

# High Dimensional Model Representation for solving Expensive Multi-objective Optimization Problems

Proteek Chandan Roy

Department of Computer Science and Engineering  
Michigan State University  
1220 Trowbridge Rd, East Lansing  
Michigan-48824, USA  
Email: roypote@msu.edu  
<http://www.coin-laboratory.com/>

Kalyanmoy Deb

Department of Electrical and Computer Engineering  
Michigan State University  
1220 Trowbridge Rd, East Lansing  
Michigan-48824, USA  
Email: kdeb@egr.msu.edu  
<http://www.egr.msu.edu/~kdeb/>

**Abstract**—Metamodel based evolutionary algorithms have been used for solving expensive single and multi-objective optimization problems where evaluation of functions consume major portion of the running time. The system can be complex, high dimensional, multi-objective and black box function. In this paper, we have proposed a framework for solving expensive multi-objective optimization problems that uses high dimensional model representation (HDMR) as a basic model. The proposed method first explores the region of interest and then exploits them by narrowing the search space. It uses Kriging to interpolate subcomponents of HDMR and NSGA-II to solve the model space. It is compared with basic NSGA-II and multi-objective Kriging method on ZDT, DTLZ and CEC09 test problem suits. The results show that this framework is able to find a good distribution of solutions which are sufficiently converged to Pareto optimal fronts with limited number of solution evaluations.

## I. INTRODUCTION

Evolutionary algorithms have been used extensively in solving multi-objective optimization problems. Calculation of objective functions can be carried out by an exact functional form, computational simulation or physical experiment. Due to high computational cost of simulation or time consuming nature of experiments, evolutionary algorithms are combined with surrogate or computationally cheap models to reduce total time of evaluations. For example, surrogate models have been applied to aerodynamic design [1] or drug design problems [2] to reduce the overall time for optimization.

In literature, different surrogate models are proposed to solve single and multi-objective optimization problems (MOPs). The popular surrogate models include response surface model (RSF) [3], radial basis function (RBF) [4], multivariate adaptive regression splines (MARS) [5], multi-layer perceptron (MLP) [6], Kriging models [7] etc. Jones *et al* [8] proposed a Kriging based method called efficient global optimization (EGO) for single objective problems. Kriging predicts the objective values along with uncertainty measures. It is then used to measure expected improvement (EI), that computes how much the solution is supposed to improve. Emmerich *et al.* introduced Kriging in multi-objective problems with expected improvement (EI) and expected hypervolume improvement (EHVI) as a selection criteria [7]. The solutions

having maximum EI or EHVI are selected for next generation and evaluated by real fitness function or high-fidelity evaluation. ParEGO [9] solves multi-objective problem by converting them into single objective with parameterized scalarizing weight vector.  $\mathcal{S}$ -metric selection based EGO (SMS-EGO) [10] uses a hypervolume based metric or  $\mathcal{S}$ -metric to select an individual for exact solution evaluation. They have used covariance matrix adaptation evolution strategy (CMA-ES) for solving the problems. Decomposition based multi-objective evolutionary algorithm (MOEA/D) [11] has been combined with EGO to evaluate multiple solutions in parallel. EGO method has also been combined with NSGA-II [12] to solve problems having more than 2 objectives [13]. However, all the Kriging based methods have difficulty in solving problems with more than 12 variables. Due to the curse of dimensionality, the number of training samples needed for modeling objective function becomes very high as the number of input dimension grows. Therefore, we introduce the model called high dimensional model representation that has some advantages in complex and high dimensional problems. Other surrogate based evolutionary algorithms can be found in [14].

High dimensional model representation (HDMR) has been introduced by Sobol [15] and Rabitz [16]. This mathematical framework is known for capturing complex input-output relationship with a set of orthogonal basis [17]. Several model fitting approaches e.g. Kriging [18], radial basis function (RBF) [19], moving least square [17] etc have been used to model the components of HDMR. In [19], RBF is used to predict the HDMR components by generating samples. After building the model with the components, an optimization algorithm finds the best predicted solutions or Pareto front in model space. A more comprehensive survey of HDMR based metamodels are given in [20]. Recently HDMR has been combined with cooperative co-evolution based algorithm for large scale problems [21] where HDMR is used to extract separable and non-separable subcomponents. In these cases, the optimization is done based on the constructed model without any explicit use of evolutionary control. This paper introduces HDMR model based evolutionary control (HDMR-EC) for solving expensive, complex and high dimensional

multi-objective problems.

Model management is an important part when it comes to expensive multi-objective problems. As the fitness landscape is not known to the algorithm beforehand, it is very common to over-fit or under-fit the landscape. Over-fitting might lead to false optima introduced by surrogates whereas under-fitting might overlook some optimal regions. Within limited amount of function evaluations, the algorithm should also focus on the part of search space that are more probable to optimize the objectives. Different methods for combining high-fidelity evaluations with the model is discussed in [14]. The main contribution of this paper is to combine an evolution control or model management framework to HDMR for solving expensive MOPs.

The next sections are organized as follows. Section II describes the mathematical framework of HDMR. Section III introduces our proposed method HDMR-EC. Experimental settings and results for 34 test instances are shown in Section IV. Section V analyzes the results. Then we conclude the paper with overall remarks, shortcomings of the method and future work.

## II. BACKGROUND

In this section we briefly discuss the theory of high dimensional model representation framework.

