

Ensemble of metamodels with optimized weight factors

E. Acar · M. Rais-Rohani

Received: 14 June 2007 / Revised: 5 December 2007 / Accepted: 18 December 2007 / Published online: 13 February 2008
© Springer-Verlag 2008

Abstract Approximate mathematical models (metamodels) are often used as surrogates for more computationally intensive simulations. The common practice is to construct multiple metamodels based on a common training data set, evaluate their accuracy, and then to use only a single model perceived as the best while discarding the rest. This practice has some shortcomings as it does not take full advantage of the resources devoted to constructing different metamodels, and it is based on the assumption that changes in the training data set will not jeopardize the accuracy of the selected model. It is possible to overcome these drawbacks and to improve the prediction accuracy of the surrogate model if the separate stand-alone metamodels are combined to form an ensemble. Motivated by previous research on committee of neural networks and ensemble of surrogate models, a technique for developing a more accurate ensemble of multiple metamodels is presented in this paper. Here, the selection of weight factors in the general weighted-sum formulation of an ensemble is treated as an optimization problem with the desired solution being one that minimizes a selected error metric. The proposed technique is evaluated by considering one industrial and four benchmark problems. The effect of different metrics for estimating the prediction error at either the training data set or a few validation points is also explored. The results show

that the optimized ensemble provides more accurate predictions than the stand-alone metamodels and for most problems even surpassing the previously reported ensemble approaches.

Keywords Metamodeling · Surrogate modeling · Optimization · Ensemble

1 Introduction

The design of modern engineering systems relies on high-fidelity computer simulations for accurate analysis of system characteristics. Venkataraman and Haftka (2004) note that analysis models of acceptable accuracy have required at least 6 to 8 h of central processing unit (CPU) time throughout the last 30 years, even though computer processing power along with memory and storage capacities have drastically increased. This perceived lack of improvement in computational efficiency can be explained by the fact that the fidelity and complexity of the analysis models have also steadily increased over the same period.

When high-fidelity simulations are combined with numerical design optimization, the computational cost tends to increase; moreover, when using gradient-based techniques, the accuracy and convergence of the optimization solution are jeopardized if the response characteristics that appear in the objective or constraint functions are noisy.

Hence, there is a growing interest in utilizing design and analysis of computer experiments methods to replace the computationally expensive simulations with smooth analytic functions (metamodels) that can serve as surrogate models for efficient response estimation.

E. Acar (✉)
Center for Advanced Vehicular Systems,
Mississippi State University, Mississippi State, MS 39762, USA
e-mail: erdem@cavs.msstate.edu

M. Rais-Rohani
Department of Aerospace Engineering,
Mississippi State University, Mississippi State, MS 39762, USA

Metamodeling techniques aim at regression and/or interpolation fitting of the response data at the specified training (observation) points that are selected using one of the many designs of experiments (DOE) techniques. There are many metamodeling techniques including polynomial response surface (PRS) approximations (Myers and Montgomery 2002), multivariate adaptive regression splines (Friedman 1991), radial basis functions (RBF; Hardy 1971; Dyn et al. 1986; Mullur and Messac 2004), Kriging (KR; Sacks et al. 1989; Armstrong and Champigny 1989; Martin and Simpson 2005), Gaussian process (GP; MacKay 1998; Daberkow and Marvis 2002; Wang et al. 2005), neural networks (Smith 1993), and support vector regression (SVR; Gunn 1997; Clarke et al. 2005).

A review of literature provides many examples where the accuracy and efficiency of various metamodels for linear, nonlinear, smooth, and noisy responses have been investigated (Giunta and Watson 1998; Simpson et al. 2001a, b; Jin et al. 2001; Papila et al. 2001; Stander et al. 2004; Clarke et al. 2005; Fang et al. 2005; Wang et al. 2006). For instance, Fang et al. (2005) found RBF gives accurate metamodels for highly nonlinear responses whereas Simpson et al. (2001b) found Kriging to be most suitable for slightly nonlinear responses in high-dimension spaces. Jin et al. (2001) proposed the use of PRS for slightly nonlinear and noisy responses, while Clarke et al. (2005) found SVR metamodels to be the best in their study. Queipo et al. (2005) provide a good review of different metamodeling techniques. In addition, a more recent and extensive review of metamodeling can be found in Wang and Shan (2007). While the studies cited above identify a single metamodel as being accurate for a particular form of response, in this study we will show that it is possible to find a more accurate metamodel by generating an ensemble based on optimized weight factors.

The lack of sufficient information describing the relationship between the response and the input variables makes it difficult for an engineer to know which metamodel is the best for a specific response. In addition, due to the dependence of metamodel accuracy on the selected DOE type, the number of design points in the training data set, and the form (e.g., linear, nonlinear, noisy, smooth) of the response, there is uncertainty in metamodel predictions as noted by Goel et al. (2007). Therefore, as an alternative to using a single metamodel for a response of interest, it would be beneficial to combine multiple metamodels in a weighted-sum formulation. The resulting hybrid metamodel takes advantage of the prediction ability of each individual stand-alone metamodel to enhance the accuracy of the response predictions.

The idea of combining different approximate models into a single hybrid model can be traced to the development of committees of neural networks by Perrone and Cooper (1993) with further refinement by Bishop (1995). More recently, Zerpa et al. (2005) and Goel et al. (2007) have extended this approach for developing ensembles of metamodels. In a related field, the recent development of ensemble of Kalman filters (Evensen 2003) is also a good example. It is possible to consider an ensemble approach as an alternative to model selection in statistics; there is a large body of work in this area including those by Madigan and Raftery (1994) and Buckland et al. (1997).

Motivated by the previous research, this paper offers a different approach for building an ensemble of metamodels by finding the optimized values of weight factors that would minimize a selected error metric (e.g., root mean square error). To demonstrate the capability of the proposed approach, the ensemble is assumed to be made up of five different stand-alone metamodels (i.e., PRS, RBF, KR, GP, and SVR) with a summary of each technique provided in the “Appendix.”

The remainder of the paper is organized as follows. “Section 2” presents the basic weighted-sum formulation and the different approaches that can be followed for selecting the weight factors for the individual metamodels. “Section 3” describes the example problems considered and the numerical procedure for finding an ensemble with optimized weight factors. The presentation and discussion of results appear in “Section 4,” followed by the summary of important conclusions in “Section 5.”

2 Ensemble of metamodels

2.1 Weighted-sum formulation

If all the stand-alone metamodels developed for a given response happen to have the same level of accuracy, then an acceptable form for the ensemble would be a simple average of the metamodels. However, this is not generally the case because some metamodels tend to be more accurate than others. Hence, in attempting to enhance the accuracy of the ensemble, the stand-alone metamodels (members of the ensemble) have to be multiplied by different weight factors. By using a weighted-sum formulation (Bishop 1995), an ensemble of metamodels for approximation of response $y(\mathbf{x})$ is expressed as

$$\hat{y}_e(\mathbf{x}) = \sum_{i=1}^M w_i(\mathbf{x}) \hat{y}_i(\mathbf{x}) \quad (1)$$

where \hat{y}_e is the ensemble-predicted response, M is the number of metamodels in the ensemble, w_i is the weight factor for the i th metamodel, \hat{y}_i is the response estimated by the i th metamodel, and \mathbf{x} is the vector of independent input variables. The weight factors in (1) are calculated while satisfying the requirement

$$\sum_{i=1}^M w_i(\mathbf{x}) = 1 \tag{2}$$

In general, the weight factors are selected such that the metamodels with high accuracy have large weight factors and vice versa.

2.2 Weight factors selection based on error correlation

The training data set represents a collection of N design points (i.e., training points) that are identified according to the DOE technique used. The response values at these training points are used to fit a stand-alone metamodel. For forming a committee of neural networks, Bishop (1995) proposed selecting the weight factors as

$$w_i = \frac{\sum_{j=1}^M (\mathbf{C}^{-1})_{ij}}{\sum_{m=1}^M \sum_{j=1}^M (\mathbf{C}^{-1})_{mj}} \tag{3}$$

where \mathbf{C} is the error correlation matrix whose elements are calculated from

$$C_{ij} = \frac{1}{N} \sum_{k=1}^N (\hat{y}_i^k - y^k)(\hat{y}_j^k - y^k) \tag{4}$$

where y^k is the true response value corresponding to input vector \mathbf{x}_k , with \hat{y}_i^k and \hat{y}_j^k the corresponding predicted values by the i th and j th neural networks, respectively. Equations (3) and (4) can also be used in generating an ensemble of metamodels.

