage of the metamodel, and thus greater potential benefits from concentrating efforts on the remaining, non-convergent domains. To avoid the extreme case of getting no experiments from the adaptive DoE when no configuration is retained ($n_d^{(k)} = 0$), a certain percent of experiments (20% chosen here) is always chosen from the adaptive DoE, while among the remaining ones (80%), the adaptive DoE also accounts for a portion proportional to $n_d^{(k)}/n_p^{(k)}$.

5 Optimization implementation

This section reviews the overall iterative optimization approach starting with some remaining implementation details.

5.1 Initial DoE and selection stochastic simulation characteristics

The remaining implementation details pertain to the initial DoE strategy and the IS density and number of samples N selection. For the former, the space-filling DoE approach described in Section 4.4 is adopted, within an adaptive setting for selecting a sufficient number of experiments (Zhang et al. 2016). An initial number of training points n_{init} is obtained; then, the metamodel is developed and its accuracy is evaluated using leave-one-out cross validation. If some minimum desired accuracy is not achieved, n_{inita} additional training points are obtained and the latter two tasks are repeated.

For the IS densities, the density for ith performance function is selected as a mixture of optimal IS densities $\pi_i(\theta|\mathbf{x})$ [IS densities defined in Appendix 3 in (40)] for the design configurations in the precedent Pareto set $\mathbf{X}_p^{(k-1)}$:

$$q_i^{(k)}(\theta) = \frac{1}{n_p^{(k-1)}}\left[\sum_{r=1}^{n_p^{(k-1)}} \pi_i\left(\theta|\mathbf{x}_p^{r\,(k-1)}\right)\right] \tag{20}$$

This mixture corresponds to an appropriate density across the entire current Pareto set and is efficiently approximated through samples from it using kernel density estimation (KDE) (Scott 1992): using rejection sampling (Robert and Casella 2004), n_{is} samples are first obtained from each density $\pi_i\left(\theta|\mathbf{x}_p^{r\,(k-1)}\right), r = 1, \cdots, n_p^{(k-1)}$; these samples are then combined to yield a total of $n_{is} \times n_p^{(k-1)}$ samples defining set $\{\theta_{IS}\}_i^{(k)}$ which is finally used for the KDE approximation [yielding, ultimately, a similar expression as in (38)]. For N_i, a sufficiently large value should be chosen to guarantee high accuracy in the stochastic simulation, something that is computationally affordable due to the numerical efficiency of the Kriging metamodel. This value can be adaptively adjusted if a sufficient accuracy is not achieved. For this purpose, the coefficient of variation for the current selection of the IS density can be utilized, approximated (Robert and Casella 2004) as

$$\delta_i^{cv}\left(\mathbf{x}|q_i^{(k)}(\theta)\right) \approx \frac{1}{\sqrt{N_i}}\sqrt{\frac{\frac{1}{N_i}\sum_{j=1}^{N_i}\left(h_i^{krig(k)}\left(\mathbf{x},\theta_i^j\right)\frac{p\left(\theta_i^j\right)}{q_i^{(k)}\left(\theta_i^j\right)}\right)^2}{\left[\frac{1}{N_i}\sum_{j=1}^{N_i}\left(h_i^{krig(k)}\left(\mathbf{x},\theta_i^j\right)\frac{p\left(\theta_i^j\right)}{q_i^{(k)}\left(\theta_i^j\right)}\right)\right]^2}-1} \tag{21}$$

where $\left\{\theta_i^j, j = 1, \ldots, N_i\right\}$ is the sample set from $q_i^{(k)}(\theta)$. N_i can be adjusted across iterations to maintain a desired accuracy threshold δ_{th}^{cv}. For the kth iteration, and using the current Pareto set as reference (i.e., establish δ_{th}^{cv} over this set), this leads to updated value

$$N_i^{up} = N_i\left(\frac{\max\limits_{r=1,2,\ldots,n_p^{(k-1)}} \delta_i^{cv}\left(\mathbf{x}_p^{r(k-1)}|q_i^{(k-1)}(\theta)\right)}{\delta_{th}^{cv}}\right)^2 \tag{22}$$

where N_i corresponds to number of samples used for estimate $\delta_i^{cv}\left(\mathbf{x}_p^{r(k-1)}|q_i^{(k-1)}(\theta)\right)$. To guarantee adequate accuracy over the entire domain X, and since there is little gain in overall efficiency by using small N_i values (due to the metamodel computational efficiency), N_i should be chosen always larger than some minimum value N_{min}, based on the computational burden consideration of estimation of (7) (for example, N_i that allows estimation of each performance objective in less than 1 s).

This IS density formulation addresses the challenges outlined in Section 2 regarding the IS density selection by primarily relying on the metamodeling computational efficiency. Note that more advanced approaches could be adopted for the IS formulation (Medina and Taflanidis 2014), something that will create, though, additional computational burden. In the proposed approach, the improvement of the stochastic simulation accuracy is sought after by leveraging the metamodel computational efficiency (use of large N_i value) and not by trying to identifying the optimal IS density to use. An easy-to-define, but still good, IS density is considered sufficient for the latter.

It should be finally pointed out that beyond the satisfaction of accuracy threshold δ_{th}^{cv}, the adoption of exterior sampling further reduces the impact of the stochastic simulation estimation error on the identified solutions (Spall 2003).

5.2 Review of algorithm

When combining the previous ideas, one can formulate the following optimization approach (illustrated also in Fig. 2), referenced herein as MODU-AIM. First, define the predictive performance vector $\mathbf{h}^{krig}(\mathbf{x}, \theta)$ and assign the optimization function H_1 and constraint function $H_{2.}$ Set the thresholds for stopping criteria δ_{conv} and δ_{anch} and select the number of conditional realization samples N_{cr}. Set the rules for choosing the number of refinement experiments through the space-filling DoE $n_{a1}^{(k)}$ and the number of refinement experiments through the adaptive DoE $n_{a2}^{(k)}$, and for the adaptive DoE, choose the number of candidate samples n_c and the reduction ratio a_r based on metamodel accuracy. Finally, choose the number of samples for the IS density approximation n_{is}, the minimum number of samples N_{min} for the stochastic simulation and the accuracy threshold δ_{th}^{cv}.

Step 1. (Initial DoE): In the first iteration employ the initial DoE strategy to obtain total of $n^{(1)}$ training points, gradually increasing number of points till satisfactory cross-validation accuracy is achieved. Evaluate model response for these points $\{\mathbf{z}(\mathbf{y}^t);\ t = 1, \ldots, n^{(1)}\}$.

Step 2. (Kriging model): Utilize all available observations in the database to formulate the Kriging metamodel and obtain approximation $h_i^{krig(k)}(\mathbf{x}, \theta)$.

Step 3. (IS density formulation): Obtain n_{is} samples from each density $\pi_i\left(\theta|\mathbf{x}_p^{r(k-1)}\right)$ and combine them to formulate IS density $q_i^{(k)}(\theta)$ for each performance function through kernel density approximation. For elements belonging to $\mathbf{X}_d^{(k-1)}$, these samples are readily available from the previous iteration (step 9). In the first iteration, skip this step and use $q_i^{(1)}(\theta) = p(\theta)$ [no prior information available].

Step 4. (Stochastic simulation sample generation): Simulate set $\{\theta\}_i^{(k)}$ of N_i samples from $q_i^{(k)}(\theta)$ as the stochastic simulation sample set. For N_i, use N_{min} in first iteration or maximum between N_{min} and updated value given by (22).

Step 5. (Anchor point identification): Identify anchor points for the predictive Pareto front by solving optimizations described by (10). Use as initial guess for gradient optimization the previously identified anchor points $\mathbf{x}_{an,i}^{(k-1)}$.

Step 6. (Interior point identification): Calculate first the retained Pareto set of design configurations identified at previous step $\utilde{\mathbf{X}}_p^{(k-1)} = \left\{\utilde{\mathbf{x}}_p^{r(k-1)}; r = 1, \ldots, \utilde{n}_p^{(k-1)}\right\}$ that fall within the current anchor points (skip in first iteration). Identify interior points of the predictive Pareto front by solving optimizations described by (11). Use as initial guess for gradient-based algorithms the reference points $\mathbf{x}_p^{r(k-1)}$ for optimization in (11). Combine points from steps 5 to 6 to obtain candidate Pareto set $\mathbf{X}_{cp}^{(k)}$.

Step 7. (Refinement procedure of the Pareto front): Refine the Pareto set $\mathbf{X}_{cp}^{(k)}$ by resolving under-populated and over-populated regions to obtain a uniform distribution of the set.

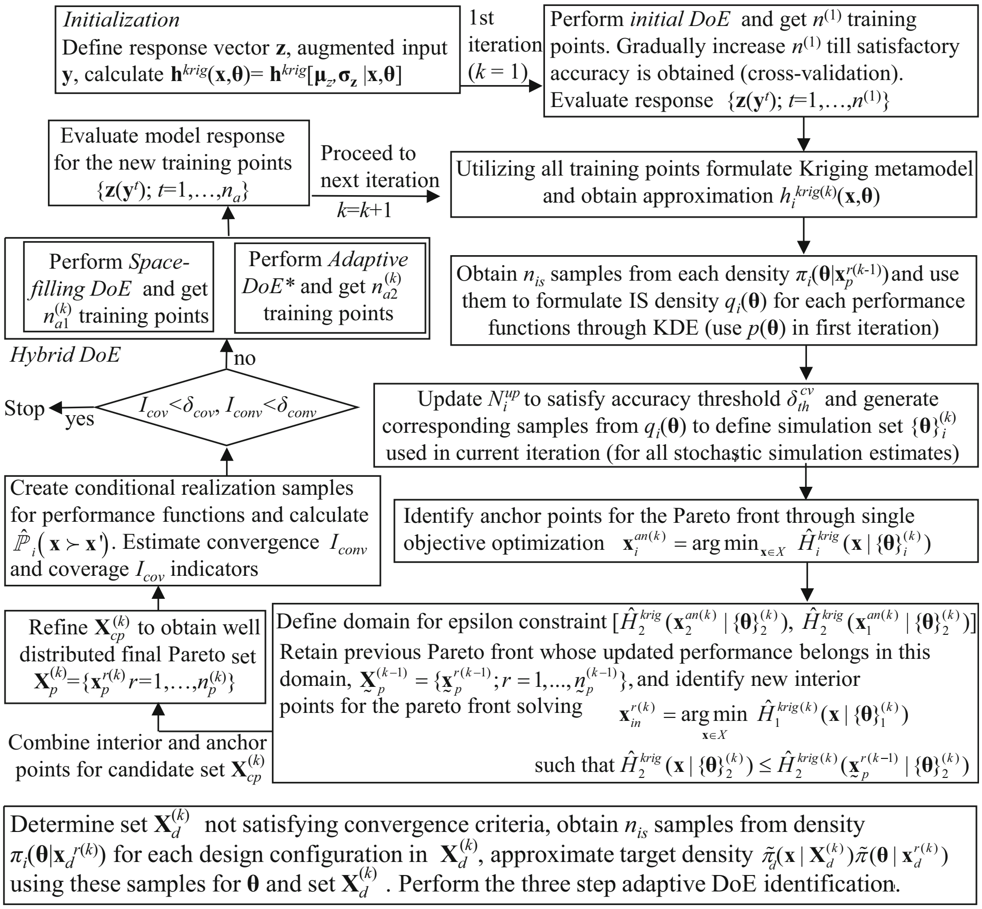
Step 8. (Stopping criteria checking): Check if the stopping criteria have been reached as detailed in Section 4.3. If they are met or computational effort has been exhausted, the optimization process is terminated.

