Multi-objective design under uncertainty using a Kriging-based evolutionary optimizer

Jize Zhang
Graduate Student, Dept. of Civil & Environmental Engineering & Earth Sciences, University of Notre Dame, Notre Dame, USA

Alexandros A. Taflanidis
Associate Professor, Dept. of Civil & Environmental Engineering & Earth Sciences, University of Notre Dame, Notre Dame, USA

ABSTRACT: Multi-objective design problems with probabilistic objectives estimated through stochastic simulation are examined in this paper. For the efficient solution of such problems a surrogate model based optimization scheme, termed MODU-AIM, was recently developed by the authors. Foundations of MODU-AIM are the formulation of the surrogate model in the augmented input space, composed of both the design variables and the uncertain model parameters, and an iterative implementation that adaptively controls surrogate model accuracy. At each iteration, a new surrogate model is developed, and a new Pareto front is identified using epsilon-constraint numerical optimization scheme. This front is then compared to the previous iteration's front to examine convergence. If convergence has not been established, a set of refinement experiments is identified for the surrogate model development and process proceeds to the next iteration. In this paper, integration of multi-objective evolutionary optimizers (MOEA) is considered for MODU-AIM. This integration extends MODU-AIM's applicability and numerical efficiency and requires a number of modifications and enhancements to address the unique traits of MOEA optimizers with respect to the Pareto front identification.

1. PROBLEM FORMULATION

Consider an engineering system with design vector $\mathbf{x} \in X \subset \mathbb{R}^{n_x}$ with admissible design space X, and uncertain parameters $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^{n_\theta}$ following a known probability density function (PDF) $p(\boldsymbol{\theta})$ with support Θ. Let $\mathbf{z}(\mathbf{x},\boldsymbol{\theta}) \subset \mathbb{R}^{n_z}$ denote the system's *response vector* (with z_m representing its mth component), obtained through a call to a deterministic computationally intensive simulator. Assume that there are multiple (n_h>1) *performance measure*s for quantifying the response vector's favorability, written as vector $\mathbf{h}(\mathbf{x},\boldsymbol{\theta})$ or $\mathbf{h}[\mathbf{z}\,|\,\mathbf{x},\boldsymbol{\theta}]$, whose ith component is $h_i(\mathbf{x},\boldsymbol{\theta}):\mathbb{R}^{n_x \times n_\theta} \rightarrow \mathbb{R}$. We are interested in the following multi-objective optimization under uncertainty problem:

$$\min_{X} \left\{ \mathbf{H}(\mathbf{x}) = \left[H_1(\mathbf{x}), \cdots, H_{n_h}(\mathbf{x}) \right] \right\} \tag{1}$$

where the ith objective is the expected performance under the probability model $p(\boldsymbol{\theta})$:

$$H_i(\mathbf{x}) == \int_{\Theta} h_i(\mathbf{x},\boldsymbol{\theta})\, p(\boldsymbol{\theta}) d\boldsymbol{\theta}; \quad i = 1,\ldots,n_h \tag{2}$$

If these probabilistic objectives are competing then the design problem in Eq. (1) does not have a unique solution, but rather a set of Pareto-optimal solutions. These solutions are the dominant (or equivalently not dominated) feasible designs, meaning that there is no other design that simultaneously improves on all objectives. The set of all Pareto optimal configurations is denoted as the Pareto set (PS) $\mathbf{X}_p$ with its elements as $\mathbf{x}_p$. The Pareto front (PF) is the representation of the PS on the objective space $\mathbf{H}(\mathbf{X}_p)$.