Selecting weight factors from (3) minimizes the error in the whole domain of input variables (Bishop 1995) based on the assumption that the errors of different neural networks are uncorrelated and unbiased (that is, with zero mean), which is not always true. In addition, even though this approach may be suitable for application to committee of neural networks, it may be unsuitable for ensembles based on other metamodeling techniques. In the case of neural networks, because there is a difference between the true response computed at each training point and the prediction of a neural network, \hat{y}_i^k , the error correlation matrix \mathbf{C} is nonvanishing. However, if the error metric is chosen as the difference between the predicted and true responses at the training points (see (4)), then for some

types of metamodels (e.g., RBF and KR) the difference is zero and the correlation matrix \mathbf{C} becomes a null matrix. A possible solution to this problem is to use k -fold cross-validation errors instead.

2.3 Weight factors selection based on prediction variance

Motivated by the work of Bishop (1995), Zerpa et al. (2005) proposed the use of a weighted-sum model of different metamodels (RS, KR, and RBF) for the optimization of an alkali surfactant-polymer flooding process. They chose the prediction variance as the error metric and set the value of weight factor for each metamodel to be inversely proportional to the point-wise estimate of the prediction variance as

$$w_i = \frac{1}{V_i} / \sum_{j=1}^M \frac{1}{V_j} \tag{5}$$

where V_i is the prediction variance of the i th metamodel. The selection of weights via (5) minimizes the prediction variance of the weighted-sum model based on the assumption that the metamodel predictions are unbiased and uncorrelated.

2.4 Weight factors selection based on GMSE or RMSE

The generalized mean square cross-validation error (GMSE) is very similar to prediction error sum of squares statistic. If there are N training points, then a metamodel is constructed N times, each time leaving out one of the training points. Then the difference between the exact response at the omitted point and that predicted by each variant metamodel is used to evaluate the global error as

$$GMSE = \frac{1}{N} \sum_{k=1}^N (y^k - \hat{y}^{(k)})^2 \tag{6}$$

where y^k is the true response at \mathbf{x}_k and $\hat{y}^{(k)}$ is the corresponding predicted value from the metamodel constructed using all except the k th design point. As evident by (6), the greater the number of training points the higher the cost of calculating the GMSE metric. This metric gives an average error in the estimated response at the selected training points. Therefore, depending on the number and distribution of training points, GMSE may not necessarily provide an evidence of global error in the whole domain of input variables.

Goel et al. (2007) considered an ensemble of three metamodels (RS, KR, and RBF), used the GMSE of the individual metamodels, and selected the appropriate weight factors as

$$w_i = w_i^* / \sum_{j=1}^M w_j^* \quad (7a)$$

$$w_i^* = (E_i + \alpha \bar{E})^\beta \quad (7b)$$

$$\bar{E} = \frac{1}{M} \sum_{i=1}^M E_i \quad (7c)$$

where E_i is the GMSE of the i th metamodel with $\beta < 0$ and $\alpha < 1$. In this approach, the analyst specifies the values of parameters α and β based on the importance of E_i and \bar{E} . According to (7), a metamodel with a large GMSE shall have a small role in the ensemble by receiving a small weight factor and vice versa.

Whereas the freedom of selecting the parameters α and β gives the analyst some flexibility, the optimal choice of these parameters requires experimentation. For example, Goel et al. (2007) found that $\alpha = 0.05$ and $\beta = -1$ leads to a good model in their study. Here, we suggest that these parameters could have been selected so as to minimize the GMSE of the ensemble. In fact, when the selection of weight factors is based on the minimization of GMSE, then a parametric model such as (7b) and (7c) is not necessary as will be shown later.

In general, Goel et al. (2007) showed that the GMSE is a reasonably good substitute for the root mean square error (RMSE); however, they noted that GMSE overestimated RMSE for PRS while underestimated it for KR and RBF metamodels. Similarly, Martin and Simpson (2005) found that if a metamodel's predicted R^2 (based on GMSE) is greater than 0.90, then the metamodel's actual R^2 (based on RMSE) is also greater than 0.90.

Besides GMSE, it is also possible to use the RMSE as the error metric, E_i in (7b) and (7c). In that case, RMSE is calculated based on data at some randomly selected design points belonging to a test data set; these test points are separate from the training points and as such provide a less biased measure of global error in the model. Depending on the precision level sought for estimating the error, the number of test points, denoted by T , can vary with a typical value in the range of $0.1N$ to $0.4N$. Whereas for GMSE calculation, N variants of the same metamodel are constructed based on

responses evaluated at N design points, in RMSE only a single metamodel is constructed, but the response values at $N(\text{training}) + T(\text{test})$ points are needed. Hence, for the former error metric, the computational cost is tied to the development of multiple variants of the metamodel, while for the latter error metric the cost burden is in the additional simulations for response determination. Depending upon the type of metamodel and the computational cost of response calculation, one error metric would be less expensive to evaluate than the other.

2.5 Weight factors selection based on error minimization

Here, the weight factors in (1) are selected by solving an optimization problem of the form

Find w_i , $i \in M$ that would

$$\begin{aligned} \min \quad & \varepsilon_e = \text{Err}\{\hat{y}_e(w_i, \hat{y}_i(\mathbf{x}^k))y(\mathbf{x}^k), k = 1 \in N\} \\ \text{s.t.} \quad & \sum_{i=1}^M w_i = 1 \end{aligned} \quad (8)$$

where $\text{Err}\{\}$ is the selected error metric that measures the accuracy of the ensemble-predicted response, \hat{y}_e .

It is worth noting that the availability and diversity of metamodels are the two basic criteria for membership in the ensemble (1). Should one stand-alone metamodel be significantly less accurate than the rest, its corresponding weight factor would be reduced accordingly in the solution of (8).

Because only one (global) ensemble is sought to describe the response variation over the range of all input variables, the problem in (8) is solved only once. Hence, the computational cost of solving (8) depends only on the form of the objective function (i.e., error metric) and the number of design variables, which is the same as the number of members in the ensemble; as such, it would constitute only a small fraction of the overall computational cost. Optimization cost is on par with fitting a response surface model. The weight selection in (8) is also different from the adaptive metamodeling approaches (Wang et al. 2001; Wujek and Renaud 1998a, b) where the metamodels are updated during the process as new data points are used.

If, for example, GMSE is chosen as $\text{Err}\{\}$ in (8), then its value for the ensemble is found using (6) with $\hat{y}_e^{(k)}$ replacing $\hat{y}^{(k)}$. Likewise, if RMSE is chosen as the error metric, then the weight factors are evaluated by minimizing this error at an arbitrary set of points

referred to here as the validation points. In this context, RMSE can be defined as

$$RMSE^v = \frac{1}{N_v} \sqrt{\sum_{i=1}^{N_v} [y(\mathbf{x}_i^v) - \hat{y}_e(\mathbf{w}, \mathbf{x}_i^v)]^2} \tag{9}$$

where N_v is the number of validation points and \mathbf{x}_i^v is the vector of input variables corresponding to the i th validation point. After the optimal values of the weight factors are found, (1) is used to estimate the response of interest. When using $RMSE^v$, the accuracy of the resulting ensemble depends on the value of N_v , and as such, the best value for N_v would be problem dependent as will be shown in the example problems.

The general procedure based on the formulation in (8) is independent of the error metric selected. For example, it is possible to use any of the other global error metrics reported in the literature such as correlation coefficient between the predicted and the actual responses, coefficient of multiple determination (R^2) and also its adjusted value (R_{adj}^2), average absolute error, or maximum absolute error. Likewise, it is also possible to make the selection via local error measures (e.g., prediction variance) as was done by Sanchez et al. (2006). However, it is important to note that the accuracy of the resulting ensemble does depend on the error metric used. In the following example problems, we present the solution to (8) using two global error metrics (i.e., GMSE and $RMSE^v$).

3 Example problems

To test the performance of the proposed ensemble technique with optimized weight factors, five example problems are considered. In the first four, the true response is described by analytic functions that are commonly used as benchmark problems in the literature whereas in the last one the true responses are obtained from nonlinear transient dynamic finite element simulations of an automobile model in frontal crash with two barriers, one rigid and one deformable.

3.1 Benchmark problems

The benchmark problems are defined by the following four analytical functions:

Branin–Hoo

$$y(x_1, x_2) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos(x_1) + 10 \tag{10}$$

where $x_1 \in [-5, 10]$, and $x_2 \in [0, 15]$.

Camelback

$$y(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \tag{11}$$

where $x_1 \in [-3, 3]$, and $x_2 \in [-2, 2]$.

