

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Computational Intelligence and Its Applications in Uncertainty-Based Design Optimization

Ali Asghar Bataleblu

Abstract

The large computational cost, the curse of dimensionality and the multidisciplinary nature are known as the main challenges in dealing with real-world engineering optimization problems. The consideration of inevitable uncertainties in such problems will exacerbate mentioned difficulties as much as possible. Therefore, the computational intelligence methods (also known as surrogate-models or metamodels, which are computationally cheaper approximations of the true expensive function) have been considered as powerful paradigms to overcome or at least to alleviate the mentioned issues over the last three decades. This chapter presents an extensive survey on surrogate-assisted optimization (SAO) methods. The main focus areas are the working styles of surrogate-models and the management of the metamodels during the optimization process. In addition, challenges and future trends of this field of study are introduced. Then, a comparison study will be carried out by employing a novel evolution control strategies (ECS) and recently developed efficient global optimization (EGO) method in the framework of uncertainty-based design optimization (UDO). To conclude, some open research questions in this area are discussed.

Keywords: computational intelligence, metamodeling, surrogate-assisted optimization, uncertainty-based design optimization

1. Introduction

Motivated by industrial demands and development of more powerful optimization techniques, the engineering design community has undergone a major transformation. They are continually seeking new optimization challenges and to solve increasingly more complicated real-world engineering problems in the shortest feasible time. In order to achieve the best solution in dealing with complex real-world engineering optimization, the classical optimization methods are weak in convergence. In solving these design problems, an evolutionary algorithm may require thousands of function evaluations in order to provide a satisfactory solution, whereas each evaluation requires hours of computer run-time. To overcome such difficulties, researchers have applied sampling-based learning methods such as artificial neural networks, radial basis functions, and polynomial model. These methods can 'learn' the problem behaviors and approximate the function value. These approximation models can speed up the function evaluation as well as

estimation the function value with an acceptable accuracy. Also, they can improve optimization performance and provide a better final solution. However, the application of computational intelligence methods to expensive optimization problems is not straightforward. It is important to note that accuracy is the most important criterion for evaluating a metamodel, since metamodels with a low accuracy may lead to local optima, or may even fail to obtain a satisfactory solution. Nevertheless, the choice of the surrogate model is depending on the design problem [1].

Viana et al. [2] and Jin [3] have presented a survey study about metamodeling techniques and their application in the design and analysis of computer experiments. Moreover, they proposed the future work directions to handle more complex simulations. The metamodeling/surrogate-modeling techniques approximate the real model in the entire design space bound that could help to reduce the running time of a complex problem considerably [4]. The simulation-based design problems using metamodels is reviewed extensively in [5–8]. Furthermore, researchers at Boeing and Rice University have proposed a number of mathematical techniques for application of metamodels in optimization problems [2, 9]. They have introduced some software packages to expedite the design and optimization process by using metamodels. These packages include “Optimus” developed by Tzannetakis et al. [10] and “DAKOTA” developed by Adams et al. [11]. Some of the common metamodeling techniques are including response surface method (RSM), artificial neural networks (ANN), Kriging, radial basis functions (RBF), and support vector regression (SVR) [12].

In the recent years, the large number of research and literature indicates the importance of using metamodeling techniques in the optimization. Horng and Lin [13] have proposed an evolutionary algorithm optimizer using metamodels in the ordinal optimization framework. They have used their algorithm to solve a stochastic optimization problem with a huge discrete design space. Sóbester et al. [14] have presented a research on improving the accuracy of metamodels in engineering design problems. Gong et al. [15] have proposed a metamodel with the small computational cost for design by evolutionary optimization algorithms. In order to consider the low and high fidelity model’s information and make a trade-off between accuracy and computational cost, Zhou et al. [16] have introduced an active learning strategy for application of metamodeling. Belyaev et al. [17] have presented a new tool namely GTApprox to generate medium-scale metamodels for industrial design. Sun et al. [18] have introduced a swarm optimization algorithm based on the surrogate models. Recently, a strategy for reducing the running time has presented by Sayyafzadeh [19] that is based on a self-adaptive metamodeling approach. Also, a number of metamodeling strategies that could be used for the uncertainty-based design optimization have reviewed by Chatterjee et al. [20].

Despite the recent advances in the design optimization tools, researchers are still trying to surmount some other issues such as curse of dimensionality, the numerical noise, and handling mixed discrete/continuous variables. Surrogate-assisted optimization (SAO) and the evolution control strategies (ECS) are two newly developed methods in this field of area [1]. Both of these strategies can be applied offline or online. The main difference between these two methods is the management of using metamodels instead of real models during the optimization process. In the SAO strategy, metamodels are substituted for real models directly, but in the ECS strategy, metamodels are substituted for real models in some of the optimizer design points. Furthermore, metamodels that are used offline are not updated while the optimization is ongoing, whereas online metamodels are adaptively updated during the optimization process and can progressively improve the accuracy of the metamodels [12]. One of the known SAO strategies is known as the efficient global

optimization (EGO) that operates based on the approximation of responses using a Gaussian process (e.g. Kriging or SVR) [1].

In order to build globally accurate metamodels, offline SAO methods may require more sample points and they may be computationally expensive. Instead, the online SAO methods could train with fewer sample points. One of the weaknesses of online SAO methods is that the few numbers of sample points in the first iterations can lead to a poor predictive capability of the metamodel. Therefore, this can entice the optimizer into a local optimum or infeasible regions in the design space [1]. To surmount them all, researchers have presented some techniques to call real models and metamodels beside each other during the optimization. For instance, the real model can be used to correct the fitness value of some/all individuals in some generations of evolutionary optimization algorithms. This is known as the management of the metamodels or the evolution control and has been applied in many literatures [12]. However, the best time for calling the real model or the metamodel is still a major challenge for the metamodels' management.

This chapter is introduced for the wide field of research and can be applied by readerships who are interested in the development of computational intelligence techniques for nowadays' expensive optimization problems. Moreover, a novel ECS that benefits from managing the use of metamodels for increasing the optimization accuracy is proposed in this chapter. The performance benefits of this proposed strategy include decreasing the computational cost as well as providing a global or near-global optimum solution. It is important to note that this strategy can be applied for both deterministic and non-deterministic optimization problems, with any optimization algorithms [1].

This chapter is organized as follows. Section 2 introduces the metamodeling approximation. Section 3 presents the proposed ECS strategy. Section 4 presents applications of the proposed strategy to some mathematical benchmark problems, and numerical results are discussed in detail. Section 5 presents the research conclusions as well as some directions for the future research.

2. Meta-modeling

Extensive research on design and optimization of engineering problems using metamodeling techniques has been done. These research fields are including sampling, metamodeling, validation, management and application, and so on. Over the years it has become prove that metamodeling provides a decision criteria role for designers [7].

Metamodeling involves (a) choosing an experimental design for generating design points, (b) function evaluation of generated design points, and then (c) choosing a model to represent the data and fitting the model to the observed data (see **Figure 1**).

After building the metamodels from the available dataset, the accuracy of the models should carefully be checked. When the metamodel is found to have acceptable accuracy, it can be employed for considered design and optimization studies. The metamodel type that is suitable for the approximation could vary depending on the intended use or the underlying problem's physic and design space. Different datasets could be appropriate for building metamodels. The process of where pick out the design points in the design space, i.e. how to spread the design points within the complete design space, is called the design of experiments (DOE) [21].

There are several options for each of metamodeling steps as shown in **Figure 2**, and three predominant ones are highlighted. For example, response surface methodology usually employs central composite designs, second order polynomials, and

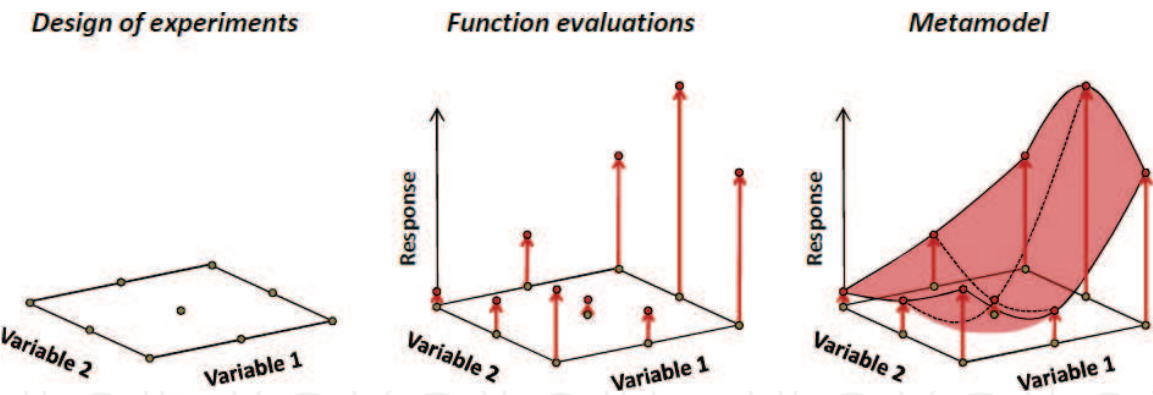


Figure 1. The concept of metamodel creation [21].