### A. High Dimensional Model Representation

High dimensional model representation is a mathematical framework that approximates a system response without considering the complicated underlying principles. It captures non-linear system outputs as a summation of variable correlations in a hierarchical order. In a full model expansion, it considers all possible variable interactions and their contribution to the original function. The first term describes the average value of the fitness landscape near a reference point called cut-center. Second order terms express the independent effects of each variable if decision variables are deviated from the cut-center. Higher order terms denote the residual group effect of variables. Therefore, the terms are independent or orthogonal to each other to form a mean convergent series. The function  $f(\mathbf{x})$  can be written as a series expansion in the following manner.

$$f(\mathbf{x}) = g_0 + \sum_{i=1}^n g_i(x_i) + \sum_{1 \leq i < j \leq n} g_{ij}(x_i, x_j) + \dots + g_{12\dots n}(x_1, x_2, \dots, x_n) \quad (1)$$

Here  $g_0$  is the mean response in the neighborhood of  $\mathbf{x}$  in decision variable space and it is constant.  $g_i(x_i)$  gives the independent effect of variable  $x_i$  on the output  $f(\mathbf{x})$  when acting alone.  $g_{ij}(x_i, x_j)$  contributes with pair-wise interaction of variables  $x_i$  and  $x_j$ . Note that, this term doesn't contain any effect of previous terms, i.e. constant effect and effect of single variables. This series approaches in a hierarchical manner up to interaction of all variables  $g_{12\dots n}(x_1, x_2, \dots, x_n)$ . This term gives the residual correlated behavior over all decision variables. In order to compute the component functions, we

need a reference point  $\bar{\mathbf{x}}$  as a cut-center and a set of sample points in the neighborhood of  $\bar{\mathbf{x}}$  to find all the effects. The terms will later be used to predict function values of unknown points. First three terms can be computed by the following equations.

$$\begin{aligned} g_0 &= g(\bar{\mathbf{x}}) = f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \\ g_i(x_i) &= f(\bar{x}_1, \bar{x}_2, \dots, x_i, \bar{x}_{i+1}, \dots, \bar{x}_n) - g_0 \\ g_{ij}(x_i, x_j) &= f(\bar{x}_1, \dots, \bar{x}_{i-1}, x_i, \bar{x}_{i+1}, \dots, x_j, \bar{x}_{j+1}, \dots, \bar{x}_n) - g_i(x_i) - g_j(x_j) - g_0 \end{aligned} \quad (2)$$

$f(\bar{x}_1, \bar{x}_2, \dots, x_i, \bar{x}_{i+1}, \dots, \bar{x}_n)$  denotes the value of the function  $f$  at the point where all the decision variables are at reference point  $\bar{\mathbf{x}}$  except  $\bar{x}_i$  is replaced by  $x_i'$  which is a close value of  $\bar{x}_i$ . By subtracting the lower order terms we can identify the unique effect of the corresponding variable interaction. First order approximation is given by the equation below.

$$f(\mathbf{x}) = g_0 + \sum_{1 \leq i \leq n} g_i(x_i) + \mathcal{R} \quad (3)$$

Here  $\mathcal{R}$  is the error term produced by the truncation of the series up to second term. Given reference point  $\bar{\mathbf{x}}$ , regularly spaced points are sampled in decision space coordinates. Effect of an arbitrary point  $\mathbf{x}'$  can be found with the help of all sampled effects and an interpolation method. Function value is predicted by the component ( $g$ ) functions. In this study we have used Kriging to interpolate  $g$  functions (Fig. 1).

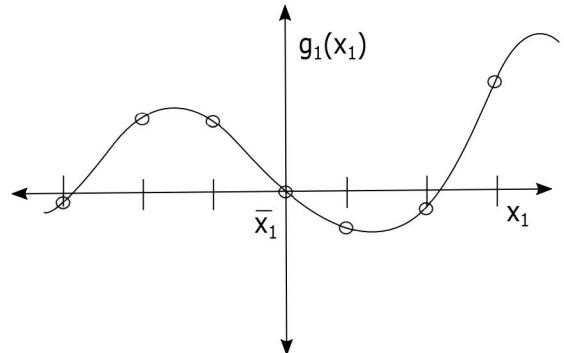


Fig. 1: Interpolating linear effects of regularly spaced sample points for variable  $x_1$ .  $\bar{x}_1$  is the first coordinate value of  $n$ -dimensional reference point  $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ . Six points are sampled around  $x_1$  coordinate and their effects are interpolated with Kriging.

## III. THE PROPOSED ALGORITHM

In the previous section, we have discussed the process of building HDMR model with an interpolation method and sampling. Here we propose an optimization scheme for expensive multi-objective problems.

Many surrogate based methods convert a multi-objective problem into a single objective one as discussed previously. Here we are proposing a method, namely HDMR-EC, that will retain the multi-objective nature of the problem and use a

multi-objective problem solver to solve the constructed model. Since the method is supposed to be applied for expensive problems, we use only first two terms (Eq. 3) of HDMR to reduce number of samples needed to construct the model. We will use Kriging to model each first order  $g$  functions. Uncertainty measure of the Kriging model is simply added (assuming minimization problem) to the predicted function value to discourage over-fitting. It means that, an optimization algorithm will not over-emphasize some uncertain regions.

There are two main parts of the proposed scheme— 1) Initial Exploration and 2) Exploitation. Each of them consists of more than one sub-steps. The steps are discussed below. Here we assume that the optimization problem has  $n$  decision variables,  $M$  objectives and  $FE_{max}$  maximum number of solution evaluations.

