

# A Kriging Metamodel Assisted Multi-Objective Genetic Algorithm for Design Optimization

M. Li

Graduate Research Assistant

G. Li

Graduate Research Assistant

S. Azarm<sup>1</sup>

Professor

e-mail: azarm@umd.edu

Department of Mechanical Engineering,  
University of Maryland,  
College Park, MD 20742

*The high computational cost of population based optimization methods, such as multi-objective genetic algorithms (MOGAs), has been preventing applications of these methods to realistic engineering design problems. The main challenge is to devise methods that can significantly reduce the number of simulation (objective/constraint functions) calls. We present a new multi-objective design optimization approach in which the Kriging-based metamodeling is embedded within a MOGA. The proposed approach is called Kriging assisted MOGA, or K-MOGA. The key difference between K-MOGA and a conventional MOGA is that in K-MOGA some of the design points are evaluated on-line using Kriging metamodeling instead of the actual simulation model. The decision as to whether the simulation or its Kriging metamodel should be used for evaluating a design point is based on a simple and objective criterion. It is determined whether by using the objective/constraint functions' Kriging metamodels for a design point, its "domination status" in the current generation can be changed. Seven numerical and engineering examples with different degrees of difficulty are used to illustrate applicability of the proposed K-MOGA. The results show that on the average K-MOGA converges to the Pareto frontier with an approximately 50% fewer number of simulation calls compared to a conventional MOGA. [DOI: 10.1115/1.2829879]*

## 1 Introduction

A significant challenge in the applications of genetic algorithm (GA) based optimization methods to engineering design problems has been the high computational cost due to the large number of simulation (or function) calls required by these methods [1]. A common strategy to reduce the computational effort for such optimization methods when integrated with simulation models is to use metamodeling techniques. Researchers have been quite active in developing models and methods that improve the efficiency of the GAs in terms of the number of simulation calls [2–18]. Some of these approaches are based on fitness approximations in which neural network [3–5], response surface [6], Kriging [7], and radial basis function [8] methods are used for metamodeling. Others use fitness inheritance approaches [9,10] in which the fitness of an offspring is inherited from its parents. A comprehensive review of fitness approximation and metamodeling approaches can be found in Ref. [16] and Refs. [17–19], respectively.

The fitness approximation methods are of two types: off-line (nonadaptive) and on-line (adaptive). In off-line approaches, metamodels are developed separately and prior to the start of an optimization algorithm [4,6–8,20,21]. The shortcoming of the off-line methods is that it is difficult to obtain both a good fidelity metamodel over the entire design space and at the same time maintain a low number of simulation calls [18,20]. The on-line approaches use a combination of metamodeling with the simulation model during the optimization procedure while adaptively improving the metamodel [2,3,5,11–14]. Most of the on-line methods developed so far are focused on single-objective optimization. The research on how to embed metamodeling within multi-objective genetic algorithms (MOGAs) remains sparse [2,3,13].

In on-line approaches [2,3,11–13], in the initial stages of the GA, rough metamodels are constructed. These metamodels are then gradually improved as more simulation data become available. Most of this type of approaches utilize neural network, which is known to require a large number of simulation calls [18]. Another unresolved issue in the current adaptive methods is how to objectively decide when to switch to the metamodel instead of using the simulation during the optimization [11,12,16]. Usually, the switching between the actual simulation model and the corresponding metamodel is subjectively decided [13]. Moreover, the fidelity of the metamodel may vary significantly during the optimization process and this can cause oscillation [16].

In this paper, we have attempted to address all of the above mentioned shortcomings. More specifically, a new on-line Kriging metamodel assisted MOGA, hereafter called K-MOGA, is presented. We use an objective criterion to determine whether a simulation model or its Kriging metamodel replacement should be used to evaluate design points. The proposed criterion is developed based on the metamodel's predicted error, which can be easily obtained as a by-product from Kriging. In the proposed approach, the Kriging metamodels for objective and constraint functions are built and adaptively improved within a MOGA. The approach is general and requires no additional simulation calls prior to the start of the optimization procedure to build the Kriging metamodels. Our current results show that the proposed approach resolves the problem often reported in the literature, that is, the metamodel may be of low fidelity and that it may produce false optima [12,16].

In Sec. 2, the terminology and background are reviewed. Details of K-MOGA are presented in Sec. 3. Seven examples and corresponding results are given in Sec. 4 to illustrate the applicability of the proposed approach. Concluding remarks and some directions for future research are presented in Sec. 5.

## 2 Terminology

In this section, we present the terminology and background used throughout the paper.

<sup>1</sup>Corresponding author.

Contributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received May 30, 2006; final manuscript received July 16, 2007; published online February 4, 2008. Review conducted by Timothy W. Simpson. Paper presented at the ASME 2006 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2006), Philadelphia, PA, September 10–13, 2006.

**2.1 Multi-Objective Optimization.** A multi-objective optimization problem is given as follows:

$$\begin{aligned} \min_{\mathbf{x}} f_m(\mathbf{x}) \quad m = 1, \dots, M \\ \text{s.t. } g_j(\mathbf{x}) \leq 0 \quad j = 1, \dots, J \\ \mathbf{x}^{\text{lower}} \leq \mathbf{x} \leq \mathbf{x}^{\text{upper}} \end{aligned} \quad (1)$$

The objective  $f_m(\mathbf{x})$  is the  $m$ th element in the vector  $\mathbf{f} = (f_1, \dots, f_M)^T$ , where  $T$  refers to the transpose of the row vector.  $\mathbf{x} = (x_1, \dots, x_N)^T$  is the design variable vector.  $\mathbf{x}^{\text{lower}}$  and  $\mathbf{x}^{\text{upper}}$  are the lower and upper bounds on  $\mathbf{x}$ , respectively. The functions  $g_1, \dots, g_J$  are the inequality constraints. We assume that there are trade-offs among at least two of the  $M$  objective functions. As such, the optimization problem in Eq. (1) has more than one solution. These solutions are optimal in a Pareto sense, and the set of all such solutions forms the Pareto set or Pareto frontier [1]. In this paper, a simulation call refers to calculation of objective and constraint values together for a single design point.

In the context of MOGA, a point is said to be “nondominated” if no other point in the current generation (or set of points under consideration) is better than that point with respect to all objectives [1]. The set of all nondominated points in the current generation forms a nondominated set. The remaining points in the current generation form a “dominated” set. The “domination status” of a point determines whether a point is dominated or nondominated. In MOGA, the set of nondominated points eventually evolves to form the Pareto frontier (or an estimate of it) when MOGA converges.

**2.2 Kriging Metamodeling.** For completeness, a brief description of the Kriging metamodeling is given in this section [18,19]. Kriging [18,22] has been widely used in recent years for metamodeling of computationally expensive deterministic simulations [21,23,24]. Kriging predicts the response of unobserved points (i.e., those whose response has not been obtained by the simulation) based on all of the observed points (i.e., those whose response has already been obtained). The Kriging method used in this paper is also called ordinary Kriging in the literature [18,22] and it is often used for predicting the simulation’s response values at discrete input locations (or design points), which is the situation for the proposed approach. The reason we have used the Kriging approach is that the predicted error of an estimated response value can be easily obtained as a by-product with the Kriging metamodel. However, the Kriging metamodeling needs to perform matrix inversions for predicting the response, which can increase the computational expense when the dimension of the problem is high [16].