Step 9. (Hybrid DoE for refinement): If optimization process is not terminated, employ the hybrid DoE strategy detailed in Section 4.4 to obtain total of n_a training points, $n_{a1}^{(k)}$ from the space-filling DoE and $n_{a2}^{(k)}$ from the adaptive DoE.

Step 10. (Evaluation of the response): Evaluate the model response $\{\mathbf{z}(\mathbf{y}^t);\ t = 1, \ldots, n_a\}$ for the newly identified training points at step 9 and combine with the previous observations in a database over all iterations. Proceed back to step 2 advancing to $k + 1$st iteration.

With respect to selection of the different characteristics of the algorithm (recall δ_{conv}, δ_{anch}, and N_{min} have been already discussed), the following recommendations are provided. The number of conditional realization samples N_{cr} should be chosen higher than 100; for the case that $\mathbb{P}_i\left(\mathbf{x} \succ \mathbf{x}'\right) = 0.5$ (designs having identical performance), this establishes a coefficient of variation below 5%. The number of the initial experiments n_{init} and the additional refinement experiments n_a should be chosen

Fig. 2 Schematic of MODU-AIM algorithm


Initialization
Define response vector **z**, augmented input **y**, calculate $\mathbf{h}^{krig}(\mathbf{x},\boldsymbol{\theta})=\mathbf{h}^{krig}[\boldsymbol{\mu}_z,\boldsymbol{\sigma}_z|\mathbf{x},\boldsymbol{\theta}]$
1st iteration ($k=1$)
Perform *initial DoE* and get $n^{(1)}$ training points. Gradually increase $n^{(1)}$ till satisfactory accuracy is obtained (cross-validation). Evaluate response $\{\mathbf{z}(\mathbf{y}^t);\ t=1,\ldots,n^{(1)}\}$
Utilizing all training points formulate Kriging metamodel and obtain approximation $h_i^{krig(k)}(\mathbf{x},\boldsymbol{\theta})$
Obtain n_{is} samples from each density $\pi_i(\boldsymbol{\theta}|\mathbf{x}_p^{r(k-1)})$ and use them to formulate IS density $q_i(\boldsymbol{\theta})$ for each performance functions through KDE (use $p(\boldsymbol{\theta})$ in first iteration)
Update N_i^{up} to satisfy accuracy threshold δ_{th}^{cv} and generate corresponding samples from $q_i(\boldsymbol{\theta})$ to define simulation set $\{\boldsymbol{\theta}\}_i^{(k)}$ used in current iteration (for all stochastic simulation estimates)
Identify anchor points for the Pareto front through single objective optimization $\mathbf{x}_i^{an(k)}=\arg\min_{\mathbf{x}\in X}\hat{H}_i^{krig}(\mathbf{x}|\{\boldsymbol{\theta}\}_i^{(k)})$
Define domain for epsilon constraint $[\hat{H}_2^{krig}(\mathbf{x}_2^{an(k)}|\{\boldsymbol{\theta}\}_2^{(k)}),\ \hat{H}_2^{krig}(\mathbf{x}_1^{an(k)}|\{\boldsymbol{\theta}\}_2^{(k)})]$
Retain previous Pareto front whose updated performance belongs in this domain, $\mathbf{X}_p^{(k-1)}=\{\mathbf{x}_p^{r(k-1)};r=1,\ldots,n_p^{(k-1)}\}$, and identify new interior points for the pareto front solving $\mathbf{x}_{in}^{r(k)}=\arg\min_{\mathbf{x}\in X}\hat{H}_1^{krig(k)}(\mathbf{x}|\{\boldsymbol{\theta}\}_1^{(k)})$
such that $\hat{H}_2^{krig}(\mathbf{x}|\{\boldsymbol{\theta}\}_2^{(k)})\leq\hat{H}_2^{krig(k)}(\mathbf{x}_p^{r(k-1)}|\{\boldsymbol{\theta}\}_2^{(k)})$
Combine interior and anchor points for candidate set $\mathbf{X}_{cp}^{(k)}$
Refine $\mathbf{X}_{cp}^{(k)}$ to obtain well distributed final Pareto set $\mathbf{X}_p^{(k)}=\{\mathbf{x}_p^{r(k)}\ r=1,\ldots,n_p^{(k)}\}$
Create conditional realization samples for performance functions and calculate $\hat{\mathbb{P}}_i(\mathbf{x}\succ\mathbf{x}')$. Estimate convergence I_{conv} and coverage I_{cov} indicators
$I_{cov}<\delta_{cov},\ I_{conv}<\delta_{conv}$
yes
Stop
no
Perform *Space-filling DoE* and get $n_{a1}^{(k)}$ training points
Perform *Adaptive DoE** and get $n_{a2}^{(k)}$ training points
Hybrid DoE
Evaluate model response for the new training points $\{\mathbf{z}(\mathbf{y}^t);\ t=1,\ldots,n_a\}$
Proceed to next iteration $k=k+1$
Determine set $\mathbf{X}_d^{(k)}$ not satisfying convergence criteria, obtain n_{is} samples from density $\pi_i(\boldsymbol{\theta}|\mathbf{x}_d^{r(k)})$ for each design configuration in $\mathbf{X}_d^{(k)}$, approximate target density $\tilde{\pi}(\mathbf{x}|\mathbf{X}_d^{(k)})\tilde{\pi}(\boldsymbol{\theta}|\mathbf{x}_d^{r(k)})$ using these samples for $\boldsymbol{\theta}$ and set $\mathbf{X}_d^{(k)}$. Perform the three step adaptive DoE identification.

**Adaptive DoE process*

proportional to the dimension of input space n_y following the recommendations in Jones et al. (1998). A 7- to 30-fold increase has been found effective in the applications examined: it avoids incremental only updates in the metamodel due to addition of small only number of experiments, something that can lead to premature convergence, while it also facilitates the adaptive characteristics of the algorithm, allowing a gradual increase of experiments. For the IS sampling density, n_{is} should be chosen in the range of $[10n_\theta, 50n_\theta]$ to facilitate adequate accuracy for the kernel-based approximation (Scott 1992). Finally, the number of candidate samples n_c should be much larger than n_a to make both reduction and clustering effective. Appropriate selection for n_c is $200n_a$ whereas for the reduction a_r, value of 5–10% has been shown to work well in the applications examined. It should be noted that besides n_a, these selections have not been found to have a strong impact on algorithmic efficiency.

6 Illustrative examples

The framework is illustrated next with two examples. The first one corresponds to a simpler, low-dimensional problem with analytical performance measures and the second one to a challenging engineering application examining the design of bi-linear passive dampers for the suspension of a half-car nonlinear model riding on a rough road. These two examples will be herein referenced, respectively, as "analytical" and "car suspension design". All characteristics of the numerical optimization that are independent of n_y (or n_θ and n_x), i.e., independent of the dimension of the design variables and the uncertain model characteristics, are selected common for both examples. For the stopping criteria characteristics, the thresholds δ_{conv} and δ_{anch} are taken as 15% and 65% respectively, whereas $N_{cr}=500$ is assumed for the conditional realization samples. The number of candidate samples n_c is taken as $200n_a$ with the reduction a_r taken as 10%, whereas the number of DoE experiments $n_{a1}^{(k)}$ and $n_{a2}^{(k)}$ are chosen based on the criteria discussed at the end of Section 4.4. The stochastic simulation characteristics (dependent on n_θ) and the number of experiments (dependent on n_y) are reported individually for each of the examples. As pointed out earlier besides the number of experiments, these selections are not anticipated to have a strong impact on the efficiency of the algorithm and sensitivity studies performed have verified this.

For each problem, a reference (benchmark) solution is obtained (whose procedure will be elaborated later) in order to compare the quality of the identified Pareto front. For evaluating the computational efficiency of the proposed approach, the problem is also solved by another surrogate model aided optimizer appropriate for multi-objective problems, SOCEMO (Müller 2017). SOCEMO is a derivative-free algorithm developed for efficiently solving deterministic multi-objective problem with expensive, black-box performance functions, like the ones discussed in this manuscript. It allows for a direct control of the number of evaluations of the expensive performance objectives. It has been already shown (Müller 2017) to offer enhanced efficiency for applications with expensive functions when compared to many other standard approaches for solving multi-objective problems, like the NSGA-II algorithm (Deb et al. 2002). This is the reason it is selected here (instead of such, more popular, alternatives). As SOCEMO is not directly applicable for design under uncertainty problems, it is applied here adopting a "double-loop," i.e., "nested" formulation, a common approach for such problems (Eldred et al. 2002): for each design configuration evaluation of the expensive design objectives is performed by stochastic simulation utilizing IS, as described by (3), adopting an exterior sampling approach. The total number of model evaluations for SOCEMO is the product of the number of required expensive evaluations of the design objectives and the sample size utilized in the stochastic simulation. For such double-loop approaches, an inevitable trade-off exists between the overall computational efficiency and the stochastic simulation accuracy. Higher sample size directly reduces computational efficiency but also reduces the stochastic simulation accuracy, lowering potentially quality of the identified solutions (Spall 2003). Since the inner loop is treated as a black-box, adaptive control of this accuracy is not possible. For the stochastic simulation, the IS densities are selected as the efficient ones identified in the final iteration of the MODU-AIM implementation, given by (20). In the overall comparisons of the computational efficiency, this provides a relative advantage to SOCEMO: formulation of good IS densities requires a considerable computational burden, especially in the context of design optimization, since such densities are not common across the whole design configurations examined (Medina and Taflanidis 2013). Here, efficient IS densities are provided for the inner loop of the SOCEMO implementation with no computational cost.