The numerical solution of the multi-objective problem of Eq. (1) aims at identifying a (discrete) subset of the PF that represent a comprehensive

picture of trade-offs between the performance objectives (Coello et al. 2007). This solution has a considerable computational burden (compared to single-objective optimization problems) due to the need to thoroughly examine the entire design space to evaluate the objective functions trade-offs. This burden is substantially increased for design-under uncertainty applications, especially when the simulator evaluation of $\mathbf{z}(\mathbf{x},\boldsymbol{\theta})$ is expensive, since estimation of each objective entails calculation of the multi-dimensional integral of Eq. (2), which can be reliably performed only though stochastic simulation. To alleviate this burden, surrogate modeling techniques can be employed to replace the complex simulator with a fast-to-evaluate emulator (Eldred et al. 2002). Here the development of the metamodel in the so-called augmented input space (Taflanidis and Beck 2008), composed of both the design variables and the uncertain model parameters is considered, a formulation that can provide substantial computational efficiency benefits. Kriging is adopted as metamodel and the recently developed MODU-AIM (Zhang and Taflanidis 2018) scheme is adopted and enhanced. Next the Kriging-based optimization is reviewed before discussing the proposed enhancements.

2. KRIGING-BASED OPTIMIZATION

2.1. Fundamentals for Kriging approximation

As in (Zhang and Taflanidis 2018) the metamodel output is chosen to correspond to the response vector **z**, instead of performance objective h_i, and the input, denoted herein **y**, to combination of both **x** and **θ**, $\mathbf{y}=[\mathbf{x};\boldsymbol{\theta}]$. For forming the 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 Gaussian process approximation for z_m based on observations $\mathbf{Y}=[\mathbf{y}^1\ldots\mathbf{y}^n]^T$ (Sacks 1989):

$$M_m(\mathbf{y}\,|\,\mathbf{Y}) \sim N\left(\mu_m(\mathbf{y}),\sigma_m^2(\mathbf{y})\right) \tag{3}$$

where $N(a,b)$ stands for Gaussian distribution with mean a and variance b, and $\mu_m(\mathbf{y})$ and $\sigma_m^2(\mathbf{y})$ correspond to the predictive mean and variance for the output z_m. Once the Kriging metamodel is established, the predictive mean and variance can be provided with small computational burden (Lophaven et al. 2002). For the entire vector **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 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.

Based on the response approximation $M_{\mathbf{z}}(\mathbf{y}\,|\,\mathbf{Y})$, the following *predictive performance measure* h_i^{krig} can be derived to approximate h_i:

$$\begin{aligned} h_i^{krig}(\mathbf{x},\boldsymbol{\theta}) &= \mathbb{E}_M\left[h_i[\mathbf{z}\,|\,\mathbf{x},\boldsymbol{\theta}]\,|\,\mathbf{Y}\right] \\ &= \int_{\mathbb{R}^{n_z}} h_i[\mathbf{z}\,|\,\mathbf{x},\boldsymbol{\theta}]\phi\left(\frac{\mathbf{z}-\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y})}{\boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{y})}\right)d\mathbf{z} \\ &= h_i^{krig}[\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y}),\boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{y})\,|\,\mathbf{x},\boldsymbol{\theta}] \end{aligned} \tag{4}$$

where $\mathbb{E}_M[.\,|\,\mathbf{Y}]$ denotes conditional expectation under $M_{\mathbf{z}}(\mathbf{y}\,|\,\mathbf{Y})$, ϕ corresponds to the standard Gaussian PDF and last equality stresses fact that integral of Eq. (4) can be analytically calculated for many practical applications [further discussions included in (Zhang and Taflanidis 2018)]. By taking conditional expectation in Eq. (4), the metamodel error has been explicitly incorporated into the metamodel-based approximation for h_i, something that can offer substantial benefits for design under uncertainty problems (Zhang et al. 2017).

Approximation of Eq. (4) leads to the following metamodel-based predictive objective H_i^{krig} for H_i:

$$H_i^{krig}(\mathbf{x}) = \int_{\Theta} h_i^{krig}(\mathbf{x},\boldsymbol{\theta})\,p(\boldsymbol{\theta})\,d\boldsymbol{\theta} \tag{5}$$

and to estimate using Monte Carlo with Importance Sampling (IS):

$$\hat{H}_i^{krig}\left(\mathbf{x}\,|\,\{\boldsymbol{\theta}\}_i\right) = \frac{1}{N_i}\sum_{j=1}^{N_i}\left(h_i^{krig}(\mathbf{x},\boldsymbol{\theta}_i^j)\frac{p(\boldsymbol{\theta}_i^j)}{q_i(\boldsymbol{\theta}_i^j)}\right) \tag{6}$$

