

A Taxonomy for Metamodeling Frameworks for Evolutionary Multiobjective Optimization

Kalyanmoy Deb¹, *Fellow, IEEE*, Rayan Hussein², *Student Member, IEEE*,
 Proteek Chandan Roy³, *Student Member, IEEE*,
 and Gregorio Toscano-Pulido⁴, *Member, IEEE*

Abstract—One of the main difficulties in applying an optimization algorithm to a practical problem is that evaluation of objectives and constraints often involve computationally expensive procedures. To handle such problems, a metamodel is first formed from a few exact (high-fidelity) solution evaluations and then optimized by an algorithm in a progressive manner. However, in solving multiobjective or many-objective optimization problems involving multiple constraints, a simple extension of the idea to form one metamodel for each objective and constraint function may not constitute the most efficient approach. The cumulative effect of errors from each metamodel may turn out to be detrimental for the accuracy of the overall optimization procedure. In this paper, we propose a taxonomy of different plausible metamodeling frameworks for multiobjective and many-objective optimization and provide a comparative study by discussing advantages and disadvantages of each framework. The results presented in this paper are obtained using the well-known Kriging metamodeling approach. Based on our extensive simulation studies on proposed frameworks, we report intriguing observations about the behavior of each framework, which may provide salient guidelines for further studies in this emerging area within evolutionary multiobjective optimization.

Index Terms—Evolutionary multiobjective optimization (EMO), Kriging, metamodel, surrogate model, taxonomy.

I. INTRODUCTION

MOST practical optimization problems face with a common difficulty: objective and constraint functions are computationally expensive to evaluate. No matter how efficient and intelligent an optimization algorithm is, every method must compare a requisite number of solutions from the search space before arriving at a reasonably good solution.

Manuscript received December 23, 2016; revised May 2, 2017, September 16, 2017, December 29, 2017, and February 25, 2018; accepted April 9, 2018. Date of publication April 18, 2018; date of current version January 28, 2019. (*Corresponding author: Kalyanmoy Deb.*)

K. Deb, R. Hussein, and P. C. Roy are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: kdeb@egr.msu.edu; husseinr@egr.msu.edu; royprote@egr.msu.edu).

G. Toscano-Pulido is with the Information Technology Laboratory, CINVESTAV-IPN-Tamaulipas, Victoria 87130, Mexico (e-mail: gtoscano@cinvestav.mx).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. This consists of a PDF file which contains additional results in terms of figures and tables and extension of the metamodeling taxonomy to unconstrained and single-objective problems. This material is 7.79 MB in size.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2018.2828091

While this can be a time consuming process, in most occasions, practitioners cannot wait too long to find such a solution. Although the advent and advances of parallel and distributed computing certainly helps in reducing the overall computational time, algorithmic efficacies are must and unavoidable.

The use of metamodels (or surrogate models) to approximate the functional form of exact objective and constraints by using a few high-fidelity solution evaluations is a common approach [1]. Among various methods, the Kriging method is one of the widely used metamodel, which can provide an estimated function value and also simultaneously provide an error estimate of the approximation [2]. Developments in optimization methods have recently led to an increasing interest in metamodeling efforts. Some researchers have made efforts to classify different metamodeling approaches [2], but most such studies exist in the realm of single-objective optimization, although a few studies are in multiobjective optimization area, which we discuss in the next paragraph. Santana-Quintero *et al.* [3] have classified surrogates according to the type of model used (e.g., Kriging, radial basis function, and polynomial regression). Jin [4] proposed a classification based on the way single-objective evolutionary algorithms incorporated surrogate models. Shi and Rasheed [5] classified different metamodeling approaches according to direct or indirect fitness replacement methods. Díaz-Manríquez *et al.* [6] classified surrogate models depending on the accuracy between metamodel approaches and which approach was best suited for use.

Most metamodeling efforts in multiobjective optimization, so far, seem to have taken a straightforward extension of single-objective metamodeling approaches and did not consider constraints in much details. First, every objective and constraint function is metamodeled independently. Thereafter, a standard evolutionary multiobjective optimization (EMO) methodology is applied to the metamodels, instead of the original objective and constraint functions, to find the nondominated front. In some studies, the above metamodeling-EMO combination is repeated in a progressive manner so that refinement of the metamodels can occur with iterations [7]. However, with the possibilities of a combined constraint violation function that can be formulated by combining violations of all constraints in a normalized manner [8] and a combined scalarized objective function (weighted-sum, achievement scalarization function (ASF), Tchebyshev function, or