In Kriging, a one-dimensional response value from a simulation is globally estimated by a known polynomial and a random departure from the polynomial:

$$y(\mathbf{x}) = p(\mathbf{x}) + Z(\mathbf{x}) \quad (2)$$

where  $y(\mathbf{x})$  is the unknown response of interest,  $p(\mathbf{x})$  is a known polynomial, and  $Z(\mathbf{x})$  is assumed as a realization of a Gaussian random process with a mean of zero, variance of  $\sigma^2$ , and a non-zero covariance [18,22]. The  $p(\mathbf{x})$  term provides a “global” approximation of the design space (the value of  $p(\mathbf{x})$  is given in Eq. (6)); the  $Z(\mathbf{x})$  term creates a “localized” deviation so that the Kriging metamodel interpolates with respect to the  $n_o$  observed points. The covariance matrix of  $Z(\mathbf{x})$  is given by

$$\text{cov}[Z(\mathbf{x}^i), Z(\mathbf{x}^j)] = \sigma^2 \mathbf{R} \quad \mathbf{R} = [R(\mathbf{x}^i, \mathbf{x}^j)] \quad (3)$$

where  $\mathbf{R}$  is an  $n_o \times n_o$  symmetric correlation matrix with ones along diagonal, and  $R(\mathbf{x}^i, \mathbf{x}^j)$ , which is the correlation function between any two observed points  $\mathbf{x}^i$  and  $\mathbf{x}^j$  for off-diagonal elements. The correlation function  $R(\mathbf{x}^i, \mathbf{x}^j)$  used in this paper is

$$R(\mathbf{x}^i, \mathbf{x}^j) = \exp \left[ - \sum_{n=1}^N \theta_n |x_n^i - x_n^j|^2 \right] \quad (4)$$

where  $\theta_n$  is an unknown correlation parameter. The quantities  $x_n^i$  and  $x_n^j$  are the  $n$ th components of the observed points  $\mathbf{x}^i$  and  $\mathbf{x}^j$ , respectively. The estimate,  $\hat{y}(\mathbf{x}^*)$ , of the response  $y(\mathbf{x}^*)$  at an unobserved point  $\mathbf{x}^*$  is given by

$$\hat{y} = \hat{\beta} + \mathbf{r}^T(\mathbf{x}^*) \mathbf{R}^{-1}(\mathbf{y} - \mathbf{p}\hat{\beta}) \quad (5)$$

where  $\mathbf{y}$  is the column vector of length  $n_o$ , which contains the values of the response at each observed point, and  $\mathbf{p}$  is a column vector with  $n_o$  components, which is filled with ones when  $p(\mathbf{x})$  is constant. In Eq. (5),

$$\hat{\beta} = (\mathbf{p}^T \mathbf{R}^{-1} \mathbf{p})^{-1} \mathbf{p}^T \mathbf{R}^{-1} \mathbf{y}$$

$$\mathbf{r}^T(\mathbf{x}^*) = [R(\mathbf{x}^*, \mathbf{x}^1), R(\mathbf{x}^*, \mathbf{x}^2), \dots, R(\mathbf{x}^*, \mathbf{x}^{n_o})]$$

The estimate of variance  $\hat{\sigma}^2$  for Eq. (2) is given by

$$\hat{\sigma}^2 = (\mathbf{y} - \mathbf{p}\hat{\beta})^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{p}\hat{\beta}) / n_o \quad (7)$$

The mean squared error (MSE)  $s^2$  for an unobserved point  $\mathbf{x}^*$  using this Kriging metamodel predictor is given by

$$s^2(\mathbf{x}^*) = \hat{\sigma}^2 \left( 1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{(1 - \mathbf{p}^T \mathbf{R}^{-1} \mathbf{r})^2}{\mathbf{p}^T \mathbf{R}^{-1} \mathbf{p}} \right) \quad (8)$$

Statistically, the root mean squared error (RMSE) or the standard deviation  $s(\mathbf{x}^*)$  represents the predicted deviation of the Kriging metamodel from the actual response. In this paper, it is assumed that the predicted deviation from the Kriging metamodel has a conditional normal distribution with a mean that is equal to the prediction and variance equal to the Kriging variance. This normally distributed standard deviation  $s$  will be used in the paper to decide on a prediction interval (e.g., one standard deviation  $s$  is used for objective functions, see Sec. 3.2; it could be  $2s$  or  $4s$ , or  $6s$  depending on the confidence requirement). However, even if this normal distribution assumption does not hold, it is often possible to find a transformation that makes the random process approximately normal [25].

The maximum likelihood estimate of  $\theta_n$  in Eq. (4) can be obtained by maximizing the following expression:

$$\max_{\theta > 0} \frac{-[n_o \ln(\hat{\sigma}^2) + \ln(\mathbf{R})]}{2} \quad (9)$$

Some new schemes that are used to update Kriging metamodel parameters  $\theta_n$  have been reported in the literature, e.g., Ref. [26].

### 3 Kriging Metamodel Assisted Multiobjective Genetic Algorithm

The basic idea behind K-MOGA is to ensure that in each generation the nondominated design points obtained using the Kriging metamodels (for objective and constraint functions) remain the same as the one obtained using the simulation. In this regard, the RMSE (recall Eq. (8)) is used as the main component in a criterion that ensures a nondominated set is not changed whether the simulation model or alternatively the Kriging metamodel is used for evaluating a design point.

For simplicity, we first describe the K-MOGA approach mainly for handling objective functions. After that, we briefly describe how to handle constraint functions in the approach. For completeness, we begin by presenting an overview of a conventional MOGA.

**3.1 Conventional Multiobjective Genetic Algorithm.** The conventional MOGA used here is based on NSGA [1] combined with an elitism strategy. As shown in the “MOGA” block in Fig. 1 (left dashed block), the conventional MOGA begins with a current

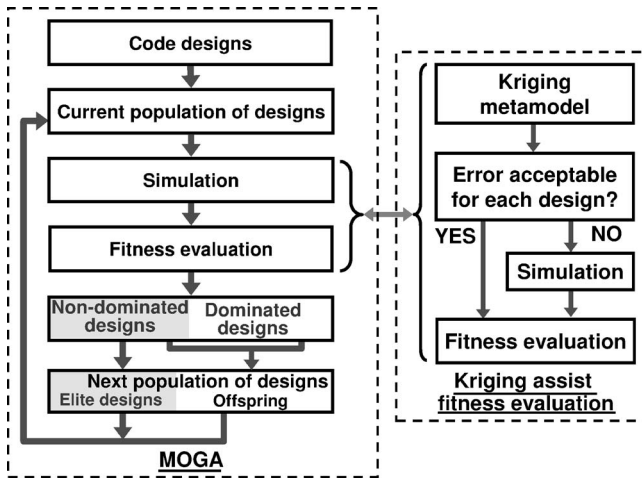


Fig. 1 Flowchart of MOGA (left dashed block) and proposed addition (right dashed block) of K-MOGA

population of individuals (or design points) and is composed of two parts (see the “Next population” in Fig. 1): elite and offspring points. The elite points are nondominated points that are directly inherited from the previous generation. In this paper, a strategy similar to NSGA-II [1] has been used to ensure that the number of nondominated points is not more than a prespecified percentage (e.g., 70% of the population) of the population. The remaining points are offspring design points that are produced by genetic operations, such as crossover and mutation. Such a strategy ensures that a prespecified percentage (e.g., 30%) of individuals in the population is generated by genetic operations. For offspring design points, we use a probability of 0.95 for crossover and a probability of 0.05 for mutation. For constraints, a previously reported constraint handling approach [27,28] has been used. Essentially, the constraint handling method is based on a penalty function, which takes into account both the amount and the number of violated constraints. Moreover, using this method, the feasible solutions always outrank infeasible ones.