which can be efficiently estimated exploiting Kriging's numerical efficiency. In Eq. (6) $q_i(\boldsymbol{\theta})$ is the IS proposal density introduced to enhance the stochastic simulation accuracy for the estimation of the *i*th objective and $\{\boldsymbol{\theta}\}_i = \{\boldsymbol{\theta}_i^j, j = 1,\ldots,N_i\}$ is the sample set from $q_i(\boldsymbol{\theta})$. Selection of N_i and $q_i(\boldsymbol{\theta})$ that establishes a target accuracy for the estimate in Eq. (5) within the optimization framework are discussed in detail in (Zhang and Taflanidis 2018).

Utilizing the approximated objectives in Eq. (6), the following approximate multi-objective problem is established:

$$\min_{X}\left\{\hat{\mathbf{H}}^{krig}(\mathbf{x}) = \left[\hat{H}_1^{krig}(\mathbf{x}),\cdots,\hat{H}_{n_h}^{krig}(\mathbf{x})\right]\right\} \qquad (7)$$

This problem can be solved by any appropriate multi-objective optimizer. The important question is how to efficiently obtain a predictive PS for Eq. (7) that approximates the actual PS of Eq. (1)well. This efficiency is directly quantified by the number of experiments needed for the metamodel development. The adaptive iterative MODU-AIM scheme proposed recently by the authors (Zhang and Taflanidis 2018) is utilized for this purpose.

2.2. MODU-AIM optimization overview

At each MODU-AIM iteration (*k*), the metamodel is constructed from all current available experiments and is used to create the current approximation to the performance objectives $\hat{H}_i^{krig(k)}$ and subsequently, identify the design configurations belonging to the predictive PS $\mathbf{X}_p^{(k)} = \{\mathbf{x}_p^{r(k)}; r = 1,...,n_p^{(k)}\}$. This is established by solving optimization of Eq. (7) through epsilon-constraint approach (Mavrotas 2009), which decomposes the multi-objective optimization into a number of single-objective constrained optimization problems. Then the current PS are compared to the PS in the previous iteration $\mathbf{X}_p^{(k-1)} = \{\mathbf{x}_p^{r(k-1)}; r = 1,...,n_p^{(k-1)}\}$. If convergence is not reached, 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, and total computational effort can be controlled by iteratively evaluating convergence properties and adaptively controlling metamodel accuracy by identifying regions of importance for the multi-objective problem at hand.

The applicability of MODU-AIM is somewhat constrained by the adoption of epsilon-constraint as optimizer. This limits the extension beyond bi-objective problems, as computational burden of method drastically increases (Miettinen 2012), as well as the size of the PS, since identification of each member requires a separate optimization. The reliance on epsilon-constrain is addressed next.

3. EVOLUTIONARY-BASED MODU-AIM

Over the past decades, a number of multi-objective evolutionary algorithms (MOEAs) have been suggested (Coello et al. 2007). MOEAs directly identify a population of PS candidate solutions under some fitness assignment and selection operators. The primary reason for their popularity is due to their inherent parallelism and their capability to discover multiple Pareto-optimal solutions in a single optimization run, as well as their easily scalability to many-objective problems (n_h>2). To benefit from such advantages of MOEAs, we proposed an improved version of MODU-AIM that uses one of the popular MOEA techniques, the NSGA-II solver (Deb et al. 2002) to find predictive Pareto-optimal solutions in each iteration. In what follows, we provide several modification and enhancement strategies to MODU-AIM that takes into account the use of MOEA optimizers.

3.1. Importance sampling proposal density

In original MODU-AIM, the predictive PS in every iteration are always well-distributed and relatively small (10-20 solutions). Consequently, the IS proposal density can be chosen as the mixture of KDE optimal IS densities for all $\mathbf{x}_p$ in the predictive PS. Under MOEA, two issues need to be addressed before using this approach. First, the predictive PS obtained with MOEA lacks guarantee to be as well distributed as ones obtained using epsilon-constraints optimizations.