Goldstein–Price

$$y(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 4x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \tag{12}$$

where $x_1, x_2 \in [-2, 2]$.

Hartman

$$y(\mathbf{x}) = - \sum_{i=1}^m c_i \exp \left[- \sum_{j=1}^n a_{ij} (x_j - p_{ij})^2 \right] \tag{13}$$

where $x_i \in [0, 1]$. Both the three-variable ($n = 3$) and the six-variable ($n = 6$) models of this function are considered. The values of function parameters c_i , a_{ij} and p_{ij} for Hartman-3 and Hartman-6 models, taken from Goel et al. (2007), are given in Tables 1 and 2. The value of the parameter m for both cases is taken as four.

Table 1 Parameters used in Hartman-3 function, $j \in 1, 3$

i	a_{ij}			c_i	p_{ij}		
1	3.0	10.0	30.0	1.0	0.3689	0.1170	0.2673
2	0.1	10.0	35.0	1.2	0.4699	0.4387	0.7470
3	3.0	10.0	30.0	3.0	0.1091	0.8732	0.5547
4	0.1	10.0	35.0	3.2	0.03815	0.5743	0.8828

Table 2 Parameters used in Hartman-6 function, $j \in 1, 6$

i	a_{ij}						c_i	p_{ij}					
1	10.0	3.0	17.0	3.5	1.7	8.0	1.0	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.05	10.0	17.0	0.1	8.0	14.0	1.2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	3.0	3.5	1.7	10.0	17.0	8.0	3.0	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	17.0	8.0	0.05	10.0	0.1	14.0	3.2	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

3.2 Automobile crash problem

A finite element (FE) model of a c-class passenger car shown in Fig. 1 is used for full frontal impact (FFI) and offset-frontal impact (OFI) simulations using the FE code, LS-DYNA. Here, metamodels are sought that can accurately estimate the intrusion distances and average peak accelerations at the floor pan, the driver seat, and steering wheel locations in FFI and OFI scenarios for a crash duration of 100 ms; hence, there are 12 responses of interest.

The input-variable vector consists of the geometric parameters that control the shape (x_1 to x_4) and wall thickness (x_5) of the two side rails (see Fig. 1) as well as the parameters that define variability or uncertainty in the material stress–strain relationship (x_6), offset distance (x_7), impact speed (x_8), and occupant mass (x_9). In an earlier investigation by Rais-Rohani et al. (2006), these responses were modeled using the multiquadric formulation of RBF for subsequent use in the reliability-based design optimization of the side rails. Here, we will consider different metamodeling techniques to compare the estimation accuracies of the stand-alone and ensemble of metamodels for the 12 responses of interest.

3.3 Design and analysis of computer experiments

For the benchmark problems in (10) through (13), Latin hypercube sampling (LHS) technique is used to

select the locations of the training points such that the minimum distance between the design points is maximized. The MATLAB® routine *lhsdesign* and *maximin* criterion with a maximum of 20 iterations is used to obtain the locations of the training points.

To reduce the effect of random sampling, 1,000 different training sets are used for all the benchmark problems except the Hartman-6, which is based on 100 different training sets. Depending on the number of input variables, the training set for each benchmark problem is composed of 12 to 56 design points. Hence, all the corresponding metamodels (stand-alone and ensemble) are constructed multiple times with the error estimate being the average value corresponding to multiple versions (replicates) of the same metamodels. The low computational cost is the reason for considering such a large quantity of training sets and replicates. Ordinarily, it would not be necessary or practical to perform such an elaborate evaluation.

The accuracy of each stand-alone and ensemble model for the benchmark problems is measured using the mean and the coefficient of variation (COV) of GMSE and RMSE error metrics. The COV of error metrics for different examples specifies the variation of function values on data set for different examples. While the GMSE error metric is calculated at the design points in each independent training set and then averaged over the multiple training sets, the RMSE is calculated using the response prediction errors at the individual test points in a single test data set and

Fig. 1 Perturbed geometry of the right side rail at the upper and lower limits of x_1 through x_4

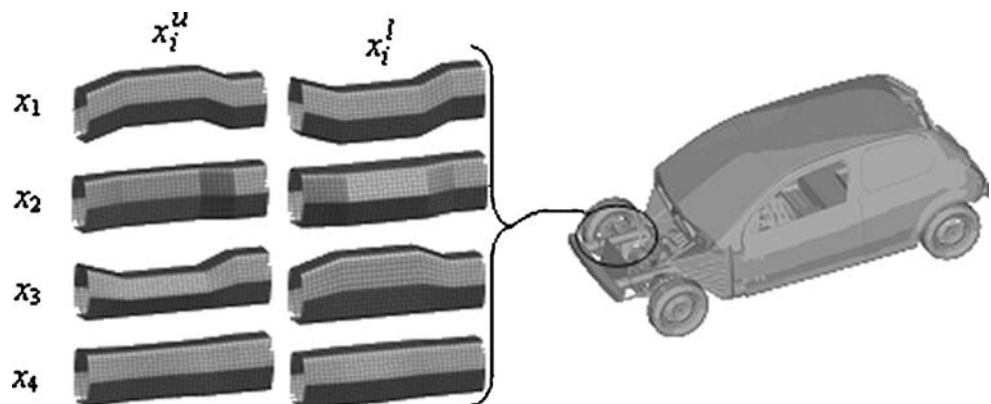


Table 3 Summary of training and test data used in each problem

Problem	Training sets	Design points in a training set	Test points
Branin–Hoo	1,000	12	441
Camelback	1,000	12	441
Goldstein–Price	1,000	12	441
Hartman-3	1,000	20	512
Hartman-6	100	56	496
Automobile crash	1	100	40

then averaged over the multiple training sets. Similar to the locations of the training points, the locations of validation points used in (9) are also selected in random based on LHS.

For the automobile crash problem, the LHS technique is also used; however, because of the computational cost of each high-fidelity simulation (13 CPU hours for FFI and 17 CPU hours for OFI using a 32-processor IBM Corp. Linux Cluster with Intel Pentium III 1.266-GHz processors and 607.5-GB random-access memory), the meta-modeling calculations are done using only a single training set with 100 training points (~ 11 times the number of input variables). Therefore, a total of 200 simulations are performed to generate the pool of true responses for the two crash scenarios.

For the ensemble models found using (8), the *fmincon* function (optimizer) of MATLAB® based on the sequential quadratic programming algorithm is used to solve the optimization problem. In all the example problems, the initial values for the weight factors in both the GMSE and RMSE minimization problems are chosen to be $1/M = 0.2$. Because the optimization routine is a gradient-based optimizer and the objective function being minimized is not necessarily convex, there is a possibility for the solution to represent only a local optimum.

Additional information about the training and test data sets is provided in Table 3. Depending upon the size of input-variable vector, the number of test points is adjusted to assure accuracy of the error estimates. In the benchmark problems, the number of design points in each training set is equal to twice the number of coefficients in the corresponding quadratic PRS metamodel, while the number of design points in the automobile crash problem is the same as that used in Rais-Rohani et al. (2006).

3.4 Metamodeling techniques

Five different metamodeling techniques are considered here; they include: PRS, RBF, KR, GP, and SVR.

These metamodels are also used as the five members of the ensemble that is developed based on the four previously described techniques.

The PRS metamodel is represented by a second-order (i.e., fully quadratic polynomial) model. The RBF metamodel is based on the multiquadric formulation with the constant, $c = 1$. A Gaussian correlation function and a linear trend model are used in KR metamodel. The covariance function in GP metamodel is selected as the sum of a squared exponential function with automatic relevance determination and covariance function for the input-independent noise (i.e., white noise). In SVR metamodel, a Gaussian kernel function is used with the size of the insensitive zone selected as 1% of the response range. The mathematical descriptions of the five metamodels are provided in the “Appendix.”

While for metamodeling of the benchmark problems, the range of values for each input variable is that shown below the function in (10) to (13), for the automobile crash problem, the nine input variables are normalized in the scale of -1.0 to 1.0 .