6.1 Analytical problem

6.1.1 Design problem formulation and optimization details

The expressions of two performance measures are:

$$
\begin{aligned}
h_1(\mathbf{x},\theta) &= \log\left[(x_1+2\theta)^2+(2x_2-x_1)^2+3(4\theta-x_2)^2\right]\\
h_2(\mathbf{x},\theta) &= 15-\exp\left(-\frac{x_1+2x_2+2\theta}{20}\right)
\end{aligned}
\tag{23}
$$

where $\mathbf{x}=[x_1\ x_2]$ is the design vector and $\boldsymbol{\uptheta}=\theta$ the single uncertain parameter, taken as a standard Gaussian variable. For simplicity, the response vector corresponds directly to the performances measures, leading to $\mathbf{z}=[h_1(\mathbf{x},\theta)\ h_2(\mathbf{x},\theta)]^T$. The predictive performance measure in (5), the partial derivative in (34), and the linearized prediction variances of (18) can be expressed in closed forms as ($i=1,2$):

$$
\begin{aligned}
&h_i^{krig}(\mathbf{y}) = \int_{\Re} z_i\cdot\phi\left(\frac{z_i-\mu_i(\mathbf{y})}{\sigma_i(\mathbf{y})}\right)dz_i = \mu_i(\mathbf{y})\\
&\frac{\partial h_i^{krig}(\mathbf{y})}{\partial x_l} = \frac{\partial \mu_i(\mathbf{y})}{\partial x_l}\\
&\mathrm{VAR}\left[h_i^{krig}(\mathbf{y})\right] = (\sigma_i(\mathbf{y}))^2
\end{aligned}
\tag{24}
$$

The design domains X has upper bounds [10, 10], and lower bounds [− 10, − 10] and the starting point of the algorithm (first iteration) is set to be $\mathbf{x}^{(1)}=[0, 0]$, the center of this domain. The pre-specified number of constraints n_{dp} is taken at 20, while the refinement procedure guarantees the Pareto set to have 10 to 40 well-distributed members. For the DoE characteristics, the number of the initial support points n_{init} and the additional refinement support points n_a are chosen as 20, whereas the number of incremental support points with unsatisfied cross-validation accuracy n_{inita} is taken as 10. The number of samples from the target IS density n_{is} is taken as 20, the number of minimum samples N_{min} for the stochastic simulation as 1000, and the accuracy threshold δ_{th}^{cv} as 5%.

The reference solution is obtained by solving problem analytically and in this case corresponds to the actual Pareto front, whereas for the SOCEMO implementation two variants are examined: one (denoted as SOCEMO 50^2) terminates after 50 evaluations of the actual performance objectives with a IS sample size of 50 (total 2500 expensive model evaluations per optimization), while the other (SOCEMO 100^2) terminates after 100 evaluations of the actual performance objectives with a IS sample size of 100 (10,000 expensive evaluations per run).

6.1.2 Results and discussion

Due to the relative simplicity of this example, focus is placed here on the final results only. Behavior across the iterations of the algorithm will be discussed in the next example. Results for a sample run of the proposed algorithm (MODU-AIM) and the two SOCEMO variants are shown in Table 1. The proposed algorithm converged in 4 iterations, requiring a total of 80 evaluations of the system response. This provides a considerable computational efficiency compared to any of the two SOCEMO variants examined. For comparing quality of the obtained solutions, once the Pareto sets are identified, the performance objectives, and corresponding Pareto fronts, are analytically evaluated (to avoid the influence of any errors stemming from the stochastic simulation in this comparison).

Table 1 Comparison of quality of the identified Pareto fronts by all considered implementations for the first illustrative example (analytical problem)

Metric	MODU-AIM	SOCEMO 50^2	SOCEMO 100^2
GD	9.01×10^{-5}	8.01×10^{-3}	8.71×10^{-4}
MS	0.987	0.929	0.878

Figure 3 presents the identified Pareto fronts along with the reference one. Apart from the visual comparison, a quantitative comparison is established using two common metrics for assessing quality of approximate solutions to multi-objective problems (Goh and Tan 2006). The first metric is the generational distance, *GD*, which compares the identified Pareto front to the actual one:

$$GD = \frac{1}{n_p} \sum_{r=1}^{n_p} d_r^2 \tag{25}$$

where n_p is the number of members in the Pareto set, and d_r^2 is the squared Euclidian distance in the objective space between the *r*th member and its closes neighbor in the actual Pareto front. Lower values of *GD* indicate smaller deviations from the reference front and therefore higher quality of the identified solution. The second metric is the maximum spread, *MS*, which computes the portion of the reference Pareto front that covered by the identified one:

$$MS = \sqrt{\frac{1}{n_h} \sum_{i=1}^{n_h} \left[\frac{\min\left(\max_r \left[H_i\left(\mathbf{x}_p^r\right)\right], H_{i,\max}\right) - \max\left(\min_r \left[H_i\left(\mathbf{x}_p^r\right)\right], H_{i,\min}\right)}{H_{i,\max} - H_{i,\min}} \right]^2} \tag{26}$$

where $H_{i,\max}$, $H_{i,\min}$ represents the max/min value of the *i*th objective across the actual Pareto front, and $\max_r \left[H_i\left(\mathbf{x}_p^r\right)\right]$, $\min_r \left[H_i\left(\mathbf{x}_p^r\right)\right]$ represents the max/min value of that objective across the identified Pareto front. *MS* is bounded in [0, 1], and higher values for it indicate better coverage (i.e., higher quality identification). Note that *MS* metric can be implemented by comparing simply to anchor points whereas *GD* metric requires comparison to the actual Pareto front, or at least a reference discrete front with large number of solutions. This stems from the need to identify the closest neighbor, which assumes a very well populated front to compare to.

The results in Fig. 3 (qualitative, visual comparison) and Table 1 (quantitative comparison) show that the proposed implementation (MODU-AIM) identifies a Pareto front close to the actual one (*GD* value) and with the same coverage (*MS* value). In contrast, the Pareto fronts identified by both SOCEMO implementations include non-dominant design configurations (evident in Fig. 3 comparisons) and, in general, identify lower quality Pareto fronts (Table 1 measures). Consideration, additionally, of the difference in the number of evaluations of the performance measures required by the two approaches, 80 for MODU-AIM compared to 2500 or 10,000 by SOCEMO, reveals the advantages offered by the proposed approach: higher computational efficiency and higher identification quality for the Pareto front are accomplished. The double-loop approach employed for implementation of SOCEMO cannot outperform the efficiency offered by the formulation of a surrogate model in the augmented input space. This should not be considered as a deficiency of SOCEMO, rather than a limitation of generalized, nested optimization approaches for design under uncertainty design problems.

Another important question is what are the computational efficiency benefits provided by the iterative scheme discussed in Section 4. In other words, whether the advantages reported for MODU-AIM simply stem from the development of the metamodel in the augmented input space. Exploiting the numerical simplicity of this analytical example considered here, a non-iterative implementation is considered; this is accomplished by considering only the first iteration of the proposed algorithm with a predetermined number of experiments. This number is gradually increased till the same quality (similar values as the ones reported in Table 1) is established for the Pareto front identification. The non-iterative implementation required over 500 evaluations of the performance measures. Comparison to the computational effort required by the iterative implementation, only 80 evaluations, makes it evident that the iterative scheme, as expected, facilitates significant computational benefits.

Finally, the low dimensionality of this analytical example is leveraged to demonstrate the DoE sampling density for the design variables $\mathbf{x}$ in Fig. 4. The converged/non-converged parts of the Pareto set, as well as 1000 DoE candidate samples are shown in this figure for the 2nd, 3rd and 4th iterations of the algorithm. One thousand samples, as opposed to the 20 used in the algorithm, are utilized to better illustrate the sampling density characteristics. In earlier iterations, the DoE samples are distributed more evenly across the entire design domain *X*. This is desirable as discussed earlier, to avoid premature convergence due to locally poor metamodel accuracy. As iterations evolve, design subdomains close to the non-converged parts of the Pareto set are prioritized in the DoE, with samples populating more densely these regions. This facilitates, ultimately, an improvement of the convergence speed in these regions, without an impact on the other, converged already, domains.

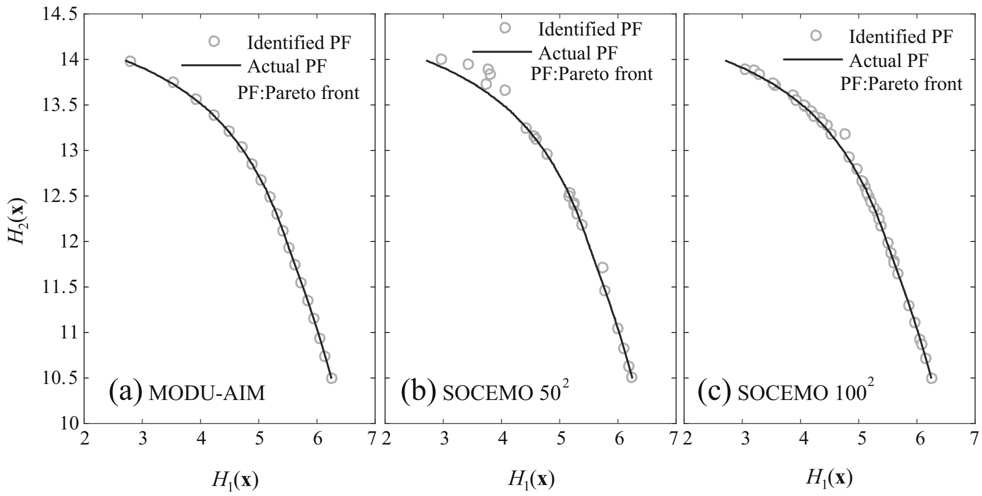


Fig. 3 Comparison of solutions for the analytical problem identified by the metamodel-aided optimization to the exact solution for (**a**) proposed approach (MODU-AIM), (**b**) SOCEMO 50^2 and (**c**) SOCEMO 100^2

6.2 Car suspension design problem

6.2.1 Design problem formulation

The schematic of the half-car model is shown in Fig. 5. A complete description of the numerical model is included in Medina and Taflanidis (2014). For the suspension damper, a passive nonlinear implementation is considered, allowing a different stroke reaction when the suspension is moving downwards as opposed to when moving upward (Verros et al. 2005). The average damping coefficients C^l and the percentage increase of damping r^n for upwards movement of the suspension are taken as the design variables. They are allowed to be separately selected for the front and rear dampers (distinguished by subscript f or r), leading to the definition of design variable vector as $\mathbf{x} = \left[C^l_f \;\; r^n_f \;\; C^l_r \;\; r^n_r \right]^T$. The road surface is modeled as a zero-mean Gaussian stationary stochastic process with power spectral density parameterized by the roughness coefficient, κ_i. A time-domain realization for this input is obtained by the spectral representation method assuming that the car drives with a constant horizontal velocity v_c. Apart from the car length, all other parameters for its model are assumed to be uncertain; this includes the eccentricity between the chassis geometric center and center of mass, e_x; the masses for the chassis and front and rear tires, m_c, m_{tf}, m_{tr}, respectively; the chassis moment of inertia, I_c; the spring and dashpot coefficient for front and rear tire, K_{tf}, C_{tf}, K_{tr}, C_{tr}, respectively; and the linear and nonlinear spring coefficients for the front and rear suspension K^l_f, K^n_f,K^l_r, K^n_r, respectively. This leads to definition for the uncertain parameter vector $\boldsymbol{\theta} = [\kappa_i \; v_c \; e_x \; m_c \; m_{tf} \; m_{tr} \; I_c \; K_{tf} \; C_{tf} \; K_{tr} \; C_{tr} \; K^l_f \; K^n_f \; K^l_r K^n_r]$ with $n_\theta = 15$. The probability models for $\boldsymbol{\theta}$ are discussed in detail in Medina and Taflanidis (2014).