*1) Initial Exploration Step:* In this step the proposed algorithm builds HDMR model using full domain of each input variable. Initial model is important to guide the search towards optimal region. This step consists of sampling the total space, building the model and optimizing based on the model.

(a) **Sampling:** The algorithm takes a cut-center or reference point whose coordinates are middle values of each decision variable domain. For example, if  $\mathbf{x} = (x_1, x_2) \in [0, 1]$ , then the center point will be  $(0.5, 0.5)$ . Now we take samples by changing the value of each variable  $x_i$  with the help of an interval vector  $\mathbf{d}$ . The change is done from lower bound to upper bound of  $x_i$  while other variables remain the same. For example, if we take 5 samples for  $x_1$  variable, the samples would be  $\{(0.0, 0.5), (0.25, 0.5), (0.5, 0.5), (0.75, 0.5), (1.0, 0.5)\}$ . Here the difference between samples in  $x_1$  is  $d_1 = 0.25$ . For  $s$  samples, the interval vector  $\mathbf{d} = (\mathbf{ub} - \mathbf{lb})/(s - 1)$  where  $\mathbf{ub}$  and  $\mathbf{lb}$  are the upper bound and lower bound of decision variables. The exploration parameter  $e$  signifies the proportion of initial exploration.  $e = 0.5$  denotes that 50% of total solution evaluations will be considered for exploration. First, we fix the number of samples for exploration, then the number of samples per variable  $s$  can be derived.

(b) **Model Building:** In this step, we calculate the  $g$  functions for each objective. Note that, simple arithmetic is sufficient to get the  $g$  function values from objectives. We build Kriging model over those functions. As we have discussed, variables are sampled by changing only one variable at a time. So there will be  $n$  models each of them consists of only one variable. We can use other interpolation techniques e.g. spline, radial basis function network or moving least squares as a model.

(c) **Optimization:** Using a multi-objective evolutionary algorithm (MOEA) we can find a Pareto front predicted by the model. For our experiment, we have chosen NSGA-II algorithm for fair comparison among three different methodologies. NSGA-II it is quite successful in two objective optimization problems [22]. However, one can choose other sophisticated algorithms to improve the performance. Here we assume that other components

of optimization process take negligible time compare to fitness evaluations. Standard parameter settings can be used for optimization algorithm.

At the end of this step, we would identify the regions that create predicted Pareto front. Initial model might not provide us the optimal points. But it is more probable if we narrow down the search space. As we will be dealing with the expensive problems, number of solution evaluations will be limited. The number of search regions can be controlled by the number of clusters discussed in the next step.

*2) Exploitation Step:* In this step, the algorithm continues to focus on the search regions to facilitate further exploitation.

- (a) **Model validation:** At the end of first step, we get a predicted Pareto front. Our goal is to find a well-distributed set of solutions. For this purpose, the algorithm takes  $K$  points by clustering those solutions. It uses fuzzy  $K$ -means clustering method in the variable space. It starts with  $K$  number of points placed randomly on the search space and assign points to each cluster with random weights or degree of belongings. Then we compute the centroids of the clusters and update degree of belongings until convergence or the stopping criteria is met. We estimate fitness values of cluster centers by expensive evaluations. It is possible that some of them are dominated by the others. Dominated points will not be considered for further exploitation. It is denoted as model validation. Therefore, it is possible that the algorithm exploits less than  $K$  regions in each iteration. The reason for model validation is to avoid false optima.
- (b) **Sampling, Model Building and Optimization:** Valid (predicted optimal) points from the previous step are used as cut-centers of HDMR and sampled again with reduced interval of  $\mathbf{d}$  (see Fig. 2). This reduction is called learning rate  $l$ . This parameter denotes proportion of change in intervals and it is given by the user.  $l = 0.2$  indicates that upper and lower bound are shrunk by 2% from previous bounds. Therefore,  $\mathbf{d} \leftarrow 0.8 \times \mathbf{d}$  for  $l = 0.2$ . We build the models for each objective and run MOEA to find a combined predicted Pareto front for all regions. We follow step (a) and (b) until it exceeds maximum number of evaluations  $FE_{max}$ .
- (c) **Output:** To provide a well distributed Pareto front to the user, the algorithms can evaluate few more points from the last predicted front. We can fix the number of evaluations to be done at the end. These solutions are also chosen by clustering method in order to find a diverse solution set.

Some properties of this approach are described below.

- The metamodeling part is very efficient in HDMR-EC. Building several Kriging models with only one variable is less time consuming compare to building the model with several variables. For that reason, model parameters can also be optimized very efficiently. We have used fuzzy  $K$ -means clustering method that also converges fast. Space complexity increases linearly with the number

---

**Algorithm 1:** HDMR-EC

---

**Data:** Number of variables  $n$  and objectives  $M$ , Total solution evaluation  $FE_{max}$ , Exploration parameter  $e$ , Number of clusters  $K$ , learning rate  $l$ , upper and lower bound of variables  $\mathbf{ub}$  and  $\mathbf{lb}$ , a multi-objective evolutionary algorithm A

