

A Generative Kriging Surrogate Model for Constrained and Unconstrained Multi-objective Optimization*

Rayan Hussein and Kalyanmoy Deb

Computational Optimization and Innovation (COIN) Laboratory

Department of Electrical and Computer Engineering

Michigan State University

East Lansing, MI 48824, USA

{husseinr,kdeb}@egr.msu.edu, <http://www.egr.msu.edu/~kdeb/>

COIN Report Number 2016007

March 28, 2016

Abstract

Surrogate models are effective in reducing the computational time required for solving optimization problems. However, there have been a lukewarm interest in finding multiple trade-off solutions for multi-objective optimization problems using surrogate models. The literature on surrogate modeling for constrained optimization problems is also rare. The difficulty lies in the requirement of building and solving multiple surrogate models, one for each Pareto-optimal solution. In this paper, we first provide a brief introduction of the past studies and suggest a computationally fast, Kriging-based, and generative procedure for finding multiple near Pareto-optimal solutions in a systematic manner. The expected improvement metric is maximized using a real-parameter genetic algorithm for finding new solutions for high-fidelity evaluations. The approach is computationally fast due to the interlinking of building multiple surrogate models and in its systematic sequencing methodology for assisting one model with another. In standard two and three-objective test problems with and without constraints, our proposed methodology takes only a few hundreds of high-fidelity solution evaluations to find a widely distributed near Pareto-optimal solutions compared to the standard EMO methods requiring tens of thousands of high-fidelity solution evaluations. The framework is generic and can be extended to utilize other surrogate modeling methods easily.

*Accepted in GECCO-2016 Conference (20-24 July, 2016), Denver, Colorado, USA.

1 Introduction

In most practical optimization problems, the evaluation of a solution involves computationally expensive softwares and procedures, which remain as one of the main bottlenecks of completing an optimization run within a reasonable amount of time. However, it is clear that since solutions created in the beginning of an optimization simulation are expected to be away from the optimal region and an exact and precise evaluation of the solution may not be necessary. As the solutions progress towards the optimal region, a more precise evaluation may be invoked.

The recent developments of optimization methods have led to an increasing interest of approximation models or *surrogate* models [1, 2, 3]. A surrogate model approximates the original objective or constraint function by computing a few high-fidelity solutions – solutions that are evaluated using the original but computationally expensive functions. Most surrogate models are designed in a way so as to pass through these high-fidelity solutions, such as Kriging models [4], but there exist some methods that makes a regression fit of the high-fidelity solutions. The optimization process is then continued using the surrogate model until the model is deemed to be accurate enough in the current region of focus. Due to this reason, surrogate models are effective in reducing the overall computational time required in real-world design optimization.

Despite the existence of a plethora of surrogate modeling study in the context of single-objective optimization problems, there is not much efforts spent on extending the ideas to multi-objective optimization. The reasons for the lack of study, except a few [5, 2] is that in multi-objective optimization, the aim is to find multiple trade-off Pareto-optimal solutions, instead of a single optimal solution. Most surrogate modeling studies have not considered modeling a multi-modal function in the spirit of locating multiple optimal solutions simultaneously. Hence, there is not much clue that can be borrowed from the single-objective literature. However, Emmerich et al [5] have generalized the probability of improvement and the expected improvement concept to multi-objective optimization. Single-objective EAs are used for maximizing one of these metrics for determining which point should be evaluated next. Although one can select multiple test points with high-metric values at each iteration, these scalar metrics on their own could not predict the most likely shape and location of the whole PF. Therefore, they may not be stochastically sound for locating multiple test points. Another study [2] used a generative method of approximating the objective function for a single Pareto-optimal solution at a time. However, the study did not consider any constraint, which is a major issue in making surrogate modeling methods practical. It is ironical that while surrogate models are used to make optimization methods practical, but on the other hand the methods are flexible enough to handle other important practicalities, such as constraint and uncertainty handling. This work is motivated by the later study, but enables the use of constraints and other efficient features that make the overall algorithm computationally fast and applicable to larger-sized problems than the above existing methodologies have demonstrated.

In the remainder of this paper, we provide a brief summary of the efficient global optimization (EGO) procedure and some past surrogate modeling studies in the context of multi-objective optimization in Section 2. The proposed multi-objective EGO procedure is then described in detail in Section 3. Results on unconstrained and constrained test prob-

lems are then presented in Section 4. Finally, conclusions and extensions of this study are discussed in Section 5.

2 Efficient Global Optimization (EGO)

One of the popular surrogate models used in approximating computationally expensive functions is the well-known Kriging model. Kriging methodology was proposed by Daniel G. Krige [6] to predict the spatial patterns for gold mines. Then, more improvements to the Kriging Model were developed by Matheron [7]. On the other hand, Sacks et al [8] utilized the Kriging model to improve the approximation of computer experiments. Later, Kriging methodology was also called the design and analysis of computer experiments (or DACE) stochastic process model [9].

The DACE is one of the efficient tools for dealing with expensive single-objective optimization problems. In this methodology, the approximation of a function in terms of a design variable is considered as a sample of Gaussian stochastic process. Then, the distribution of the function value at any untested point can be estimated using Kriging model. Jones et al [10] proposed a practical approach to determine the location of additional sample points which improves the Kriging model accuracy. This is known as the Efficient Global Optimization (EGO) procedure. On the Kriging model, EGO searches for the location where the expected improvement of the original function is maximized, and then reconstructs the surrogate model by adding a new sample point at this location. This approach consists of the following steps:

Step 1: Build an initial Kriging model for the objective function.

Step 2: Use a cross-validation method to ensure that the Kriging prediction and measure of uncertainty are satisfactory.

Step 3: Find the location that maximizes the expected improvement (EI) function.

Step 4: Evaluate the point at which the EI function was maximum. Update the Kriging model using the new point and move to Step 3.

There exists a large number of studies to incorporate Kriging model into evolutionary algorithm (EA) for only single-objective optimization problems [11]. Therefore, our proposed method is dealing with Multi-objective optimization problems based on Kriging model. A brief overview of Kriging methodology is presented in the following sections.

2.1 Overview of Kriging Method

The Kriging approach treats the function of interest as a realization of a random function (stochastic process) $y(x)$. For this reason, the mathematical model of Kriging method has been presented as a linear combination of a global model plus its departure:

$$y(x) = f(x) + Z(x), \tag{1}$$

where $y(x)$ is the unknown deterministic response, $f(x)$ is a known function of x , and $Z(x)$ is a realization of a stochastic process with zero mean, σ^2 variance, and having non-zero covariance values. The procedure starts with obtaining a sample data of limited size (i.e. n -design sets each having k -variables), $X = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}^T$, and a corresponding vector of scalar responses, $Y = \{y^{(1)}, y^{(2)}, \dots, y^{(n)}\}^T$. It is assumed that if any two design points, e.g. $x^{(i)}$ and $x^{(j)}$, are positioned close together in the design space, their respective function values $y^{(i)}$ and $y^{(j)}$ are also expected to be similar. Usually, the Latin hypercube technique is used to create initial point, ensuring a diverse set of points along each variable dimension [1].