4 Results and discussion

With the exception of the automobile crash example, the error values reported in this section represent the average value over the number of training sets used in each case. The abbreviated symbols introduced previously are used to identify the individual metamodels. For the ensembles, the model based on simple averaging is labeled as EA, the one based on the technique of Goel et al. (2007) is denoted by EG, the one based on GMSE minimization (at the training points) is labeled as EP, and the one derived from RMSE minimization (at the validation points) is denoted by EV_{N_v} . The effectiveness of each model is measured according to its ability to reduce the selected error metric. To facilitate the comparison of different models, the error values are normalized with respect to the stand-alone metamodel having the lowest error among the five meta-

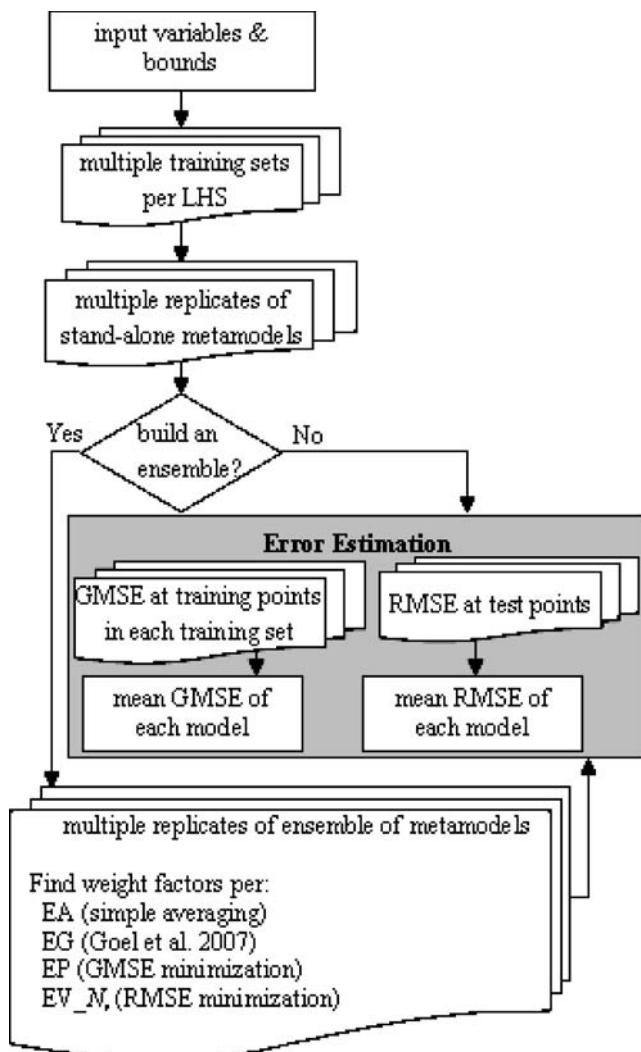


Fig. 2 Development and evaluation steps for each metamodel

models considered. Henceforth, the word ‘normalized’ is dropped when referring to the error. The flowchart in Fig. 2 shows the steps in development and accuracy assessment of a metamodel for each of the benchmark problems. For the automobile crash problem, the multiple overlapping boxes in Fig. 2 should be treated as a single box.

4.1 Branin–Hoo

The results for this problem are given in Table 4, where the lowest error value in each category is shown in bold for ease of comparison. According to the average value of GMSE, SVR is found to be the best stand-alone metamodel for this function whereas RBF is the best when considering the average value of RMSE. Here, EP based on the GMSE minimization yields the smallest GMSE overall. With an average GMSE of 0.81, EP is 20% more accurate than the best stand-alone metamodel and ~15% more accurate than the EG ensemble. However, when considering the RMSE metric, EV_{N_v} ensemble based on RMSE minimization at N_v validation points gives better results, and the improvement is amplified by increasing the number of validation points (N_v = 2, 3, or 5) with EV₅ being the best ensemble in Table 4. The fact that EP gives the lowest mean GMSE value is not surprising as the corresponding weight factors were optimized for that condition. However, when the value of RMSE at the test points is used as the error criterion, EV_{N_v} gives a superior ensemble. It should also be mentioned that the results presented in Table 4 are sensitive to the number of test points used as well as the possibility of encountering outliers. Therefore, the selection of test points requires additional care and oversight.

Recall that the mean and COV values in Table 4 are calculated based on 1,000 different training sets. Hence, the mean values of the GMSE and the RMSE over the selected population sample has a COV of $1/\sqrt{1,000}$ times that of the native COV of the GMSE and the RMSE (reported in rows 4 and 6 of Table 4), respectively. For instance, the COV of the mean GMSE for PRS model is $0.41/\sqrt{1,000} = 0.013$. This number provides an estimate of the standard error in the prediction of mean GMSE over 1,000 training sets, which is fairly small in this case.

The mean values of weight factors over multiple replicates of each ensemble are listed in Table 5. The values inside parentheses in Table 5 are the standard

Table 4 Comparison of accuracies of stand-alone and ensemble models for Branin–Hoo

Error metric	Stand-alone					Ensemble					
	PRS	RBF	KR	GP	SVR	EA	EG	EP	EV_2	EV_3	EV_5
GMSE (mean)	1.25	1.02	1.30	1.06	1.00	0.99	0.95	0.81	1.09	1.09	1.07
GMSE (COV)	0.41	0.31	0.33	0.31	0.26	0.31	0.32	0.35	0.35	0.33	0.33
RMSE (mean)	1.22	1.00	1.18	1.01	1.07	0.97	0.97	1.02	0.95	0.94	0.91
RMSE (COV)	0.15	0.21	0.38	0.38	0.23	0.27	0.27	0.23	0.29	0.28	0.28

Table 5 Means and standard deviations of weight factors in ensemble models for Branin–Hoo

Type of ensemble	Weight factors				
	PRS	RBF	KR	GP	SVR
EA	0.200 (0)	0.200 (0)	0.200 (0)	0.200 (0)	0.200 (0)
EG	0.189 (0.057)	0.214 (0.027)	0.171 (0.030)	0.210 (0.040)	0.216 (0.034)
EP	0.251 (0.296)	0.189 (0.285)	0.029 (0.119)	0.209 (0.320)	0.322 (0.353)
EV_3	0.157 (0.283)	0.144 (0.296)	0.195 (0.334)	0.254 (0.376)	0.249 (0.367)

deviations. For the Branin–Hoo function, SVR has the largest weight factor in all the weighted-sum ensemble models. For the EG, EP, and EV_3 models, the preference for the other metamodels is mixed. Whereas after SVR, EG ensemble favors the RBF metamodel; EP and EV_3 ensembles favor PRS and KR metamodels, respectively. The standard deviation values show relatively large variability in the individual weight factors in EP and EV_3 models but a somewhat small variation in the case of EG.

When considering the results in Tables 4 and 5, it appears that the stand-alone metamodel with the lowest GMSE value is most often the one with the highest weight factor in the weighted-sum ensemble models. However, no trend is found in the relationship between GMSE and weight factors for the other four metamodels.

Figures 3 and 4 show the boxplots for the error metrics, GMSE and RMSE, respectively, corresponding to the nine different metamodels for the Branin–Hoo function. The boxplots provide a graphical depiction of how the normalized value of each metric varies over the range of training sets used. The bottom and top of each box represent the lower and upper quartile values, respectively, with the interior line representing the median. The broken line (whiskers) extending from each

end of the box indicates the extent of the remaining data relative to the lower and upper quartiles. Here, the maximum whisker length is set at 1.5 times the interquartile range, and the data beyond this limit (if present) are characterized as outliers and represented by the + symbols.

When comparing the effectiveness of EP and EV_3 ensembles, it is important to note the difference in the number of design points (and responses) used in each case. For GMSE minimization in this problem, responses at 12 training points are used whereas for RMSE minimization with $N_v = 3$, there are three additional responses (15 total).

To determine what happens under an equivalent condition, we evaluated a test case in which 15 training points are used in GMSE minimization while 12 training plus 3 validation points are used in RMSE^v minimization.