In the conventional MOGA (in this paper, the improved NSGA, i.e., NSGA [1] together with an elitism strategy), the response values of the points in the initial population are calculated by a simulation model (i.e., for objective functions). Our conventional MOGA is different from NSGA-II with respect to the elitism strategy. NSGA-II requires more computational effort since it combines the offspring population with the parent population and then nondominated sorting is used to classify the entire population. In the MOGA used here, only nondominated points (i.e., those eventually converge to Pareto points) are migrated to the next generation.

### 3.2 Kriging Assisted Multi-Objective Genetic Algorithm.

As mentioned before, the response value from the Kriging metamodel has a predicted error. Using this predicted error, as long as it is determined that the domination status of design points in the current generation is not changed because of using the Kriging metamodel, it is acceptable to use the Kriging metamodel instead of simulation. If the domination status is changed, then the design points that are predicted to contribute to this change are observed (i.e., their objective function values are computed using the simulation model); otherwise, the metamodel is used to obtain the response values.

We use a quantitative measure of domination as part of a criterion to determine whether the predicted value from the Kriging metamodel should be accepted. This measure is called minimum of minimum distance (MMD), as described next.

*Minimum of Minimum Distance.* In any generation, except the initial population where all individuals are observed, the Kriging

metamodel can be used to obtain the predicted response of individuals. Based on these predicted response values, the domination status of individuals can be determined. To do this, the current population is partitioned into two sets: dominated and nondominated sets. Note that this partitioning is based on the Kriging metamodel values, that is, no simulation calls are used at this stage.

MMD is defined, in the objective space, as the minimum distance between all pairs of nondominated  $x_{nd}$  and dominated  $x_d$  points and calculated as follows. First, divide individuals in the current population into two sets: nondominated and dominated. Then, compute MMD by

$$\text{MMD} = \min\{\|f(x_{nd}) - f(x_d)\|_2\}$$

$$x_{nd} \in \{\text{nondominated set}\}$$

$$x_d \in \{\text{dominated set}\}$$

where the norm is defined in the  $f$  space:  $(f_1, f_2, \dots, f_M)$ . MMD is then projected along each objective function axis to obtain  $\text{MMD}f_m$ ,  $m=1, \dots, M$ .

*Predicted Error for Objective Functions.* Recall Eq. (8) in that the MSE of an unobserved point  $x^*$  for objective function  $f_m$  is  $s_m^2(x^*)$ . We define the predicted error (PE) of  $x^*$  as

$$\text{PE}_m(x^*) = s_m(x^*)/2, m=1, \dots, M$$

which gives a confidence level of one-half of a standard deviation  $s_m$  from the actual response for each objective function on each side of a design point.

For objective functions, the criterion for the K-MOGA is defined as follows:

- (1) Since  $\text{PE}_m(x^*)$  estimates a deviation from the actual response, the sum of PE values for the  $m$ th objective function of any pair of nondominated point  $x_{nd}$  and dominated point  $x_d$ ,  $\text{PE}_m(x_{nd}) + \text{PE}_m(x_d)$ , is the possible error when calculating the distance between the two points by that objective. In the worst case, this sum should be less than  $\text{MMD}f_m$ , for all objectives,  $m=1, \dots, M$ ; otherwise, the predicted error of any pair of  $x_{nd}$  and  $x_d$  may change the domination status. For instance, as shown in Fig. 2(a) (whereby both objective functions being minimized and Point  $a$  dominates Point  $i$ ), if  $\text{PE}_m(a) + \text{PE}_m(i) \leq \text{MMD}f_m$  for all  $m=1, 2$ , then the domination status of Points  $a$  and  $i$  should not change. In fact, this criterion should hold between any  $x_{nd}$  from the nondominated set and any  $x_d$  from the dominated set.
- (2) Mathematically,  $\text{PE}_m(x_{nd}) + \text{PE}_m(x_d) \leq \text{MMD}f_m$  implies that  $\max(\text{PE}_m(x_{nd}), \text{PE}_m(x_d)) \leq \text{MMD}f_m/2$ .
- (3) In the worst case, if  $2 \times \text{PE}_m(x) \leq \text{MMD}f_m$  and thus  $s_m(x) \leq \text{MMD}f_m$  is true for all  $m=1, \dots, M$  and for design point  $x$ , then the domination status of point  $x$  should not change in the current population.

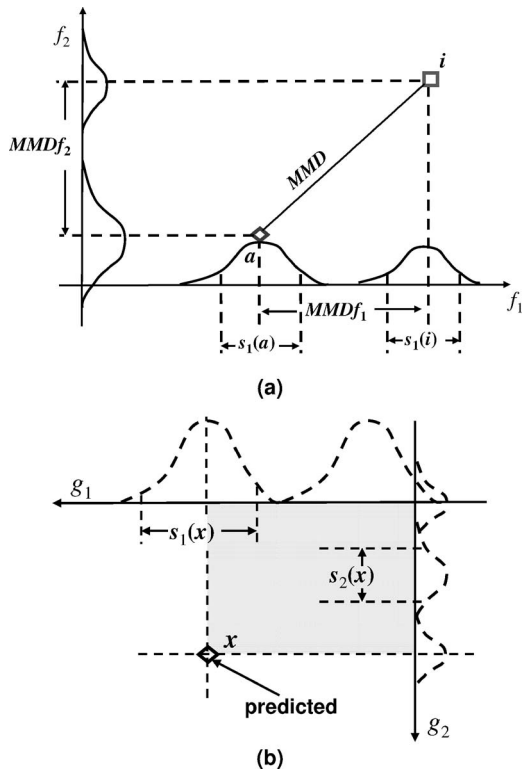
In short, if for any design point  $x$  the following holds:

$$s_m(x) \leq \text{MMD}f_m$$

for all  $m=1, \dots, M$ , then the predicted response values (as obtained by Kriging metamodels) for  $x$  will be considered as “good” values. For those points for which the threshold imposed by Eq. (12) does not hold, the simulation will be used to calculate the actual responses. In this regard, the simulation values will help to improve the fidelity of the subsequent Kriging metamodels.

Note that there are two main reasons why a point with a “large predicted error” must be observed. First, if the predicted error is too large for a design point, then that point should be evaluated by the simulation so that its domination status would not change. Second, a point with a large predicted error implies that the Kriging metamodel does not have enough sample points in its vicinity [22]. In other words, evaluating the point by the simulation and





**Fig. 2 Predicted errors in (a) objective space and (b) constraint space**

thus using it as a new sample point would improve the accuracy of the Kriging metamodel. As a by-product of our approach, our criterion provides the Kriging metamodel with a self-improving mechanism. This is based on the fact that a point with a large error is a potentially good choice for sampling. Note that it is possible that either of  $x_{nd}$  or  $x_d$  has already been evaluated by the simulation in a previous generation. In that case, following Eq. (12), it can be easily verified that there will have to be a coefficient of 2 on the right hand side of the inequality in Eq. (12).

**Predicted Error for Constraint Functions.** Each constraint function can be estimated by a Kriging metamodel as well. Here, the criterion that is used as to whether the Kriging metamodel or simulation should be used is even more critical than that for the objective functions. That is, the Kriging metamodel can be used to determine whether or not a design point is feasible. More precisely, if by using the Kriging metamodel it is determined that the design point is infeasible, then the point is observed. On the other hand, if the point is determined to be feasible by the Kriging metamodel, then the criterion in Eq. (12) has to be verified, as discussed in the next paragraph.