**Result:** Solution set  $P$

```

1  $s_1 = FE_{max} \times e;$ // Initial population size
2  $s_2 = \lfloor \frac{s_1}{n} \rfloor;$ // samples per variable
3  $\mathbf{c} \leftarrow (\mathbf{ub} - \mathbf{lb})/2;$ // cut-center
4 Evaluate  $\mathbf{c}$  and calculate mean response  $g_0^j$  at  $\mathbf{c}$  for each objective  $j \in \{1, 2, \dots, M\}$ ;
5  $\mathbf{d} \leftarrow (\mathbf{ub} - \mathbf{lb})/(s_2 - 1);$ // calculate  $\mathbf{d}$ 
6 for  $i = 1$  to  $n$  do // for each variable
7    $P \leftarrow P \cup \{\text{Sample around } x_i \text{ with interval } \mathbf{d}\};$ 
8   Calculate subcomponent  $g_i^j(x_i)$  for each objective  $j \in \{1, 2, \dots, M\}$ ;
9   Build surrogate models with  $g_i^j$ ;
10 end
11  $F = |P|;$ // total solution evaluation
12  $Q_1 \leftarrow \text{Predict Pareto front by A};$ 
13  $Q_1 \leftarrow \text{Find } K \text{ cluster centers from } Q_1, \text{ evaluate them and exclude dominated solutions};$ 
14  $P \leftarrow P \cup Q_1;$ 
15 while  $F < FE_{max}$  do
16    $Q_1 \leftarrow \emptyset;$ // predicted Pareto front
17    $\mathbf{d} \leftarrow (1 - l) \times \mathbf{d};$ // update  $\mathbf{d}$ 
18    $s_3 \leftarrow \lfloor \frac{FE_{max} - |F|}{n \times K} \rfloor;$ // samples per variable per cluster
19   if  $s_3 > 1$  then // more than one sample per variable
20     for  $k = 1$  to  $K$  do // for each cluster
21       for  $i = 1$  to  $n$  do // for each variable
22          $F \leftarrow F + s_3;$ // total evaluations
23         Sample around  $x_i$  with interval  $\mathbf{d}$ ;
24         Calculate subcomponent
25          $g_i^j(x_i) \quad \forall j \in \{1, 2, \dots, M\}$ ;
26         Build surrogate models with  $g_i^j$ ;
27     end
28      $Q_2 \leftarrow \text{Predict Pareto front by A};$ 
29      $Q_1 \leftarrow Q_1 \cup Q_2;$ 
30   end
31    $Q_1 \leftarrow \text{Find Non-dominated front from } Q_1;$ 
32    $Q_1 \leftarrow \text{Find } K \text{ cluster centers from } Q_1, \text{ evaluate them and exclude dominated solutions};$ 
33    $P \leftarrow P \cup Q_1;$ 
34 else
35    $s_4 \leftarrow FE_{max} - |F|;$ 
36    $F \leftarrow F + s_4;$ 
37   if  $s_4 > 0 \& Q_1 \neq \emptyset$  then
38      $Q_1 \leftarrow \text{Find } s_4 \text{ cluster centers from } Q_1 \text{ and evaluate};$ 
39      $P \leftarrow P \cup Q_1;$ 
40   end
41 end
42  $P \leftarrow \text{Find Non-dominated front from } P;$ 
43 return  $P$ ;

```

---

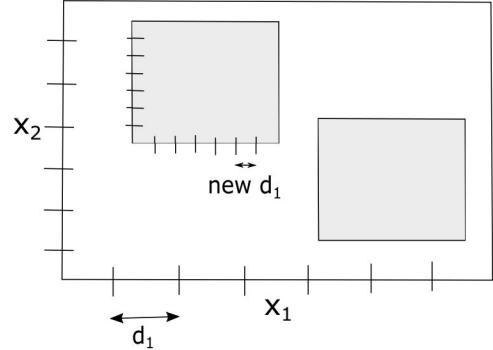


Fig. 2: After exploration step, two good regions are identified ( $K = 2$ ). The search domain of those regions are narrowed.

of input dimensions because of linear increase of model parameters.

- First order HDMR method can predict both separable and non-separable functions. The nature of interpolated  $g$  functions (Fig. 1) can be non-linear and sum of their effects might be highly non-linear. Thus it can handle complex functions. We are relying on an optimization algorithm to predict Pareto optimal front in a highly non-linear multi-dimensional model space thus reducing the computational cost of evaluating real fitness functions.
- If we fix the order of HDMR, number of samples needed to represent a function increases polynomially with the number of input dimensions [17]. But the number of terms in HDMR is exponential to the number of input variables thus limits the efficacy of this model. In practice, higher order variable interactions are rare [20].

#### IV. EXPERIMENTAL RESULTS

The proposed framework is compared with basic NSGA-II [12] and Kriging based NSGA-II algorithm [13] in 17 test problems known as ZDT{1,2,3,4,6} [23], DTLZ{1,2,3,4,7} [24], and CEC09{1,2,3,4,5,6,7} [25] with 15 and 30 variables and two objectives. Here Kriging based NSGA-II, namely M-KRIG is used because it models each function separately (no scalarization) and uses NSGA-II algorithm to optimize the model space similar to the proposed method. It is also necessary to see how a basic algorithm i.e. NSGA-II performs on the same problem set. Further analysis and comparison with other methods is left as future work.

Maximum  $FE_{max} = 500$  number of solution evaluations are allowed for comparing each algorithms. Exploration parameter is taken  $e = 0.3$  for all test instances. It denotes that at most  $0.3 \times 500 = 150$  solution evaluations are allowed for initial explorations. This number of kept same for initialization of algorithm M-KRIG. Number of samples for exploitation is kept  $s = 5$ . Number of clusters is kept  $K = 2$ . For fuzzy  $K$ -means clustering, we have used MATLAB's built-in function  $fcm$  with stopping criteria of 100 iterations. The learning parameter ( $l$ ) is kept 0.2. This parameter is chosen in adhoc manner. For final output, 10 samples are evaluated

from the last predicted front for both HDMR-EC and M-KRIG algorithm.