They can be either over- or under populated in some particular sub-regions. Secondly, in order to fully utilize the advantage of MOEA, PS size are usually chosen significantly bigger than the ones under epsilon-constraints (several hundreds). It is then unnecessary to construct and mix optimal IS densities from all members in PS as neighboring solutions should have similar IS densities.

Our remedy is to use clustering. We first cluster current PS by their predictive objectives $\hat{\mathbf{H}}^{krig(k)}(\mathbf{x})$ (pre-normalized into [0, 1] range) with K-Means clustering (Hartigan and Wong 1979) into n_c clusters. Those solutions whose predictive objective $\hat{\mathbf{H}}^{krig(k)}(\mathbf{x})$ is nearest to cluster centroids are further considered. The final IS proposal only uses KDE optimal IS densities for those solutions as the mixture component. Such clustering ensures that the selected subset of points are diverse in the objective space and can represent the whole PF well.

3.2. Stopping criteria

Stopping criteria, evaluating whether convergence has been established, are critical for the computational efficiency of MODU-AIM. This is established by assessing the performance discrepancy between previous and current PF. Upon convergence, the precedent PF should also be near Pareto-optimal in current iteration. In original MODU-AIM (Zhang and Taflanidis 2018), this performance discrepancy was *probabilistically* assessed through pairwise comparisons between members in precedent and current PS, leveraging conditional realization (CR) of the metamodel predictions to incorporate the metamodel error in the estimation (leading to a *probabilistic* assessment as mentioned above). However, the computational complexity for CR approach makes it impractical to the MOEA-based implementation, as the latter might have hundreds of members in PS.

To circumvent this challenge, stopping criteria utilizing directly the predictive performance objectives of Eq. (5), instead of CR of these objectives, are established here. Specifically, we compare the performance of previous PS $\mathbf{X}_p^{(k-1)}$ to current one $\mathbf{X}_p^{(k)}$ under two popular measures for MOEAs (Yen and He 2014). The first one is the maximum spread (MS) measuring how well the previous PS $\mathbf{X}_p^{(k-1)}$ can cover the PF in current iteration. The second one is the ratio of non-dominated individuals (RNI), measuring how many previous PS members are non-dominated by current PS. Both measure are bounded in the interval of [0, 1]. As iterations of MODU-AIM progress, and convergence is gradually achieved, the current PF should not significantly extend to new regions. As such, convergence is assessed by checking whether MS and RNI of previous PS exceed some pre-specified values (e.g., 0.8) when compared to the current PS.

3.3. Design of experiments

If convergence has not been achieved, the current experiments needs to be enriched to inform the development of a more accurate metamodel. We adopt the adaptive sampling-based DoE framework proposed in (Zhang and Taflanidis 2018) with a few enhancements and modifications to make it compatible with MOEA. The sampling-based DoE has the ability of identifying multiple new experiments simultaneously. It achieves so by first locating a large population of candidate experiments using a target density $\pi(\mathbf{x},\boldsymbol{\theta})$ to represent domains of interests in the input domain, and then using a utility function to incorporate metamodel accuracy in the DoE, by prioritizing candidate experiments with larger predicted error. Finally clustering is conducted on the prioritized candidates to eliminate close-proximity solutions.

The modifications here focus on the definition of target densities. For $\mathbf{x}$, design candidates should be generated around non-convergent PS regions. In original MODU-AIM, such non-convergent regions are discretely represented by current Pareto optimal solutions that probabilistically dominate any member in the previous PS, with probability of dominance evaluated using conditional realizations of the metamodel predictions. If the set of such solutions is denoted as $\mathbf{X}_d^{(k)} = \{\mathbf{x}_d^{r(k)}; r = 1,...,n_d\}$, then the