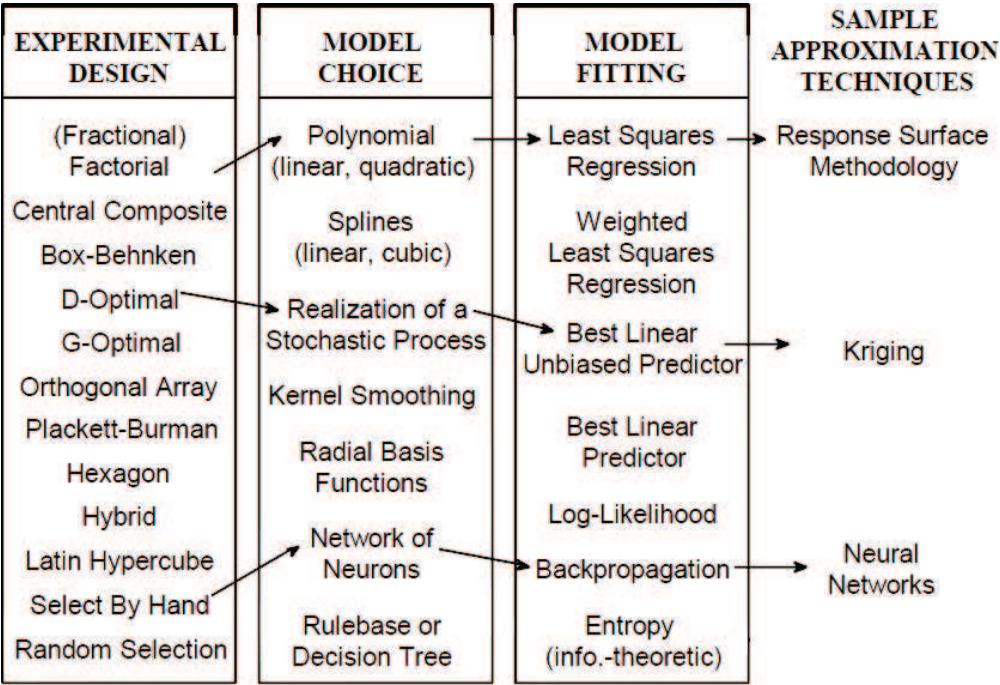


Figure 2. Metamodeling techniques [5].

least squares regression analysis while building a neural network involves fitting a network of neurons by means of back-propagation to data which is typically hand selected [5].

Several strategies exist for finding the optimal solution using metamodel-based design optimization (MBDO). In what follows, a brief overview of several DOEs, metamodel choice and metamodel fitting, and some strategies of MBDO will be explained, respectively.

2.1 Design of experiments

The gathering a set of input-output data is the first step to build a metamodel and this dataset is known as the training set. The DOE is also the theory that helps to select best samples from the design space to cover everywhere. Based on the DOE theory, it is better that the training sets be space-filling and non-collapsing [1]. This signifies the importance of sampling efficiency in the generation of the training set for building an appropriate metamodel. This field has been a challenging research area among metamodeling researchers.

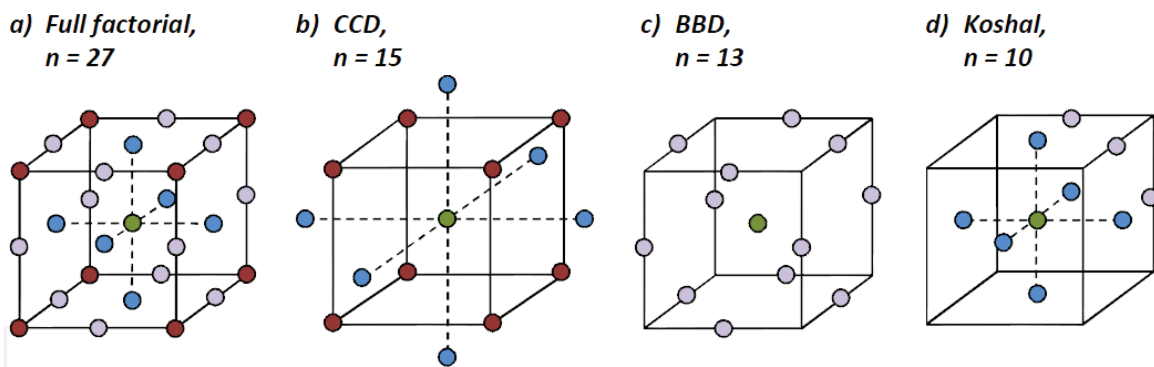


Figure 3. Experimental designs in three variables for fitting second order models (Full factorial, central composite design (CCD), box-behnken design (BBD) and Koshal) [21].

2.1.1 Classical experimental designs

The idea of the classical DOE is to reach as much information as possible from a limited number of experiments. The focus of these methods is on planning the experiments so that the random error from the physical experiments has minimum influence in the approval or disapproval of a hypothesis. Therefore, a classical experimental design represents a sequence of experiments to be performed, expressed in terms of factors (design variables) set at specified levels (predefined values) [5].

Widely used “classical DOE” include factorial or fractional factorial designs (FFD), central composite designs (CCD), Box-Behnken designs (BBD), and Koshal designs (KD). Schematic illustration of these methods is presented in **Figure 3**. These classic methods tend to pick out the sample points around boundaries of the design space and leave a few at the center of the design space. To view the details of these methods, one may refer to [21].

2.1.2 Experimental designs for complex metamodels

As computer experiments involve mostly systematic error rather than random error as in physical experiments, researchers stated that in the presence of such sources of error, a good experimental design has to be space filling and non-collapsing rather than to concentrate on the boundary [5]. Also, in dealing with a complex design space, the metamodel’s training samples should be spread the design points within the complete design space so that no prediction be too far from training points. Four types of space filling sampling methods that relatively more often used in the literature are orthogonal arrays, various Latin hypercube designs, Hammersley sequences, and uniform designs. Details of these methods are presented in Ref. [21].

2.2 Metamodel choice and metamodel fitting

After selecting an appropriate DOE strategy and performing the necessary computer runs, the next step is to choose a metamodel and fitting method. As alluded to earlier in the introduction, many machine learning methods such as ANN, Kriging, RBF and SVR have been used to approximate complex relations between a set of inputs and outputs, and can thus be used as a metamodel.

Despite the various metamodel types that have been introduced so far, which model is suitable for use? Different metamodels have their unique properties and consequently, there is no universal model that always is the best choice. Instead, the

suitable metamodel depends on the problem at hand [21]. They are bound to their special domains, and thus no comparative studies have been conducted on them.

On the other hand, the performance of metamodels is depending on the problem to be addressed, and multiple criteria need to be considered. Model accuracy is probably the most important criterion, since approximate models with a low accuracy may lead the optimization process to local optima or even diverge from the optimal solution. Model accuracy also should be evaluated based on the new random sample points instead of the training data set points. The reason for this is that for some models overfitting is a common difficulty. In the case of overfitting, the model yields good accuracy on training data but may have poor performance on new sample points. The optimization process could easily go in the wrong direction if it is assisted by models with low accuracy [12].

There are some accuracy measures that may be used to evaluate the metamodels. The coefficient of determination R^2 is a measure of how well the metamodel is able to capture the variability in the dataset. Other common ways for accuracy measures include: the maximum absolute error (MAE), the average absolute error (AAE), the mean squared error (MSE) and the root mean squared error (RMSE) [21].

2.3 Metamodel-based design optimization

Metamodel-based design optimization can be applied using different strategies. The main issue with MBDO is the error that is introduced when approximating the real simulations with metamodels. The optimization process can be performed using the detailed simulation model, using its surrogate model, or both of them. Most common types of MBDO strategies are illustrated in **Figure 4**. In the first strategy (**Figure 4a**), a global metamodel will be built and then will be used during optimization. This approach uses a relatively large number of sample points at the outset and is commonly seen in the literature. The second strategy (**Figure 4b**) is based on the online metamodeling and involves the validation and/or optimization in the loop in deciding the resampling and remodeling strategy. In this strategy, samples will be generated iteratively to update the train data and related metamodel to maintain the model accuracy. In the third strategy (**Figure 4c**), the optimization is performed by adaptive sampling alone and no formal optimization process is used. This strategy directly generates new sample points toward the optimum with the guidance of a metamodel [7].