The performance of the suspension system is evaluated assuming that car drives on the rough road for a lengthy stretch, so that response reaches stationary characteristics that can be quantified for design purposes through root mean square (RMS) statistics (Verros et al. 2005). The only source of uncertainty impacting the response of the suspension

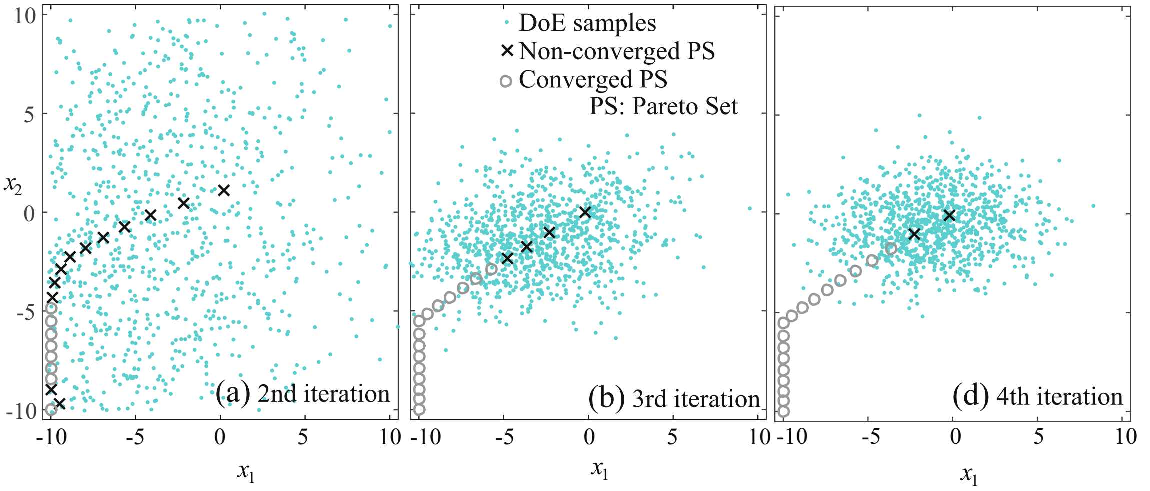
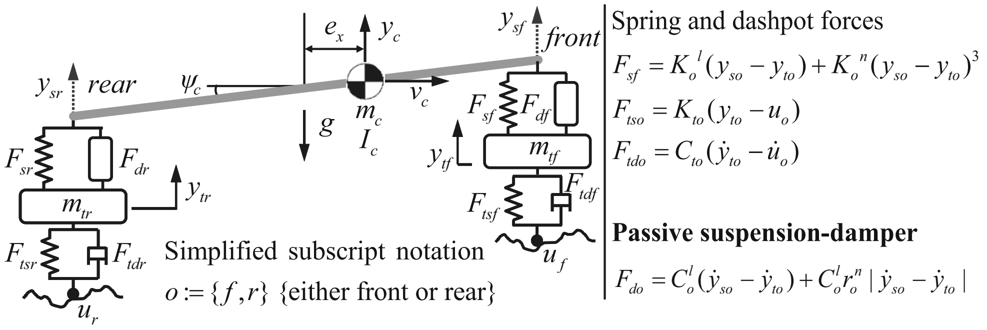


Fig. 4 Converged/non-converged parts of the Pareto set and 1000 samples from the DoE density for **x** for (**a**) 2nd, (**b**) 3rd, and (**c**) 4th iterations of the algorithm. Results are for the analytical problem



e_x
y_c
y_{sf}
front
y_{sr}
rear
ψ_c
v_c
g
m_c
I_c
F_{sf}
F_{df}
y_{tf}
m_{tf}
F_{tdf}
F_{tsf}
u_f
F_{sr}
F_{dr}
y_{tr}
m_{tr}
F_{tsr}
F_{tdr}
u_r
Simplified subscript notation
$o := \{f, r\}$ {either front or rear}
Spring and dashpot forces
$F_{sf} = K_o{}^l(y_{so} - y_{to}) + K_o{}^n(y_{so} - y_{to})^3$
$F_{tso} = K_{to}(y_{to} - u_o)$
$F_{tdo} = C_{to}(\dot{y}_{to} - \dot{u}_o)$
Passive suspension-damper
$F_{do} = C_o^l(\dot{y}_{so} - \dot{y}_{to}) + C_o^l r_o^n \mid \dot{y}_{so} - \dot{y}_{to} \mid$

Fig. 5 Half-car model schematic and forces involved

system is, therefore, the uncertainty associated with the model parameter vector θ (no influence of the stochastic excitation characteristics, since stationary conditions achieved). The competing performance objectives for formulating the multi-objective design problem are related to the ride comfort and road holding, both common objectives for car suspension design (Dahlberg 1978). For the ride comfort the fragility of the RMS vertical acceleration at the center of mass RMS_{ac} is used, whereas for the road holding the sum of RMS dynamic forces developed between the ground and tires (RMS_{tf}, RMS_{tr}) is adopted. These forces are calculated for each tire considering both the spring and dashpot connecting it to the ground, whereas dynamic characteristics pertain to deviation from the static force developed to compensate for the car weight. For the acceleration fragility, log-normal characteristic are assumed (Medina and Taflanidis 2014) with threshold $b = 1$ m/s^2 defining acceptable performance and coefficient of variation $\sigma_b = 5\%$ for the fragility. This leads to the following performance measures

$$\begin{aligned} h_1(\mathbf{x}, \theta) &= \Phi\left[\frac{\ln(RMS_{ac}) - \ln(b)}{\sigma_b}\right] \\ h_2(\mathbf{x}, \theta) &= RMS_{tf} + RMS_{tr} \end{aligned} \tag{27}$$

The response vector is taken to correspond to the log of the RMS acceleration and the RMS tire forces, representing the quantities needed in the performance measures of (27), $\mathbf{z} = [\ln(RMS_{ac})\ RMS_{tf}\ RMS_{tr}]^T$. The RMS response is evaluated through time-domain numerical simulation (Medina and Taflanidis 2014) to address the various source of nonlinearities included in the car model.

6.2.2 Kriging implementation characteristics

The predictive performance measures are evaluated as

$$\begin{aligned} h_1^{krig}(\mathbf{y}) &= \int_{\mathfrak{R}} \Phi\left[\frac{z_1 - \ln(b)}{\sigma_b}\right] \phi\left(\frac{z_1 - \mu_1(\mathbf{y})}{\sigma_1(\mathbf{y})}\right) dz_1 = \Phi\left[\frac{\mu_1(\mathbf{y}) - \ln(b)}{\sqrt{\sigma_b^2 + \sigma_1^2(\mathbf{y})}}\right] \\ h_2^{krig}(\mathbf{y}) &= \int_{\mathfrak{R}^2} [z_2 + z_3] \phi\left(\frac{\mathbf{z} - \boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y})}{\boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{y})}\right) dz_2 dz_3 = \mu_2(\mathbf{y}) + \mu_3(\mathbf{y}) \end{aligned} \tag{28}$$

which lead ultimately to the desired analytic expressions for them. The partial derivatives in (34) are

$$\frac{\partial h_1^{krig}(\mathbf{y})}{\partial x_l} = \phi\left(\frac{\mu_1(\mathbf{y}) - \ln(b)}{\sqrt{\sigma_b^2 + \sigma_1^2(\mathbf{y})}}\right) \times \left(\frac{1}{\sqrt{\sigma_b^2 + \sigma_1^2(\mathbf{y})}} \frac{\partial \mu_1(\mathbf{y})}{\partial x_l} - \frac{1}{2} \frac{\mu_1(\mathbf{y}) - \ln(b)}{(\sigma_b^2 + \sigma_1^2(\mathbf{y}))^{3/2}} \frac{\partial \sigma_1^2(\mathbf{y})}{\partial x_l}\right) \tag{29}$$

$$\frac{\partial h_2^{krig}(\mathbf{y})}{\partial x_l} = \frac{\partial \mu_2(\mathbf{y})}{\partial x_l} + \frac{\partial \mu_3(\mathbf{y})}{\partial x_l}$$

In addition, the linearized prediction variances of (18) are calculated as

$$\begin{aligned} \mathrm{VAR}\left[h_1^{krig}(\mathbf{y})\right] &= \left(\frac{1}{\sigma_b} \phi\left(\frac{\mu_1(\mathbf{y}) - \ln(b)}{\sigma_b}\right) \cdot \sigma_1(\mathbf{y})\right)^2 \\ \mathrm{VAR}\left[h_2^{krig}(\mathbf{y})\right] &= (\sigma_2(\mathbf{y}))^2 + (\sigma_3(\mathbf{y}))^2 \end{aligned} \tag{30}$$

6.2.3 Numerical optimization details

The design domains X has upper bounds [4000 Ns/m, 1, 4000 Ns/m, 1], and lower bounds [400 Ns/m, − 1, 400 Ns/m, − 1]. The starting point of the algorithm (first iteration) is set to be $\mathbf{x}^{(1)} = [500$ Ns/m, 0, 500 Ns/m, 0] which correspond to small damper configurations with pure linear component. The pre-specified number of constraints n_{dp} is taken at 10, while the refinement procedure guarantees the Pareto set to have 5 to 20 well-distributed members. Additionally, for the Pareto front refinement, the log of the first objective log ($H_1(\mathbf{x})$) was utilized since the variation of that objective across the Pareto is large (ranging from 0.05 to 20%). This choice facilitates a better representation across the front of small failure probabilities, deemed of importance for the application here. For the DoE characteristics, two values will be examined for the number of the initial support points n_{init} and the additional refinement support points n_a, either (a) 500 or (b) 200, whereas the number of incremental support points with unsatisfied cross-validation accuracy n_{inita} is taken as 100. The two different instances for n_a examined will be referenced as DoE500 and DoE200 herein. The number of samples from the target IS density n_{is} is taken as 200, while the number of minimum samples N_{min} for the stochastic simulation as 5000 and the accuracy threshold δ_{th}^{cv} as 5%.

The reference (benchmark) solution cannot be obtained in this application analytically. Instead, it is approximated by a reference Pareto set $\mathbf{X}_{ref} = \left\{\mathbf{x}_{ref}^{r}, r = 1, \ldots, n_{ref}\right\}$ containing n_{ref}= 15 well-distributed solutions. This reference solution is obtained here by solving the original design optimization problem of (2) using the adaptive weighted sum algorithm discussed in Kim and De Weck (2006) to convert the design problem to an unconstrained single-objective optimization, and the efficient global optimization algorithm (EGO) (Jones 2001) to solve the latter. These choices were promoted due to the high computational burden associated with estimation of the performance objectives, though any other appropriate methodology could had been adopted for identifying the reference solution. Like in the SOCEMO implementation, the performance is evaluated through stochastic simulation [i.e., through (3)] utilizing the exact numerical model and exterior sampling. The IS densities for generating the respective sample set were chosen as the ones used in the final iteration of the iterative Kriging-based optimization, described in (20). The number of samples N was selected to be sufficiently large (N= 5000) to guarantee high quality of the identified solutions. This sample sets is denoted as the reference sample set $\{\theta\}_i^{ref}, i = 1, 2$. To further reduce such computational cost, an initial set of common candidate solutions for EGO was constructed, consisting of 200 design configurations determined by LHS in the whole design domain. The total number of function evaluations to obtain the reference Pareto set is 3,590,000.