2.2 Kriging Procedure

Without going to the detailed mathematics, here, we provide the Kriging predictor, as follows:

$$\hat{y}(x) = \hat{\mu} + r(x^*, x)^T R^{-1}(y(x) - \mathbf{1}\hat{\mu}), \quad (2)$$

where $r(x^*, x)$ is the linear vector of correlation between the unknown point x to be predicted and the known sample points x^* . R denote the $n \times n$ matrix with (i, j) whose entry is $Corr[y^{(i)}, y^{(j)}]$, and $\mathbf{1}$ denote an n -vector of ones. The optimal values of μ and $\hat{\sigma}^2$, expressed as function of R are given below[10]:

$$\hat{\mu} = \frac{\mathbf{1}^T R^{-1} y}{\mathbf{1}^T R^{-1} \mathbf{1}}, \quad (3)$$

$$\hat{\sigma}^2 = \frac{(\vec{y} - \mathbf{1}\hat{\mu})^T R^{-1} (\vec{y} - \mathbf{1}\hat{\mu})}{n}. \quad (4)$$

Moreover, Kriging is attractive because of its ability to provide error estimates of the predictor:

$$s^2(x) = \hat{\sigma}^2 \left[1 - r^T R^{-1} r + \frac{(1 - r^T R^{-1} r)^2}{\mathbf{1}^T R^{-1} \mathbf{1}} \right]. \quad (5)$$

2.3 The Expected Improvement Procedure

As described in the previous section, using Kriging technique in optimization requires fitting the Kriging model, finding the point that maximizes expected improvement, evaluating the function at this point, and ultimately iterating. The second step of this procedure is based on the fact that Kriging helps in estimating the model uncertainty and stresses on exploring points where we are uncertain. In order to do this, Kriging method treats the value of the function at x as if it were the realization of a stochastic process $y(x)$, with the mean giving by the predictor $\hat{y}(x)$ and variance $s(x)$, The expected improvement function is given as follows [10, 4]:

$$E[I(x)] = (y_{best} - \hat{y}(x)) \Phi \left(\frac{y_{best} - \hat{y}(x)}{s(x)} \right) + s(x) \phi \left(\frac{y_{best} - \hat{y}(x)}{s(x)} \right), \quad (6)$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the normal cumulative distribution function and probability density function, respectively. It is worth noting here that the implementation of Kriging (DACE) is based on universal Kriging, where it is possible to use other regression models as well.

The decision of to select Kriging method as an approximator depends on various factors. One important factor is the size of the search space. It has been observed that Kriging methodology does not work very well when the number of variables exceeds about 10.

As mentioned above, Kriging and subsequent EGO methodology are standards in the context of single-objective unconstrained problems, but they have not been extended well neither to constrained single-objective problems nor to multi-objective optimization problems. A recent study [12] suggested a way to modify the expected improvement function for constrained problems. It selects the best feasible objective function value instead of y_{best} in Equation 6. This method has been applied only to single objective optimization problems. The expected improvement function is modified as follows:

$$E_c[I(x)] = E[I(x)]\prod_{j=1}^J F_j(x), \quad (7)$$

where $F_j(x)$ is given as follows for j -th constraint:

$$F_j(x) = \begin{cases} 0.5 + 0.5\text{erf}\left(\frac{\bar{g}_j(x)}{s_j(x)}\right), & \text{if } \text{erf}\left(\frac{\bar{g}_j(x)}{s_j(x)}\right) \geq 1, \\ 2 - \text{erf}\left(\frac{\bar{g}_j(x)}{s_j(x)}\right), & \text{if } 0 < \text{erf}\left(\frac{\bar{g}_j(x)}{s_j(x)}\right) < 1, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where $\bar{g}_j(x)$ is the normalized version of j -th constraint function, and s_j is the MSE prediction of the j -th constraint function. It is clear that if any constraint is violated, $E_c[I(x)]$ is zero and for near constraint boundary solutions $E_c[I(x)]$ has a large value, thereby emphasizing near-boundary solutions during simulations, and for points that are well inside the feasible region $E_c[I(x)] = E[I(x)]$. Here, we do not use this method, instead, handle constraint directly through a modeling of the selection function, as discussed in Section 3.

In the following subsection, we provide a few existing studies of Kriging method for multi-objective optimization problems.

2.4 Past Studies on Surrogate-based EMO

ParEGO, proposed by Knowles [13], applies the EGO algorithm and the selected aggregation function randomly for finding a point to evaluate the next point in the search space. The major drawback of the ParEGO algorithm is that it considers one aggregation function at each iteration and it is not able to generate multiple candidate points at one iteration. S-Metric Selection based EGO (SMS-EGO) [14] extends the idea of Emmerich et al. [5] and optimizes the S-metric, a hypervolume-based metric, by the covariance matrix adaptation evolution strategy (CMS-ES) algorithm to decide which point will be evaluated next. Like ParEGO, SMS-EGO evaluates only one single test point at each iteration.

MOEA/D-EGO, proposed by Zhang et al. [2], integrates the EGO algorithm with decomposition based multi-objective EAs (MOEA/D) [15]. MOEA/D is based on conventional aggregation approach. It decomposes an MOP into a number of single-objective optimization subproblems. At each iteration in MOEA/D-EGO, a Gaussian stochastic process model for each subproblem is built based on data obtained from the previous search, and the expected improvement of these subproblems are optimized simultaneously by using MOEA/D procedure for generating a set of candidate solutions. Then, a few of them are selected for

evaluation. This algorithm was applied to unconstrained multi-objective ZDT problems [16] having only eight variables, although the original test problems were defined for 30 variables. It is likely that Kriging method could not approximate a 30-dimensional problem well and the overall method suffered from the curse of dimensionality. Moreover, the method did not use any constraint in the study; hence an extension to multi-objective constrained problems is not clear.

3 Proposed Multi-objective EGO Method

With the brief description of the EGO method for single-objective optimization problems discussed above, we are now ready to provide details of our proposed procedure.

Our proposed approach is a generative multi-objective optimization procedure in which one Pareto-optimal point is determined to be found at a time. The development of a suitable surrogate modeling and its solution are achieved for one targeted Pareto-optimal solution at a time. However, the surrogate building process for multiple Pareto-optimal solutions are not all independent to each other, rather the whole process is interlinked, as described by the following step-by-step procedure.

- Step 1:** Create an archive having H random points based on Latin hypercube sampling (LHS) [1] from the entire variable search space. Evaluate each of these solutions using the objective and constraint functions (referred to as ‘high-fidelity’ evaluations).
- Step 2:** Generate R reference directions on a normalized unit hyperplane in the objective space using Das and Dennis’s method [17]. Then, for each reference direction chosen using a `Diversity_Preserver` procedure, execute the following steps.
- Step 3:** Select K points closest to the reference direction based on projected Euclidean distance of all H archive points in the objective space using a `Points_Selector` procedure.
- Step 4:** Build a local surrogate model (`Surrogate_Model` procedure) using the chosen αH points executed already with high-fidelity evaluations (for example $\alpha = 0.7$).
- Step 5:** Use the developed surrogate model to find the best possible solution of the model by a specific (`Optimization`) procedure.
- Step 6:** Perform a high-fidelity evaluation of the newly created solution and add it to the archive.
- Step 7:** Repeat Steps 3 to 6 p times to add p new solutions to the archive.
- Step 8:** Move to a new reference line according to the `Diversity_Preserver` procedure and Go to Step 3 until all reference lines are considered or another termination criterion is satisfied.

The above step-by-step procedure is a generic surrogate-based EMO procedure that can be applied with any surrogate modeling technique and optimization algorithm. It also has a few other flexibilities which we describe next.