Similarly as in the objective function case, the MSE of an unobserved point  $x^*$  for the constraint function  $g_j$  is  $s_j^2(x^*)$ , whereby  $s_j(x^*)$  estimates the deviation of the constraint's value from a mean for a presumed normal distribution (recall Sec. 2). Since it is critical that the Kriging metamodel provides a good estimate of the feasibility of a design point (i.e., an infeasible point should not be incorrectly predicted as a feasible one), we increase the confidence level to two times the standard deviation from each side of the mean. That is, the quantity  $2 \times s_j(x^*)$  provides a 97.7% prediction accuracy for the Kriging metamodel along the positive (or increasing) direction of a constraint for a design point.

On the other hand, the absolute value of  $\hat{g}_j(x)$  provides a cushion for the predicted error along  $g_j$  dimension, as shown in Fig. 2(b). If this absolute value is greater than  $2 \times s_j(x^*)$ , then the

predicted constraint value has a very little chance (i.e., less than 3%) to change feasibility of the design  $x^*$ . Thus, if for any design point  $x$  the following criterion holds:

$$2 \times s_j(x) + \hat{g}_j(x) \leq 0 \quad (13)$$

for all  $j=1, \dots, J$ , then the predicted constraint value  $\hat{g}_j(x)$ ,  $j=1, \dots, J$  of that design  $x$  will be considered to be acceptable. We only check Eq. (13) for the predicted feasible designs (i.e., with  $\hat{g}_j(x) \leq 0$ ,  $j=1, \dots, J$ ). If a design point is predicted to be infeasible, it is observed by the simulation. Note that although we used a higher confidence level in the constraint case compared to the objective case so that infeasible points are correctly predicted to be infeasible, the determination as to whether one or multiple standard deviations  $s_j$  to be used is not all that critical. We have observed that by using a different setting, e.g., one or three times instead of two times  $s_j$  as in Eq. (13), the performance of K-MOGA does not change significantly in terms of the number of simulation calls.

**Proposed Kriging Metamodel Assisted Multi-Objective Genetic Algorithm.** We combine Eq. (12) with Eq. (13), as shown in Eq. (14):

$$s_m(x) \leq \text{MMD}f_m \quad m = 1, \dots, M \quad (14)$$

$$2 \times s_j(x) + \hat{g}_j(x) \leq 0 \quad j = 1, \dots, J$$

The designs in the current population can be divided into two groups. In the first group that satisfies Eq. (14), the predicted responses will not change the domination status of the designs and also these designs are predicted to be feasible and therefore their predicted response values can be accepted. For the designs in this first group, no simulation is required. Designs in the second group do not satisfy Eq. (14). As such, the designs in this second group are calculated by the simulation to obtain their objective/constraint values, as shown in the "Kriging assist fitness evaluation" block in Fig. 1.

In K-MOGA, the responses from all points in the initial population are calculated by the simulation in order to build the initial Kriging metamodels, i.e., a Kriging metamodel is constructed for each objective/constraint function. Since the initial points may be far away from the Pareto frontier or they may not sample the design space well, the initial Kriging metamodels may not be sufficiently accurate. However, these Kriging metamodels are adaptively improved as the algorithm evolves. During the early generations of the K-MOGA, the percentage of design points for which the Kriging metamodels are used is small (e.g., for the initial population, this percentage is zero). However, as more observed points are added to the Kriging metamodel, the predicted error of the unobserved points is going to gradually improve and the percentage of the points for which Kriging metamodels are used is going to increase as the subsequent generations are evolved. Note that, according to the criteria in Eq. (14), points with a large predicted error or points that are predicted infeasible are required to be observed and used as sample points to improve the Kriging metamodels. Thus, to some extent, the general concern in using a Kriging metamodel that it should have high fidelity during the entire optimization algorithm and particularly during the initial stages can be avoided. Based on our observation, in K-MOGA, eventually the Kriging metamodels can achieve high fidelity and when that happens no more simulations are required as K-MOGA gets close to the Pareto frontier.

Figure 1 shows the complete K-MOGA procedure, including the Kriging assist fitness evaluation block, for a single generation. Note that although the dominated points, which have been observed, might be removed from the population during the evolutionary process, they will not be removed from the sample set of the Kriging metamodel. In other words, all observed points will be used for the Kriging metamodel and the size of the sample set is monotonically increasing.

**3.3 Stopping Criteria.** Since a comparison of the performance of the conventional MOGA and K-MOGA is important, appropriate and consistent stopping criteria for MOGA and K-MOGA should be determined. The following two stopping criteria are used and both have to be satisfied.

- (1) When the number of nondominated points is more than some prespecified percentage of the population size (e.g., 80% for the examples in this paper) and when it becomes steady (e.g., the number of nondominated points is more than “ $0.8 \times$  population size” for five generations for the examples in this paper), it is concluded that the algorithm has converged to the Pareto frontier.
- (2) When the iteration history, i.e., the curve representing the number of simulation calls versus the number of generations becomes flat, it can be concluded that the algorithm has been converged.

One may also employ other criteria or metrics for quality assessment of the Pareto frontier [29] as additional stopping criteria. We applied two quality metrics proposed previously in the literature [29] to compare the performance of the conventional MOGA and K-MOGA in terms of convergence and diversity of solutions, as discussed in Sec. 4.5.

**3.4 Kriging Metamodel Assisted Genetic Algorithm Steps.**

The steps for K-MOGA are as follows:

*Step 1.* Initialize. Start with generating an initial population. Simulation models are called to calculate the responses (i.e., objective/constraint functions) for individuals in the initial population and these are added to a sample set to build the initial Kriging metamodels, one for each objective/constraint function. The nondominated (or elite) points in the initial population are identified and migrated into the next generation. The remaining (or dominated) points for the next generation are generated by the GA operations.

*Step 2.* The algorithm evolves into the next generation.

*Step 3.* Apply the current Kriging metamodels to predict response values for the current population. By way of Eq. (14), the individuals in the current generation are partitioned into two parts as follows: (i) Calculate the response values (from objective/constraint functions) and RMSE (recall Eq. (8)) for each design point in the current generation using the Kriging metamodel. (ii) Identify the nondominated and dominated points and calculate MMD. (iii) Apply Eq. (14) to each individual. Individuals whose predicted response values satisfy Eq. (14) will be considered as good: The simulation model for these points is not used rather their metamodel is used. For the individuals that do not satisfy Eq. (14), the simulation model is used to calculate their responses.

*Step 4.* Calculate the fitness value of each point. Nondominated sorting algorithm [1] is used to calculate the fitness of each point.

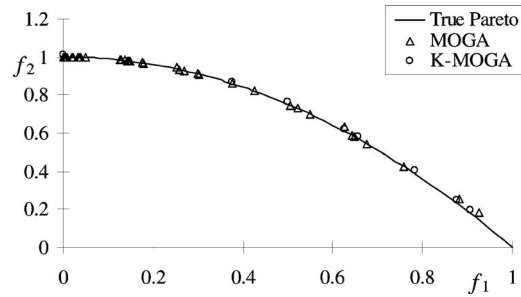
*Step 5.* Identify nondominated points and update the Kriging metamodels. Nondominated points in the current population are identified. Points (dominated or nondominated) whose response values are calculated by the simulation are added to the sample set to update the Kriging metamodels.

*Step 6.* Check the stopping criteria. Check the stopping criteria described in Sec. 3.3. If both stopping criteria are satisfied, stop the algorithm; otherwise, continue.

*Step 7.* Form the next population. The next population includes two parts: elite and offspring points. Go to Step 2.