For comparing computational efficiency, the SOCEMO implementation is again considered. The variant with only 50 samples for the stochastic simulation provided poor results in this case (significant errors in identifying parts of the Pareto front close to one of the anchor points), so only the implementation SOCEMO 100^2 is discussed further. Recall that a single run of SOCEMO 100^2 requires 10,000 evaluations of the computational expensive system model.

6.3 Results and discussion

Five runs of the proposed algorithm (MODU-AIM) and one run of the SOCEMO approach are conducted. The proposed implementation converged in 4–8 iterations for the DoE500 implementation and 8–14 iterations for the DoE200. The average computational effort is then 3100 model runs (6 iterations) for DoE500 and 2000 model runs for DoE200 (10 iterations). Upon convergence, the performance functions for all identified Pareto sets are evaluated using the exact numerical model, utilizing the reference sample set $\{\theta\}_i^{ref}, i = 1, 2$ to facilitate consistent comparisons.

Discussion first focuses on a sample run of the proposed MODU-AIM algorithm. Figure 1 in Section 4.2 depicts the progression of the Pareto front in the second iteration, including the set identified in the first iteration $\hat{H}_i^{krig(1)}\left(\mathbf{x}_p^{(1)}|\{\theta\}_i^{(1)}\right)$, its update using the new metamodel, $\hat{H}_i^{krig(2)}\left(\mathbf{x}_p^{(1)}|\{\theta\}_i^{(2)}\right)$, and the set identified in the second iteration $\hat{H}_i^{krig(2)}\left(\mathbf{x}_p^{(2)}|\{\theta\}_i^{(2)}\right)$. Comparison between $\hat{H}_i^{krig(2)}\left(\mathbf{x}_p^{(1)}|\{\theta\}_i^{(2)}\right)$ and $\hat{H}_i^{krig(1)}\left(\mathbf{x}_p^{(1)}|\{\theta\}_i^{(1)}\right)$ shows the prediction accuracy improvement with the updated metamodel, especially in regions corresponding to smaller values for H_1, whereas comparison of $\hat{H}_i^{krig(2)}\left(\mathbf{x}_p^{(2)}|\{\theta\}_i^{(2)}\right)$ to $\hat{H}_i^{krig(2)}\left(\mathbf{x}_p^{(1)}|\{\theta\}_i^{(2)}\right)$ shows the evolution of the identified solutions towards higher quality (i.e., dominant) designs. The bigger discrepancies shown for objective H_1 should be attributed to its special characteristics; for certain parts of the front, objective H_1 represents events with small likelihood of occurrence, for which an adaptive DoE strategy that target the

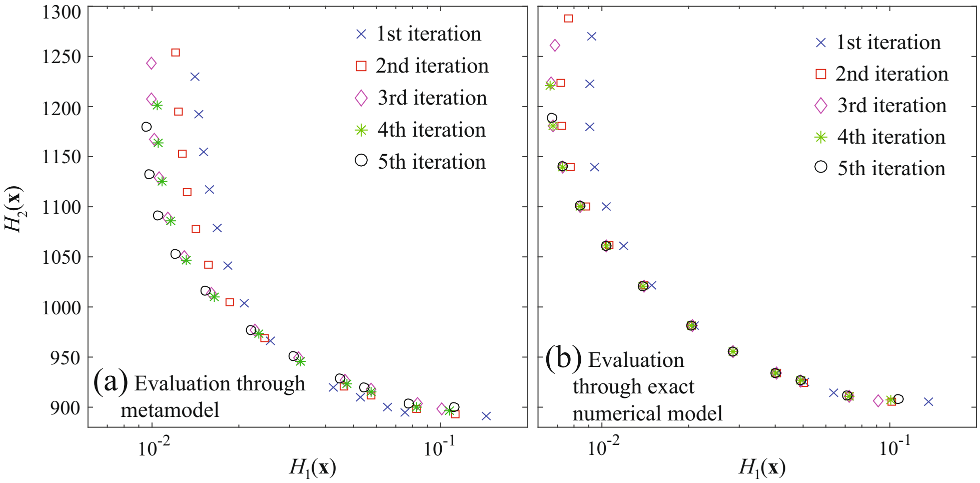


Fig. 6 Evolution of Pareto front across the different iterations of the optimization algorithm for the car suspension design problem. Evaluation is performed either through (**a**) the metamodel or (**b**) the exact numerical model

domains in Θ contributing to failure can provide substantial benefits (Zhang et al. 2016). The proposed framework adaptively adjust to this challenge and improves metamodel accuracy and subsequently the quality of the identified solutions across the iterations, as also shown in Fig. 6 which depicts the evolution of the Pareto front across the different iterations of the algorithm using either [part (a)] the metamodel established at each iteration, i.e., $\hat{H}_i^{krig(k)}\left(\mathbf{x}_p^{(k)}|\{\boldsymbol{\theta}\}_i^{(k)}\right)$ or [part (b)] the exact numerical model $\hat{H}_i\left(\mathbf{x}_p^{(k)}|\{\boldsymbol{\theta}\}_i^{(k)}\right)$. As the algorithm progresses, discrepancies between the identified fronts gradually decrease and convergence is established with great computational savings, as pointed out earlier. This convergence holds for both the metamodel-based evaluation [part (a)], which corresponds to what the algorithm approximates as behavior of the objective function, as well as for and the exact numerical model based evaluation [part (b)]. Comparison between $\hat{H}_i^{krig}$ and $\hat{H}_i$ shows greater differences for small values of objective H_1, agreeing with the aforementioned discussion of greater challenges in accurately approximating rare events. Note, though, that the proposed framework does not place any emphasis on agreement between $\hat{H}_i^{krig}$ and $\hat{H}_i$, and despite any differences existing in this comparison, it still converges to the correct front (see discussion on Fig. 7 next). This supports the philosophy for targeting convergence to correct solutions by an adaptive iterative scheme, rather than requiring development of accurate metamodels.

The focus moves now to the performance comparisons among all implementations. Results across the five different runs of MODU-AIM are reported in Fig. 7 for either the DoE500 [Fig. 7, part (a)] or DoE200 [Fig. 7, part (b)], and results for SOCEMO are reported in Fig. 8. In all instances, the reference Pareto front is included for comparison. These figures facilitate the visual comparison for the quality of the obtained solution. The results demonstrate good agreement between the reference and the identified solutions in both DoE cases for MODU-AIM. For SOCEMO, a significant portion of the identified Pareto set is non-dominant with respect to the reference solution. This behavior is especially evident for lower values of H_1 objective and therefore should be attributed to the inaccurate stochastic simulation because of using 100 only samples, despite the efficient IS densities. This lower accuracy impacts disproportionally domains where H_1 has characteristics of a rare event. An evident remedy is to increase N_1 for SOCEMO (it was found that a 10-fold increase to $N_1 = 1000$ greatly improves results). However, as discussed previously, this larger value will directly impact computational efficiency. For example, for maintaining the same total computational effort with an increase value of N_1, the number of expensive evaluations of the design objectives would proportionally decrease. Beyond the visual comparison the quantitative comparison results are reported in Table 2, adopting similar metrics as in the previous example. In this case, the generational distance *GD* is no longer meaningful since only a discrete reference Pareto front is available. For this reason, the *MS* metric is only reported. For the DoE200 and DoE500 the coefficient of variation for *MS* over the 5 different trials is also reported in parenthesis. The comparison of the results in Figs. 6 and 7 and Table 2 shows that both variants of the proposed scheme, DoE200 and DoE500, outperform SOCEMO. This is established, similar to the previous example, with a remarkable computational efficiency (check also summary results in Table 3 later), with the number of expensive model evaluations not exceeding 4000 for DoE500, and 2800 for DoE200 compared to SOCEMO (10000) or the EGO for obtaining the reference solutions (3,590,000).

Focusing on MODU-AIM, for both DoE500 and DoE200 implementations, some discrepancies with respect to the second objective exist for regions close to the minimum for H_1. These trends are anticipated: for most multi-objective problems, high sensitivity exists for the Pareto front close to the anchor points, with small improvements of the primary objective (the one that corresponds to the single-objective minimum) coming at large “sacrifice” for the secondary objective. This feature creates the differences between the approximate and reference solutions and can be resolved, if desired, by further refinement of the optimization around these domains, either by using a smaller value for δ_{anch}, or through a second optimization stage focusing explicitly at the anchor points, for example, through the single-objective optimization framework described in Zhang et al. (2016), leveraging existing experiments for the metamodel development. These challenges are compounded for regions close to the anchor point of H_1 due to the rare event characteristics. Such challenges are greater for the DoE200 implementation, with DoE500 demonstrating better robustness across the different runs. This should be attributed to the fact that addition of only 200 experiments per iteration might not significantly alter the metamodel across consecutive iterations and therefore might lead to premature convergence. Remedy for this challenge could be to rely on the evaluation of the convergence across multiple iterations, i.e., not comparing only *k*th to *k*-1st iterations. Nevertheless, the overall Pareto front identified in all DoE implementations shows good agreement with the reference solution.

Finally, the discussion moves to the effect on different choices of n_a (the number of refinement experiments), based on the overall optimization efficiency and accuracy for DoE500 and DoE200. The following metrics are used for evaluating the performance across the different runs: (i) computational efficiency metrics, evaluated by the average (N_{avg}^{tot}) or worst-case (N_{wc}^{tot}) total numbers of calls to the exact system model, and (ii) accuracy metrics, evaluated by the average ($\mathrm{Err}_{i,avg}^{ref}$), and worst-case ($\mathrm{Err}_{i,wc}^{ref}$) *error rate* of *i*th

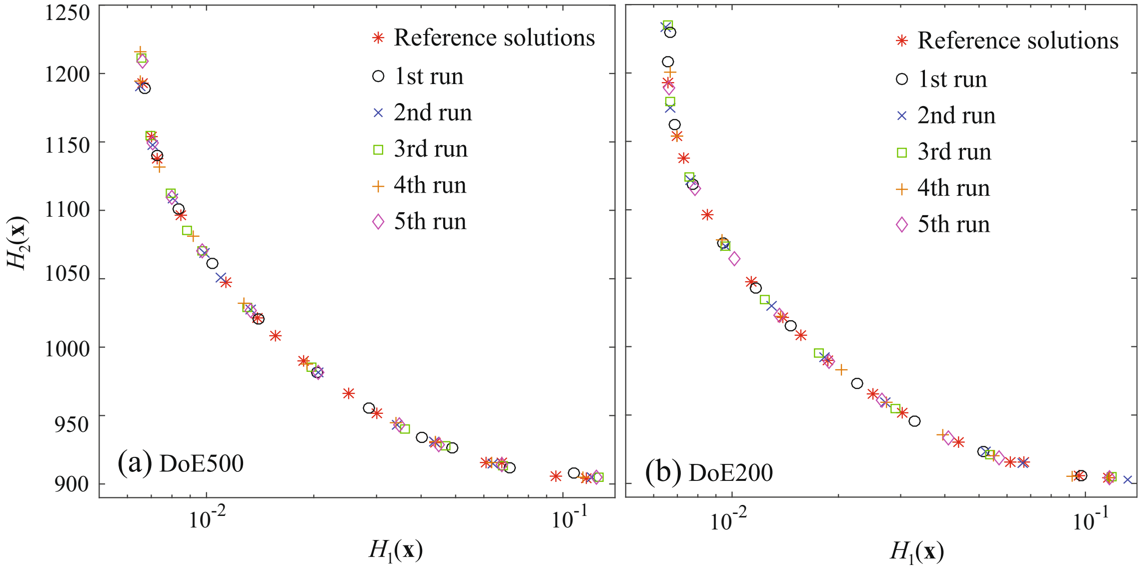


Fig. 7 Comparison of solutions identified by the metamodel-aided MODU-AIM optimization to the reference solutions across all five different runs for (**a**) DoE500 and (**b**) DoE200 for the car suspension design problem

performance function. The latter calculates the average relative absolute error of the metamodel across the entire front [i.e., comparing parts (a) and (b) of Fig. 6] and is given by

$$\mathrm{Err}_i^{ref} = \frac{1}{n_p}\sum_{r=1}^{n_p}\left|\frac{\hat{H}_i\left(\mathbf{x}_p^r|\{\theta\}_i^{ref}\right)-\hat{H}_i^{krig}\left(\mathbf{x}_p^r|\{\theta\}_i^{ref}\right)}{\hat{H}_i\left(\mathbf{x}_p^r|\{\theta\}_i^{ref}\right)}\right| \quad (31)$$

Lower error rate indicates higher accuracy over the critical regions in multi-objective optimization, namely the true Pareto set. In all instances, average results correspond to the mean over the five runs, whereas worst case corresponds to the least favorable performance over these five runs.