Parameter settings of Kriging based EGO (M-KRIG) is taken from [13]. Estimation (EST), which is predicted by the Kriging model, is used as the selection criteria of NSGA-II algorithm. To reduce the running time, initially 150 solutions are generated by latin hypercube model and one solution is picked up in each generation by fuzzy K-means clustering method with  $K = 1$  from predicted Pareto front. Model parameters are optimized while building the model. Source code of Kriging is taken from [26]. For basic NSGA-II algorithm, we use population size 25 with 20 generations. The population size is kept limited to maximize the evolution effect. Simulated Binary Crossover and polynomial mutation [22] are used with probabilities 0.9 and (1/number of variables) respectively. Crossover index and mutation index are kept 10 and 20 respectively. Inverted generational distance (IGD) and hypervolume metric (HV) [22] is used for performance comparison. HV is calculated by taking the nadir (worst) point of all dataset in that test problem. Each test problem is run 10 times in Intel Xeon E5-2697 v3 processor with 64-bit Windows machine.

## V. DISCUSSION

Test problems considered here have different properties such as non-separable variables, discrete or disconnected Pareto front or complicated variable space. IGD metric of Fig. 3 shows that HDMR-EC performs better in most of the ZDT and DTLZ problems with 15 and 30 variables compare to multi-objective Kriging (M-KRIG) and NSGA-II. For ZDT4 problem, it has reached very near to actual Pareto front while other algorithm didn't converge. In some cases, e.g. DTLZ4 and CEC09 problems, NSGA-II is better than M-KRIG in IGD as well as hypervolume (Fig. 4). In some complicated problems e.g. UF5, HDMR-EC has larger standard deviation in hypervolume than comparing algorithms. There is no significant difference between 15 and 30 variables problem instances for HDMR-EC method except the standard deviation is increased on 30 variable problems. On the other hand, performance deteriorates for M-KRIG method when dimensionality of input variable increases. Excluding some cases (e.g. UF1 and UF2), HDMR-EC gives the highest hypervolume (Fig. 4) for most of the problems. M-KRIG performs worst in most of the DTLZ and CEC09 problems.

Fig. 5 shows the distribution of solutions of 10 runs. We have computed the Pareto set combining the data of all runs. It is clear from the figures that, M-KRIG couldn't converge to optimal front within limited evaluations. HDMR-EC finds part of the optimal front even in 30 variable case. As the number of cluster is  $K = 2$ , HDMR-EC finds only some parts of the disconnected Pareto optimal regions in ZDT3 and DTLZ7. For complicated UF{3,4,5,6,7} test problems, we should increase number of solution evaluations to make HDMR-EC converge to optimal front.

Fig. 6 shows the parameter sensitivity of the proposed method. For three different settings of  $K$  and  $e$ , the algorithm

is run for 10 times in ZDT and CEC09 problems. These are H1 ( $K = 1, e = 0.3$ ), H2 ( $K = 2, e = 0.4$ ), and H3 ( $K = 2, e = 0.5$ ). We see from the figure that, 1-cluster approach tends to perform better because it facilitates more solution evaluations for exploitation, while 2-cluster approach tries to improve IGD by having a good distribution. The parameter  $e$  has no significant effect on IGD metric. Further experiments should be done to get the actual effects of the parameters.

## VI. CONCLUSION

In this paper, we have successfully integrated high dimensional model representation (HDMR) with evolutionary algorithm providing a methodology for solving expensive multi-objective optimization problems. HDMR-EC gives a good prediction of a complex function with a reasonable number of training samples. We have applied our model to 34 test problem instances with maximum of 500 high-fidelity solution evaluations. As seen from the results, we need to increase number of evaluations in case of 30 variables and complicated problems. Cluster based technique is incorporated with HDMR-EC to find a good distribution. For many objective problems, reference point based approaches [27] can be better choice. Kriging based prediction gives us the uncertainty measure of subcomponents. Uncertainty measure reduces model over-fitting which is essential for avoiding false optima. However, for complicated problems, it might not be suitable to have only first order components. Number of subcomponents of the model increases exponentially with input variables. Higher order components are excluded to save solution evaluations which is a limitation of the approach. In the exploitation step, it is possible that the cluster centroids and their bounds overlap with one another. Exploration parameter and number of clusters can also be made adaptive and problem dependent. Other criteria such as expected improvement or expected hypervolume improvement can be used in the Kriging model. In future, we would like to extend this model to expensive single and multi-objective constrained optimization problems.

## ACKNOWLEDGMENT

This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

- [1] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *Computational Intelligence Magazine, IEEE*, vol. 4, no. 3, pp. 62–76, Aug 2009.
- [2] D. Douquet, "e-lea3d: a computational-aided drug design web server," *Nucleic Acids Research*, vol. 38, no. suppl 2, pp. W615–W621, 2010.
- [3] Y. Lian and M.-S. Liou, "Multiobjective optimization using coupled response surface model and evolutionary algorithm," *AIAA Journal*, vol. 43, no. 6, pp. 1316–1325, Jun. 2005.
- [4] H. Fang and M. H. Masoud Rais-Rohani, "Multiobjective crashworthiness optimization with radial basis functions," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, August 2004.