others) [9], different metamodeling frameworks can certainly be explored. While the straightforward approach requires construction of many metamodels, the suggested metamodels for combined objective and constraint violations will reduce the number of needed metamodels. However, the flip side is that each metamodel of the combined functions is likely to be more complex having discontinuous, non-differentiable and multimodal landscapes. Thus, the success of these advanced metamodeling frameworks is closely tied with the advancements in the metamodeling techniques and among different optimization methods, evolutionary methods remain as potential candidates for handling such metamodels. A recent study [10] provided a taxonomy of model-based multiobjective optimization mainly for unconstrained problems. The taxonomy is based on modeling of independent or aggregation of objectives and how new solutions are created and selected (in-fill or otherwise) in an algorithm. Based on the taxonomy, authors have classified seven existing metamodeling multiobjective methods (including Pareto-based efficient global optimization (ParEGO) [11], decomposition-based multi-objective evolutionary algorithms with efficient global optimization (MOEA/D-EGO) [7], and S-metric selection with efficient global optimization [12]). Parallel and batch use of the methods were also highlighted and extended [13]. While these advancements are in progress, in this paper, we outline, for the first time, a number of different and interesting metamodeling frameworks for *constrained* multiobjective optimization utilizing combined and aggregated approaches of objectives independently, constraints independently, and objectives and constraints together. Our taxonomy is unified to cover single and multiobjective optimization problems with and without constraints. It includes one framework that requires as many as $(M + J)$ metamodels (where M and J are number of objectives and constraints) to another extreme method that requires only one metamodel. To demonstrate the behavior of each of these metamodeling frameworks, we develop and implement a representative algorithm for each framework and present results by using the Kriging metamodeling approach. The taxonomy proposed here is generic for it to degenerate for single-objective unimodal, multimodal, and importantly constrained optimization problems.

In the remainder of this paper, Section II briefly introduces Kriging predictor. Section III describes the proposed taxonomy to classify the different multiobjective metamodeling frameworks. The proposed frameworks use the so-called in-fill strategy for creating and selecting new solutions. This section also classifies many existing multiobjective metamodeling studies into six categories proposed in this paper. Sections IV–VII present detail description of each of the six frameworks. Thereafter, our extensive results on unconstrained and constrained test problems are presented in Section VIII. No effort is made to maximize each algorithm’s performance by individual parameter setting, but a representative set of results showing performance of each framework is presented in this paper. Finally, salient conclusions of this extensive study and plausible future extensions of this paper are discussed in Section IX.

II. KRIGING PROCEDURE

Without going to the detailed mathematics, here, we provide the Kriging predictor, which we have used here as the acquisition function or the metamodeling function

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

where $r(\mathbf{x}^*, \mathbf{x})$ is the linear vector of correlation between the unknown point \mathbf{x} to be predicted and the known sample points \mathbf{x}^* . R denote the $n \times n$ matrix with (i, j) whose entry is $\text{Corr}[y^{(i)}, y^{(j)}]$, and $\mathbf{1}$ denote an n -vector of ones. The optimal values of $\hat{\mu}$ and $\hat{\sigma}^2$, expressed as function of R are given as follows [14]:

$$\hat{\mu} = \frac{\mathbf{1}^T R^{-1} y}{\mathbf{1}^T R^{-1} \mathbf{1}} \quad (2)$$

$$\hat{\sigma}^2 = \frac{(\bar{y} - \mathbf{1}\hat{\mu})^T R^{-1} (\bar{y} - \mathbf{1}\hat{\mu})}{n}. \quad (3)$$

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

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - r^T R^{-1} r + \frac{(1 - r^T R^{-1} r)^2}{\mathbf{1}^T R^{-1} \mathbf{1}} \right]. \quad (4)$$

In-fill criterion for choosing new point(s) based on the optimization of the metamodel(s) is different for each metamodeling framework proposed below. In-fill criterion for each framework is discussed in its description below.

III. PROPOSED TAXONOMY

Multiple and many-objective optimization problems involve a number of (say, M) objective functions ($f_i(\mathbf{x})$) as a function of decision variables (\mathbf{x}) and a number of (say, J) constraint functions ($g_j(\mathbf{x})$), each as a function of \mathbf{x} . For brevity, we do not consider equality constraints in this paper, but with certain modifications, they can be handled in the same way as discussed here.