3.1 Diversity_Preserver Procedure

This procedure provides the linking between model building and its solution among several reference lines. Given K reference directions spanning the entire first quadrant of the M -dimensional objective space, this procedure determines the sequence of choosing reference lines. Figure 1 illustrates the proposed procedure in which $H = 20$ random points denote the archive and for each of five reference directions, $K = 4$ points are chosen.

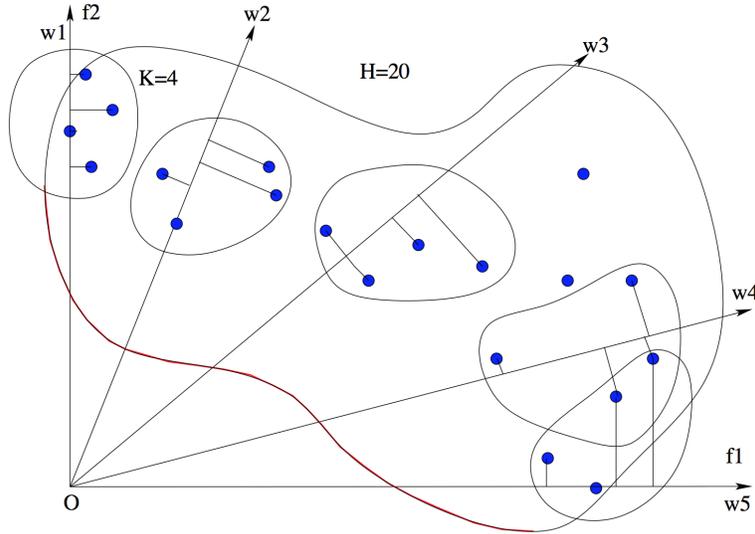


Figure 1: An illustration of the proposed surrogate-based generative EMO procedure

This procedure involves fixing the following three entities:

1. The starting reference direction,
2. The sequence of choosing reference directions, and
3. Repeat pattern of the sequencing.

The surrogate modeling can start anywhere, but either one of the extreme reference directions (lying on one of the objective directions) or in the middle of the objective space (equi-distance from all objective directions) are two unbiased approaches. In all our study here, we consider the former approach of choosing one of the objective axis direction as the initial reference direction.

The second entity is probably the most important factor. Once the initial direction is chosen, a model is built and one or more solutions are obtained using the model, which reference direction to choose next is a relevant question.

In the *neighborhood* approach, we can choose one of the nearest reference direction to the already chosen directions as the next direction. Since there is one or more models already built around a neighboring reference direction in this approach, inclusion of their best solutions in the neighboring model building process should create a better model. However, the flip side is that each model then have a local perspective and the performance of this approach will depend on the mapping between variable and objective spaces. In the

maximum diversity approach, a reference direction which is maximally away from all chosen reference directions is selected for building the next surrogate model. In this approach, initial few models are independent to each other, but together they will span the search space well. Later surrogate models tend to have a more global perspective than those in the neighborhood approach. We use both approaches in this study.

Another aspect of the diversity preserving procedure is whether the above sequencing process needs to be repeated after their first pass on all reference directions! This can be decided adaptively, based on whether a satisfactory set of non-dominated solutions have been found for each reference direction. If it is to be repeated, an exactly the same sequencing or a different sequencing operation can be chosen as well. The repetition need not for all reference directions, the ones that generated a dominated solution can be repeated. For repetition, it will be a good idea to preserve all surrogate models generated in the first pass of the procedure. In this study, we repeat the first pass in the same sequence one more time.

3.2 Points_Selector Procedure

In this procedure, a set of K points will be chosen from the archive having H points. Here, we suggest to compute the orthogonal distance of each archive point to the given reference direction and select K shortest orthogonal distance solutions. But other distance metrics (such as the achievement scalarization function (ASF) value or its augmented version) or Tchebyshev metric or other L_p norms can be used, instead.

3.2.1 Achievement Scalarization Function (ASF)

One of the common ways to solve the generic multi-objective optimization problem is to solve a parameterized achievement scalarization function (ASF) optimization problem repeatedly for different parameter values. The ASF approach was originally suggested by Wierzbicki[18]. For a specified reference point z and a weight vector w (parameters of the ASF problem), the ASF problem is given as follows:

$$\begin{aligned} \text{Minimize}_x \quad & \text{ASF}(x, z, w) = \max_{i=1}^M \left(\frac{f_i(X) - z_i}{w_i} \right), \\ \text{Subject to} \quad & g_j(X) \leq 0, \quad j = 1, 2, \dots, J. \end{aligned} \tag{9}$$

The reference point $z \in R^M$ is any point in the M -dimensional objective space and the weight vector $w \in R^M$ is an M -dimensional unit vector for which every $w_i \geq 0$ and $|W| = 1$. To avoid division by zero, we shall consider strictly positive weight values. It has been proven that for above conditions of z and w , the solution to the above problem is always a Pareto-optimal solution [19]. Figure 2 illustrates the ASF procedure of arriving at a weak or a strict Pareto-optimal solution.

For illustrating the working principle of the ASF procedure, we consider specific reference vector z (marked in the figure) and weight vector w (marked as w in the figure). For any point x , the objective vector f is computed (shown as F). Larger of two quantities $(f_1 - z_1)/w_1$ and $(f_2 - z_2)/w_2$ is then chosen as the ASF value of the point x . For the point G, the above two quantities are identical, that is, $(f_1 - z_1)/w_1 = (f_2 - z_2)/w_2 = p$ (say). For points on line GH, the first term dominates and the ASF value is identical to p . For points on line

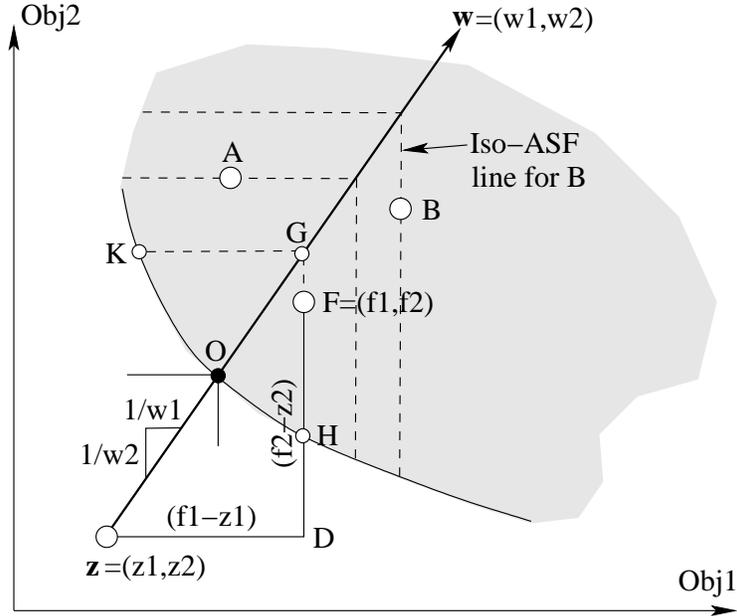


Figure 2: ASF procedure of finding a Pareto-optimal solution is illustrated.