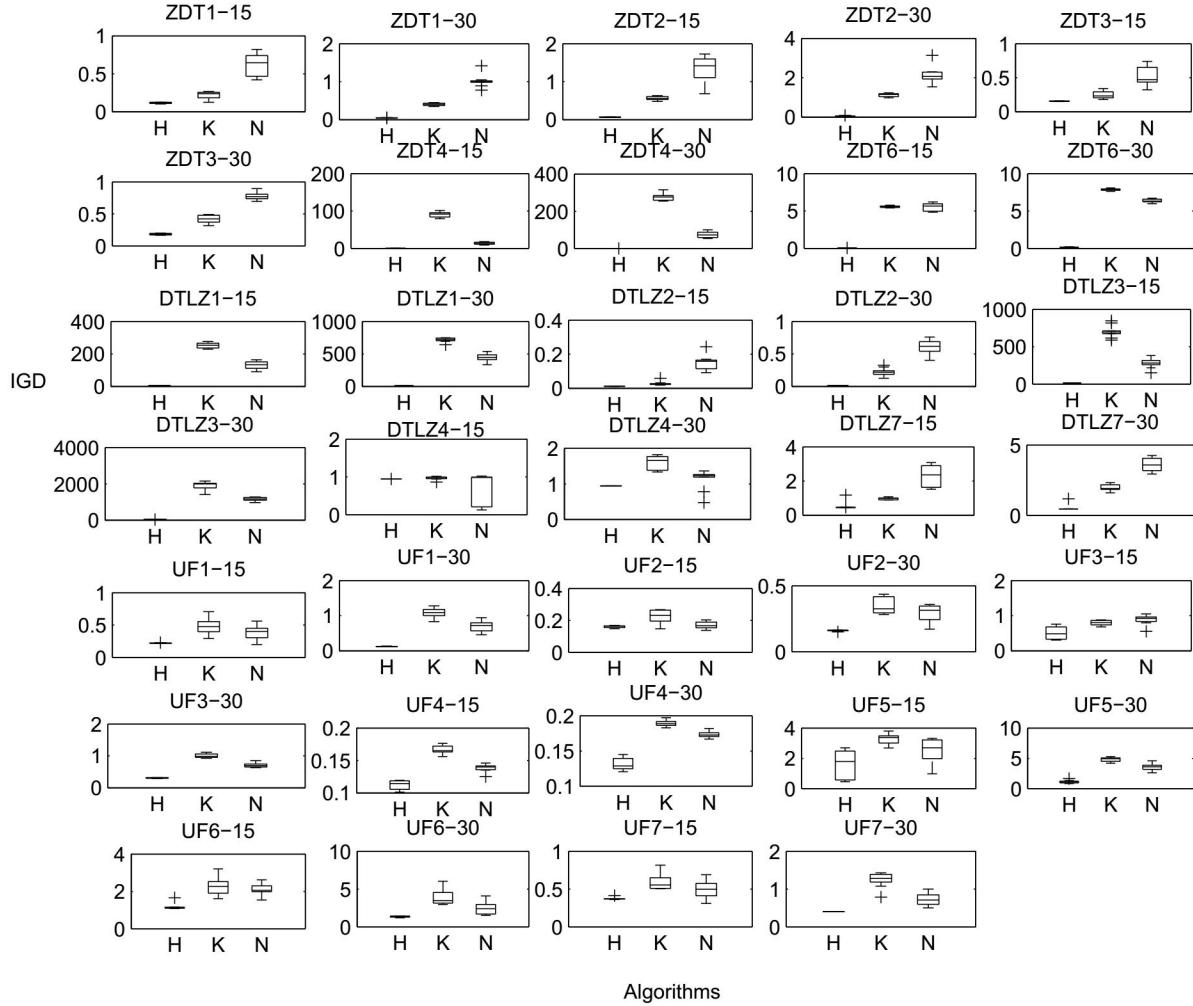


Fig. 3: IGD values of HDMR-EC (H), multi-objective Kriging (K) and NSGA-II (N) algorithm on different problems containing 15 and 30 variables.

- [5] J. H. Friedman, "Multivariate adaptive regression splines," *Annals of Statistics*, vol. 19, pp. 1–67, 1991.
- [6] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 5, pp. 481–494, Oct 2002.
- [7] M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 4, pp. 421–439, Aug 2006.
- [8] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec. 1998.
- [9] J. Knowles, "Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 1, pp. 50–66, Feb 2006.
- [10] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted  $S$ -metric selection," in *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature: PPSN X*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 784–794.
- [11] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by moea/d with gaussian process model," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 3, pp. 456–474, June 2010.
- [12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, Apr 2002.
- [13] C. Luo, K. Shimoyama, and S. Obayashi, "Kriging model based many-objective optimization with efficient calculation of expected hypervolume improvement," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 1187–1194.
- [14] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
- [15] I. M. Sobolá, "Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates," *Math. Comput. Simul.*, vol. 55, no. 1–3, pp. 271–280, Feb. 2001.
- [16] H. Rabitz and d. F. Alis, "General foundations of high-dimensional model representations," *Journal of Mathematical Chemistry*, vol. 25, pp. 197–233, 1999.
- [17] R. Chowdhury and S. Adhikari, "High dimensional model representation