In this section, we propose a taxonomy of various frameworks for using metamodeling approach in multiple and many-objective optimization algorithms. Our taxonomy finds six different broad frameworks (M1–M6), as illustrated in Fig. 1. Our approach is based on the cardinality of metamodels for objectives and constraints. In the first framework (M1), all objectives and constraints are metamodeled independently, thereby requiring a total of $(M + J)$ metamodels before a multiobjective optimization approach can be applied. This framework is a straightforward extension of the single-objective metamodeling studies, applied to approximate each objective and constraint functions. Once all such metamodels are constructed, an EMO algorithm can use them to a find one Pareto-optimal solution at a time (like the generative method used in classical optimization literature [9]) and we call this framework M1-1, or they can be used to find a number of Pareto-optimal solutions simultaneously (like in the EMO literature) and we call this framework M1-2.

The next metamodeling framework can approximate an overall estimation function of all constraint violations together as one quantity, thereby reducing the overall number of metamodels to $(M + 1)$. The well-known normalized, bracket-operator-based constraint violation functions [8], [15]

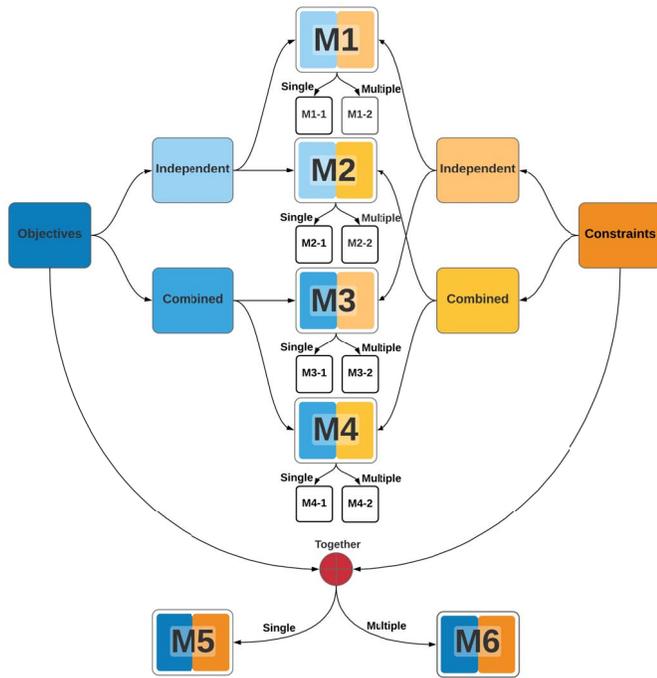


Fig. 1. Proposed taxonomy of six different metamodeling approaches for multiple and many-objective optimization.

can be used for this purpose. Like in M1, the constructed metamodels can also be used to find one Pareto-optimal solution at a time as a generative approach (we call it M2-1) or simultaneously like in an EMO approach (we call it M2-2).

The next metamodeling framework approximates each constraint function independently, but metamodels a combined or aggregated objective function involving all M objectives together, similar to ParEGO approach [11]. Both M3 and ParEGO methods use parameterized scalarization methods to target one optimal point at a time. While ParEGO uses Tchebyshev function as scalarization, our M3 framework uses the ASF, which is identical to Tchebyshev function under certain conditions. M3 uses a real-valued GA to optimize the metamodel, while ParEGO uses a steady-state EA where parents are replaced by better offsprings. While M3-1 proposes to find a single Pareto-optimal solution in one run, thereby requiring multiple runs to generate Pareto-optimal solutions, a multimodal combined landscape, similar to framework M6 described later, but with objective functions alone, can be metamodeled as M3-2.

Then, our fourth classification (M4) requires only two metamodels to be constructed for finding one Pareto-optimal solution, in which one metamodel is for a combined objective function and the second metamodel is made for a combined constraint violation (like in M2 approach). A similar multimodal approach (M4-2) can also be constructed. The frameworks (M1-1, M2-1, M3-1, and M4-1) are ideal for classical point-based optimization algorithms, each requiring multiple applications to find multiple Pareto-optimal solutions. However, frameworks (M1-2, M2-2, M3-2, and M4-2) are ideal for EMO approaches.