The results in Table 6 indicate that, in this case, GP is the best stand-alone metamodel for both error metrics, which is different from the previous results in Table 4. This finding shows that the performance of metamodels has a strong dependency on the number of training points used. Similarly, the performance of the ensemble of metamodels is altered as the number of training points is changed. In Table 6, EP has the

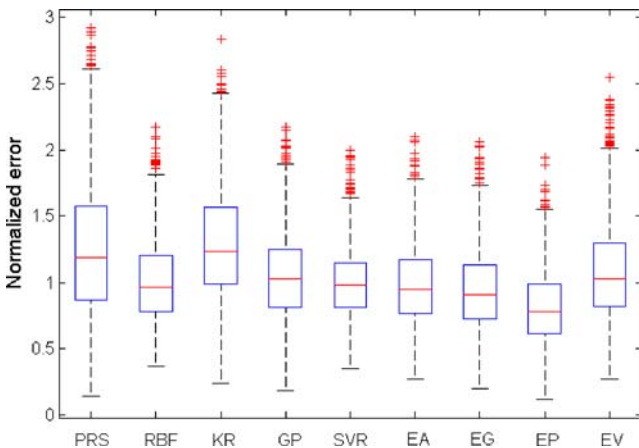


Fig. 3 Boxplots of normalized GMSE over 1,000 training sets

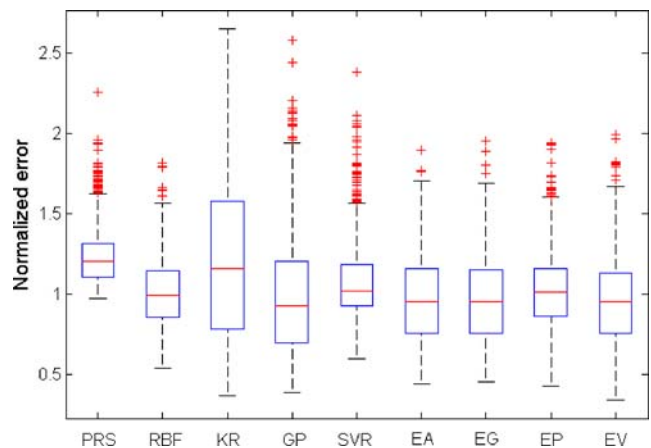


Fig. 4 Boxplots of normalized RMSE over 1,000 training sets

Table 6 Comparison of accuracies of stand-alone and ensemble models for Branin–Hoo based on an equivalent number of data points

Error metric	Stand-alone					Ensemble			
	PRS	RBF	KR	GP	SVR	EA	EG	EP	EV_3
GMSE (mean)	1.42	1.09	1.24	1.00	1.09	1.03	0.98	0.84	1.03
GMSE (COV)	0.34	0.31	0.35	0.37	0.30	0.32	0.33	0.37	0.32
RMSE (mean)	1.52	1.15	1.11	1.00	1.27	1.06	1.04	1.07	1.06
RMSE (COV)	0.12	0.22	0.51	0.37	0.22	0.30	0.31	0.31	0.30

best performance overall. Note also that the values in the last column of Table 6 (for EV_3) are different from the corresponding values in Table 4. The reason for this difference is the use of different data sets for Tables 4 and 6. This difference shows the high variability in function values and dependence on data sets.

4.2 Other benchmark problems

For the Camelback, Goldstein–Price, Hartman-3, and Hartman-6 functions, the values of GMSE and RMSE for the stand-alone and ensemble of metamodells are

compared in Fig. 5. The general trend in Fig. 5 is that the best metamodell based on GMSE is different from that based on RMSE metric. When the GMSE metric is used, the EP ensemble clearly works the best. On average, the error of EP ensemble is 13% less than that of the most accurate stand-alone metamodell. When the RMSE metric is used, then the EV_3 ensemble provided the best model; of course, the performance of EV_3 improves as the number of validation points is increased. Average weight factors (average over 1,000 DOEs) of metamodells in different ensembles for Camelback, Goldstein–Price, Hartman-3, and Hartman-6 functions are presented in Fig. 6. Note that

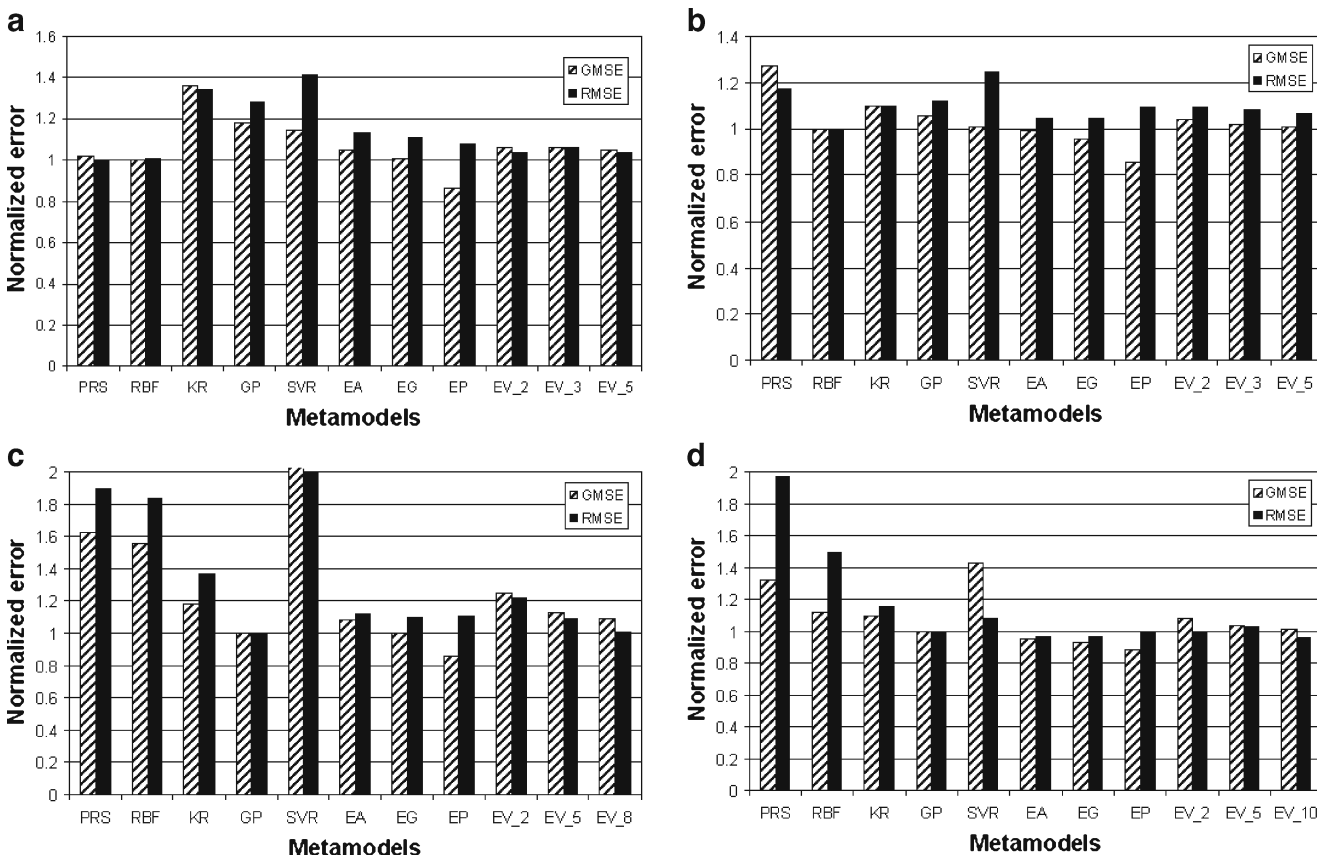


Fig. 5 Comparison of stand-alone and ensemble of metamodells for **a** Camelback, **b** Goldstein–Price, **c** Hartman-3, and **d** Hartman-6 functions