GK, the second term dominates and the ASF value is also identical to p . Thus, it is clear that for any point on lines GH and GK, the corresponding ASF value will be the same as p . This is why an iso-ASF line for an objective vector at F traces any point on lines KG and GH. For another point A, the iso-ASF line is shown in dashed lines passing through A, but it is important to note that its ASF value will be larger than p , since its iso-ASF lines meet at the w -line away from the direction of minimum of both objectives. Since the ASF function is minimized, point F is considered better than A in terms of the ASF value. Similarly, point A will be considered better than point B. A little thought will reveal that when the ASF problem is optimized the point O will correspond to the minimum ASF value in the entire objective space (marked with a shaded region), thereby making point O (an efficient point) as the final outcome of the optimization problem stated in Equation 9 for the chosen z and w vectors. By keeping the reference point z fixed and by changing the weight vector w (treating it like a parameter of the resulting scalarization process), different points on the efficient front can be generated by the above ASF minimization process.

The ASF procedure calculated from an ideal or an utopian point may result in a weak Pareto-optimal solution [20]. To avoid finding weak points, the following augmented ASF computation is used [19]:

$$\begin{aligned}
 \text{Minimize}_x \quad & \text{AASF}(x, z, w) = \max_{i=1}^M \left(\frac{f_i(X) - z_i}{w_i} \right) \\
 & + \rho \sum_{j=1}^M \left(\frac{f_j(X) - z_j}{w_j} \right), \\
 \text{Subject to} \quad & g_j(X) \leq 0, \quad j = 1, 2, \dots, J.
 \end{aligned} \tag{10}$$

Here, the parameter ρ takes a small value ($\sim 10^{-3}$). The additional term on the objective function has an effect of making the iso-AASF lines inclined to objective axes.

3.3 Surrogate Model Procedure

This is one of the main procedures that will affect the performance of the overall algorithm. Here, we use the Kriging methodology [4] to model the selection function used for comparing solutions in the presence/absence of constraints. The selection function is given below:

$$\mathcal{S}(x) = \begin{cases} f(x), & \text{if } x \text{ is feasible,} \\ f_{\max} + \text{CV}(x), & \text{otherwise.} \end{cases} \quad (11)$$

It is assumed that the objective function is minimized and there are J constraints of type $g_j(x) \leq 0$ ($j = 1, 2, \dots, J$). Equality constraints are converted into two inequality constraints of the above type. Here, the parameter f_{\max} is the worst objective function value of all feasible solutions of K solutions used for building the model. The function $\text{CV}(x)$ is the overall constraint violation, defined as follows:

$$\text{CV}(x) = \sum_{j=1}^J \langle \bar{g}_j(x) \rangle, \quad (12)$$

where the bracket operator $\langle \alpha \rangle$ is $-\alpha$ if $\alpha < 0$ and zero, otherwise. The function \bar{g}_j is a normalized version of constraint function g_j [21].

Thus, for a given set of K points to build the Kriging model, the above unconstrained selection function $\mathcal{S}(x)$ for each point x is computed and the standard Kriging modeling technique [4] is used to find the estimator and the mean squared error (MSE) function. Thereafter, the expected improvement function [10] is formulated for the next step.

Here, other surrogate modeling methods can also be used, instead of the Kriging methodology. We are currently working with a few other surrogate models for this purpose.

3.4 Optimization Procedure

Once the surrogate model is built, the next step is to optimize the model to find the best possible solution of the model. For this purpose, we use a real-parameter genetic algorithm (rGA) which uses simulated binary crossover [22] and polynomial mutation operator [20]. The population is started with a random population and rGA is run for 100 number of generations. Here, the rGA population can be initialized by the K points used to build the surrogate model.

Figure 1 illustrates the overall procedure. It can be observed that some of the original archive points may not have been chosen for any of the reference directions, particularly when K is very small compared to H . Also, some points can be chosen for more than one reference directions, particularly when K is comparable to H . Since all H points were already made high-fidelity evaluations, it is better to choose K in a way so that all H evaluated solutions are used for one or more reference directions. Another advantage of the proposed method is that since the points chosen for a surrogate model for a particular reference direction are close to each other, each model will be local to the reference direction and is likely find the respective Pareto-optimal solution more reliably than if the model was a global spanning the entire search space. Since each surrogate model uses a few points to build it, the proposed algorithm is also likely to be computationally faster than global models.

4 Results

In this section, we compare our proposed multi-objective EGO method with MOEA/D-EGO procedure, proposed elsewhere [2]. We mentioned in the previous section, MOEA/D-EGO solved ZDT problems with only eight variables and solved only one test problem (DTLZ2) having three objective functions. Our multi-objective EGO is applied to 30-variable version of the ZDT problems, also to the constrained test problems, and multiple three-objective test problems. The reference direction that used in most of instances test problems is neighborhood approach as we illustrated in Section 3.

The rGA control parameters are set as follows:

- Population size = $10k$, where k is a number of variables.
- Number of generations = 100.
- Crossover probability = 0.9.
- Mutation probability = $1/k$.
- Distribution index for SBX operator = 2.
- Distribution index for polynomial mutation operator = 20.

4.1 Two-Objective Unconstrained Problems

First, we compare our multi-objective EGO with MOEA/D-EGO on ZDT1 and ZDT2 problems for an identical number of variables ($k = 8$). As reported in the original study [2], MOEA/D-EGO requires 200 high-fidelity function evaluations for ZDT1 and ZDT2, while our method takes 164 function evaluations for ZDT1 and 122 function evaluations for ZDT2, respectively, to have almost a similar number of non-dominated solutions, as illustrated in Figures 3 and 4 respectively.

Next, we apply our multi-objective EGO method to solve ZDT1, ZDT2, and ZDT3 with 30 variables (the original size). In our knowledge, this is the first time a meta-modeling method has attempted to solve the original version of ZDT test problems. ZDT1 takes 858 function evaluations with 51 reference directions, ZDT2 takes 276 function evaluations with 21 reference directions, and ZDT3 takes 1,170 function evaluations with 51 reference directions. The Pareto-optimal front of these problems and obtained non-dominated solutions are shown in Figures 5 and 6 (a), respectively. For ZDT6 problem having 10 variables, our method takes 558 function evaluations with 51 reference directions, as illustrated in Figure 6 (b).

ZDT4 problem is multi-modal for any Kriging methodology. The previous study [2] did not show results and our approach was also not able to find solutions close to the true Pareto-optimal front.