target density for $\mathbf{x}$ is chosen as a kernel-based density $\tilde{\pi}(\mathbf{x}\,|\,\mathbf{X}_d^{(k)})$. In this paper, we replace the probabilistic dominance by a more efficient check for non-convergence. This is done by simply looking at the performance of current Pareto optimums in previous iteration, and retain the solutions being dominated by previous PS as non-convergent solutions:

$$\mathbf{X}_d^{(k)} = \{\mathbf{x} \in \mathbf{X}_p^{(k)} \,|\, \exists \mathbf{x}' \in \mathbf{X}_p^{(k-1)}, \forall i = 1,\cdots,n_h, \; \hat{H}_i^{krig(k-1)}(\mathbf{x}') \leq \hat{H}_i^{krig(k-1)}(\mathbf{x})\} \tag{8}$$

The principle behind definition of Eq. (8) is that if a current Pareto optimal solution is dominated with respect to the solution of the approximate multi-objective problem in the previous iteration, significant enhancements must have been locally achieved around this solution at the current iteration, which means that its domain should be regarded as non-converged and considered for further exploitation.

For $\boldsymbol{\theta}$, the domain of interest corresponds to domains in Θ that provide higher contribution towards the integrand representing performance functions for each specific $\mathbf{x}$. The optimal IS densities can be good choices here. In the original MODU-AIM, this is established by introducing the auxiliary density as an equal-weight mixture of the IS densities for each objective. This effectively means that for any design configuration, all objective are treated equally, as different IS densities generate same amount of candidate experiments. This however neglects the fact that different objectives have different levels of importance regarding the particular design configuration $\mathbf{x}$. For example, if the design is the minimizer for one objective, reducing the metamodeling error for important domains of the other objectives would be far less critical (practically inconsequential) than the objective it minimizes.

In this paper, we compute the importance of different objectives to the design configuration $\mathbf{x}$ using the Tchebycheff procedure (Miettinen 2012). This approach finds the unique weight vector ($\boldsymbol{\lambda}(\mathbf{x}_p) = [\lambda_1(\mathbf{x}_p),\cdots,\lambda_{n_h}(\mathbf{x}_p)]$) that sum up to one and associate each Pareto optimum $\mathbf{x}_p$ of Eq. (7) to the minimizer of a composited single-objective as:

$$\mathbf{x}_p = \arg\min_X \left[\max_{i=1,\cdots,n_h} \{\lambda_i(\mathbf{x})[\hat{H}_i^{krig}(\mathbf{x}) - \hat{H}_{i,\min}^{krig}]\} \right] \tag{9}$$

where $\hat{H}_{i,\min}^{krig}$ corresponds to the minimum of the ith objective. Using the PS, one can then approximately identify the weight vector $\boldsymbol{\lambda}(\mathbf{x}_p)$ for each solution, for example, through exhaustive search on a large sample set of $\boldsymbol{\lambda}$ which sum up to one and is inside the space $[0, 1]^{nh}$. For selecting the target density for $\boldsymbol{\theta}$, each optimal IS proposal density is ultimately weighted proportional to $\boldsymbol{\lambda}$:

$$\pi(\boldsymbol{\theta}\,|\,\mathbf{x}_p) = \frac{1}{n_h}\sum_{i=1}^{n_h} \lambda_i(\mathbf{x}_p)\pi_i(\boldsymbol{\theta}\,|\,\mathbf{x}_p) \tag{10}$$

where $\pi_i(\boldsymbol{\theta}\,|\,\mathbf{x})$ is the optimal IS density for the ith objective (that is, the density proportional to the integrand of the objective)

The target density is ultimately defined by combining $\tilde{\pi}(\mathbf{x}\,|\,\mathbf{X}_d^{(k)})$ for $\mathbf{x}$ (sampled first) with $\pi(\boldsymbol{\theta}\,|\,\mathbf{x}_p)$ for $\boldsymbol{\theta}$ (conditionally sampled second), where the latter is evaluated for the specific $\mathbf{x}_p$ that corresponds to the respective Kernel selected for $\tilde{\pi}(\mathbf{x}\,|\,\mathbf{X}_d^{(k)})$.