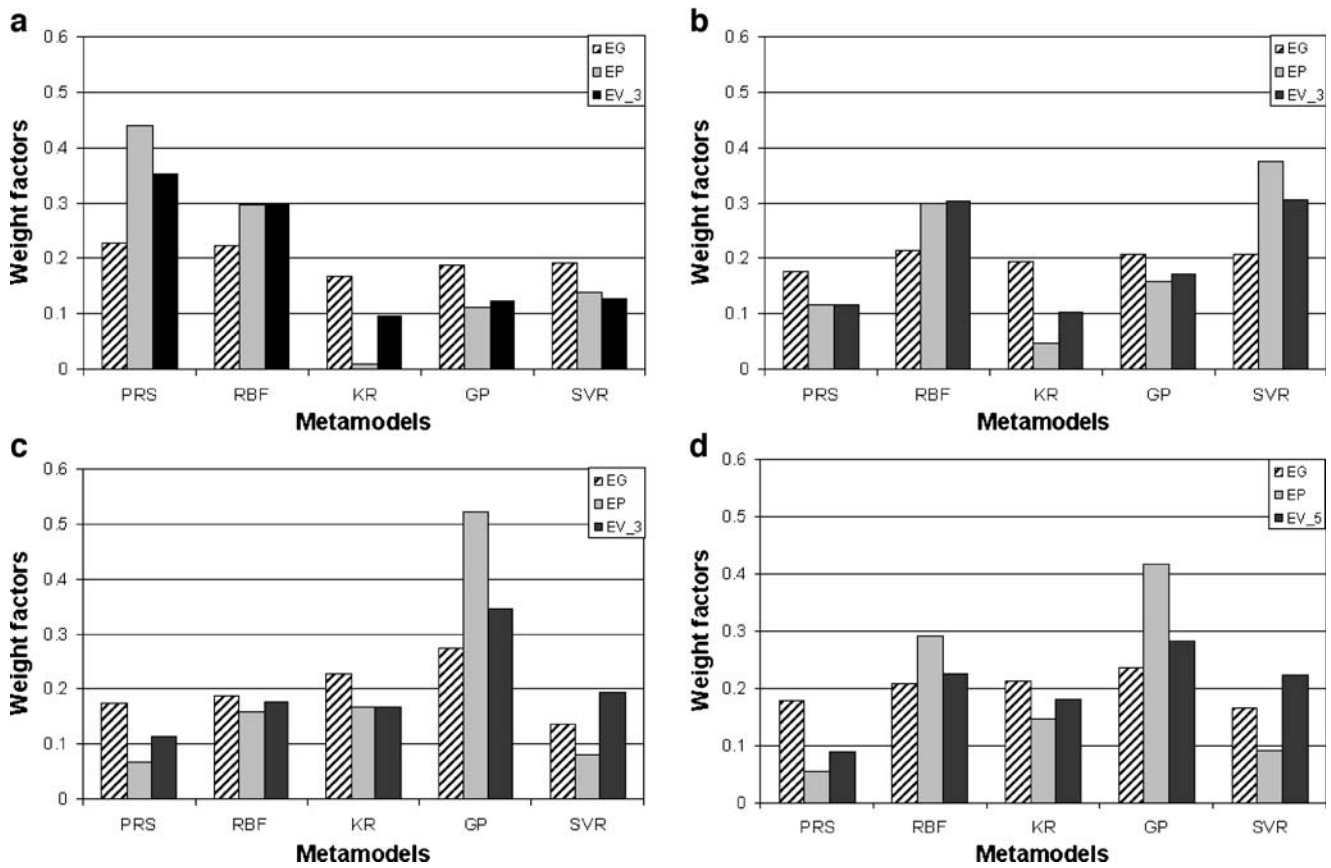


Fig. 6 Weight factors of metamodels (averaged over 1,000 training sets) in different ensembles for **a** Camelback, **b** Goldstein-Price, **c** Hartman-3, and **d** Hartman-6 functions

the weight factors of the metamodels for EA ensemble is equal to $1/M = 0.2$.

4.3 Automobile crash problem

Table 7 identifies the best stand-alone and ensemble of metamodels for the 12 responses of interest. In terms

Table 7 Accuracy of metamodels for crash problem

Metamodel type	Number of responses most accurately modeled based on	
	GMSE	RMSE
PRS	0	0
RBF	1	9
KR	4	1
GP	7	2
SVR	0	0
EA	0	0
EG	0	0
EP	12	0
EV_40	0	12

of GMSE, GP is found to be the best stand-alone metamodel for the majority of responses whereas RBF appears the best in terms of RMSE metric. As for the ensembles, EP gives the best ensemble based on GMSE metric whereas EV_40 is the best according to RMSE value. As an example, the accuracies of the metamodels

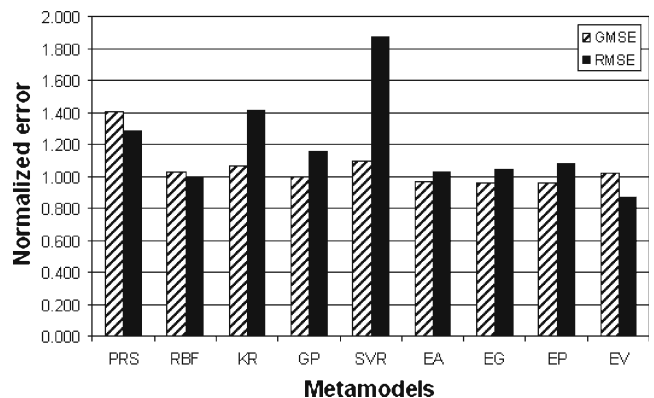


Fig. 7 Comparison of stand-alone and ensemble of metamodels for one of the responses (acceleration at driver seat under full frontal impact) in the crash problem

for one critical response (i.e., acceleration at driver seat under full frontal impact) are compared in Fig. 7 where EV_40 is shown to be the best model overall.

4.4 General findings

After the examination of results for the five example problems, the general findings can be summarized as follows:

1. An ensemble of metamodels with optimized weight factors based on EP or EV_{N_v} formulation always outperforms its stand-alone members when judged by the same error metric.
2. Although the choice of error metric does not change the overall approach, which circles around the solution to the optimization problem in (8), it can influence the accuracy of the resulting ensemble.
3. When using GMSE (i.e., focusing on average error at the training points), EP ensemble with its weight factors found through GMSE minimization outperforms the other ensembles.
4. When using RMSE (i.e., focusing on average error at the test points), EV_{N_v} ensemble with its weight factors found through RMSE^v minimization outperforms the other ensembles.
5. While the general trend indicates that the higher the value of N_v the better the ensemble, the rate of improvement varies from one problem to another. Because the best value for N_v tends to be problem dependent, it is not possible to establish strict guidelines on its selection. However, in general, the required value for N_v is far less than the number of design points in the training set.
6. While the use of EV_{N_v} ensemble can be advantageous, it tends to be more expensive due to additional simulations.
7. While it is easy to safeguard against outliers among the test points, it is not as easy to do the same with the validation points. Although not used here, the use of an outlier analysis can help alleviate this problem.

5 Conclusions

A new approach for selecting the weight factors in an ensemble of metamodels was presented. This approach extends earlier efforts on development of ensembles by treating the weight factors as design variables in

an optimization problem whereby a global or local error metric is selected as the objective function to be minimized. Although the approach is not tied to any specific error metric, we illustrated the procedure using GMSE and RMSE^v as two examples.

The effectiveness of the proposed approach was tested on four analytic functions or benchmark problems as well as an automobile crash problem requiring high-fidelity simulations of a complex model with nonlinear responses. We considered five stand-alone metamodels and four alternative ensemble techniques in these test problems.

The results generally favored the proposed ensemble approach over the other techniques considered. As to be expected, the accuracy of the ensemble and the computational cost of finding the optimum weight factors depend on the choice of error metric used. Of the two error metrics used, the one based on GMSE is more efficient as it uses information from the training set whereas the one based on RMSE^v is more accurate as it also includes additional responses at the validation points. The computational cost of using the proposed ensemble approach, similar to the other approaches, is in finding the responses at multiple design points and the fitting of stand-alone metamodels. By contrast, the solution of the optimization problem requires only a very short computational time.

Topics being contemplated for future work include: the use of other error metrics that can be considered as better representatives of error in the whole design domain, deeper investigation into the minimum number of test points and validation points for an ensemble with a specified level of accuracy, the effect of selective inclusion of specific stand-alone metamodels on the accuracy of the ensemble.

Acknowledgements The funding provided for this study by the US Department of Energy under Grant No. DE-FC26-06NT42755 is gratefully acknowledged. The authors also wish to thank Dr. Raphael T. Haftka and Dr. Tushar Goel for their helpful comments.

Appendix

A.1 Description of selected metamodeling techniques

In this appendix, a brief overview of the mathematical formulation of PRS, RBF, GP, KR, and SVR metamodeling techniques is provided.

A.2 Polynomial response surface approximations

The most commonly used PRS model is the second-order model in the form of a second-degree algebraic polynomial function as

$$\hat{f}(x) = b_0 + \sum_{i=1}^L b_i x_i + \sum_{i=1}^L b_{ii} x_i^2 + \sum_{i=1}^{L-1} \sum_{j=i+1}^L b_{ij} x_i x_j \quad (14)$$

where \hat{f} is the response surface approximation of the actual response function, f , L is the number of variables in the input vector \mathbf{x} , and b_0, b_i, b_{ii}, b_{ij} are the unknown coefficients to be determined by the least squares technique.