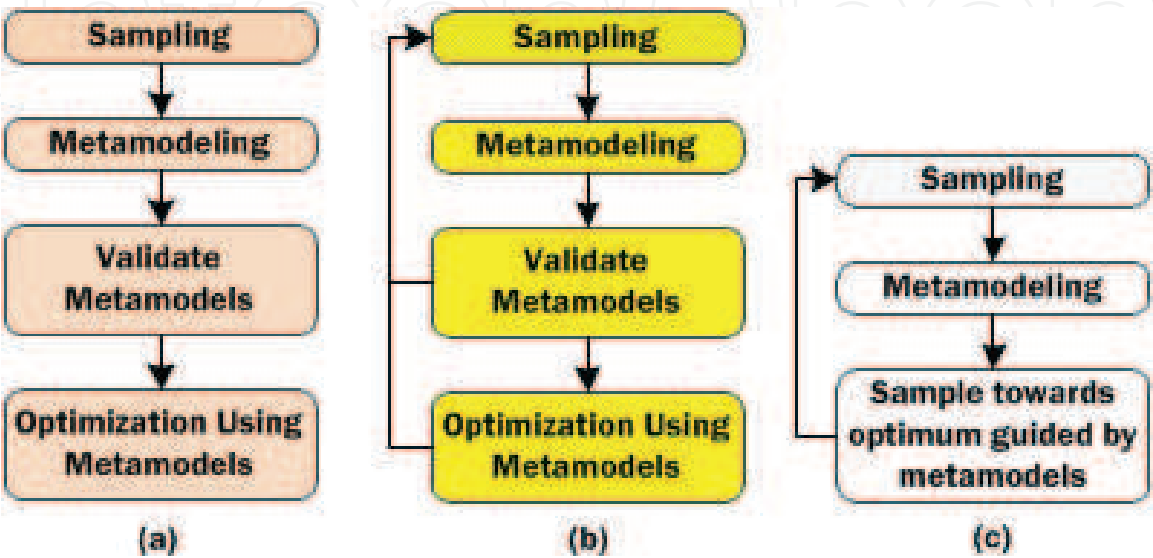


Figure 4. MBDO strategies: (a) sequential approach; (b) adaptive MBDO; and (c) direct sampling approach [7].

In complex real-world design problems, achieving a flawless metamodel is almost impossible. Therefore, in order to take advantage of metamodels in MBDO, it is better to manage using metamodels based on their accuracy in design points/spaces. As shown in **Figure 5**, evolution control and migration are two major classes of management strategies for utilizing metamodels [12]. In the evolution control class, metamodels are called beside the real-models during the optimization process where the real-models are used in some/all individuals and in some/all generations. In model Migration, the entire population is divided into several sub-populations with its local metamodel. Also, the individuals in various sub-populations can migrate into other sub-populations [1]. To study the details of evolution control and migration, readers may refer to Tenne and Goh [12]. To improve the applicability of MBDO for complex real-world design problems, a novel ECS is developed that is introduced in the next section.

3. Proposed evolution control strategy (ECS)

In this section, a novel management strategy for application of the metamodels is introduced. This strategy relies on the Mean-Squared-Displacement (MSD) concept and is based on the evolution control class of metamodel management strategies. The MSD means the deviation of a particle’s position relative to a reference position that is a statistical concept.

During the optimization process, the value of MSD for each design point must be computed that is named as calculated MSD (CMSD).

$$CMSD = \frac{1}{N_{train}} \sum_{n=1}^{N_{train}} (x_n - x_{ind})^2 \tag{1}$$

where N_{train} and X_n indicate the number and the vector of design variables of metamodel training data, respectively. X_{ind} is the vector of optimizer design variables, iteratively.

In order to use the proposed strategy in the optimization process, the MSD value of each sample that is used as metamodel’s test point based on the all of training data set has to be computed. Then, using these MSD values, two MSD values for the first and last iteration of the optimization process have to be selected. These two values that are named initial MSD (IMSD) and final MSD (FMSD) respectively indicate

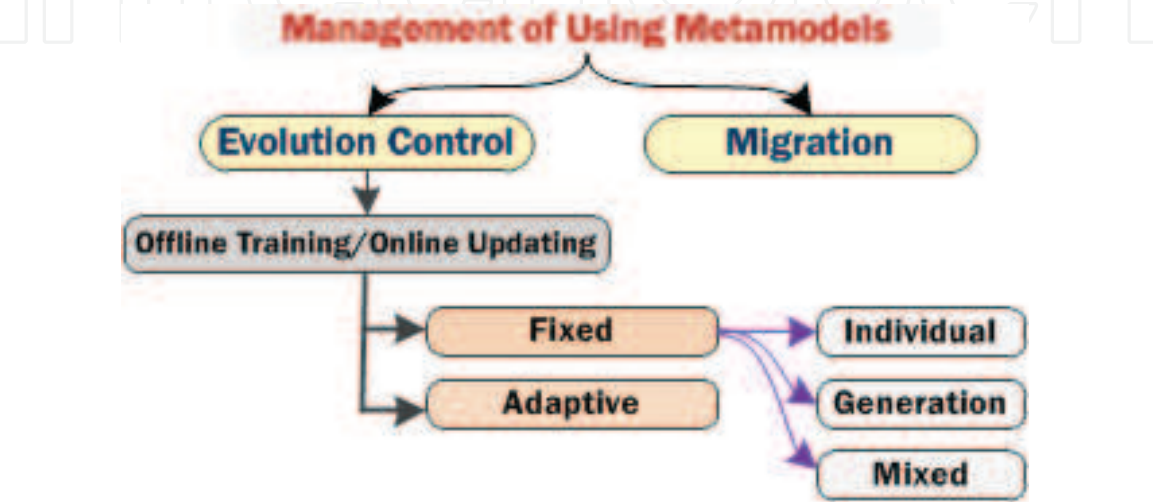


Figure 5.
 Metamodel management strategies in MBDO [12].

the acceptable accuracy of metamodels from the first until the last iteration of the optimization process. An adaptive threshold namely predetermined MSD (PMSD) is proposed for the management of the decreasing PMSD value from IMSD to FMSD. The adaptive threshold of the PMSD enables the optimizer to call more metamodels vs. real models in the first iterations of optimization. Also, it enables that while the optimization is ongoing, the number of metamodel's call functions decrease slowly and the real model's call functions will increase. Here, the proposed adaptive PMSD threshold relies on the inverse hyperbolic cosecant concept, as follows:

$$PMSD = IMSD + (IMSD - FMSD) \times a \csc h(k) \quad (2)$$

The variable k has to iteratively increase within an interval (e.g. from -12.5 to -1) while the optimization is ongoing. The introduced strategy is summarized iteratively, as follows [1]:

1. Calculate MSD value of metamodel's test points and initialize the IMSD and FMSD.
2. Determine the value of the k for PMSD estimation during the optimization process.
3. Start the optimization process using an initial design point.
4. Calculate the CMSD for each optimizer design point, compare CMSD value with PMSD and take a decision on using metamodels or real models.
5. Evaluate objective functions and constraints based on the decision in step 4.
6. Go to the next iteration of the optimizer and update the PMSD value.
7. Check the optimization convergence criterion and go to step 4.

The introduced strategy could be applied to all class of the optimization algorithms and both deterministic and non-deterministic optimization problems. In the next section, a number of benchmark problems are solved to present the ability of the proposed strategy.

4. MBDO of benchmark problems

In this section, the performance of the proposed strategy in achieving the global or at least the near-global optimum is investigated through solving some benchmark problems.

4.1 Analytical problem: one dimensional

Here, a one-dimensional nonlinear analytical example from Ref. [8] is used to illustrate the implementation of the proposed strategy. The mathematical formulation is shown as:

$$f(x) = (6x - 2)^2 \sin(12x - 4) \quad (3)$$

The design variable x follows a normal distribution $x \sim N(x, \sigma_x^2)$, where $\sigma_x = 0.08$ and $x \in [0, 1]$. The objective of this example is to find the robust optimum solution and its cost function is defined as:

$$\text{Minimize } F(x) = \mu_f(x) + 3\sigma_f(x) \tag{4}$$

Based on four initial samples at $x = [0, 0.33, 0.67, 1]$, a Kriging and ANN metamodel of Eq. (4) is constructed (Figure 6).