4. REVIEW OF ALGORITHM

When combining the previous ideas, one can formulate the following optimization approach, termed MODU-AIMe

***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 $h_i^{krig(k)}(\mathbf{x},\boldsymbol{\theta})$.

***Step* 3** (IS density formulation): Cluster the PS in previous iteration $\mathbf{X}_p^{(k-1)}$ into n_c centroids. For each centroid, identify the Pareto set member that is closest to it ($\mathbf{x}_p^{r\,(k-1)}$), and obtain n_{is}

samples from all of its optimal IS densities $\pi_i(\boldsymbol{\theta} \mid \mathbf{x}_p^{r\,(k-1)})$. For each objective, the IS density $q_i^{(k)}(\boldsymbol{\theta})$ is formulated through kernel density approximation combining all corresponding samples. In the first iteration skip this step and use $q_i^{(1)}(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ [no prior information available].

***Step* 4** (Stochastic simulation sample generation): Simulate set $\{\boldsymbol{\theta}\}_i^{(k)}$ of N_i samples from $q_i^{(k)}(\boldsymbol{\theta})$ as the stochastic simulation sample set for the current iteration.

***Step* 5** (PS identification): Solve optimizations described by Eq. (7) with NSGA-II optimizer to obtain current Pareto set $\mathbf{X}_p^{(k)}$.

***Step* 6** (Stopping criteria): Check if the stopping criteria have been reached as detailed in Section 3. 2. If they are met or any desired limit on total computational effort has been exhausted, the optimization process is terminated.

***Step* 7** (Refinement DoE): If optimization process is not terminated, employ the adaptive sampling-based DoE strategy (Section 3.3) to obtain total of n_a training points.

***Step* 8** (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 7 and combine with the previous observations in a database over all iterations. Proceed back to Step 2 advancing to k+1st iteration.

5. ILLUSTRATIVE EXAMPLE

For the illustrative example, the design of bilinear passive dampers for the suspension of a half-car nonlinear model riding on a rough road is considered.

5.1. Numerical details

Numerical details are included in (Zhang and Taflanidis 2018). The n_x=4-dimensional design variable contains average damping coefficients C^l and the percentage increase of damping r^n for upwards movement of the suspension (creating the bilinear damping force). They are allowed to be separately selected for the front and rear dampers (distinguished by subscript f or r), leading to $\mathbf{x} = [C_f^l \quad r_f^n \quad C_r^l \quad r_r^n]^T$. A total of n_θ=15 uncertain model parameters are considered, including modelling parameters for the car model, the driving velocity and the road surface roughness (modeled as a stochastic process).

Performance objectives chosen are related to the ride comfort and road holding, corresponding to typical objective examined tor suspension design (Dahlberg 1978). For the ride comfort the fragility related to the root mean square statistics (RMS) of the 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. For the acceleration fragility, log-normal characteristic are assumed with threshold b=1 m/s^2 defining acceptable performance and coefficient of variation σ_b=5% for the fragility. This leads to the following performance measures

$$h_1(\mathbf{x},\boldsymbol{\theta}) = \Phi\left[\frac{\ln(RMS_{ac}) - \ln(b)}{\sigma_b}\right] \qquad (11)$$
$$h_2(\mathbf{x},\boldsymbol{\theta}) = RMS_{tf} + RMS_{tr}$$

The response vector corresponds to the log of the RMS acceleration and the RMS tire forces with $\mathbf{z}$=[ln(RMS_{ac}) RMS_{tf} RMS_{tr}]^T. The predictive performance objectives in Eq. (4) are

$$h_1^{krig}(\mathbf{x},\boldsymbol{\theta}) = \Phi\left[\frac{\mu_1(\mathbf{y}) - \ln(b)}{\sqrt{\sigma_b^2 + \sigma_1^2(\mathbf{y})}}\right] \qquad (12)$$
$$h_2^{krig}(\mathbf{x},\boldsymbol{\theta}) = \mu_2(\mathbf{y}) + \mu_3(\mathbf{y})$$