4.2 Three-Objective Problems

Regarding the three-objective optimization problems, the MOEA/D-EGO solved the simplest problem (DTLZ2) with only six variables which takes 300 solution evaluations. Our

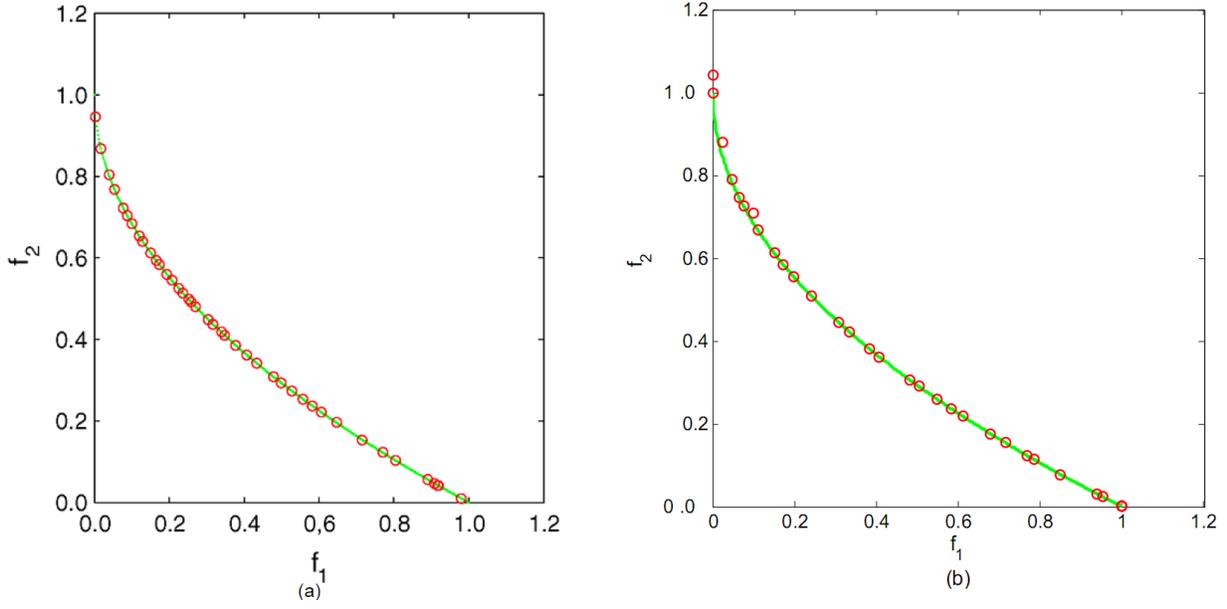


Figure 3: ZDT1 with eight decision variable (a) MOEA/D-EGO, (b) Multi-objective EGO

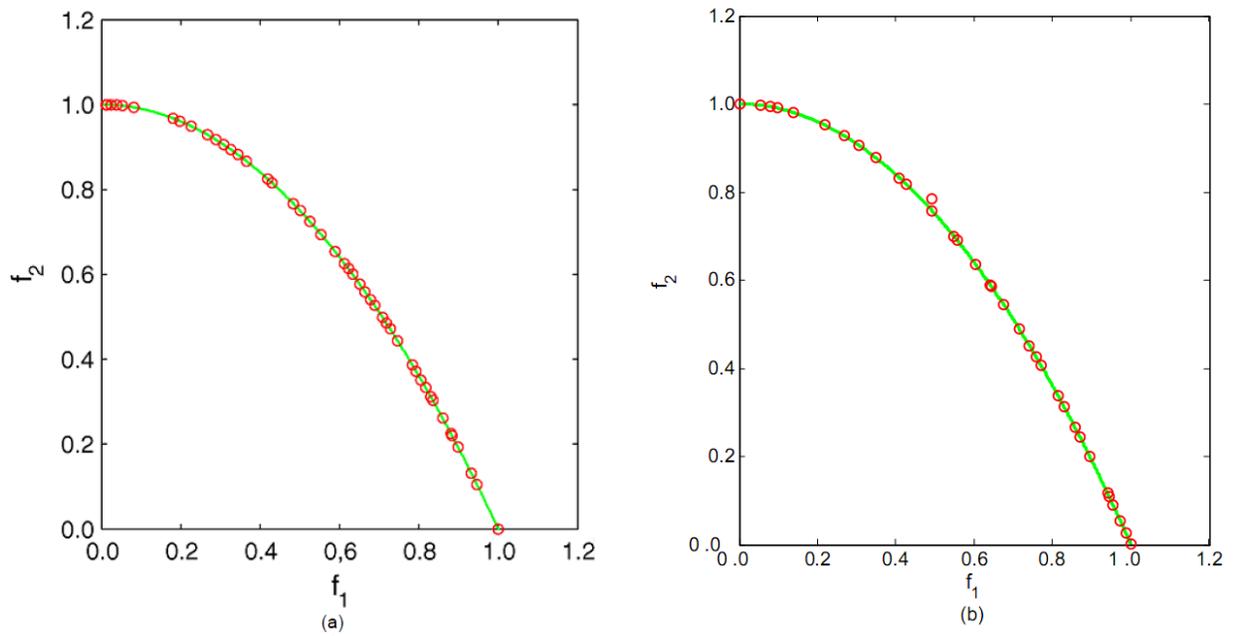


Figure 4: ZDT2 with eight decision variable (a) MOEA/D-EGO, (b) Multi-objective EGO

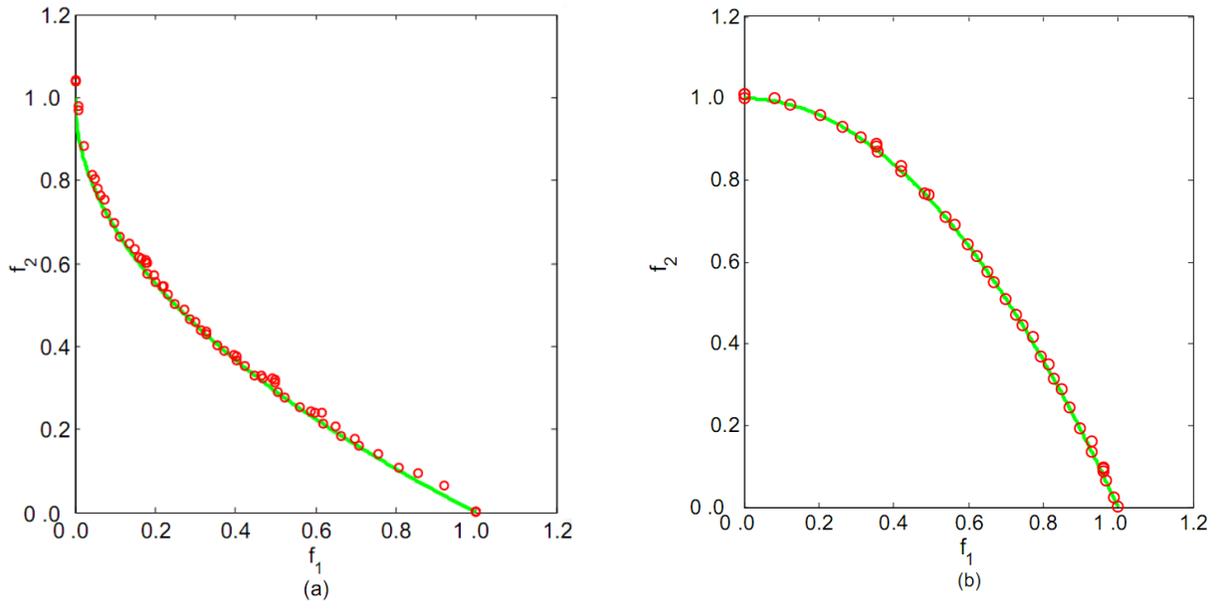


Figure 5: (a) Multi-objective EGO for thirty-variable ZDT1, (b) Multi-objective EGO for thirty-variable ZDT2