In Figure 6, the cross-mark and square mark represent the test and train sample points of metamodels, respectively. As illustrated in Figure 6, the robust optimum points resulted from Kriging and ANN metamodels are different from the real model one. The robust solution of Kriging and ANN are located at point $x = 0.38$, which are far away from the real solution $x = 0.28$. Therefore, due to the relatively large error of these metamodels, the obtained robust solution cannot be accepted. In order to resolve this issue, it is essential to add more samples to improve the prediction capability of metamodels.

Another way to overcome this problem is by using proposed ECS in this work. To do this, MSD value of all sample points that are used for metamodels test should be calculated. Then, based on these MSD values, a setting of the PMSD parameters (Eq. (2)) including IMSD and FMSD should be done. Figure 7 illustrates the MSD value related to testing sample points. According to MSD values in Figure 7, the value of IMSD and FMSD are considered as 1.3 and 0.5, respectively. Now, it is time to select the adaptive threshold variable (k in Eq. (2)). This variable should increase iteratively while optimization is ongoing and its bound has a direct impact on how the PMSD threshold decreases adaptively. For example, by considering the interval $[-12.5, -1]$ for the variable k and assuming the maximum iteration of the optimization process be 10, the PMSD adaptive threshold variation is illustrated in Figure 8.

Now, the optimization problem defined in this example is solved through proposed strategy along with simulated annealing optimizer and $x = 1$ as a start point. Convergence process in comparing with using only metamodels and real model is illustrated in Figure 9. Also, switching between real model and ANN metamodel

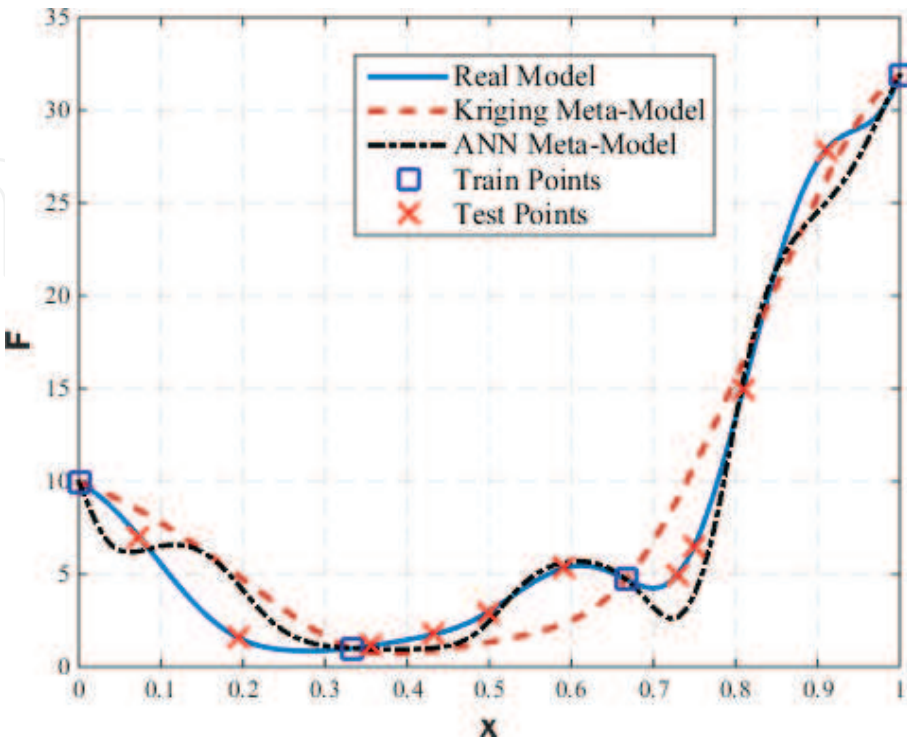


Figure 6.
Real robust function and its metamodels.

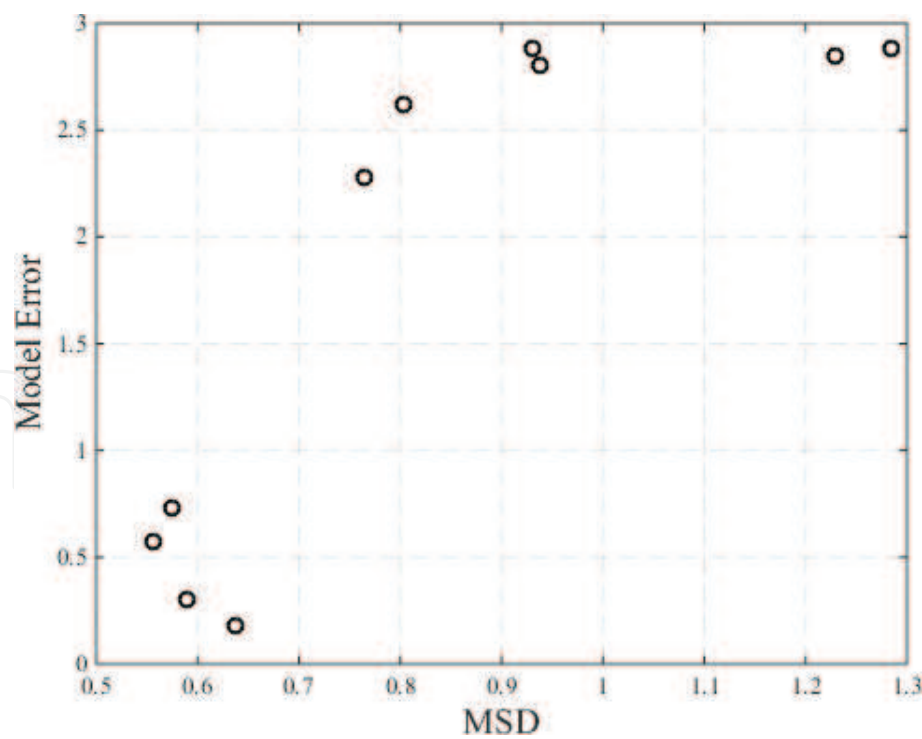


Figure 7.
MSD values related to the metamodels test points.

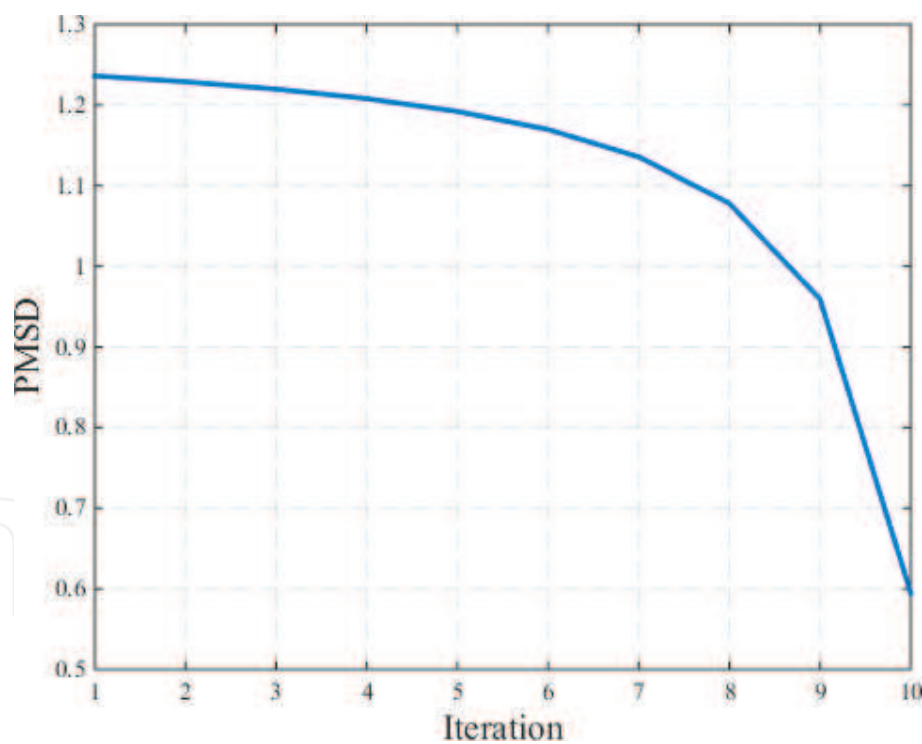


Figure 8.
PMSD variation vs. optimization iteration.

based on the CMSD value of each design point and PMSD value in the related iteration is shown in **Figure 10**. **Table 1** illustrates that proposed strategy with 3 real model call functions is achieved to near global robust optimum point compared to other methods.