For the algorithm numerical settings, the number of the initial support points n_{init} and the additional refinement support points n_a are both chosen as 200. The algorithm terminates when both the MS and RNI of precedent PS exceeds 0.8 in current iteration. For the NSGA-II optimizer, the size of final obtained PS is set to n_p = 200. The other algorithm parameter settings are identical to (Zhang and Taflanidis 2018). In the sample run, MODU-AIMe converges within 6 iterations, using a total of 1200 experiments. For verification purpose, a set of 15 well-distributed Pareto optimum are obtained with a total number of 3,590,000 function evaluations using a costly global optimization approach (Zhang and

Taflanidis 2018). This set will be further denoted as the reference PS.

5.2. Results and discussion

To show that the proposed framework can correctly identify the actual PF, we first obtain the actual objectives (i.e., estimated with the exact numerical model) of the predictive PS identified in different iterations of MODUM-AIMe, and depict them along with the reference PS in Figure 1. As the computational cost to estimate actual objective function on every predictive Pareto optimums would be prohibitive, here we first conduct clustering as described in Section 3.1 to obtain 10 representative members per iteration, and then estimate the objectives on those members only.

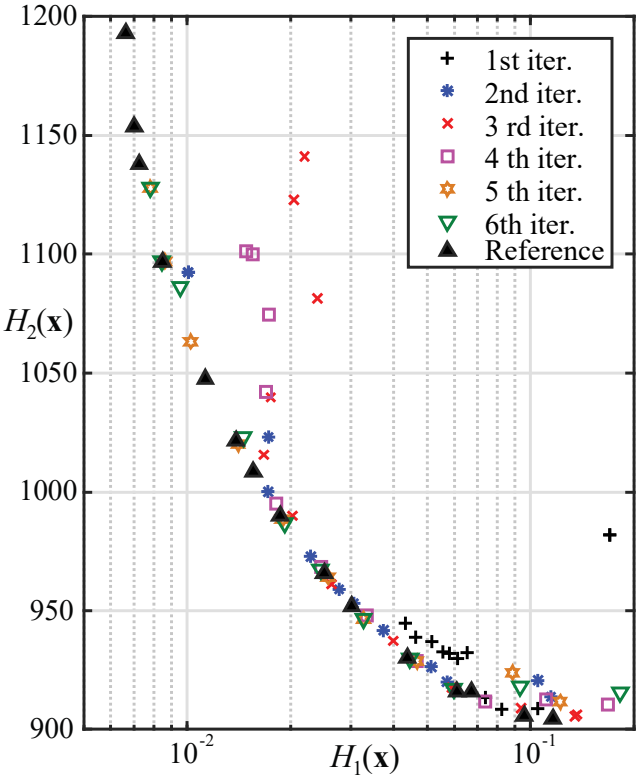

Figure 1. Evolution of Pareto front across the different iterations of MODU-AIMe. Evaluation is performed through the exact numerical model.

Is it clear from this figure that as the iteration increases better agreement is established between the MODUM-AIMe and reference PF. This demonstrates that proposed framework can indeed adaptively improve the metamodel accuracy and subsequently the quality of the identified solutions across the iterations, facilitating a convergence to reference solutions with substantial computational savings (1200 model evaluations only).