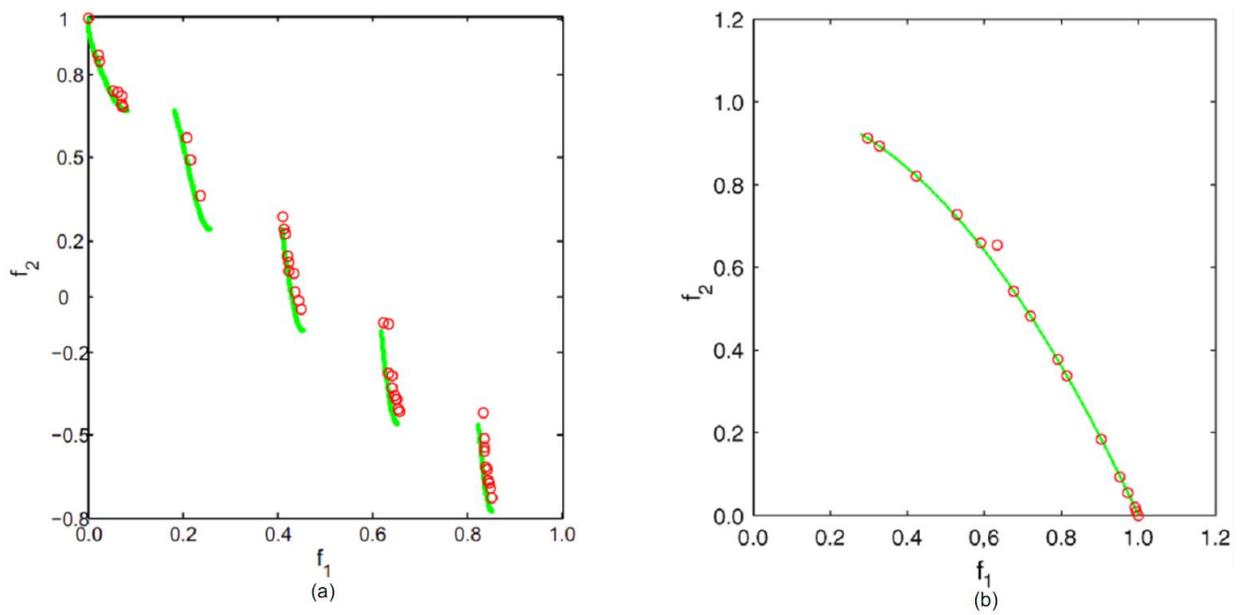


Figure 6: (a) Multi-objective EGO for thirty-variable ZDT3, (b) Multi-objective EGO for ten-variable ZDT6

proposed method used the original size (seven variables) and takes 1,246 high-fidelity solution evaluations with initial sample size of $100k$, as illustrated in Figure 7. The neighborhood approach is used with 91 reference directions.

Our multi-objective EGO method is also applied to solve DTLZ4 and DTLZ5 problems, as shown in Figures 8(a) and (b), respectively. The solution evaluations required for DTLZ4 was 1,610, because of a non-uniform density of solutions across the search space. DTLZ5 required 826 solution evaluations with 21 reference directions. The augmented achievement Scalarization Function (AASF) was used for these test problems to avoid finding the weak Pareto-optimal solutions.

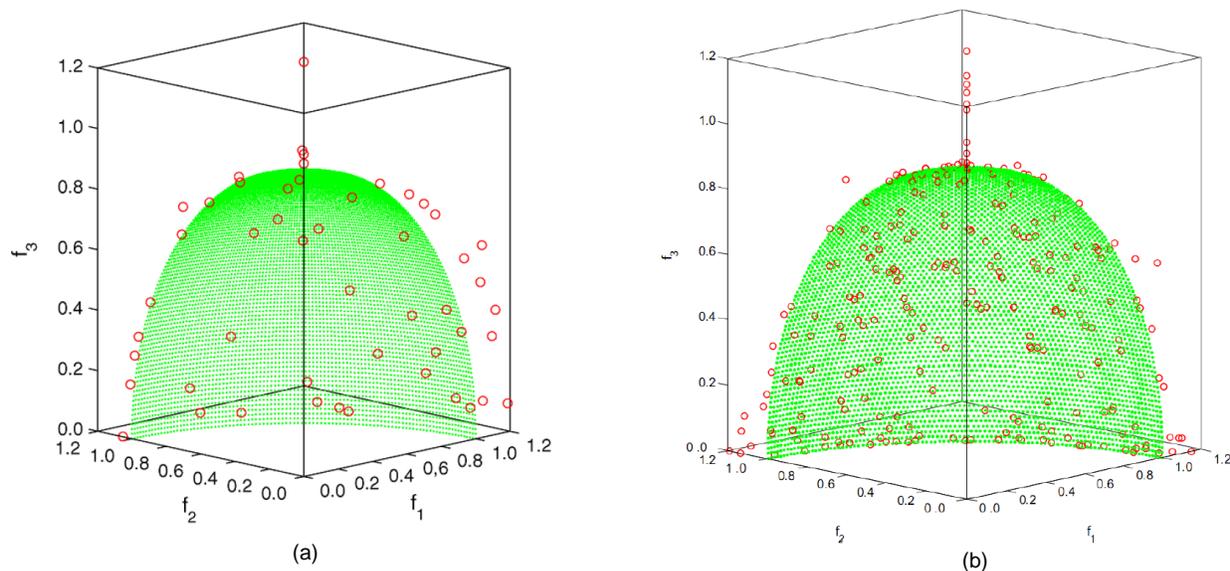


Figure 7: (a) MOEA/D-EGO for six-variable DTLZ2, (b) Multi-objective EGO for seven-variable DTLZ2

4.3 Constrained Problems

Next, we apply our multi-objective EGO method to solve constrained test problems: BNH, SRN, TNK, and OSY. This is the first time, these problems are attempted to be solved using any multi-objective surrogate modeling approach. For more information regarding those problems, refer to [20]. For the BNH problem, our method takes only 288 solution evaluations and the obtained non-dominated solutions are shown in Figure 9(a). The SRN problem requires only 248 solution evaluations, shown in Figure 9(b).

From Figures 10(a) and (b), obtained for the more difficult TNK and OSY problems, it is clear that our method could not converge well to solve them. This is because of the non-linearity and complexity of the variable interactions of the search space in defining feasible and near-optimal solutions. However, the solution set shown in the figures requires only 182 high-fidelity evaluations for TNK and 528 for OSY.

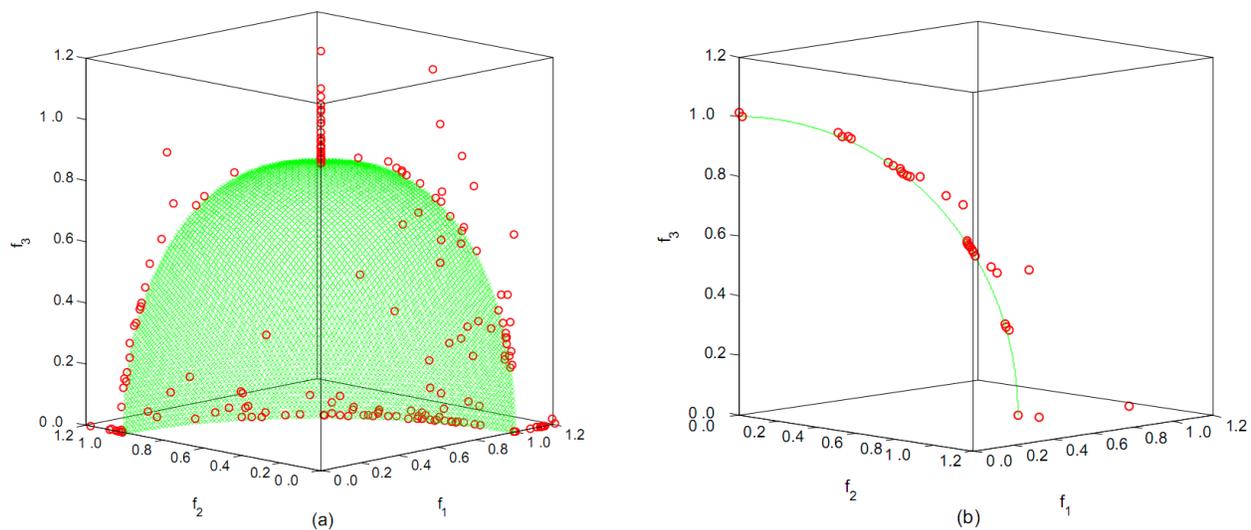


Figure 8: (a) Multi-objective EGO for seven-variable DTLZ4, (b) Multi-objective EGO for seven-variable DTLZ5