The results in Table 3 stress the computational efficiency of the proposed approach as the convergence to the Pareto front is accomplished with small computational effort, not exceeding 4000 model evaluations in either case. The DoE500 case takes extra computational effort (compare N_{tot}^{avg} and N_{wc}^{tot}), but also leads to higher quality results (compare $\mathrm{Err}_{i,avg}^{ref}$ and $\mathrm{Err}_{i,wc}^{ref}$), which can provide greater robustness in identifying better quality solutions as discussed earlier. Further focusing on the predicted performance across the identified front, described through Err_i^{ref}, low error rate is reported for the first performance function (not higher than 0.114) and an almost vanishing error rate (not higher than 0.016) for the second one. The higher error rate of the first objective function is in line with the aforementioned trends. It is also interesting to compare this error across the iterations of the optimization algorithm. For the sample run, illustrated in Fig. 6 earlier, the evolution of Err across iterations for objective 1 ($i = 1$) from 1st iteration to 5th iteration is [0.798, 0.493, 0.194, 0.137, 0.088] and for objective 2 ($i = 2$) is [0.027, 0.014, 0.013, 0.016, 0.010]. Evidently, the proposed adaptive iterative scheme facilitates a gradual improvement of the metamodel accuracy over the domains of importance, which, in turn, allows the identification of higher quality solutions as demonstrated in Fig. 6.

Overall, the results in the figures and tables demonstrate the benefits from the proposed iterative scheme for adaptively

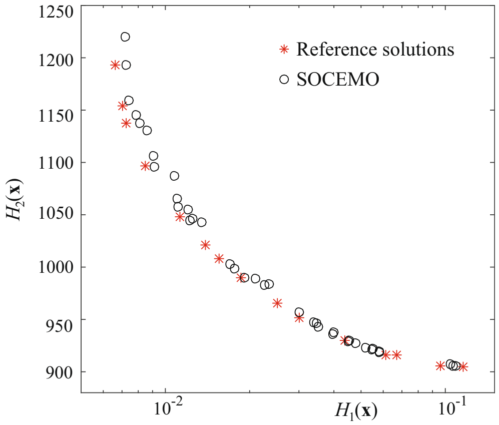


Fig. 8 Comparison of solutions identified by SOCEMO to the reference solutions for the car suspension design problem

Table 2 Comparison of quality of the identified Pareto fronts by all considered implementations for the car suspension design problem. For DoE500 and DoE200, the mean over 5 trials and coefficient of variation in parenthesis are reported

	DoE500	DoE200	SOCEMO
MS	0.998 (0.4%)	0.996 (0.49%)	0.967

Table 3 Overall computational efficiency of the algorithm for both the DoE500 and DoE200 implementations for the car suspension design example. For metrics pertaining to average statistics, the coefficient of variation for each metric over 5 runs is also reported in parenthesis

	N_{tot}^{avg}	N_{wc}^{tot}	$\mathrm{Err}_{1,avg}^{ref}$	$\mathrm{Err}_{1,wc}^{ref}$	$\mathrm{Err}_{2,avg}^{ref}$	$\mathrm{Err}_{2,wc}^{ref}$
DoE500	3100 (26.4%)	4000	0.099 (17.5%)	0.114	0.009 (54.8%)	0.016
DoE200	2000 (30.8%)	2800	0.177 (39.6%)	0.263	0.010 (54.9%)	0.018

adjusting metamodel accuracy: the algorithm gradually converges and identifies problematic regions for improving metamodel accuracy, while it demonstrates a very good agreement with reference solutions with great computational savings. Selection of smaller number of experiments to add per iteration leads to greater computational efficiency, but at the potential expense of robustness in identifying the part of Pareto front closer to the anchor points. This can be resolved through a second optimization refinement as discussed earlier.

7 Conclusions

A Kriging-based, iterative optimization framework was developed for reducing the computational burden associated with multi-objective design under uncertainty problems that involve complex numerical models and adopt stochastic simulation for evaluation of the different performance functions. The metamodel was utilized for predicting the model response, whereas its formulation was established in the augmented input space, composed of both the design variables and uncertain model parameters. The probabilistic nature of the metamodel was directly exploited to evaluate the predictive performance measure, i.e., propagate metamodel error from predicted response to system performance functions. The epsilon-constraint approach was adopted for identifying the Pareto front, whereas an iterative optimization scheme was developed to reduce the number of simulations for the expensive system model. At each iteration, a new metamodel is developed utilizing all available support points and a new predictive Pareto set is generated. This set is then compared to the one generated in the precedent iteration across different stopping criteria that evaluate whether new solutions emerged in the current iteration. If convergence has not been established, the algorithm proceeds to the next iteration and new training points (experiments) are obtained to improve accuracy of the metamodel. A hybrid design of experiment (DoE) was established for this purpose combining a space-filling DoE and an adaptive, sample-based DoE that aims at improving accuracy in regions of importance. These regions are separately identified for the design variables and the uncertain model parameters, and sample-based techniques utilizing kernel density estimation were developed to efficiently approximate the distributions defining them.

As illustrative examples, both an analytical example and an engineering example considering the design of nonlinear passive dampers for the suspension systems of a half-car model riding on a rough road were investigated. Results demonstrated the computational efficiency and robustness of the proposed algorithm as it was able to achieve convergence and coverage of the true Pareto front with small computational burden, comparing to an alternative surrogate-based multi-objective optimizer appropriate for problems with expensive design objectives. Both the formulation of the metamodel in the augmented input space and the iterative scheme were shown to contribute to the efficiency of the proposed scheme. With respect to the iterative scheme and the adaptive control of the metamodel accuracy, it was demonstrated that as the iterations progress, the Pareto front gradually approaches the reference solution with increasingly accurate objective function estimates. Solution of the design problem without the proposed iterative scheme was shown to greatly impact the computational efficiency. For the specific examples considered, the metamodel development in augmented input space setting coupled with the iterative, adaptive optimization scheme led to a significant increase in algorithmic efficiency without compromising the quality of the results. Selection of smaller number of experiments to add per iteration led to greater computational efficiency, though at the potential expense of robustness in identifying the complete Pareto front close to the anchor points.

There remain some interesting directions to fully explore and investigate, in order to improve the efficiency and applicability scope of the current algorithm. Rather than balancing the exploitation and exploration based on adaptively allocating number of experiments between space-filling and adaptive DoE, it might be beneficial to develop a more intelligent adaptive DoE that can automatically balance between exploration and exploitation. In addition, the current work uses stationary correlation function for Kriging. The use of alterative non-stationary correlation functions that incorporates problem-specific knowledge might also be helpful.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix 1: Gradient for predictive performance function

The gradient vector $\nabla H_i^{krig}(\mathbf{x})$ of the predictive performance function in (6) corresponds to

$$\begin{aligned}\nabla H_i^{krig}(\mathbf{x}) &= \nabla \mathbb{E}_{p(\boldsymbol{\theta})}\left[h_i^{krig}(\mathbf{x}, \boldsymbol{\theta})\right] \\ &= \mathbb{E}_{p(\boldsymbol{\theta})}\left[\nabla h_i^{krig}(\mathbf{x}, \boldsymbol{\theta})\right]\end{aligned} \tag{32}$$

with $\nabla h_i^{krig}(\mathbf{x}, \boldsymbol{\theta})$ representing the vector of partial derivatives with respect to $\mathbf{x}$,

$$\nabla h_i^{krig}(\mathbf{x}, \boldsymbol{\theta}) = \left[\frac{\partial h_i^{krig}(\mathbf{x}, \boldsymbol{\theta})}{\partial x_1} \cdots \frac{\partial h_i^{krig}(\mathbf{x}, \boldsymbol{\theta})}{\partial x_{n_x}}\right]^T \tag{33}$$

Each of these partial derivatives can be easily obtained based on the derivative information for the Kriging metamodel as

$$\begin{aligned}&\frac{\partial h^{krig}(\mathbf{x}, \boldsymbol{\theta})}{\partial x_l} \\ &= \sum_{m=1}^{n_z}\left(\frac{\partial h^{krig}[\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y}), \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{y})|\mathbf{x}, \boldsymbol{\theta}]}{\partial \mu_m(\mathbf{y})}\frac{\partial \mu_m(\mathbf{y})}{\partial x_l} + \frac{\partial h^{krig}[\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y}), \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{y})|\mathbf{x}, \boldsymbol{\theta}]}{\partial \sigma_m^2(\mathbf{y})}\frac{\partial \sigma_m^2(\mathbf{y})}{\partial x_l}\right)\end{aligned} \tag{34}$$

where $\partial h^{krig}[\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y}), \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{y})|\ \mathbf{x}, \boldsymbol{\theta}]/\partial \mu_m(\mathbf{y})$ and $\partial h^{krig}[\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y}), \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{y})|\mathbf{x}, \boldsymbol{\theta}]/\partial \sigma_m^2(\mathbf{y})$ are obtained through the analytical expression for $h^{krig}[\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y}), \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{y})|\ \mathbf{x}, \boldsymbol{\theta}]$, and derivatives $\partial \mu_m(\mathbf{y})/\partial x_l$ and $\partial \sigma_m^2(\mathbf{y})/\partial x_l$ are directly available through the metamodel (Lophaven et al. 2002). Using stochastic simulation an estimate for the gradient in (32) is obtained as:

$$\nabla \hat{H}_i^{krig}(\mathbf{x}|\{\boldsymbol{\theta}\}_i) = \frac{1}{N_i}\sum_{j=1}^{N}\left(\nabla h_i^{krig}(\mathbf{x}, \boldsymbol{\theta}_i^j)\frac{p(\boldsymbol{\theta}_i^j)}{q_i(\boldsymbol{\theta}_i^j)}\right) \tag{35}$$