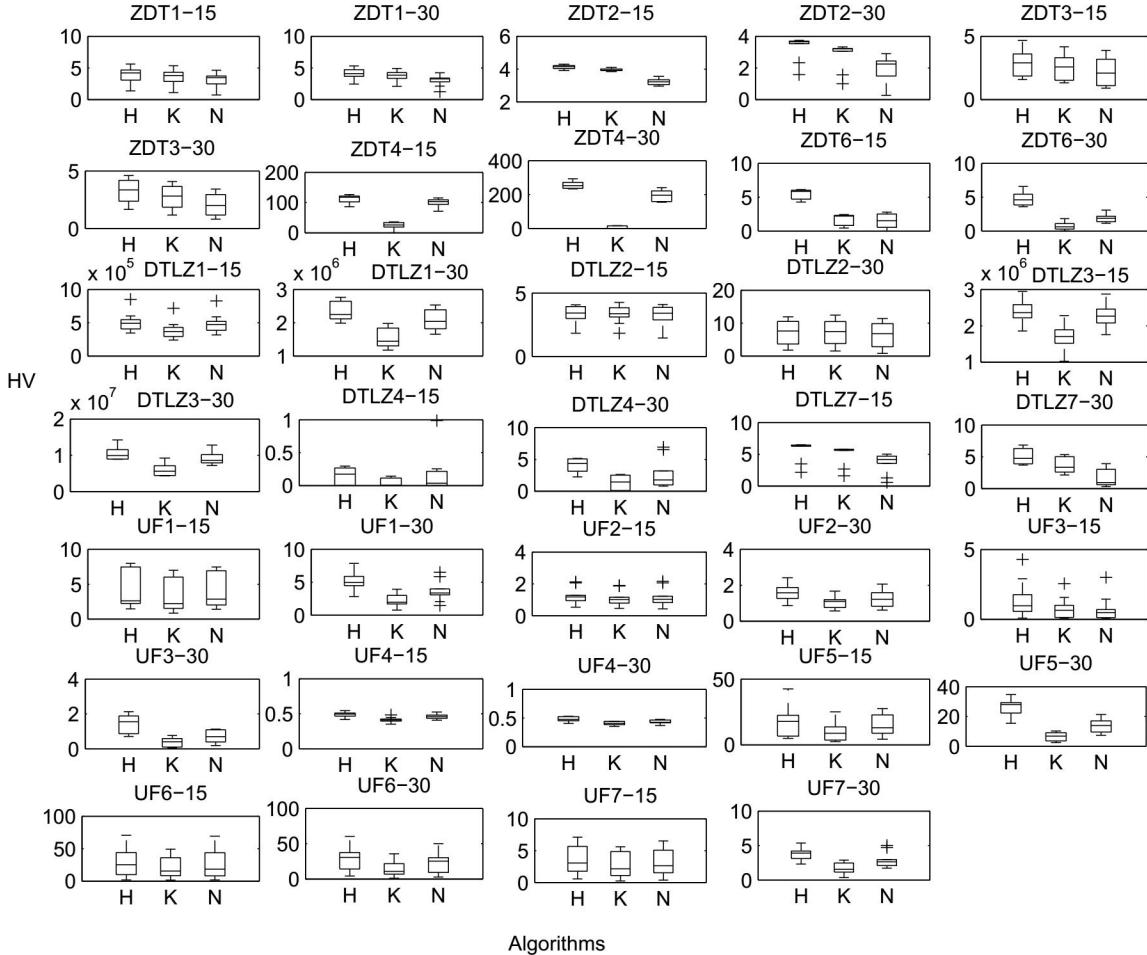


Fig. 4: Hypervolume (HV) values of HDMR-EC (H), multi-objective Kriging (K) and NSGA-II (N) algorithm on different problems containing 15 and 30 variables.

- for stochastic finite element analysis,” *Applied Mathematical Modelling*, vol. 34, no. 12, pp. 3917 – 3932, 2010.
- [18] S. Ullaganathan, I. Couckuyt, T. Dhaene, J. Degroote, and E. Laermans, “High dimensional kriging metamodeling utilising gradient information,” *Applied Mathematical Modelling*, 2015.
  - [19] S. Shan and G. G. Wang, “Metamodeling for high dimensional simulation-based design problems,” *Journal of Mechanical Design*, vol. 132, p. 051009, 2010.
  - [20] S. Shan and G. Wang, “Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions,” *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219–241, 2009.
  - [21] S. Mahdavi, M. Shiri, and S. Rahnamayan, “Cooperative co-evolution with a new decomposition method for large-scale optimization,” in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 1285–1292.
  - [22] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.
  - [23] E. Zitzler, K. Deb, and L. Thiele, “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
  - [24] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, vol. 1, May 2002, pp. 825–830.
  - [25] Q. Zhang, S. Z. A. Zhou, W. L. P. N. Suganthan, and S. Tiwari, “Multiobjective optimization test instances for the cec 2009 special session and competition,” University of Essex, Tech. Rep., apr 2009.
  - [26] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, “Aspects of the MATLAB toolbox DACE,” Technical University of Denmark, Copenhagen, Technical Report IMM-TR-2002-13, 2002.
  - [27] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints,” *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 4, pp. 577–601, Aug 2014.