In **Table 1**, the methods developed by Zhang et al. [22] that are based on the Kriging metamodel have been reached the near global optimal point through adding a new sample to the training set iteratively. For every new design point metamodel has been re-trained. Every time that a new point is added, you need to re-train and

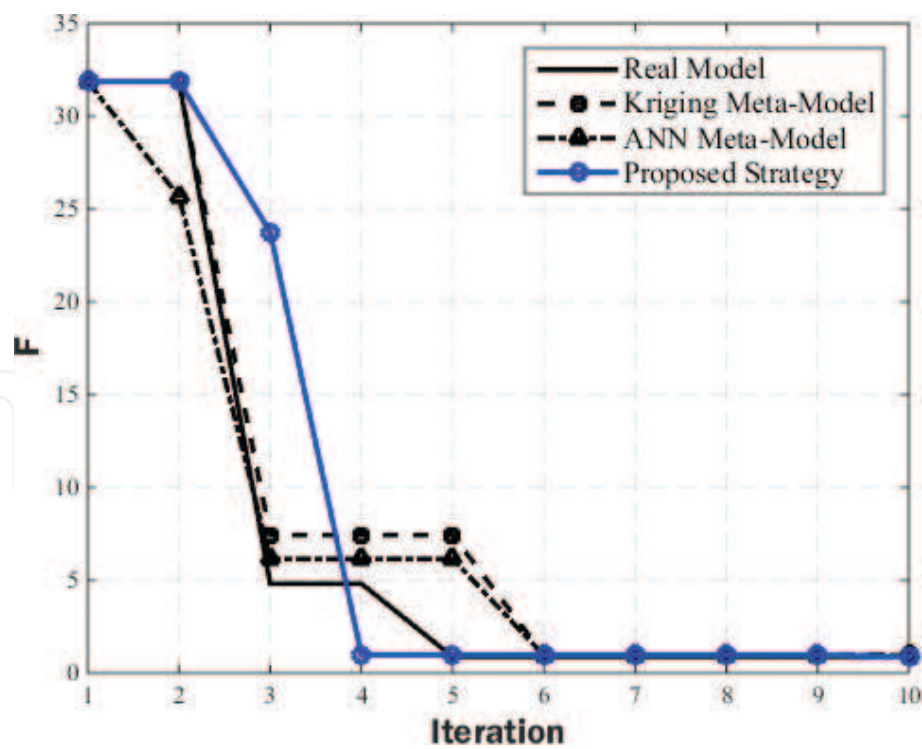


Figure 9.
Optimization convergence—one dimensional example.

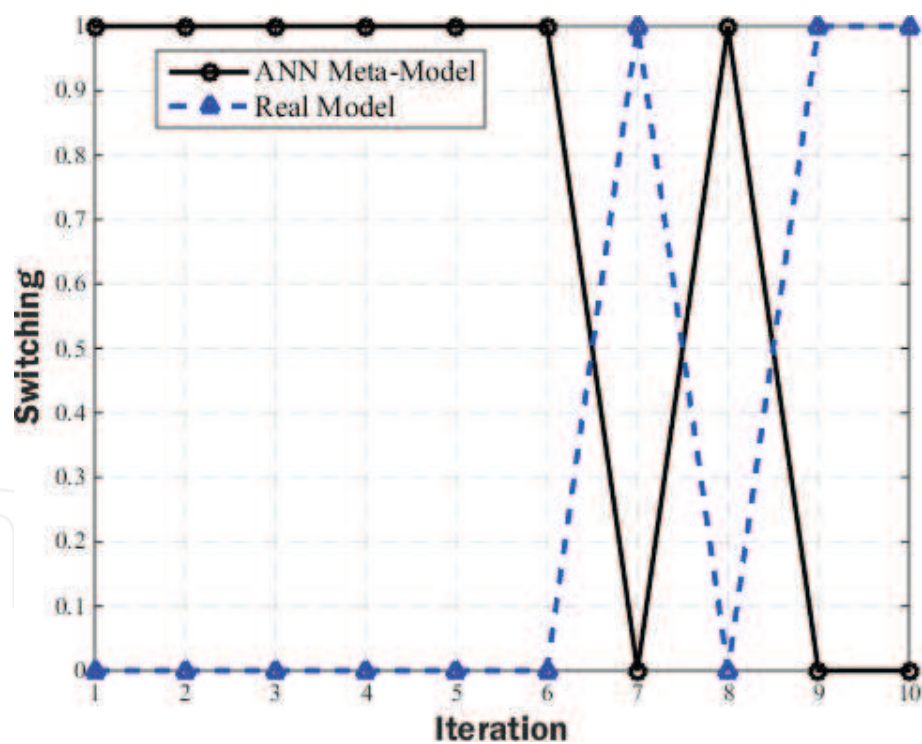


Figure 10.
Switching between ANN metamodel and real model.

re-test the metamodel. Since the metamodel is poor in the first iterations, this can mislead the optimization process into local optimum or infeasible regions in the design space.

As illustrated in **Figure 10**, the proposed strategy allows the optimizer to call metamodel in the first iterations. As optimization ongoing, metamodel accuracy in each design point will be checked and the real model will be called if necessary to prevent the optimizer from going to the wrong direction. Therefore, with proper

| Model | Robust solution | |
|---|-----------------|-------|
| | X | F |
| Real model | 0.3 | 0.88 |
| Kriging | 0.335 | 0.973 |
| ANN | 0.377 | 0.947 |
| EI-based EGO [22]—(4 extra points is added to the training samples) | 0.270 | 0.94 |
| R-EI-based EGO [22]—(2 extra points is added to the training samples) | 0.30 | 2.24 |
| Proposed Strategy—(with 7 meta-model calls and 3 real model calls) | 0.29 | 0.87 |

Table 1.
Robust solution resulted from different methods [1].

management of metamodels during the optimization process, the possibility of accessing the global or near global optimum will be increased.

4.2 Analytical problem—two dimensional

To further investigate the benefits of the proposed strategy, the robust design of two-dimensional Haupt function is presented here [22]. The Haupt function is defined as:

$$f(\mathbf{x}) = x_1 \sin(4x_1) + 1.1x_2 \sin(2x_2)$$
 (5)

In this example, both of the design variables x_1 and x_2 follow a normal distribution $\mathbf{x} \sim N(\mathbf{x}, \sigma_{\mathbf{x}}^2)$, where $\mathbf{x} = [x_1, x_2]$ with $\mathbf{x} \in [0, 4]$ and $\sigma_{\mathbf{x}} = [\sigma_{x1}, \sigma_{x2}] = [0.2, 0.2]$. Considering the effect of design variable uncertainty, the robust design formulation is defined as:

$$\text{Minimize } F(\mathbf{x}) = \mu_f(x_1, x_2) + 3\sigma_f(x_1, x_2)$$
 (6)

Based on Improved LHS (ILHS), 10 points are generated as the training sample points. The real model, Kriging and ANN metamodels of Eq. (6) are shown in **Figure 11**. It can be seen that the constructed metamodels are not sufficiently accurate and will mislead the optimizer into local optimum or non-optimal regions. To resolve this issue, Zhang et al. [22] have been proposed methods to generate new points while optimization is ongoing and increase the metamodel accuracy, iteratively. But since the predictive capability of the metamodel is poor in the first

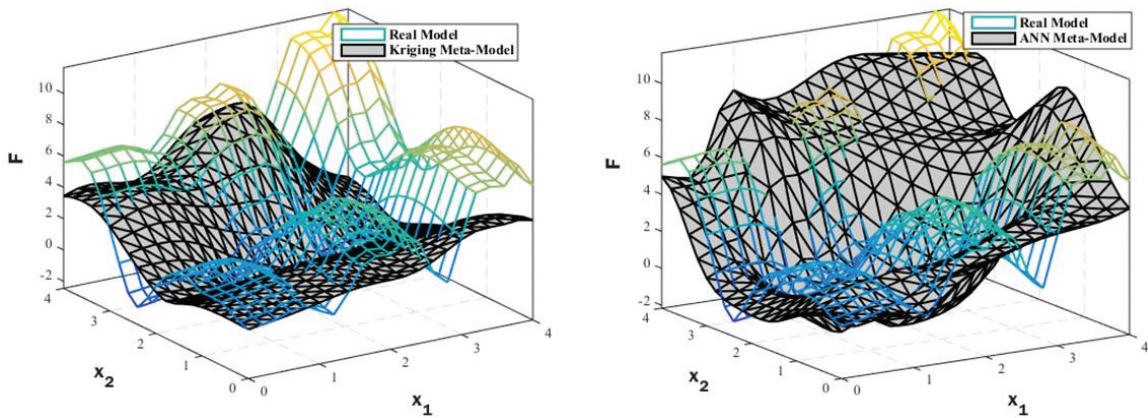


Figure 11.
Design space and different models of Haupt function.

iterations of optimization, there is not any guarantee to achieve global optimization, especially in complex real-world applications.