**4 Examples and Discussion**

In this section, we use seven numerical and engineering examples with different degrees of difficulty to illustrate the applicability of the proposed K-MOGA. As a typical example of our results, we use the first example, a simplified version of ZDT2 [1], to present a detailed comparison of the conventional MOGA and K-MOGA including the verification via (i) quality metrics, (ii) MMD, and (iii) the predicted error. Due to the space limitation,



**Fig. 3 Pareto solutions for the ZDT2 example**

we do not show the detailed verification for the other test examples which produced similar results. We also present the results for a real-world engineering example [30,31]. Finally, comparison results for the remaining five examples selected from the literature [1,20] are presented. In order to compare the conventional MOGA and our proposed K-MOGA, the same initial population size of 30 individuals is used for all experiments. Also MOGA and K-MOGA were run for 30 times each to account for randomness in these methods. The values of other genetic parameters are selected according to the description in Sec. 3.1. The same settings are used for all examples. For simplicity, the conventional MOGA is referred as “MOGA” hereafter.

**4.1 ZDT2 Example.** We applied the MOGA, which was described in Sec. 3.1, and K-MOGA to a simplified ZDT2 [1] problem. This example has two-objective functions, no constraint, and three variables:

$$\min f_1(\mathbf{x}) = x_1$$

$$f_2(\mathbf{x}) = g(\mathbf{x}) \times h(\mathbf{x})$$

where

$$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

$$h(\mathbf{x}) = 1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2 \tag{15}$$

$$n = 3$$

$$0 \leq x_i \leq 1 \quad i = 1, \dots, n$$

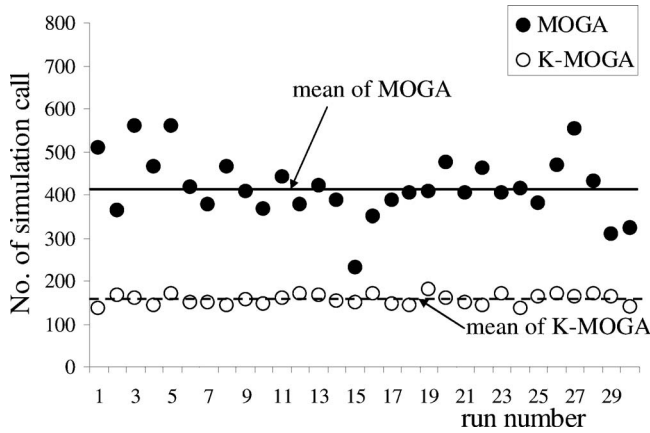
The true Pareto optimal solutions for this problem [1] are  $x_1$  being any point within the range  $[0,1]$ ,  $x_2=0$ , and  $x_3=0$ , with the true Pareto frontier as shown in Eq. (16).

$$f_2 = 1 - f_1^2 \tag{16}$$

$$0 \leq f_1 \leq 1$$

The Pareto frontier is nonconvex, as shown in Fig. 3. For this example, two separate Kriging metamodels for the two objectives are built in K-MOGA and adaptively improved to predict the response for the objective functions.

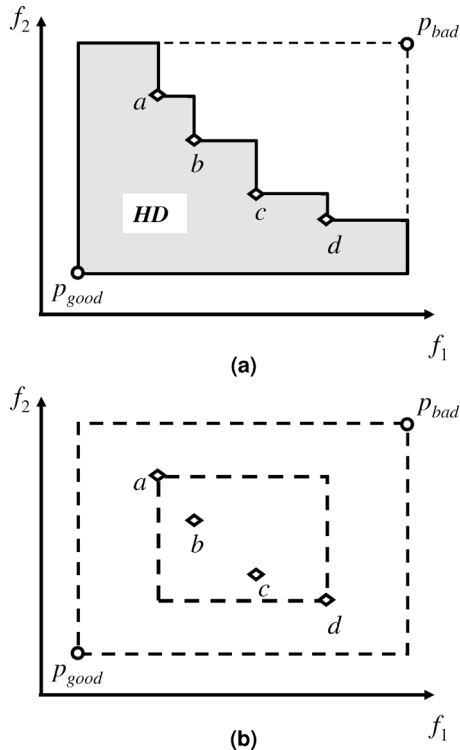
Figure 3 shows a typical set of Pareto optimal solutions as obtained from one of the 30 runs of MOGA and K-MOGA. The results from K-MOGA are in good agreement with MOGA and the true Pareto frontier. Figure 4 shows the number of simulation call for 30 different runs. As shown in Fig. 4, a MOGA run with the least number of simulation calls (i.e., 231 in run 15) requires more simulation calls than a K-MOGA run with the maximum number of simulation calls (i.e., 197). The mean value and standard deviation (STD) for all 30 runs for both the MOGA and K-MOGA are shown in Table 3. The results for the ZDT2 ex-



**Fig. 4** No. of simulation calls versus run number for the ZDT2 example

ample show that on the average the number of simulation calls has been reduced by more than 60% using the proposed K-MOGA compared to MOGA.

**4.1.1 Verification by Quality Metrics.** In order to evaluate the quality of convergence and diversity of solutions for the proposed K-MOGA and compare the results with MOGA, two quality metrics proposed in the literature [29], i.e., the hyperarea difference (HD) and overall spread (OS) metrics, are calculated and obtained



**Fig. 5** Quality metrics (a) HD and (b) OS

**Table 1** Quality metrics for MOGA and K-MOGA for the ZDT2 example

	MOGA			K-MOGA		
	30 runs	Mean	STD	30 runs	Mean	STD
HD	[0.90–0.64]	0.75	0.069	[0.50–0.95]	0.72	0.097
OS	[0.42–1.40]	0.66	0.200	[0.31–1.91]	0.69	0.331

for the ZDT2 example.

Figure 5 shows the geometrical interpretation of these two metrics in a two-objective space. Let us assume  $P=\{a,b,c,d\}$  be the current nondominated set in the objective space and  $p_{\text{bad}}$  and  $p_{\text{good}}$  are the good and bad points, respectively. The quantity HD, shown as the shaded area in Fig. 5(a), is defined as the difference between the area (hyperarea (HA) or volume if there are three or more objectives) bounded by  $p_{\text{bad}}$  and  $p_{\text{good}}$  and the area bounded by  $p_{\text{bad}}$  and the set  $P$ :

$$\text{HD}(P) = \text{HA}(p_{\text{bad}}, p_{\text{good}}) - \text{HA}(p_{\text{bad}}, a, b, c, d) \quad (17)$$

The quantity OS, shown in Fig. 5(b), is defined as the ratio between the area bounded by the two extreme points in  $P$  and the area bounded by  $p_{\text{bad}}$  and  $p_{\text{good}}$ :

$$\text{OS} = \frac{\text{HA}[\text{extremes}(P)]}{\text{HA}(p_{\text{bad}}, p_{\text{good}})} \quad (18)$$

The quantities HD and OS serve as the quality metrics of convergence and diversity, respectively, for the obtained Pareto frontier.

In the ZDT2 example, we set  $p_{\text{bad}}=[1, 1]$  and  $p_{\text{good}}=[0, 0]$  as their objective function values, and  $\text{HD}(\text{true})$ , for true Pareto frontier shown in Eq. (16), is  $2/3$  and  $\text{OS}(\text{true})$  is 1. We calculate the HD and OS values for the MOGA and K-MOGA from the 30 runs, as shown in Table 1. From Table 1, it can be seen that the convergence and diversity of the obtained Pareto frontiers from the MOGA and K-MOGA are comparable.

**4.1.2 Verification by MMDf.** As a further verification of the proposed approach, in Fig. 6, we compare  $\text{MMDf}_m$  based on both simulation and Kriging metamodels for the ZDT2 example. According to our experiments, the estimated  $\text{MMDf}_m$  as obtained from the combined Kriging metamodels and simulations is less than or equal to  $\text{MMDf}_m$  from the simulation for most generations. Moreover, this is especially the case at the later generations of K-MOGA. From these results, we can conclude that  $\text{MMDf}_m$  from Kriging provides a good estimate of the actual  $\text{MMDf}_m$ .