A.3 Radial basis function

RBF methods were originally developed to approximate multivariate functions based on scattered data. For a data set consisting of the values of input variables and response values at n sampling points, the true function $f(\mathbf{x})$ can be approximated as

$$\tilde{f}(\mathbf{x}) = \sum_{i=1}^n \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (15)$$

where \mathbf{x} is the vector of input variables, \mathbf{x}_i is the vector of input variables at the i th sampling point, $\|\mathbf{x} - \mathbf{x}_i\| = \sqrt{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}$ is the Euclidean norm representing the radial distance, r from design point \mathbf{x} to the sampling point or center \mathbf{x}_i , ϕ is a radially symmetric basis function, and $\lambda_i, i = 1, n$ are the unknown interpolation coefficients. Equation (15) represents a linear combination of a finite number of radially symmetric basis functions. Some of the most commonly used RBF formulations include: $\phi(r) = r^2 \log(r)$ (thin-plate spline); $\phi(r) = e^{-\alpha r^2}, \alpha > 0$ (Gaussian); $\phi(r) = \sqrt{r^2 + c^2}$ (multiquadric); and $\phi(r) = 1/\sqrt{r^2 + c^2}$ (inverse multiquadric). The parameter c in the multiquadrics is a constant. If the r values are normalized to the range of $(0, 1)$, then $0 < c \leq 1$. The choice of $c = 1$ is found to be suitable for most function approximations. The feature that makes these functions excellent candidates for ϕ is not simply their radial symmetry but their smoothness and certain properties of their Fourier transform (Buhmann 2003). In this study, we have chosen the multiquadric formulation of RBF because of its prediction accuracy and its commonly linear and possibly exponential rate of convergence with increased sampling points.

Given the design coordinates of n sampling points and associated responses, the unknown coefficients in (15) are found by minimizing the residual or the sum of the squares of the deviations expressed as

$$R = \sum_{j=1}^n \left[f(\mathbf{x}_j) - \sum_{i=1}^n \lambda_i \phi(\|\mathbf{x}_j - \mathbf{x}_i\|) \right]^2 \quad (16)$$

Expressed in matrix form, (16) appears as

$$[\mathbf{A}] \{\boldsymbol{\lambda}\} = \{\mathbf{f}\} \quad (17)$$

where $[\mathbf{A}] = [\phi \|\mathbf{x}_j - \mathbf{x}_i\|], j = 1, n; i = 1, n, \{\boldsymbol{\lambda}\}^T = \{\lambda_1, \lambda_2, \dots, \lambda_n\}^T$, and $\{\mathbf{f}\}^T = \{\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2), \dots, \mathbf{f}(\mathbf{x}_n)\}^T$. The coefficient vector $\boldsymbol{\lambda}$ is obtained by solving (17).

A.4 Gaussian process

Gaussian process assumes that the output variables $f_N = \{f_n(x_n^1, x_n^2, \dots, x_n^L)\}_{n=1}^N$ are related to each other with a Gaussian joint probability distribution

$$P(f_N | C_N, X_N) = \frac{1}{\sqrt{(2\pi)^N |C_N|}} \times \exp \left[-\frac{1}{2} (f_N - \mu)^T C_N^{-1} (f_N - \mu) \right] \quad (18)$$

where $X_N = \{x_n\}_{n=1}^N$ are N pairs of L -dimensional input variables $x_n = (x_n^1, x_n^2, \dots, x_n^L)$, C_N is the covariance matrix with elements of $C_{ij} = C(x_i, x_j)$. μ is the mean output vector. GP estimates the output at a prediction point $\mathbf{x}_p = (x_p^1, x_p^2, \dots, x_p^L)$ as

$$\hat{f}(\mathbf{x}_p) = k^T C_N^{-1} f_N \quad (19)$$

where $k = [C(x_1, x_p), \dots, C(x_N, x_p)]$. One of the nice properties of the GP is that the standard deviation at the prediction point is readily available without a requirement of any extra simulations. This standard deviation can be utilized as an error measure and can be calculated from

$$\sigma_{\hat{f}(\mathbf{x}_p)} = \kappa - k^T C_N^{-1} k \quad (20)$$

where $\kappa = C(x_p, x_p)$.

We notice from (19) that the GP prediction depends on the covariance matrix C_N . The elements of this matrix are calculated from

$$C_{ij} = \theta_1 \exp \left[-\frac{1}{2} \sum_{l=1}^L \frac{(x_i^{(l)} - x_j^{(l)})^2}{r_l^2} \right] + \theta_2 \quad (21)$$

$$C_{ij} = \theta_1 \exp \left[-\frac{1}{2} \sum_{l=1}^L \frac{(x_i^{(l)} - x_j^{(l)})^2}{r_l^2} \right] + \theta_2 + \delta_{ij}\theta_3 \quad (22)$$

where $\theta_1, \theta_2, \theta_3$, and r_l ($l = 1, 2, \dots, L$) are called ‘‘hyperparameters.’’ Here, δ_{ij} is the Kronecker delta and θ_3 is an independent noise parameter. The hyperparameters are selected so as to maximize the logarithmic likelihood that the model prediction matches the training response data. The logarithmic likelihood function L is given in (23).

$$L = -\frac{1}{2} \log |C_N| - \frac{1}{2} f_N^T C_N^{-1} f_N - \frac{N}{2} \log 2\pi + \ln P(\theta) \quad (23)$$

where $P(\theta)$ is the prior distribution of the hyperparameters. In most of the applications, there is no prior knowledge of the values of the hyperparameters, so the prior distribution is uniform. Then, the last term of (23), $\ln P(\theta)$, is a constant and can be taken as zero for the purpose of optimization, as we did in this work.

The covariance function given in (21) defines the interpolation mode of the GP metamodel that passes through all the training data points exactly. On the other hand, (22) defines the regression mode of the model, which allows us to build smoother surfaces for problems with noisy data.

With the noise of the output values filtered out, the predicted surface becomes less complex and may not pass through all the training points; however, it provides a better prediction at the nontraining points. In this work, we used Gaussian process code from Rasmussen and Williams (2006).

A.5 Kriging

The basic assumption of KR is the estimation of the response in the form

$$f(x) = p(x) + Z(x) \quad (24)$$

where f is the response function of interest, p is a known polynomial that globally approximates the response, and $Z(x)$ is the stochastic component that generates deviations such that the Kriging model

interpolates the sampled response data. The stochastic component has a mean value of zero and covariance of

$$COV [Z(x_i), Z(x_j)] = \sigma^2 \mathbf{R} [R(x_i, x_j)] \quad (25)$$

where \mathbf{R} is $N \times N$ correlation matrix if N is the number of data points, $R(x^i, x^j)$ is correlation function between the two data points x^i and x^j . Mostly, the correlation function is chosen as Gaussian, that is,

$$R(\theta) = \prod_{k=1}^L \exp(-\theta_k d_k^2) \quad (26)$$

where L is the number of variables, $d_k = x_k^i - x_k^j$ is the distance between the k th components of the two data points x^i and x^j , and θ_k are the unknown parameters to be determined.

Once the correlation function has been selected, the response f is predicted as

$$\hat{f}(x) = \hat{\beta} + \mathbf{r}^T(x) \mathbf{R}^{-1} (\mathbf{f} - \hat{\beta} \mathbf{p}) \quad (27)$$

where $\mathbf{r}^T(x)$ is the correlation vector of length N between a prediction point x and the N sampling points, f represents the responses at the N points and p is an L -vector of ones (in the case that $p(x)$ is taken as a constant). The vector \mathbf{r} and scalar $\hat{\beta}$ are given by

$$\begin{aligned} \mathbf{r}^T(x) &= [R(x, x^1), R(x, x^2), \dots, R(x, x^N)]^T, \\ \hat{\beta} &= (\mathbf{p}^T \mathbf{R}^{-1} \mathbf{p})^{-1} \mathbf{p}^T \mathbf{R}^{-1} \mathbf{f} \end{aligned} \quad (28)$$

The variance of the output model (which is different than the variance of the sampled output) can be estimated as

$$\hat{\sigma}^2 = \frac{(\mathbf{f} - \hat{\beta} \mathbf{p})^T \mathbf{R}^{-1} (\mathbf{f} - \hat{\beta} \mathbf{p})}{N} \quad (29)$$

The unknown parameters θ_k can be estimated by solving the following constrained maximization problem (Simpson et al. 2001a, b)

$$\begin{aligned} \text{Max } \Phi(\Theta) &= \frac{-[N \ln(\hat{\sigma}^2) + \ln |\mathbf{R}|]}{2} \\ \text{s.t. } \Theta &> 0 \end{aligned} \quad (30)$$

where Θ is the vector of unknown parameters θ_k , and both $\hat{\sigma}$ and \mathbf{R} are functions of Θ .

In this work, we use a MATLAB® Kriging toolbox developed by Lophaven et al. (2002).