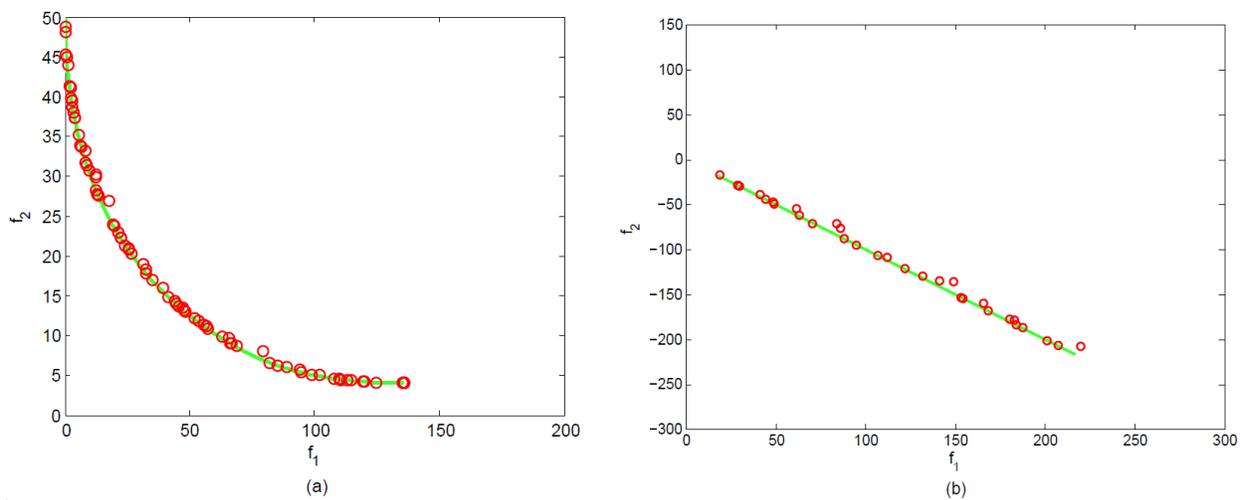


Figure 9: (a) Multi-objective EGO for BNH, (b) Multi-objective EGO for SRN

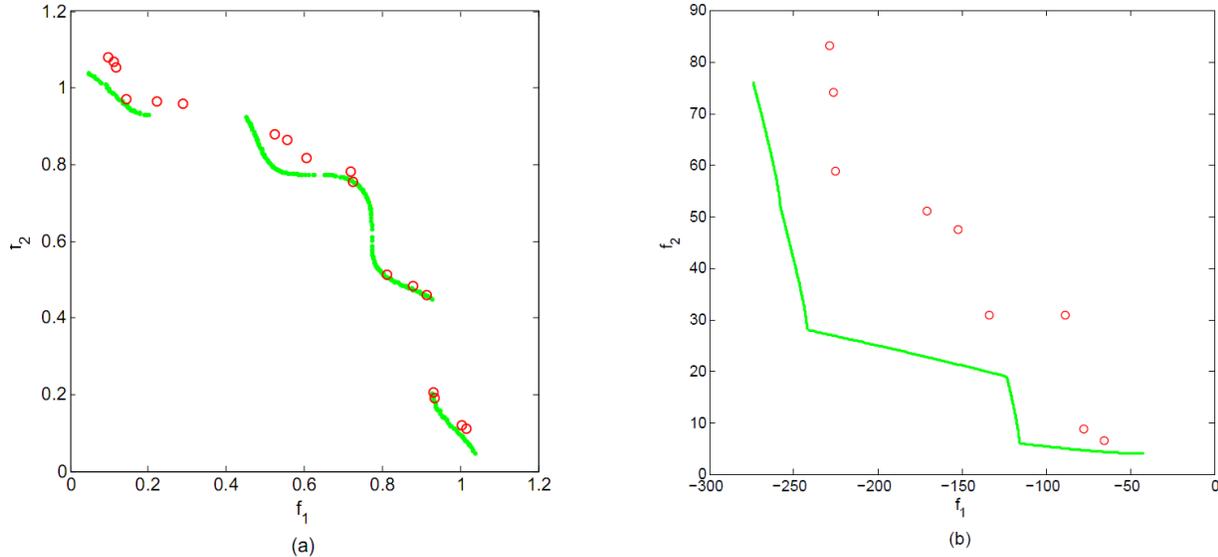


Figure 10: (a) Multi-objective EGO for TNK, (b) Multi-objective EGO for OSY

4.4 Simplified Constraints

In the above simulations, objective function and constraints are used in the Kriging and expected improvement function modeling tasks in a unique parameter-less way. In problems, where both objective function and constraint functions involve computationally expensive procedures, this is the only approach. However, in some practical problems, certain constraints may be simpler to evaluate and may not require the same or any computationally expensive method. In this case, the constraints can be directly used in the rGA method for solving the EI problem.

We resolve two of the difficult constrained problems – TNK and OSY – again by using the constraint functions in the rGA optimization, but the Kriging and EI models approximate the objective function alone. Results are shown in Figure 11(a) and (b), respectively. The problem TNK takes 182 high-fidelity solution evaluations, while OSY takes 408 solution evaluations and the obtained solutions are shown in the figure. A much more converged set of solutions is now found. A comparison of these figures with that of the previous subsection reveals that the Kriging and the EI approach find it difficult to approximate the feasible region adequately. More emphasis in research should be spent in approximating feasible region for surrogate-based constrained multi-objective optimization problems.

4.5 IGD Performance Metric

The inverted generational distance (IGD) [15] is considered as the performance metric for providing a quantitative evaluation of the obtained solutions. The IGD metric provides a combined information about the convergence and diversity of the obtained solutions.

Table 1 presents the IGD values for comparison between MOEA/D-EGO and Multi-objective EGO for ZDT1 and ZDT2 with eight variable decision variables.