**4.1.3 Verification by Predicted Error.** For verification as to whether the predicted error  $s_m(x)$  is a valid estimation of deviation in Eqs. (12) and (13), we have obtained the error, which is the absolute value of the difference between the actual and predicted values of  $f$  (for both  $f_1$  and  $f_2$  in this case) for each design point in a typical generation (e.g., the tenth generation in Fig. 7 for the ZDT2 problem). The term “Real error,” as shown in Fig. 7, is for the deviation from the predicted value (from Kriging metamodel) to the actual value (from simulation). The “Predicted error” term is equal to  $0.5 \times s_m(x)$  and calculated from the Kriging metamodel as in Eq. (11). As shown in Fig. 7, for most design points (i.e., 25 out of 30 designs) in the tenth generation, the real error is less than the predicted error, which means that  $s_m(x)$  is a valid estimation of the STD.

Similar results were observed for the errors in the constraint estimation, and for other generations and test problems.

**4.2 Cabinet Example.** The optimization of a fully enclosed vertical cabinet containing ten individual blade server racks is selected as the engineering test example. The simulation model for this example was developed by Rolander et al. [30]. This model has two isoflux blocks that act as flow obstructions, repre-

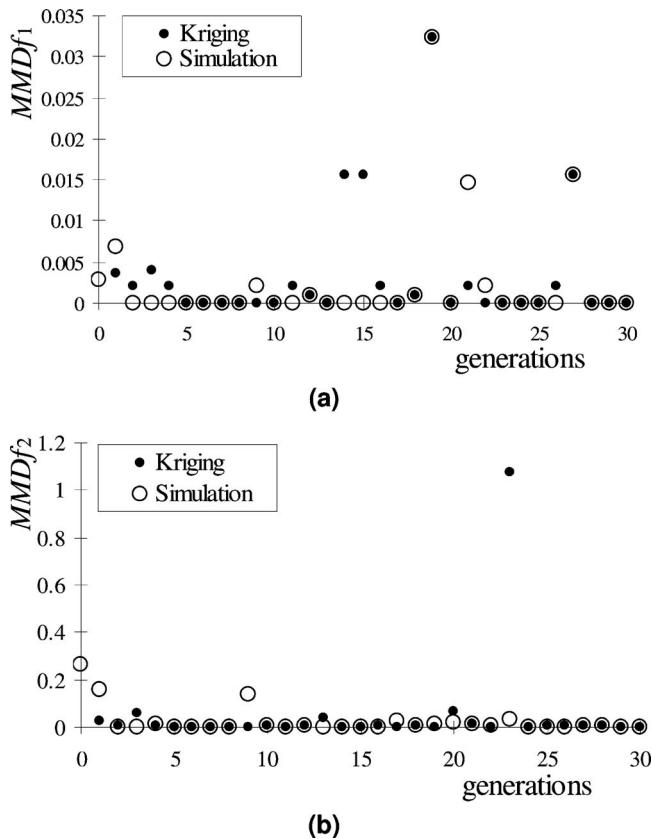


Fig. 6 (a)  $MMDf_1$  and (b)  $MMDf_2$  based on simulation and Kriging metamodel

sending a dual processor blade server. Both blocks have a constant heat generation rate  $Q$ , which is dissipated through convection to the air flowing through the server. The cabinet is divided into three sections:  $a$ ,  $b$ , and  $c$ , corresponding to the lower two, middle three, and upper five servers. The quantities  $Q_a$ ,  $Q_b$ , and  $Q_c$  denote the heat generation of servers in each respective cabinet section. In each section, all blocks have the same constant heat generation rate  $Q$ . Air flow with an inlet velocity is used to cool the cabinet. A two-dimensional heat transfer simulation model has already been built for this problem and reported in the literature [30,31].

In this thermal optimization model, inlet air velocity  $V_{in}$  and heat generation  $Q_a$ ,  $Q_b$ , and  $Q_c$  are considered as design variables. The output of the simulation model is a ten-element vector of temperatures, each for one server. The design objectives are to (1) minimize  $f_1$ , the maximum server temperatures, and (2) maximize  $f_2$ , the sum of total heat generations. The optimization problem is defined as

$$\begin{aligned}
 & \text{Min}_{V_{in}, Q_{a,b,c}} f_1 = \max(T_j) \quad j = 1, \dots, 10 \\
 & \text{Max}_{V_{in}, Q_{a,b,c}} f_2 = 2Q_a + 3Q_b + 5Q_c \\
 & \text{s.t.} \quad \max(T_j)/85 - 1 \leq 0 \quad j = 1, \dots, 10 \\
 & \quad \quad 0.2 \leq V_{in} \leq 2 \\
 & \quad \quad 2 \leq Q_{a,b,c} \leq 200
 \end{aligned} \quad (19)$$

The optimization results for this cabinet problem are shown in Fig. 8 and Table 3. Again, it is observed that the number of simulation calls used in K-MOGA is significantly fewer than MOGA, while the results, as shown in Fig. 8, are comparable.

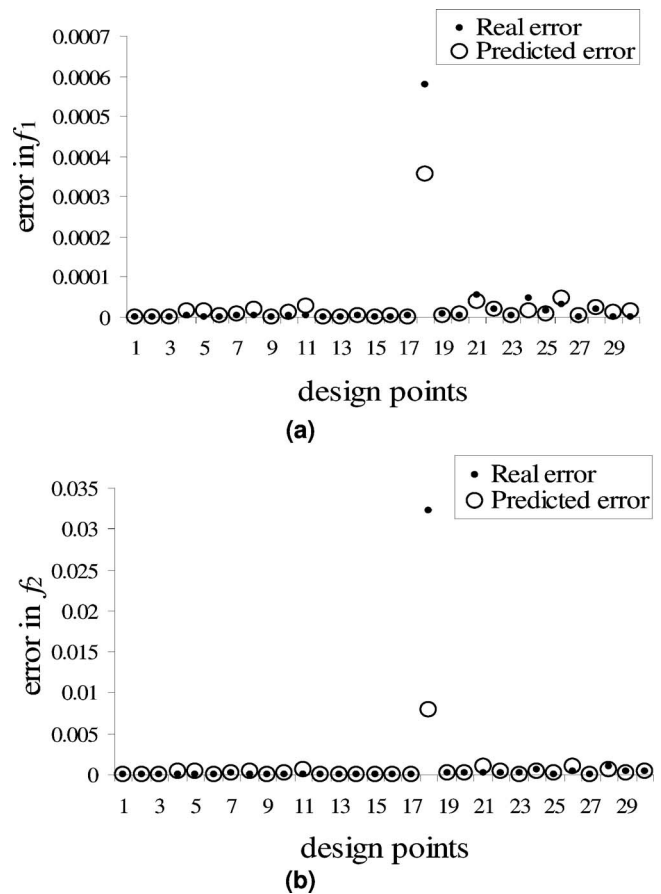


Fig. 7 Real error and predicted error for (a)  $f_1$  and (b)  $f_2$  for the ZDT2 example in the tenth generation

**4.3 Additional Numerical Examples.** In this section, five additional examples, ZDT1, ZDT3, GearTrain and OSY from [1], and TEST4 from [20], are presented to demonstrate further applicability of K-MOGA. These five test problems have different degrees of difficulty and characteristics. For instance, the Pareto frontiers for ZDT3 and TEST4 are disconnected. The GearTrain test example has integer design variables. The formulations of these test examples are listed in Table 2 with the comparison results as obtained from K-MOGA, MOGA, and the true Pareto frontier when the closed form solution is available.