A.6 Support vector regression, SVR

The prediction via SVR can be performed through linear or nonlinear regression. When linear regression is performed, then the function is predicted as

$$\hat{f}(x) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + \mathbf{b} \tag{31}$$

where $\langle \mathbf{w} \cdot \mathbf{x} \rangle$ is the dot product of \mathbf{w} and \mathbf{x} . We would like to have the approximation function as flat as possible. For that purpose, we solve the following optimization problem

$$\begin{aligned} \text{Min} \quad & \frac{1}{2} |\mathbf{w}|^2 \\ \text{s.t.} \quad & y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - b \leq \varepsilon \\ & \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b - y_i \leq \varepsilon \end{aligned} \tag{32}$$

This formulation assumes that the function $\hat{f}(x)$ can approximate all the y_i training points within an ε precision. However, this may not be true for all training points and two slack variables can be introduced to yield a modified formulation (Vapnik et al. 1997). Now, the optimization problem is

$$\begin{aligned} \text{Min} \quad & \frac{1}{2} |\mathbf{w}|^2 + C \sum_{i=1}^l \xi_i + \xi_i^* \\ \text{s.t.} \quad & y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - b \leq \varepsilon + \xi_i \\ & \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \tag{33}$$

where C determines the tradeoff between the flatness and tolerance. The second term in the objective function is referred as ε -insensitive loss function (Vapnik et al. 1997). If we write the Lagrangian function, assess the Karush–Kuhn–Tucker (KKT) conditions, and substitute KKT conditions into the Lagrangian function, we can write the optimization problem in dual form as

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \\ & - \varepsilon \sum_{i=1}^l (\alpha_i - \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad & \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ & (\alpha_i - \alpha_i^*) \in [0, C] \end{aligned} \tag{34}$$

The weights and the linear regression are then calculated through

$$\mathbf{w} = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \mathbf{x}_i, \quad \hat{f}(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + b \tag{35}$$

Instead of using linear regression, nonlinear regression can also be used by replacing the dot product of the input vectors with kernel functions. Commonly used Kernel functions include nonlinear polynomials and Gaussian and sigmoid kernel functions. In this case, the optimization function is written as

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) k(\mathbf{x}_i \cdot \mathbf{x}_j) \\ & - \varepsilon \sum_{i=1}^l (\alpha_i - \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad & \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ & (\alpha_i - \alpha_i^*) \in [0, C] \end{aligned} \tag{36}$$

Then, the support vector regression approximation is obtained through

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) k(\mathbf{x}_i \cdot \mathbf{x}) + b \tag{37}$$

More detailed information can be found in Clarke et al. (2005). In this work, we used the MATLAB® code developed by Gunn (1997).

References

Armstrong M, Champigny N (1989) A study on Kriging small blocks. *CIM Bull* 82(923):128–133

Bishop CM (1995) *Neural networks for pattern recognition*. Oxford University Press, New York, pp 364–369

Buckland ST, Burnham KP, Augustin NH (1997) Model selection: an integral part of inference. *Biometrics* 53:603–618

Buhmann MD (2003) *Radial basis functions: theory and implementations*. Cambridge University Press, New York

Clarke SM, Griebisch JH, Simpson TW (2005) Analysis of support vector regression for approximation of complex engineering analyses. *ASME J Mech Des* 127(11):1077–1087

Daberkow DD, Marvis DN (2002) An investigation of metamodeling techniques for complex systems design. In: *Proceedings of the 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization*, Atlanta, GA, September

Dyn N, Levin D, Rippa S (1986) Numerical procedures for surface fitting of scattered data by radial basis functions. *SIAM J Sci Stat Comput* 7(2):639–659

Evensen G (2003) The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dyn* 53:343–367

Fang H, Rais-Rohani M, Liu Z, Horstemeyer MF (2005) A comparative study of metamodeling methods for multi-objective crashworthiness optimization. *Comput Struct* 83: 2121–2136

Friedman JH (1991) Multivariate adaptive regression splines. *Ann Stat* 19(1):1–141

- Giunta A, Watson LT (1998) A comparison of approximation modeling techniques: polynomial versus interpolating models. In: Proceedings of the 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, St. Louis, MO, 2–4 September 1998, vol. 1, pp. 392–404.
- Goel T, Haftka RT, Shyy W, Queipo NV (2007) Ensemble of surrogates. *Struct Multidisc Optim* 33(3):199–216
- Gunn SR (1997) Support vector machines for classification and regression. Technical report. Image Speech and Intelligent Systems Research Group, University of Southampton, Southampton
- Hardy RL (1971) Multiquadratic equations of topography and other irregular surfaces. *J Geophys Res* 76:1905–1915
- Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodeling techniques under multiple modeling criteria. *Struct Multidisc Optim* 23:1–13
- Lophaven SN, Nielsen HB, Søndergaard J (2002) DACE—A MATLAB Kriging toolbox. Informatics and mathematical modeling. Technical University of Denmark, Lyngby
- MacKay DJC (1998) Introduction to Gaussian processes. In: Bishop CM (ed) *Neural networks and machine learning*, vol 168 of NATO ASI Series, 1998. Springer, Berlin, pp 133–165
- Madigan D, Raftery AE (1994) Model selection and accounting for model uncertainty in graphical models using Occam's window. *J Am Stat Assoc* 89(428):1535–1546
- Martin JD, Simpson TW (2005) Use of Kriging models to approximate deterministic computer models. *AIAA J* 43(4):853–863
- Mullur AA, Messac A (2004) Extended radial basis functions: more flexible and effective metamodeling. In: Proceedings of the 10th AIAA/ISSMO symposium on multidisciplinary analysis and optimization, Albany, NY, Aug–Sept, 2004
- Myers RH, Montgomery DC (2002) *Response surface methodology: process and product optimization using designed experiments*. Wiley, New York
- Papila N, Shyy W, Griffin LW, Dorney DJ (2001) Shape optimization of supersonic turbines using response surface and neural network methods. *J Propuls Power* 18:509–518
- Perrone MP, Cooper LN (1993) When networks disagree: ensemble methods for hybrid neural networks. In: Mammone RJ (ed) *Artificial neural networks for speech and vision*. Chapman and Hall, London, pp 126–142
- Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK (2005) Surrogate-based analysis and optimization. *Prog Aerosp Sci* 41:1–28
- Rais-Rohani M, Solanki K, Eamon C (2006) Reliability-based optimization of lightweight automotive structures for crashworthiness. In: Proceedings of the 11th AIAA/ISSMO multidisciplinary analysis and optimization conference, September 2006, AIAA Paper 2006–7004, Portsmouth, VA
- Rasmussen CE, Williams CKI (2006) *Gaussian processes for machine learning*. MIT, Cambridge
- Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Stat Sci* 4(4):409–435
- Sanchez E, Pintos S, Queipo N (2006) Toward an optimal ensemble of kernel based approximations with engineering applications. In: IEEE world congress on computational intelligence, Vancouver, Canada, July 16–21, 2006
- Simpson TW, Peplinski JD, Koch PN, Allen JK (2001a) Metamodels for computer based engineering design: survey and recommendations. *Eng Comput* 17:129–150
- Simpson TW, Mauery TM, Korte JJ, Mistree F (2001b) Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA J* 39(12):2233–2241
- Smith M (1993) *Neural networks for statistical modeling*. Von Nostrand Reinhold, New York
- Stander N, Roux W, Giger M, Redhe M, Fedorova N, Haarhoff J (2004) A comparison of metamodeling techniques for crashworthiness optimization. In: Proceedings of the 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, Albany, NY, Aug 2004
- Vapnik V, Golowich S, Smola A (1997) Support vector method for function approximation, regression estimation, and signal processing. In: Mozer M, Jordan M, Petsche T (eds) *Advances in neural information processing systems*. MIT, Cambridge, pp 281–287
- Venkataraman S, Haftka RT (2004) Structural optimization complexity: what has Moore's law done for us? *Struct Multidisc Optim* 28:375–387
- Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. *ASME J Mech Des* 129(4):370–380
- Wang G, Dong Z, Aitchison P (2001) Adaptive response surface method—a global optimization scheme for computation-intensive design problems. *J Eng Optim* 33(6):707–734
- Wang JM, Fleet DJ, Hertzmann A (2005) Gaussian process dynamical models. In: Proceedings of the 18th advanced neural information processing systems conference, December 2005, Vancouver, Canada, pp. 1441–1448
- Wang L, Beeson D, Wiggs G, Rayasam M (2006) A comparison of metamodeling methods using practical industry requirements. In: Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, Newport, RI, May 2006
- Wujek BA, Renaud JE (1998a) New adaptive move-limit management strategy for approximate optimization, Part 1. *AIAA J* 36(10):1911–1921
- Wujek BA, Renaud JE (1998b) New adaptive move-limit management strategy for approximate optimization, Part 2. *AIAA J* 36(10):1922–1934
- Zerpa L, Queipo NV, Pintos S, Salager J (2005) An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. *J Pet Sci Eng* 47:197–208