Table 2 presents the IGD values for all problems with original variables size, compared

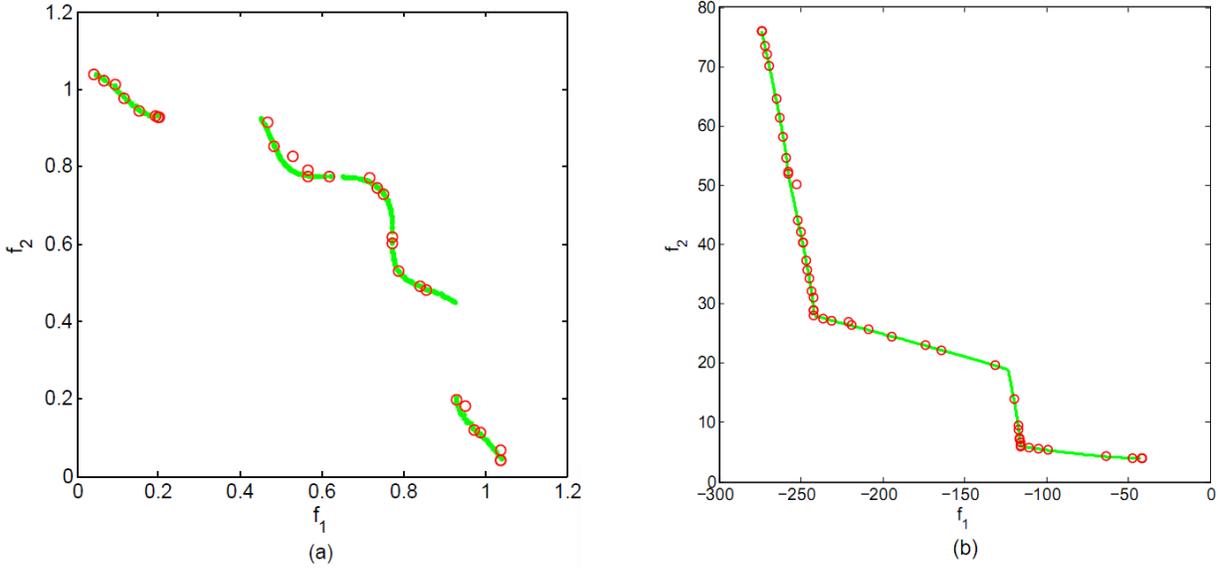


Figure 11: Multi-objective EGO for approximating objective function alone: (a) TNK, (b) OSY.

Table 1: The IGD Performance Metric of MOEA/D-EGO and Multi-objective EGO

Problem	MOEA/D			Multi-objective EGO		
	Best	Mean	Std	Best	Mean	Std
ZDT1_8	0.01200	0.01480	0.00160	0.00071	0.00082	0.00011
ZDT2_8	0.01210	0.01560	0.00220	0.00068	0.00079	0.00007

Table 2: The IGD Performance Metric of Multi-objective EGO

Problems	Lowest	Mean	Std
ZDT1	0.00063	0.00081	0.00012
ZDT2	0.00066	0.00076	0.00008
ZDT3	0.00158	0.00200	0.00026
ZDT6	0.00262	0.00407	0.00188
DTLZ2	0.00052	0.00080	0.00020
DTLZ4	0.00171	0.00186	0.00012
DTLZ5	0.00072	0.00207	0.00071
BNH	0.060012	0.06737	0.00685
SRN	0.07956	0.08418	0.000439
TNK	0.00289	0.00307	0.00018
OSY	0.79962	0.85210	0.03816
TNK-R	0.00120	0.00150	0.00031
OSY-R	0.07511	0.08319	0.00508

against a set of exact Pareto-optimal solutions.

5 Conclusions

Surrogate models are effective in reducing the computational time required to solve single or multi-objective optimization problems. However, there has not been much studies in using them for multi-objective optimization problems. The difficulty lies in finding a set of optimal solutions using one surrogate model, hence a common approach has been to use single-objective surrogate modeling approaches in a generative manner to find one solution at a time.

In this work, we have proposed a generative surrogate modeling procedure for multi-objective optimization in which one model designed for finding a particular Pareto-optimal solution helps in modeling and finding another neighboring Pareto-optimal solution. Our proposed method is generic and is now ready to be tested for other surrogate modeling approaches and for different sequence of building individual models. Our extensive simulation results are compared with an existing study which was limited to small-sized problems. It has been observed that our approach can find a well converged and well diverged set of solutions on the originally proposed large-sized version of the test problems in a few hundreds of solution evaluations. Another hallmark of our approach is that we have proposed a constrained handling method within the surrogate modeling approach that can solve difficult constrained test problems of the multi-objective optimization literature.

We are now extending the proposed approach in many ways: (i) effect of other reference direction sequencing approaches, (ii) effect of other surrogate modeling approaches, such as radial basis neural network and SVM procedures, (iii) application to more complex test problems and practical problems, and (iv) other non-generative, and multi-modal surrogate modeling approaches. Results from these studies will be communicated as they are obtained.

References

- [1] A. Forrester, A. Sobester, and A. Keane, *Engineering design via surrogate modelling: A practical guide*. John Wiley & Sons, 2008.
- [2] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, “Expensive multiobjective optimization by MOEA/D with gaussian process model,” vol. 14, no. 3, pp. 456–474, 2010-06.
- [3] K. Shimoyama, K. Sato, S. Jeong, and S. Obayashi, “Updating Kriging surrogate models based on the hypervolume indicator in multi-objective optimization,” vol. 135, 2013.
- [4] D. R. Jones, “A taxonomy of global optimization methods based on response surfaces,” *Journal of global optimization*, vol. 21, no. 4, pp. 345–383, 2001.
- [5] M. Emmerich, K. Giannakoglou, and B. Naujoks, “Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels,” vol. 10, pp. 421–439, 2006-08.
- [6] D. Krige, “A statistical approach to some basic mine valuation problems on the Witwatersrand,” *Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa*, pp. 119–139, 1951.

- [7] G. Matheron, “Principles of geostatistics,” *Economic Geology*, pp. 1246–1266, 1963.
- [8] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and analysis of computer experiments,” *Statistical science*, pp. 409–423, 1989.
- [9] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, “DACE—A Matlab Kriging toolbox, version 2.0,” Tech. Rep., 2002.
- [10] 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, 1998.
- [11] 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, 2002.
- [12] C. C. Tutum, K. Deb, and I. Baran, “Constrained efficient global optimization for pultrusion process,” *Materials and manufacturing processes*, vol. 30, no. 4, pp. 538–551, 2015.
- [13] J. Knowles, “Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 10, no.1, pp. 50–66, 2006.
- [14] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, “Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection,” in *Parallel Problem Solving from Nature—PPSN X*. Springer, 2008, pp. 784–794.
- [15] Q. Zhang and H. Li, “MOEA/D : A multiobjective evolutionary algorithm based on decomposition,” vol. 11, no. 6, pp. 712–731, 2007.
- [16] K. Deb, “Multi-objective genetic algorithms: Problem difficulties and construction of test problems,” *Evolutionary computation*, vol. 7, no. 3, pp. 205–230, 1999.
- [17] I. Das and J. E. Dennis, “Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems,” *SIAM J. on Optimization*, vol. 8, no. 3, 1998.
- [18] A. P. Wierzbicki, “The use of reference objectives in multiobjective optimization,” in *Multiple criteria decision making theory and application*. Springer, 1980, pp. 468–486.
- [19] K. Miettinen, *Nonlinear multiobjective optimization*. Springer Science & Business Media, 2012, vol. 12.
- [20] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.
- [21] —, “An efficient constraint handling method for genetic algorithms,” *Computer methods in applied mechanics and engineering*, vol. 186, no. 2, pp. 311–338, 2000.

- [22] K. Deb and A. Kumar, “Real-coded genetic algorithms with simulated binary crossover: studies on multimodal and multiobjective problems,” *Complex Systems*, vol. 9, no. 6, pp. 431–454, 1995.