The performance of MOGA and K-MOGA for all examples in terms of simulation calls are presented in Table 3. As shown in that table, on the average, K-MOGA can save over 50% in terms

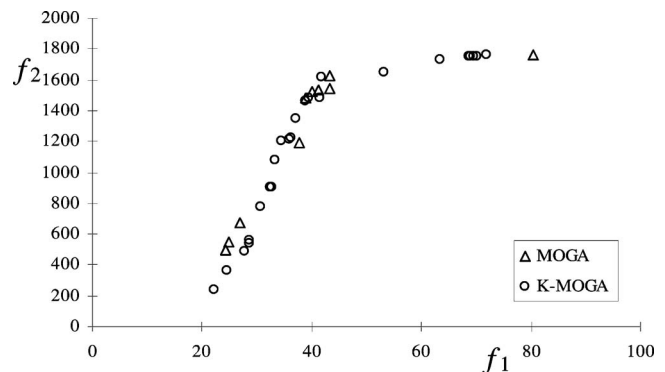


Fig. 8 Pareto solutions for cabinet problem using MOGA and proposed K-MOGA



Table 2 Formulation and solutions for additional numerical examples

	Formulation	Solutions
<b>ZDT1</b>	minimize $f_1(x) = x_1$ $f_2(x) = g(x) \times h(x)$ where $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(x) = 1 - \sqrt{f_1(x)/g(x)}$ $n = 3$ $0 \leq x_i \leq 1, \quad i = 1, \dots, n$	
<b>ZDT3</b>	minimize $f_1(x) = x_1$ $f_2(x) = g(x) \times h(x)$ where $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(x) = 1 - \sqrt{f_1(x)/g(x)} - (f_1(x)/g(x)) \sin(10 \pi f_1)$ $n = 3$ $0 \leq x_i \leq 1, \quad i = 1, \dots, n$	
<b>TEST4</b>	minimize $f_1(x_1, x_2) = (x_1 + x_2 - 7.5)^2 + (x_2 - x_1 + 3)^2 / 4$ $f_2(x_1, x_2) = (x_1 - 1)^2 / 4 + (x_2 - 4)^2 / 2$ s. t. $g_1(x_1, x_2) = (x_1 - 2)^3 / 2 + x_2 - 2.5 \leq 0$ $g_2(x_1, x_2) = x_1 + x_2 - 8(x_2 - x_1 + 0.65)^2 - 3.85 \leq 0$ $0 \leq x_1 \leq 5; 0 \leq x_2 \leq 3$	
<b>OSY</b>	minimize $f_1(x) = -[25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2]$ $f_2(x) = \sum_{i=1}^n x_i^2$ s. t. $g_1(x) \equiv 2 - (x_1 + x_2) \leq 0$ $g_2(x) \equiv (x_1 + x_2) - 6 \leq 0$ $g_3(x) \equiv (x_2 - x_1) - 2 \leq 0$ $g_4(x) \equiv (x_1 - 3x_2) - 2 \leq 0$ $g_5(x) \equiv (x_3 - 3)^2 + x_4 - 4 \leq 0$ $g_6(x) \equiv 4 - ((x_5 - 3)^2 + x_6) \leq 0$ $0 \leq x_1, x_2, x_6 \leq 10, \quad 1 \leq x_3, x_5 \leq 5 \quad 0 \leq x_4 \leq 6$	
<b>Gear-Train</b>	minimize $f_1(x) = \log_{10} \left( \left( \frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4} \right)^2 \right)$ $f_2(x) = \max(x_1, x_2, x_3, x_4)$ where $12 \leq x_1, x_2, x_3, x_4 \leq 60$ $x_i$ are integers	



**Table 3 Comparison of MOGA and K-MOGA for test examples**

Example	Pop-size	Percent average reduction in No. of simulation calls	Number of simulation calls					
			MOGA			K-MOGA		
			30 runs	Mean	STD	30 runs	Mean	STD
ZDT1	30	57%	[293–594]	398	51	[120–258]	172	38
ZDT2	30	63%	[231–561]	417	73	[134–197]	156	12
ZDT3	30	50%	[340–616]	485	69	[176–307]	243	34
TEST4	20	60%	[248–529]	372	74	[105–213]	148	29
OSY	50	50%	[1455–2288]	1819	210	[701–1210]	901	126
Gear train	30	50%	[347–633]	471	78	[159–323]	234	39
Cabinet	30	42%	[810–1125]	998	75	[434–705]	584	75

of the number of simulation calls when compared to MOGA. For most of the test examples, STDs in K-MOGA are also less than those in MOGA. Based on the results in Table 2, K-MOGA is expected to be applicable to problems with convex, nonconvex, and even those with disconnected Pareto frontier.

## 5 Summary

A new multi-objective design optimization approach called K-MOGA is presented in this paper. In the proposed approach, the Kriging metamodeling is embedded within a conventional MOGA. Compared to the conventional MOGA, K-MOGA reduces the number of simulation calls by evaluating some individuals in the population by Kriging metamodels instead of the simulation. We have introduced the concept of the MMD and derived its relation with the predicted error that is easily obtained from Kriging. This criterion is used to identify those individuals in the population that can be evaluated using Kriging metamodels. The identified individuals are those that do not change the estimated domination status in the objective space and do not change the estimated feasibility for the current generation. For other individuals in the generation, the responses are obtained from the simulation and used to adaptively update the next generation Kriging metamodels so that more individuals can be evaluated by the updated Kriging metamodels and thus an additional number of simulation calls can be saved in subsequent generations.

In K-MOGA, the general concern that the metamodel may be of low fidelity and that it may even produce false optima can be avoided. The proposed criterion is objective rather than subjective and can be applied to other population-based optimization methods using a different type of metamodels if the measure for predicted error is available. The main advantage of using on-line Kriging is that the predicted error of the estimated response can be obtained without much extra computational effort.

Seven examples of both numerical and engineering types and with different degrees of difficulty are used to demonstrate the applicability of the proposed K-MOGA. The results show that K-MOGA is able to achieve comparable convergence and diversity of the Pareto frontier as to that obtained from a conventional MOGA while at the same time significantly reduce the number of simulation calls.

One of the advantages of adaptive approaches is that those points migrated from previous generations with incorrectly estimated Kriging variance are most likely to be removed from the population by a more accurate Kriging metamodel. Therefore, the side effect of such migrated points can be diminished when the Kriging metamodels are updated adaptively in consecutive generations. In essence, the proposed K-MOGA has a self-correcting mechanism in terms of identifying “good” points for Kriging metamodeling.

Finally, it should be noticed that the relation between Kriging’s predicted error and MMD for the objective functions, and also for the constraints, is devised based on a worst case scenario and thus the proposed criterion can be considered to be conservative. Also,

the sample set used in Kriging metamodel can affect the accuracy of the metamodel (currently, they are generated by GA operations). By devising a less conservative criterion and an improved sampling strategy for Kriging, it should be possible to further improve the efficiency of K-MOGA in terms of the number of simulation calls. These aspects will be further investigated as part of our future research.

## Acknowledgment

The work presented in this paper was supported in part through a Contract from Indian Head Division, Naval Surface Warfare Center, and in part by a contract from the Office of Naval Research. Such support does not constitute an endorsement by the funding agency of the opinions expressed in the paper. Special thanks are due to Professor Yogendra Joshi and his research group at Georgia Tech (in particular, Mr. Jeffrey Rambo) for providing us with the simulation model for the cabinet example.