A deeper thought will reveal that there could be two more frameworks, in which objectives and constraints are somehow

combined to have a single overall *selection* function which when optimized will lead to one or more Pareto-optimal solutions. In M5, the combined selection function has a single optimum coinciding with a specific Pareto-optimal solution and in M6, the combined selection function is multimodal and makes multiple Pareto-optimal solutions as its optima. Both M5 and M6 frameworks involve a single metamodel in each iteration, but if K Pareto-optimal solutions are to be found, M5 needs to be applied K times, whereas M6 still involves a single multimodal metamodel in finding a set of Pareto-optimal solutions. In EMO algorithms, such as in non-dominated sorting genetic algorithm (NSGA)-II [16], NSGA-III [17], MOEA/D [18], and others, the combined action of the selection operator involving nondomination and niching operations is an ideal way of visualizing the above-mentioned selection function. The advantage of modeling the outcome of the selection function is that it takes into account multiple objectives and constraints in an integrated manner and the selection function can directly emphasize nondominated and less-crowded solutions over dominated and crowded solutions. In this spirit, we believe that M5 and M6 are intricately advantageous for EMO approaches and although there has not been paid much attention, remain as potential and fertile areas for metamodeling-based EMO algorithms.

Thus, it is observed that according to our proposed taxonomy, frameworks M1–M6 require the maximum possible metamodels ($M + J$) to single metamodel in each iteration of the multiobjective metamodeling algorithms. While M6 requires minimum number of metamodels, this does not come free and it is expected that complexity of the metamodels will become increasingly higher from M1 to M6. It then becomes an interesting research task to identify a balance between the number of metamodels and the reduced complexity of metamodels for a particular problem–algorithm combination. In this paper, we do not study the effect of algorithm per se, but present results of a particular algorithmic parameter setting on different problems using all six metamodeling frameworks to illustrate each framework’s potential in addressing different problems. In an application, all M objectives can be clustered into a smaller number of m clusters ($m < M$) and all J constraints can be clustered into smaller number of j clusters ($j < J$). In such a situation, objectives and constraints within each cluster can be combined and then each combined function can be metamodeled using our proposed taxonomy.

On a survey of many existing multiple and many-objective metamodeling studies, we have made a classification of them according to our proposed taxonomy. The majority of the existing studies used M1 framework (in which each objective and constraint function is metamodeled separately) [19], [20] and only a few studies used M2 [21], M3 [22], and M4 [23]. One of our previous initial proof-of-principle study on M5 [24] is a lone study in this category—more details are given in the supplementary material. There does not exist any study implementing our M6 approach, which seems to be an interesting and technically challenging proposition. In this paper, we propose one such framework, although other frameworks are certainly possible.

A little thought will prevail that if the multiobjective optimization problem is unconstrained, frameworks M1 and M2 becomes identical and so are M3 and M4. Interestingly, M3–M5 also become identical to each other.

The proposed taxonomy for multiobjective metamodeling frameworks also degenerates to single-objective problems. In this case, frameworks M1 and M3 are identical and so are M2 and M4. A similar taxonomy can also be derived for finding multiple optima in a single-objective optimization problem, except that M6 framework will now become relevant. Subframeworks M1-1 and M1-2 become relevant in determining whether a single optimum at a time or multiple optima simultaneously, respectively, would be found. Similarly, subframeworks M2-1 and M2-2 will also be relevant in this case. For brevity, we do not present the respective diagram for single-objective, multimodal optimization here.

We now provide further description and one representative algorithm of each of the six multiobjective frameworks in the following section.

IV. FRAMEWORKS M1 AND M2

In these two frameworks, each objective is independently modeled, but constraints are either independently modeled or modeled together. Thus, any classical generative or any EMO algorithm can be run using the objective and constraint metamodels, once they are constructed. The difference between M1 and M2 is that in the latter, one overall constraint violation function is metamodeled. The following constraint violation function $CV(\mathbf{x})$ [8], which accumulates violation of each constraint function ($g_j(\mathbf{x}) \geq 0$) is used in this paper:

$$CV(\mathbf{x}) = \sum_{j=1}^J \langle \bar{g}_j(\mathbf{x}) \rangle \quad (5)$$

where the bracket operator $\langle \chi \rangle$ is $-\chi$ if $\alpha < 0$ and zero, otherwise. The function \bar{g}_j is a normalized version of constraint function g_j [25]. In this paper, we have used NSGA-II [16] as the EMO algorithm throughout, although any other algorithm could have been used. Direct fitness replacement (DFR) [5] has been one of the most straightforward methods to embed surrogate models into MOEAs. DFR assumes that solutions assessed in the surrogate models are comparable to those assessed by the real function (high-fidelity function evaluations). DFR is further subdivided into three major model managements [5].

- 1) No evolution control, which evaluates the MOEA's generated solutions in the surrogate model exclusively (this models trains the surrogate model before the execution of the MOEA).
- 2) Fixed evolution control (FEC), which only some generations or some individuals are evaluated in the surrogate model while the remaining population is evaluated using the real test function.
- 3) Adaptive evolution control, which avoids any possible poor parameter tuning by the use of an adaptive control that adjusts the number of solutions that will be evaluated in the surrogate model.