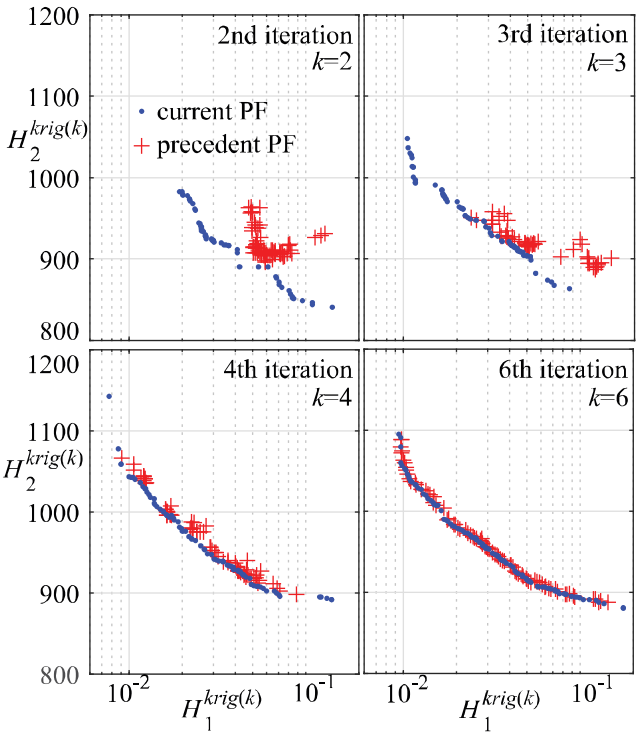

Figure 2. Comparison between precedent (red pluses) and current Pareto front (blue dots) across the different iterations of MODU-AIMe. Evaluation is performed through the current metamodel

To evaluate the efficiency of the proposed stopping criteria, Figure 2 plots the predictive objectives of current PS [i.e., $\hat{\mathbf{H}}^{krig(k)}(\mathbf{X}_p^{(k)})$] in current (kth) iteration as well as the updated objective values of the previous (k-1) PS [i.e., $\hat{\mathbf{H}}^{krig(k)}(\mathbf{X}_p^{(k-1)})$]. By visual inspection one can see that the discrepancy between precedent and current PS indeed decreases. This can be seen from the increasing maximum spread (MS) of the precedent PF (see the distance between extremes in precedent and current PS), but also by more and more non-dominated members in the precedent PS (measured by RSI). Overall, this plot justifies the use of proposed stopping criteria.

6. CONCLUSIONS

Multi-objective design problems with probabilistic objectives estimated through stochastic simulation were examined. Approach extends the authors' previous Kriging-based iterative optimization scheme MODU-AIM, by the integration of multi-objective evolutionary optimizers (MOEA). This use of MOEA has the promise of broadening the applicability and enhancing the numerical efficiency of MODU-AIM, but requires modifications to address the unique traits of MOEA optimizers, such as the lack of guarantee of well-distributed solutions and the large solution set. Adjustments to the importance sampling density selection as well as to the stopping criteria were proposed that are compatible with MOEA's large PS size. A modified design of experiment (DoE) method that is compatible with MOEA was also proposed. The modified DoE established additional enhancements to accounts for different local importance of objectives. The computational framework was applied to an engineering problem considering the design of bilinear passive dampers for the suspension of a half-car nonlinear model riding on a rough road. The example demonstrated that the proposed framework can correctly converge to the true Pareto optimums with great computational efficiency.

7. REFERENCES

Coello, C. A. C., Lamont, G. B., and Van Veldhuizen, D. A., *Evolutionary algorithms for solving multi-objective problems*, Springer, 2007

Dahlberg, T., Ride comfort and road holding of a 2-DOF vehicle travelling on a randomly profiled road. *Journal of Sound and vibration*, 58(2), 179-187, 1978.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197, 2002.

Eldred, M., Giunta, A., Wojtkiewicz, S., and Trucano, T. "Formulations for surrogate-based optimization under uncertainty." *Proc., 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization*, 5585.

Hartigan, J. A., and Wong, M. A., Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100-108, 1979.

Lophaven, S. N., Nielsen, H. B., and Sondergaard, J., DACE-A MATLAB Kriging Toolbox. Technical University of Denmark, 2002.

Mavrotas, G., Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2), 455-465, 2009.

Miettinen, K., *Nonlinear multiobjective optimization*, Springer Science & Business Media, 2012

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

Taflanidis, A., and Beck, J., Stochastic subset optimization for optimal reliability problems. *Probabilistic Engineering Mechanics*, 23(2-3), 324-338, 2008.

Yen, G. G., and He, Z., Performance metric ensemble for multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 18(1), 131-144, 2014.

Zhang, J., and Taflanidis, A. A., Multi-objective optimization for design under uncertainty problems through surrogate modeling in augmented input space. *Structural and Multidisciplinary Optimization*, 1-22, 2018.

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