Appendix 2: Generation of samples for performance functions

Since both $H_i^{cr}(\mathbf{x})$ and $H_i^{cr}(\mathbf{x}')$ correspond to probabilistic integral with respect to $\boldsymbol{\theta}$, a simplification is established here for their calculation by replacing them with their sample-based estimates in a stochastic simulation setting. The same sample set $\{\boldsymbol{\theta}\}_i^{(k)}$ is used for all estimates, as this choice facilitates consistency in comparisons, and furthermore greatly enhances computational efficiency (as will be shown next). Pairs of conditional samples for $H_i^{cr}(\mathbf{x})$ and $H_i^{cr}(\mathbf{x}')$ are obtained through the following process, termed herein as *conditional realization sampling*

Step 1. Construct the $2N_i \times n_y$ sample matrix $\tilde{\mathbf{Y}} = \left[\mathbf{y}^1, \ldots, \mathbf{y}^{N_i}, \mathbf{y}'^1, \ldots, \mathbf{y}'^{N_i}\right]$ where sample $\mathbf{y}^j$ and $\mathbf{y}'^j$ stand for the *j*th augmented input sample, each composed of $\mathbf{x}'$, respectively, along with each sample $\boldsymbol{\theta}^j$ within $\{\boldsymbol{\theta}\}_i^{(k)}$. Elements of $\tilde{\mathbf{Y}}$ are denoted $\tilde{\mathbf{y}}$. For each response component z_m, the $2N_i \times 2N_i$ correlation matrix $\mathbf{R}_m^{cr}(\tilde{\mathbf{Y}})$ is calculated with *tp*- element $R(\tilde{\mathbf{y}}^t, \tilde{\mathbf{y}}^p)$; *t*,*p* = 1, …,2$N_i$, where, recall, *R*(.,.) stands for the correlation function for the Kriging metamodel. The lower Cholesky factorization of this matrix is then obtained as $\mathbf{C}_j^{cr}(\tilde{\mathbf{Y}})$.

Step 2. $2N_i$ conditional realizations are obtained for each response quantity z_m over the set $\tilde{\mathbf{Y}}$, with each sample having mean $\mu_m(\tilde{\mathbf{y}}^l)$, variance $\sigma_m(\tilde{\mathbf{y}}^l)$, and the samples corresponding to correlation $\mathbf{R}_m^{cr}(\tilde{\mathbf{Y}})$. These samples are given by

$$z_m^{cr}(\tilde{\mathbf{y}}^l) = \mu_m(\tilde{\mathbf{y}}^l) + e_m^l \cdot \sigma_m(\tilde{\mathbf{y}}^l) \tag{36}$$

where e_m^l is the *l*th component of $\mathbf{C}_m^{cr}(\tilde{\mathbf{Y}})\mathbf{w}_m$, with $\mathbf{w}_m$ a vector of $2N_i$ independent identically distributed standard Gaussian variables. Utilizing these samples we get a pair of conditional realization sample for the performance function as:

$$\begin{aligned}\hat{H}_i^{cr}(\mathbf{x}) &= \frac{1}{N_i}\sum_{j=1}^{N_i}\left(\left[h_i\left(\mathbf{z}^{cr}\left(\tilde{\mathbf{y}}^j\right)|\mathbf{x}, \boldsymbol{\theta}_i^j\right)\right]\frac{p(\boldsymbol{\theta}_i^j)}{q_i(\boldsymbol{\theta}_i^l)}\right) \\ \hat{H}_i^{cr}(\mathbf{x}') &= \frac{1}{N_i}\sum_{j=1}^{N_i}\left(h_i\left(\mathbf{z}^{cr}\left(\tilde{\mathbf{y}}^{j+N}\right)|\mathbf{x}', \boldsymbol{\theta}_i^j\right)\frac{p(\boldsymbol{\theta}_i^j)}{q_i(\boldsymbol{\theta}_i^j)}\right)\end{aligned} \tag{37}$$

where $\mathbf{z}^{cr}(\tilde{\mathbf{y}}^l)$ is the vector composed of components $z_m^{cr}(\tilde{\mathbf{y}}^l)$.

Step 3. Step 2 is repeated N_{cr} times utilizing different samples $\mathbf{w}_m$ and therefore obtaining different conditional samples for $\left[\hat{H}_i^{cr}(\mathbf{x})\right]^t$ and $\left[\hat{H}_i^{cr}(\mathbf{x})\right]^t$; $t = 1, \ldots N_{cr}$ This repetition involves minimal computational burden since it merely requires use of different samples for $\mathbf{w}_m$. This efficiency stems from the decision discussed above to use same set for $\boldsymbol{\theta}$ samples (no need to repeat step 1).

Appendix 3: Definition of target densities for adaptive DoE

The domains of interest for $\mathbf{x}$ for defining the target density in DoE context are evidently close to the current Pareto set, since the latter represents the domain where the numerical optimization is expected to converge. Rather than considering, though, the entire set, domains with sufficient metamodel accuracy should be neglected as establishing improvement in those is not expected to provide any additional benefits. These domains ultimately correspond to the ones for which the Pareto set has been already demonstrating convergence based on the criteria discussed in Section 4.3. Based on this observation, the set of retained configurations for DoE $\mathbf{X}_d^{(k)}=\left\{\mathbf{x}_d^{r(k)}; r=1,\ldots,n_d\right\}$ is defined as the elements of $\mathbf{X}_p^{(k)}$ that correspond to (i) $I_{conv}\left(\mathbf{x}|\mathbf{X}_p^{(k-1)}\right)>\delta_{conv}$ or (ii) $I_{cov}\left(\mathbf{x}|\left\{\mathbf{x}_{an,i}^{(k-1)}\right\}_{i=1,2}\right)>\delta_{anch}$. The target density may be ultimately approximated based on $\mathbf{X}_d^{(k)}$ through any desired approach. Here, this is established through multivariate product kernel density estimation (KDE):

$$\tilde{\pi}_d\left(\mathbf{x}|\mathbf{X}_d^{(k)}\right)=\frac{1}{n_d}\sum_{r=1}^{n_d}\tilde{\pi}_d\left(\mathbf{x}|\mathbf{x}_d^{r(k)}\right);\qquad \tilde{\pi}_d\left(\mathbf{x}|\mathbf{x}_d^{r(k)}\right)=\prod_{l=1}^{n_x}\left[\frac{1}{b_l}K\left(\frac{x_l-x_{d,l}^{r(k)}}{b_l}\right)\right] \tag{38}$$

where $K(.)$ is the chosen kernel (for example Gaussian kernel) with b_l the bandwidth parameter defining the spread of the kernel and subscript l is used to describe the l_{th} component of design vector $\mathbf{x}$ (Scott 1992). An illustration of this density is included in the first illustrative example examined in Section 6.

For the uncertain model parameters $\boldsymbol{\theta}$, the domain of interests corresponds to domains in Θ that provide higher contribution towards the integrand representing the two performance functions (in other word IS densities). This is established by introducing the auxiliary density for design configuration $\mathbf{x}$:

$$\pi(\boldsymbol{\theta}|\mathbf{x})=\frac{1}{2}\sum_{i=1}^{2}\frac{|h_i^{krig}(\mathbf{x},\boldsymbol{\theta})|}{\int_\Theta|h_i^{krig}(\mathbf{x},\boldsymbol{\theta})|p(\boldsymbol{\theta})d\boldsymbol{\theta}}p(\boldsymbol{\theta})=\frac{1}{2}\sum_{i=1}^{2}\pi_i(\boldsymbol{\theta}|\mathbf{x}) \tag{39}$$

where the optimal IS density for each performance function, representing the domain of importance for $\boldsymbol{\theta}$ for that function, is defined as

$$\pi_i(\boldsymbol{\theta}|\mathbf{x})=\frac{|h_i^{krig}(\mathbf{x},\boldsymbol{\theta})|p(\boldsymbol{\theta})}{\int_\Theta|h_i^{krig}(\mathbf{x},\boldsymbol{\theta})|p(\boldsymbol{\theta})d\boldsymbol{\theta}}\propto|h_i^{krig}(\mathbf{x},\boldsymbol{\theta})|p(\boldsymbol{\theta}) \tag{40}$$

with ∝ denoting proportionality. Density $\pi(\boldsymbol{\theta}|\,\mathbf{x})$ corresponds to the mixture of these IS densities.

Combining (38)–(39), the DoE target density $\pi_s^{(k)}(\mathbf{x},\boldsymbol{\theta})$ is finally defined as:

$$\pi_s^{(k)}(\mathbf{x},\boldsymbol{\theta})=\tilde{\pi}_d\left(\mathbf{x}|\mathbf{X}_d^{(k)}\right)\pi(\boldsymbol{\theta}|\mathbf{x}) \tag{41}$$

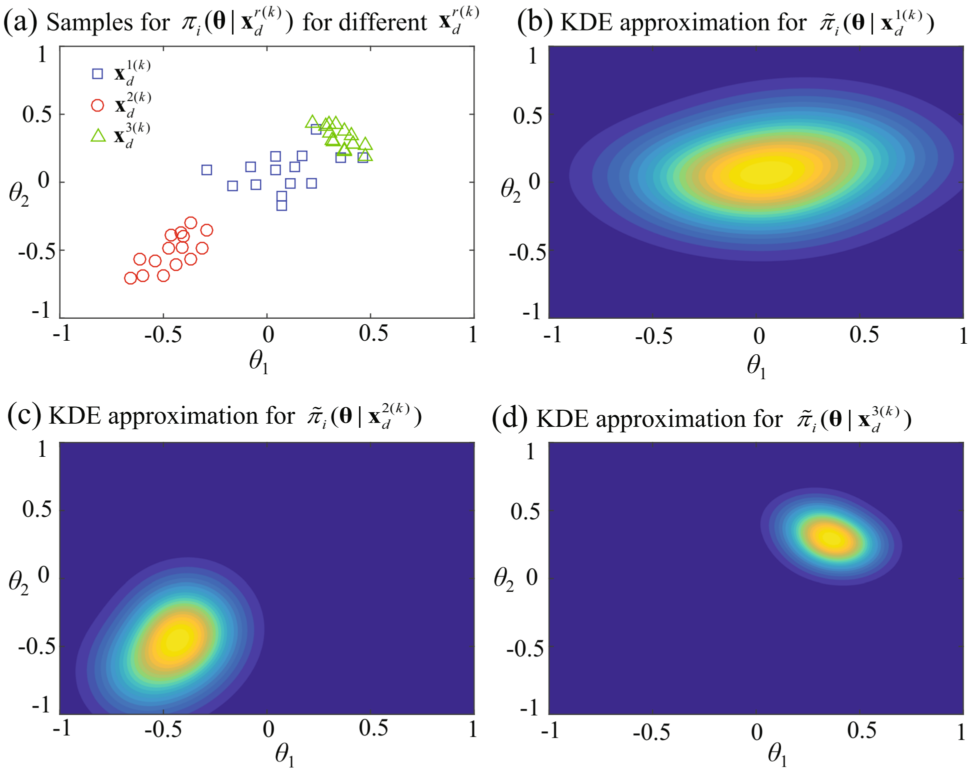


Fig. 9 Demonstration of approximation for π_i $(\boldsymbol{\theta}|\mathbf{x})$ for case with $n_d=3$: (**a**) samples for $\tilde{\pi}_i\left(\boldsymbol{\theta}|\mathbf{x}_d^{r(k)}\right)$ for three different $\mathbf{x}_d^{r(k)}$ and (**b**–**d**) KDE approximations established for each $\tilde{\pi}_i\left(\boldsymbol{\theta}|\mathbf{x}_d^{r(k)}\right)$; $r=1,2,3$