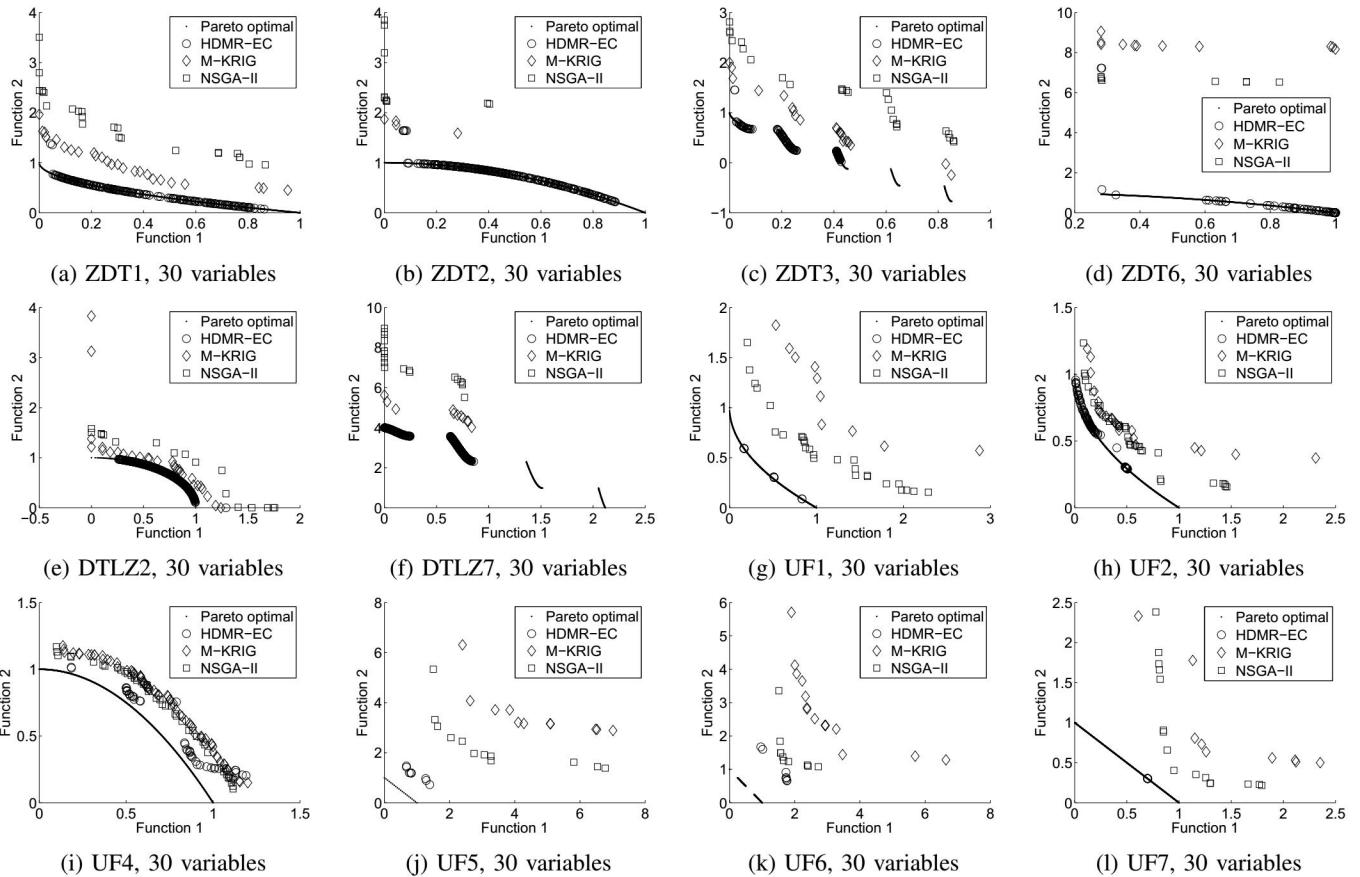


Fig. 5: Distribution of solutions on different test instances after 500 solution evaluations by HDMR-EC, multi-objective Kriging (M-KRIG) and NSGA-II algorithms.

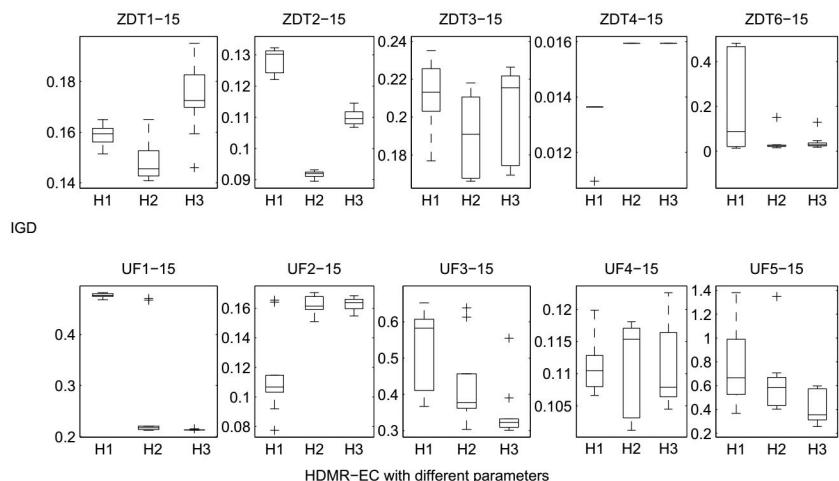


Fig. 6: IGD values of three different settings H1 ( $K = 1, e = 0.3$ ), H2 ( $K = 2, e = 0.4$ ), and H3 ( $K = 2, e = 0.5$ ) of HDMR-EC method is shown.