In order to the implementation of the proposed strategy to moderate this issue, PMSD equation parameters (Eq. (2)) including IMSD and FMSD should be determined based on the MSD amount of metamodels test points. Therefore, in this example, IMSD and FMSD values are considered as 2.4 and 0.8, respectively. As alluded to in the previous analytical example, to reduce the amount of PMSD threshold slowly, the interval of the variable k is assumed as $[-12.5, -1]$. According to the considered set of proposed strategy along with simulated annealing optimizer and $x = [4, 4]$ as a start point, the robust design problem defined in Eq. (6) is solved using different methods. Optimization convergence process and switching between models are shown in Figures 12 and 13, respectively. Also, resulted robust optimums of different methods are presented in Table 2.

As presented in Table 2, one-stage sampling and sequential sampling methods that are based on the Kriging metamodel and proposed by Zhang et al. [22] have been reached the near global optimal point through 30 and 19 training sample points, respectively. But proposed strategy with checking the accuracy of the metamodel during optimization process through 5 switching between real model and metamodel (see Figure 13) is able to achieve near global optimum.

4.3 Engineering problem—two-bar truss structure

Uncertainty based design optimization of truss and frame structures is a popular topic in mechanical, civil, and structural engineering due to the complexity of problems and benefits to industry. In this section, the popular two-bar truss structure problem (Figure 14) is used as a benchmark problem for the multi-objective Robust Design Optimization (RDO) under epistemic uncertainties. The test case is adapted from Ref. [23].

As illustrated in Figure 14, the cross-section diameter (d) and the structure height (H) are as the design variables. The uncertain design parameters are including

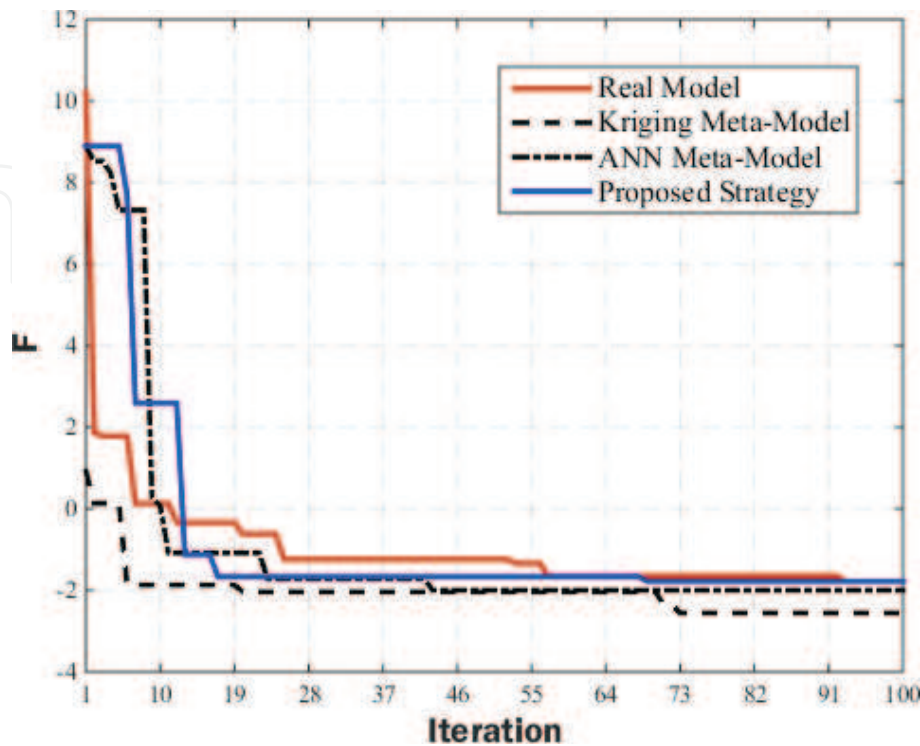


Figure 12.
Optimization convergence—Haupt function robust design.

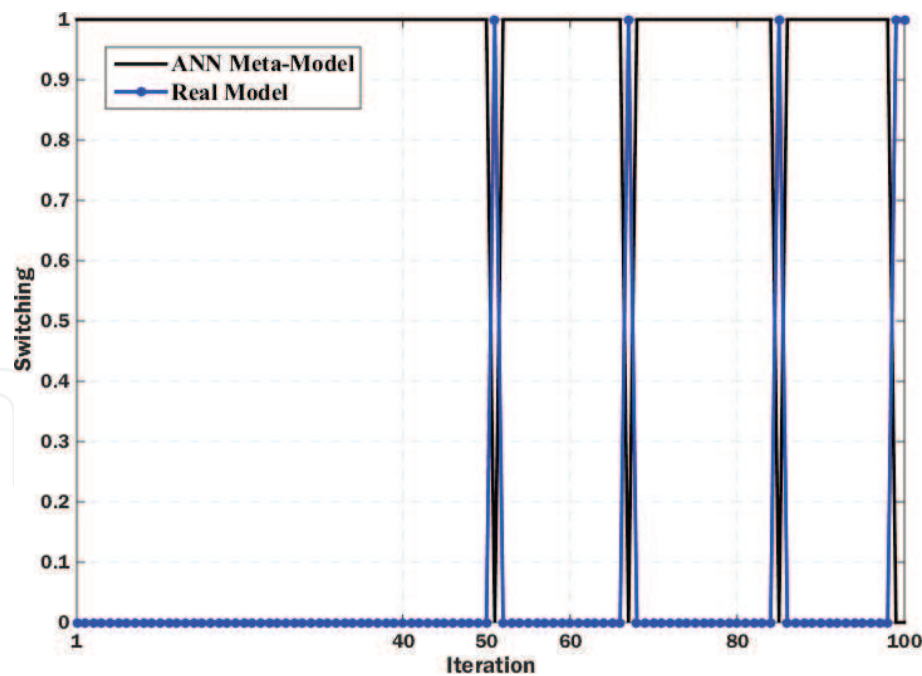


Figure 13.
Switching between metamodel and real model of Haupt function.

| Model | Robust solution | |
|---|-----------------|-------|
| | X | F |
| Real model | [1.18, 2.45] | −1.78 |
| Kriging | [2.72, 2.53] | −2.56 |
| ANN | [1.74, 1.76] | −1.99 |
| One-stage sampling method [22]—(30 training sample points) | [1.19, 2.44] | −1.74 |
| sequential sampling method [22]—(19 training sample points) | [1.2, 2.47] | −1.68 |
| Proposed strategy—(with 95 meta-model calls and 5 real model calls) | [1.2, 2.4] | −1.77 |

Table 2.
Robust solution resulted from different methods for Haupt function [1].

vertical force ($P \sim N(150, 5)$ kN), structure width ($B \sim N(750, 10)$ mm), Elastic modulus ($E \sim N(2.1e5, 5e3)$ N/mm²), and member thickness ($t \sim N(2.5, 0.4)$ mm). The RDO problem is formulated in the following equation to minimize volume, vertical displacement and robustness criteria of the structure subject to constraints of stress and buckling.

In this design problem, the robustness measure F_{RC} given in Eq. (7) is defined as follows, with P , B , E and t as the four uncertain parameters.

$$\begin{aligned} & \text{Minimize} \quad \left\{ \mu_{\text{volume}}, \mu_{\text{deflection}}, F_{RC}(\sigma_{\text{volume}}, \sigma_{\text{deflection}}, \sigma_S) \right\} \\ & \text{Subject to} \quad g_1 : \mu_S \leq S_{\max} \\ & \quad \quad \quad g_2 : \mu_S \leq S_{\text{crit}} \\ & \text{With respect to} \quad 20 \leq d \text{ (mm)} \leq 80, 200 \leq H \text{ (mm)} \leq 1000 \\ & \text{volume} = 2\pi dt \sqrt{B^2 + H^2}; \text{deflection} = \frac{P(B^2 H^2)^{\frac{3}{2}}}{(2\pi EdH)^2} \\ & S = \frac{P \sqrt{B^2 + H^2}}{2\pi t d H}, \quad S_{\text{crit}} = \frac{\pi^2 E (t^2 + d^2)}{8(B^2 + H^2)}, \quad S_{\max} = 400 \text{ MPa} \end{aligned} \tag{7}$$