## References

- [1] Deb, K., 2001, *Multiobjective Optimization Using Evolutionary Algorithms*, Wiley, New York.
- [2] Farina, M., 2001, “A Minimal Cost Hybrid Strategy for Pareto Optimal Front Approximation,” *Evolutionary Optimization*, an international journal on the internet, 3(1), pp. 41–52 (available online at [www.jeo.org](http://www.jeo.org)).
- [3] Farina, M., 2002, “A Neural Network Based Generalized Response Surface Multiobjective Evolutionary Algorithm,” *Proceedings of Congress on Evolutionary Computation*, IEEE, New York, pp. 956–961.
- [4] Papadrakakis, M., Lagaros, N., and Tsompanakis, Y., 1999, “Optimization of Large-Scale 3D Trusses Using Evolution Strategies and Neural Networks,” *Int. J. Space Struct.*, 14(3), pp. 211–223.
- [5] Hong, Y.-S., Lee, H., and Tahk, M.-J., 2003, “Acceleration of the Convergence Speed of Evolutionary Algorithms Using Multilayer Neural Networks,” *Eng. Optimiz.*, 35(1), pp. 91–102.
- [6] Lian, Y., and Liou, M., 2004, “Multiobjective Optimization Using Coupled Response Surface Model and Evolutionary Algorithm,” *Proceedings of the Tenth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, Aug. 30–Sept. 1, AIAA-2004-4323.
- [7] Chung, H., and Alonso, J., 2004, “Multiobjective Optimization Using Approximation Model-Based Genetic Algorithms,” *Proceedings of the Tenth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, Aug. 30–Sept. 1, AIAA-2004-432.
- [8] Fang, H., Rais-Rohani, M., and Horstemeyer, M., 2004, “Multiobjective Crashworthiness Optimization With Radial Basis Functions,” *Proceedings of the Tenth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, Aug. 30–Sept. 1, AIAA-2004-4487.
- [9] Chen, J.-H., Goldberg, D. E., Ho, S.-Y., and Sastry, K., 2002, “Fitness Inheritance in Multi-Objective Optimization,” *Proceedings of the Genetic and Evolutionary Computation Conference*, New York, July 9–13, Morgan Kaufmann, New York.
- [10] Smith, R., Dike, B., and Stegmann, S., 1995, “Fitness Inheritance in Genetic Algorithms,” *Proceedings of the ACM Symposium on Applied Computing*, ACM, Nashville, TN, February 26–28, pp. 345–350.
- [11] Jin, Y., Olhofer, M., and Sendhoff, B., 2001, “Managing Approximate Models in Evolutionary Aerodynamic Design Optimization,” *Proceedings of IEEE Congress on Evolutionary Computation*, Vol. 1, pp. 592–599.
- [12] Jin, Y., Olhofer, M., and Sendhoff, B., 2002, “A Framework for Evolutionary Optimization With Approximate Fitness Functions,” *IEEE Trans. Evol. Comput.*, 6(5), pp. 48–494.
- [13] Nain, P. K. S., and Deb, K., 2003, “Computationally Effective Search and Optimization Procedure Using Coarse to Fine Approximations,” *Proceedings of the Congress on Evolutionary Computation (CEC-2003)*, Canberra, Australia.

- lia, pp. 2081–2088.
- [14] Nair, P. B., and Keane, A. J., 1998, “Combining Approximation Concepts With Genetic Algorithm-based Structural Optimization Procedures,” *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, 39th, and AIAA/ASME/AHS Adaptive Structures Forum*, Long Beach, CA, Apr. 20–23, Collection of Technical Papers, Pt. 2, A98-25092 06-39, AIAA-1998-1912.
- [15] Oduguwa, V., and Roy, R., 2002, “Multi-Objective Optimization of Rolling Rod Product Design Using Meta-Modeling Approach,” *Proceedings of the Genetic and Evolutionary Computation Conference*, New York, July 9–13, Morgan Kaufmann, New York, pp. 1164–1171.
- [16] Jin, Y., 2005, “A Comprehensive Survey of Fitness Approximation in Evolutionary Computation,” *Soft Comput.*, **9**(1), pp. 3–12.
- [17] Wang, G. G., and Shan, S., 2007, “Review of Metamodeling Techniques in Support of Engineering Design Optimization,” *ASME J. Mech. Des.*, **129**(4), pp. 370–380.
- [18] Simpson, T. W., Peplinski, J., Koch, P. N., and Allen, J. K., 2001, “Metamodels for Computer-based Engineering Design: Survey and Recommendations,” *Eng. Comput.*, **17**(2), pp. 129–150.
- [19] Simpson, T. W., Booker, A. J., Ghosh, D., Giunta, A. A., Koch, P. N., and Yang, R.-J., 2004, “Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion,” *Struct. Multidiscip. Optim.*, **27**, pp. 302–313.
- [20] Wilson, B., Cappelleri, D., Frecker, M., and Simpson, T. W., 2001, “Efficient Pareto Frontier Exploration Using Surrogate Approximations,” *Optim. Eng.*, **2**, pp. 31–50.
- [21] Koch, P. N., Wujek, B. A., Golovidov, O., and Simpson, T. W., 2002, “Facilitating Probabilistic Multidisciplinary Design Optimization Using Kriging Approximation Models,” *Proceedings of the Ninth AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, Sept. 4–6, AIAA 2002-5415.
- [22] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., 1989, “Design and Analysis of Computer Experiments,” *Stat. Sci.*, **4**(4), pp. 409–435.
- [23] Martin, J., and Simpson, T. W., 2006, “A Methodology to Manage System-level Uncertainty During Conceptual Design,” *ASME J. Mech. Des.*, **128**(4), pp. 959–968.
- [24] Sasena, M. J., Parkinson, M., Reed, M. P., Papalambros, P. Y., and Goovaerts, P., 2005, “Improving an Ergonomics Testing Procedure via Approximation-Based Adaptive Experimental Design,” *ASME J. Mech. Des.*, **127**(5), pp. 1006–1013.
- [25] Albada, S. J., and Robinson, P. A., 2007, “Transformation of Arbitrary Distributions to the Normal Distribution With Application to EEG Test-Retest Reliability,” *J. Neurosci. Methods*, **161**(2), pp. 205–211.
- [26] Gano, S. E., Renaud, J. E., Martin, J. D., and Simpson, T. W., 2006, “Update Strategies for Kriging Models for Using in Variable Fidelity Optimization,” *Struct. Multidiscip. Optim.*, **32**(4), pp. 287–298.
- [27] Narayanan, S., and Azarm, S., 1999, “On Improving Multiobjective Genetic Algorithms for Design Optimization,” *Struct. Optim.*, **18**, pp. 146–155.
- [28] Kurapati, A., Azarm, S., and Wu, J., 2002, “Constraint Handling Improvements for Multi-Objective Genetic Algorithms,” *Struct. Multidiscip. Optim.*, **23**(3), pp. 204–213.
- [29] Wu, J., and Azarm, S., 2001, “Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set,” *ASME J. Mech. Des.*, **123**(1), pp. 18–25.
- [30] Rolander, N., Rambo, J., Joshi, Y., and Mistree, F., 2005, “Towards Sustainable Design of Data Centers: Addressing the Lifecycle Mismatch Problem,” *Proceedings of IPACK05 International Electronic Packaging Technical Conference and Exhibition*, San Francisco, CA, Jul. 17–22.
- [31] Li, G., Li, M., Azarm, S., Rambo, J., and Joshi, Y., 2007, “Optimizing Thermal Design of Data Center Cabinets with A New Multi-Objective Genetic Algorithm,” *Distributed and Parallel Databases*, **21**(2–3), pp. 167–192.