Generating a sample $[\mathbf{x}, \theta]$ from this density, as required in the first step of the adaptive DoE procedure, is decomposed into two tasks: (i) simulate the design variable sample $\mathbf{x}^j$ from mixture-density $\tilde{\pi}_d\left(\mathbf{x}|\mathbf{X}_d^{(k)}\right)$ by first randomly choosing $\mathbf{x}_d^{j(k)}$ from set $\mathbf{X}_d^{(k)}$ and then generating sample $\mathbf{x}^j$ from $\tilde{\pi}_d\left(\mathbf{x}|\mathbf{x}_d^{r(k)}\right)$ and (ii) simulate the uncertain model parameter sample θ^j according to the auxiliary density conditioned on the simulated $\mathbf{x}^j$, $\pi(\theta|\, \mathbf{x}^j)$. Task (ii) is, though, non-trivial, especially when repeated for each different $\mathbf{x}^j$. For this reason, a relaxation is introduced by first replacing $\pi_i(\theta|\, \mathbf{x})$ with $\pi_i\left(\theta|\mathbf{x}_d^{j(k)}\right)$ and then approximating the latter through KDE utilizing samples from it. This approximation requires generation of n_{is} samples from density $\pi_i\left(\theta|\mathbf{x}_d^{r(k)}\right)$ $i = 1, 2$, which can be performed through rejection sampling (Robert and Casella 2004), and then forming approximation $\tilde{\pi}_i\left(\theta|\mathbf{x}_d^{r(k)}\right)$ [yielding similar expression as in (38)]. It can be also seamlessly integrated with the IS density formulation discussed in Section 5.1. Finally, $\pi\left(\theta|\mathbf{x}_d^{r(k)}\right)$ can be substituted by its KDE approximation $\tilde{\pi}\left(\theta|\mathbf{x}_d^{r(k)}\right)$.

Figure 9 demonstrates the DoE target density for θ for a case where $n_d = 3$ (i.e., set $\mathbf{X}_d^{(k)}$ includes three elements) for specific iteration k of the MODU-AIM algorithm and specific performance objective H_i. Part (a) shows samples for $\tilde{\pi}_i\left(\theta|\mathbf{x}_d^{r(k)}\right)$ separately for each $\mathbf{x}_d^{r(k)}$(different symbols) and parts (b–d) the KDE approximations established for $\tilde{\pi}_i\left(\theta|\mathbf{x}_d^{r(k)}\right)$; $r = 1,2,3$. Sample θ^j is ultimately obtained by generating a sample from the specific KDE approximation [from parts (b–d)] that corresponds to the same $\mathbf{x}_d^{r(k)}$ utilized to generated sample $\mathbf{x}^j$ [using approximation $\tilde{\pi}_d\left(\mathbf{x}|\mathbf{x}_d^{r(k)}\right)$]. Note that the combination of the KDE approximations in parts (b–d) of this figure also facilitates the IS approximation in (20). As the algorithm progresses, the density $\tilde{\pi}_i\left(\theta|\mathbf{x}_d^{r(k)}\right)$ is modified iteratively due to variation of the set $\mathbf{X}_p^{(k)}$ or reduction of the retained configurations $\mathbf{X}_d^{(k)}$ due to local convergence of the front. For example, if the front in vicinity of $\mathbf{x}_d^{3(k)}$ has converged for some iteration k, then part (d) of Fig. 9 is no longer considered for the definition of the DoE density for θ. The bigger impact across the iterations of MODU-AIM is on the DoE target density for $\mathbf{x}$, $\tilde{\pi}_d\left(\mathbf{x}|\mathbf{X}_d^{(k)}\right)$, which, as discussed earlier, is illustrated in the first example examined in Section 6.

References

Beck AT, Santana Gomes WJ (2012) A comparison of deterministic, reliability-based and risk-based structural optimization under uncertainty. Probabilistic Engineering Mechanics 12:18–29

Beck JL, Taflanidis A (2013) Prior and posterior robust stochastic predictions for dynamical systems using probability logic. Journal of Uncertainty Quantification 3(4):271–288

Bichon BJ, Eldred MS, Mahadevan S, McFarland JM (2013) Efficient global surrogate modeling for reliability-based design optimization. J Mech Design 135(1):011009

Coelho RF, Lebon J, Bouillard P (2011) Hierarchical stochastic metamodels based on moving least squares and polynomial chaos expansion. Struct Multidiscip Optim 43(5):707–729

Dahlberg T (1978) Ride comfort and road holding of a 2-DOF vehicle travelling on a randomly profiled road. J Sound Vib 58(2):179–187

Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol 16. John Wiley & Sons

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

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

Dubreuil S, Bartoli N, Gogu C, Lefebvre T, Colomer JM (2018) Extreme value oriented random field discretization based on an hybrid polynomial chaos expansion-Kriging approach. Comput Methods Appl Mech Eng

Eldred MS, Giunta AA, Wojtkiewicz SF, Trucano T (2002) Formulations for surrogate-based optimization under uncertainty. Paper presented at the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA,

Gidaris I, Taflanidis AA, Lopez-Garcia D, Mavroeidis GP (2016) Multi-objective risk-informed design of floor isolation systems. Earthquake Engineering & Structural Dynamics 45(8):1293–1313

Goh CK, Tan KC (2006) Noise handling in evolutionary multi-objective optimization. In: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. IEEE, pp 1354–1361

Haimes YY, Ladson L, Wismer DA (1971) Bicriterion formulation of problems of integrated system identification and system optimization. IEEE Transactions on Systems Man and Cybernetics (3):296-&

Hartigan JA, Wong MA (1979) Algorithm AS 136: a K-means clustering algorithm. J R Stat Soc: Ser C: Appl Stat 28(1):100–108

Helton JC, Johnson JD, Oberkampf WL (2004) An exploration of alternative approaches to the representation of uncertainty in model predictions. Reliab Eng Syst Saf 85(1):39–71

Janusevskis J, Le Riche R (2013) Simultaneous kriging-based estimation and optimization of mean response. J Glob Optim 55:313–336

Jia G, Taflanidis AA (2013) Kriging metamodeling for approximation of high-dimensional wave and surge responses in real-time storm/hurricane risk assessment. Comput Methods Appl Mech Eng 261-262:24–38

Jin R, Du X, Chen W (2003) The use of metamodeling techniques for optimization under uncertainty. Struct Multidiscip Optim 25(2):99–116

Jones D, Schonlau M, Welch W (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13:455–492

Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. J Glob Optim 21(4):345–383

Kim IY, De Weck O (2006) Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. Struct Multidiscip Optim 31(2):105–116

Kleijnen JPC (2009) Kriging metamodeling in simulation: a review. Eur J Oper Res 192(3):707–716

Kouri DP, Heinkenschloss M, Ridzal D, van Bloemen Waanders BG (2014) Inexact objective function evaluations in a trust-region algorithm for PDE-constrained optimization under uncertainty. SIAM J Sci Comput 36(6):A3011–A3029

Lagaros ND, Papadrakakis M (2007) Robust seismic design optimization of steel structures. Struct Multidiscip Optim 33(6):457–469

Leotardi C, Serani A, Iemma U, Campana EF, Diez M (2016) A variable-accuracy metamodel-based architecture for global MDO under uncertainty. Struct Multidiscip Optim 54(3):573–593

Liang C, Mahadevan S (2017) Pareto surface construction for multi-objective optimization under uncertainty. Struct Multidiscip Optim 55(5):1865–1882

Lophaven SN, Nielsen HB, Sondergaard J (2002) DACE-A MATLAB Kriging Toolbox. Technical University of Denmark

Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 26:369–395

Mavrotas G (2009) Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. Appl Math Comput 213(2):455–465

Medina JC, Taflanidis A (2014) Adaptive importance sampling for optimization under uncertainty problems. Comput Methods Appl Mech Eng 279:133–162. https://doi.org/10.1016/j.cma.2014.06.025

Medina JC, Taflanidis A (2015) Probabilistic measures for assessing appropriateness of robust design optimization solutions. Struct Multidiscip Optim 51(4):813–834

Medina JC, Taflanidis AA (2013) Adaptive Importance Sampling for Optimization under Uncertainty Using Stochastic Simulation. In: 54th Structures, Structural Dynamics, and Materials Conference AIAA Conference, Boston, MA, April 8-11

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

Müller J (2017) SOCEMO: surrogate optimization of computationally expensive multiobjective problems. INFORMS J Comput 29(4): 581–596

Papadimitriou D, Papadimitriou C (2016) Robust and reliability-based structural topology optimization using a continuous adjoint method. J Risk Unceràinty Eng Part A: Civil Eng. https://doi.org/10.1061/AJRUA1066.0000869

Poles S, Lovison A (2009) A polynomial chaos approach to robust multiobjective optimization. In: Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum für Informatik,

Ren X, Rahman S (2013) Robust design optimization by polynomial dimensional decomposition. Struct Multidiscip Optim 48(1):127–148

Robert CP, Casella G (2004) Monte Carlo statistical methods, 2nd edn. Springer, New York, NY

Ruiz R, Taflanidis AA, Lopez-Garcia VC (2016) Life-cycle based design of mass dampers for the Chilean region and its application for the evaluation of the effectiveness of tuned liquid dampers with floating roof. Bull Earthq Eng 14(3):943–970

Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci 4(4):409–423

Scott DW (1992) Multivariate density estimation: theory, Practise and Visualization. Wiley-Interscience, New York, N.Y

Spall JC (2003) Introduction to stochastic search and optimization. Wiley-Interscience, New York

Teich J (2001) Pareto-front exploration with uncertain objectives. In: International Conference on Evolutionary Multi-Criterion Optimization. Springer, pp 314–328

Verros C, Natsiavas S, Papadimitriou C (2005) Design optimization of quarter-car models with passive and semi-active suspensions under random road excitation. J Vib Control 11(5):581–606

Wang Q, Stengel RF (2000) Robust nonlinear control of a hypersonic aircraft. J Guid Control Dyn 23(4):577–585

Yang B, Yeun Y-S, Ruy W-S (2002) Managing approximation models in multiobjective optimization. Struct Multidiscip Optim 24(2):141–156

Zhang J, Taflanidis AA, Medina JC (2016) Sequential approximate optimization for design under uncertainty problems utilizing Kriging metamodeling in augmented input space. Comput Methods Appl Mech Eng 315(369–395)

Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195