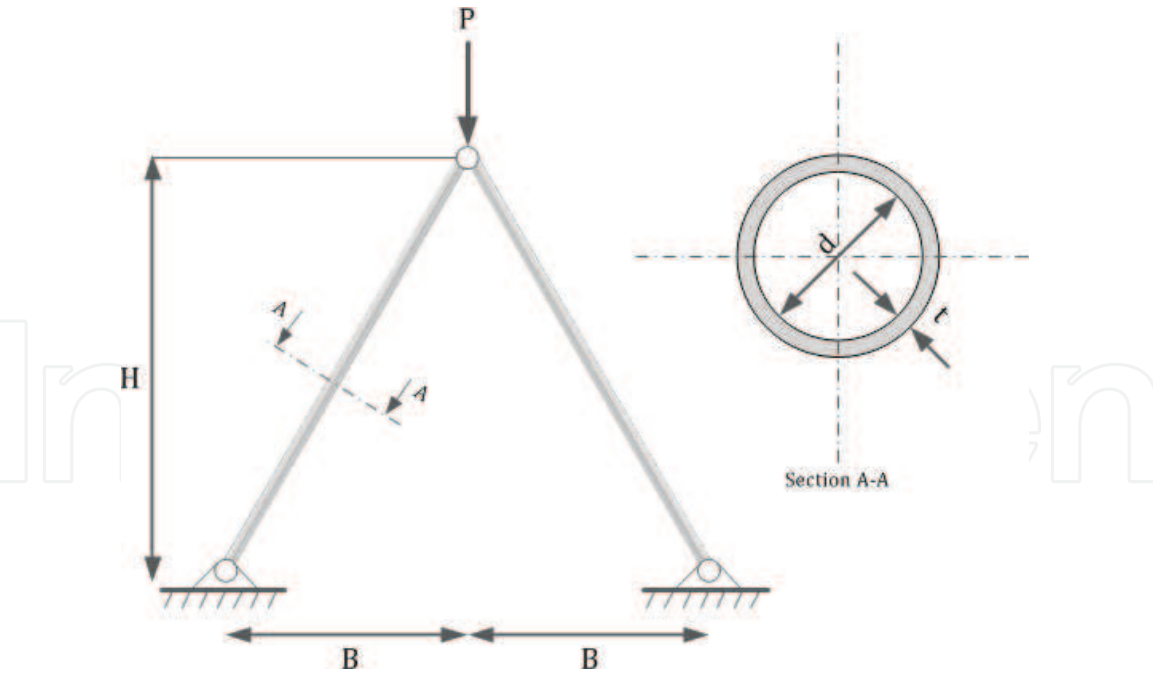


Figure 14.
Two-bar truss structure [23].

$$F_{RC} = \frac{1}{3 \times 4} \left\{ \begin{aligned} &\left(\frac{\sigma_{volume}}{\sigma_t} + \frac{\sigma_{volume}}{\sigma_w} \right) \\ &+ \left(\frac{\sigma_{deflection}}{\sigma_t} + \frac{\sigma_{deflection}}{\sigma_w} + \frac{\sigma_{deflection}}{\sigma_P} + \frac{\sigma_{deflection}}{\sigma_E} \right) \\ &+ \left(\frac{\sigma_S}{\sigma_t} + \frac{\sigma_S}{\sigma_w} + \frac{\sigma_S}{\sigma_P} \right) \end{aligned} \right\} \quad (8)$$

Based on the procedure summarized in Section 2 and 3, four different ANN metamodels are constructed for computing normal stress, buckling stress, volume and deflection. For this purpose, the training set is provided by 100 sampling using ILHS and testing set is generated using 1000 random sampling on the design variables and uncertain parameters bounds. Metamodeling creation involves making a decision on the appropriate number of layer(s) and the number of neurons in the hidden layer(s) and selecting the best model with minimum MSE. The architecture and approximating capability of these metamodels on training and testing sets are shown in **Table 3**.

As illustrated in **Table 3**, the existence of some areas without enough accuracy is inevitable, so using metamodels in optimization process requires a management strategy. In order to implement the developed management strategy, we need the value of IMSD and FMSD parameters. To make a decision on the value of these parameters, the random testing set is utilized to compute the MSD value of each design point using created metamodels. Increasing in MSD values led to an increase in

| Meta-model | Unit | N. neurons in each layer | Train | Test |
|-----------------|-----------------|--------------------------|---------|---------|
| | | | MSE | MSE |
| Normal stress | MPa | [5 5 5] | 9.30e−6 | 81.22 |
| Buckling stress | MPa | [5 5 5] | 2.45e−5 | 2.28e+2 |
| Volume | mm ³ | [5 5 5] | 5.38e−6 | 0.0374 |
| Deflection | mm | [5 5 5] | 1.22e−5 | 4.34e+7 |

Table 3.
Approximation capability of metamodels for two-bar truss problem.

metamodel error. So, during the optimization process, if the amount of the MSD in each individual (i.e. CMSD) be less than PMSD, the metamodel has enough accuracy and can be used to functions evaluation. Contrariwise, when the accuracy of the metamodel is not sufficient (i.e. MSD value is greater than PMSD), the real-model should be used. As stated above, here, the amount of IMSD and FMSD are considered as 2.5 and 0.5, respectively. Also, it is assumed that the adaptive threshold variable k in Eq. (2) be increase from -12.5 to -1 until final optimization generation.

The explained problem is solved through NSGA-II optimizer. The LHS based on the correlation criterion is used to generate the initial population of the optimization process. The optimization setting including population size, generation, crossover, and mutation are 50, 150, 0.8 and 0.15, respectively. The ILHS method with 1000 points is employed for uncertainty propagation and analysis. Finally, the Pareto frontier with two optimality criteria and one robustness criteria using both proposed strategy and real-model simulation is illustrated in **Figure 15**. The number of metamodel/real-model call function and PMSD value are shown in **Figure 16**, iteratively.

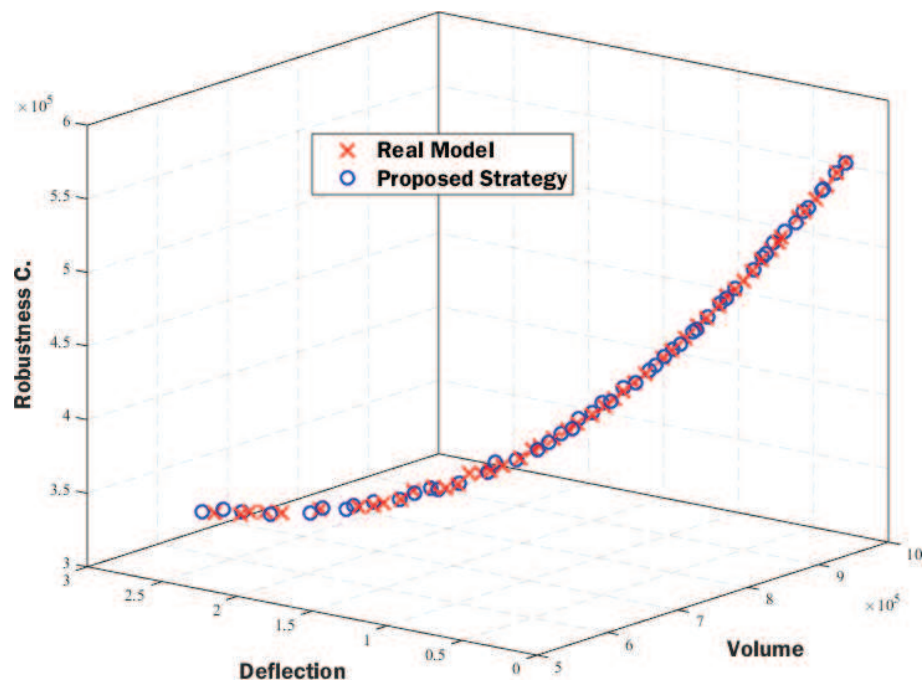


Figure 15.
Pareto frontiers resulted from RDO for two-bar truss problem.

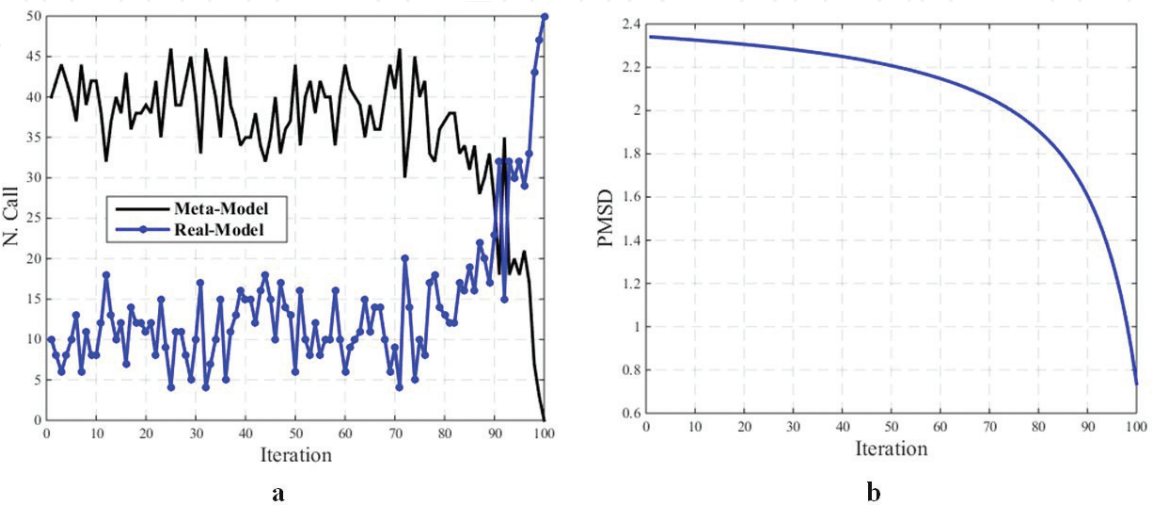


Figure 16.
Number of call functions (a) and PMSD threshold (b) for two-bar truss problem.

According to the iterative results shown in **Figure 16a**, in the early iterations of the optimization process, the number of metamodels call functions are more than real-model ones and in the final iterations, real-model call functions are increased. Therefore, control of call functions in metamodel based optimization process could lead to increasing the accuracy and globality of convergence. In addition, during the optimization process, the tuned adaptive threshold (**Figure 16b**) is slowly decreased to increase the real-model call functions, iteratively.

5. Concluding remarks and future work

Despite recent surrogate-based design books [8, 12, 24] for engineers and extensive investigations conducted in this field, many researchers are still making efforts to push the boundaries of metamodeling. It can be noted that, despite the numerous research carried out over the last few years, the computational complexity still remains as a major challenge of this field of study. Also, today's engineering design problems with multidisciplinary nature are extremely complex (e.g., uncertainty-based multidisciplinary design optimization). Therefore, metamodels management and their accuracy over the design space are another challenges and open fields for research. Therefore, in this chapter, a novel ECS is proposed to improve computational efficiency and make better decisions for function evaluation when facing the metamodel based design optimization problems.

The assessment of the benchmark problems revealed both the efficiency and the effectiveness of the proposed strategy. For all case studies, ANN and Kriging metamodels are used to create metamodels based on ILHS. Also, the ILHS is utilized for uncertainty propagation and analysis. Results illustrate that the proposed strategy could lead to improving the computational efficiency, accuracy, and globality of the convergence process in MBDO problems.

Future researches could be include extensions of the problem to higher dimensional with high fidelity analysis modules and considering different sources of uncertainties. The sensitivity of the proposed strategy to other metamodeling techniques (i.e. RSM, RBF, Kriging, etc.) can be considered in the future. Also, the use of metamodels in co-simulation works to replace high fidelity analysis with inexpensive surrogate models might be an interesting research field. Determining appropriate criteria for extracting or selecting new points to update the metamodels training set during online metamodeling is another challenge of this field of study. Also, the presence of data mining approaches along with computational intelligence methods could provide the basis for the emergence of new metamodeling techniques, which could be very significant.

IntechOpen

IntechOpen


Author details

Ali Asghar Bataleblu

Department of Aerospace Engineering, K. N. Toosi University of Technology,
Tehran, Islamic Republic of Iran

*Address all correspondence to: ali.bataleblu@gmail.com

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Roshanian J, Bataleblu AA, Ebrahimi M. A novel evolution control strategy for surrogate-assisted design optimization. *Structural and Multidisciplinary Optimization*. 2018; **58**:1255. DOI: 10.1007/s00158-018-1969-4
- [2] Viana FA, Simpson TW, Balabanov V, Toropov V. Special section on multidisciplinary design optimization: Metamodeling in multidisciplinary design optimization: How far have we really come? *AIAA Journal*. 2014; **52**(4): 670-690. DOI: 10.2514/1.J052375
- [3] Jin Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*. 2011; **1**(2): 61-70. DOI: 10.1016/j.swevo.2011.05.001
- [4] Sudret B. Meta-models for structural reliability and uncertainty quantification. In: *Proc. 5th Asian-Pacific Symp. Structural Reliab, Its Appl.* Singapore: APSSRA; 2012. pp. 53-76
- [5] Simpson TW, Peplinski J, Koch PN, Allen JK. Meta-models for computer-based engineering design: Survey and recommendations. *Engineering Computations*. 2001; **17**(2):129-150. DOI: 10.1007/PL00007198
- [6] Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang RJ. Approximation methods in multidisciplinary analysis and optimization: A panel discussion. *Structural and Multidisciplinary Optimization*. 2004; **27**(5):302-313. DOI: 10.1007/s00158-004-0389-9
- [7] Wang GG, Shan S. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*. 2007; **129**(4):370-380. DOI: 10.1115/1.2429697
- [8] Forrester AIJ, Keane AJ. Recent advances in surrogate-based optimization. *Progress in Aerospace Science*. 2009; **45**(1-3):50-79. DOI: 10.1016/j.paerosci.2008.11.001
- [9] Booker AJ, Dennis JE Jr, Frank PD, Serafini DB, Torczon V, Trosset MW. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*. 1999; **17**(1): 1-13. DOI: 10.1007/BF01197708
- [10] Tzannetakis N, Van de Peer J. Design optimization through parallel-generated surrogate models, optimization methodologies and the utility of legacy simulation software. *Structural and Multidisciplinary Optimization*. 2002; **23**(2):170-186. DOI: 10.1007/s00158-002-0175-5
- [11] Adams BM, Bohnhoff WJ, Dalbey KR, Eddy JP, Eldred MS, Gay DM, et al. DAKOTA, a Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 User's Manual. Tech. Rep. SAND2010-2183. Sandia National Laboratories; 2009
- [12] Tenne Y, Goh CK. Computational intelligence in expensive optimization problems. Vol. 2. Berlin: Springer Science & Business Media; 2010
- [13] Horng SC, Lin SY. Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation. *Information Sciences*. 2013; **233**:214-229. DOI: 10.1016/j.ins.2013.01.024
- [14] Sóbester A, Forrester AI, Toal DJ, Tresidder E, Tucker S. Engineering design applications of surrogate-assisted optimization techniques. *Optimization*

and Engineering. 2014;**15**(1):243-265.
DOI: 10.1007/s11081-012-9199-x

[15] Gong W, Zhou A, Cai Z. A multioperator search strategy based on cheap surrogate models for evolutionary optimization. *IEEE Transactions on Evolutionary Computation*. 2015;**19**(5): 746-758. DOI: 10.1109/TEVC.2015.2449293

[16] Zhou Q, Shao X, Jiang P, Gao Z, Zhou H, Shu L. An active learning variable-fidelity metamodeling approach based on ensemble of metamodels and objective-oriented sequential sampling. *Journal of Engineering Design*. 2016;**27**(4-6): 205-231. DOI: 10.1080/09544828.2015.1135236

[17] Belyaev M, Burnaev E, Kapushev E, Panov M, Prihodko P, Vetrov D, et al. GTApprox: Surrogate modeling for industrial design. *Advances in Engineering Software*. 2016;**102**:29-39. DOI: 10.1016/j.advengsoft.2016.09.001

[18] Sun C, Jin Y, Cheng R, Ding J, Zeng J. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*. 2017;**21**: 644-660. DOI: 10.1109/TEVC.2017.2675628

[19] Sayyafzadeh M. Reducing the computation time of well placement optimisation problems using self-adaptive metamodeling. *Journal of Petroleum Science and Engineering*. 2017;**151**:143-158. DOI: 10.1016/j.petrol.2016.12.015

[20] Chatterjee T, Chakraborty S, Chowdhury R. A critical review of surrogate assisted robust design optimization. *Archives of Computational Methods in Engineering*. 2017:1-30. DOI: 10.1007/s11831-017-9240-5

[21] Ryberg AB, Domeij Bäckryd R, Nilsson L. *Metamodel-Based Multidisciplinary Design Optimization for Automotive Applications*. Linköping: Linköping University Electronic Press; 2012

[22] Zhang SL, Zhu P, Arendt PD, Chen W. Extended objective oriented sequential sampling method for robust design of complex systems against design uncertainty. In: *Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*. 2012. pp. 12-15

[23] Martinez J, Marti P. Metamodel-based multi-objective robust design optimization of structures. In: *12th International Conference on Optimum Design of Structures and Materials in Engineering*; New Forest, UK. 2012

[24] Dellino G, Meloni C. *Uncertainty Management in Simulation Optimization of Complex Systems: Algorithms and Applications*. New York: Springer; 2015