For the sake of simplicity and performance, we have adopted DFR-FEC model to implement M1 and M2 frameworks.

Furthermore, NSGA-II's Pareto dominance and population-based nature are able to find multiple solutions in a single run, easing the implementation of frameworks M1-2 and M2-2. Thus, we have not used M1-1 and M2-1 in this paper.

The description of M1-2 and M2-2 frameworks is given as follows. The metamodeling algorithm starts with an archive of initial population created using the Latin hypercube method on the entire search space. Then, metamodels are constructed for all M objectives ($f_i(\mathbf{x}), i = 1, 2, \dots, M$). For M1, each constraint function is metamodeled separately and for M2, one constraint violation function ($CV(\mathbf{x})$) is metamodeled. Then, NSGA-II procedure is run for τ generations with these metamodels. Each nondominated solution after τ generations of an NSGA-II run is considered as an in-fill point and is included in the archive. New metamodels are created again using the archive and the process is repeated until termination. A basic structure of frameworks M1-2 and M2-2 are outlined in Algorithm 1. For the use of M1-1 and M2-1, instead of an EMO, a single-objective optimization algorithm can be used repeatedly.

V. FRAMEWORKS M3 AND M4

In these two frameworks, we transform the multiobjective optimization problem into a number of parameterized single-objective optimization problems. We use the ASF [26] using a set of H reference points $\mathbf{z}^{(h)}$ and a corresponding reference direction \mathbf{w} , identical for every \mathbf{z} . The reference direction is an equally angled direction from each objective axis ($\mathbf{w} = (1, 1, \dots, 1)^T / \sqrt{M}$). Reference points $\mathbf{z}^{(h)}$ for $h = 1, 2, \dots, H$ are initialized as equi-spaced points on a unit hyperplane making equal angle to each objective axis. In this paper, we have used Das and Dennis's method to create H equi-spaced points on the hyperplane, but any other method or any other biased set of points, if desired, can also be used. Objective values of solutions are normalized ($\hat{f}_i(\mathbf{x})$) using the population-maximum and population-minimum objective values so that reference points on the normalized hyperplane ($z_i^{(h)} \in [0, 1]$ for all i and h) can be compared with the normalized objective values of the population members. The ASF formulation is given as follows:

$$ASF(\mathbf{x}) = \max_{i=1}^M \frac{f_i(\mathbf{x}) - z_i^{(h)}}{w_i}. \quad (6)$$

In M3 and M4, we construct one metamodel $ASF(\mathbf{x})$, instead of constructing one metamodel for each objective function $f_i(\mathbf{x})$ independently. In M3, each constraint function $g_j(\mathbf{x})$ is modeled separately, but in M4, only the overall constraint violation function $CV(\mathbf{x})$ is metamodeled, as described in (5). Since a parameterized scalarization of multiple objectives are used in both M3-1 and M4-1, we use a single-objective evolutionary optimization algorithm (real-coded genetic algorithm (RGA) [15]). The RGA uses a penalty parameter-less approach [25] to handle constraints. Like in M1 and M2, both algorithms start with an archive of randomly created solutions created using the Latin hypercube method. Each archive member is then evaluated exactly and then suitable metamodels are constructed for a specific objective scalarization parameter values after every τ generations. Constraint functions are

Algorithm 1: Frameworks M1 and M2

Input : Objectives: $(f_1, \dots, f_M)^T$, constraints: $(g_1, \dots, g_J)^T$, n (variables), ρ (sample size), τ (EMO generations per metamodel), SE_{\max} (maximum high fidelity solution evaluations), EMO (multi-objective evolutionary algorithm), μ (EMO's population size), Γ (other parameters of EMO), CV (constrained violation function),

Output: P_T

```

1  $t \leftarrow 0$ 
2  $k \leftarrow (t \% \tau)$ 
3  $P_t \leftarrow \emptyset$ 
4  $\hat{P}_k \leftarrow \text{LHS}(\rho, n)$  // Initialize surrogate model's training set
5  $eval \leftarrow 0$ 
6 while  $eval < SE_{\max}$  do
7   if  $(t \bmod \tau) = 0$  then
8      $\hat{P}_{k+1} \leftarrow \hat{P}_k \cup P_t$ 
9      $\hat{F}_{k+1}^m \leftarrow f_m(\hat{P}_{k+1}), \forall m \in \{1, \dots, M\}$  // high-fidelity evaluations (functions)
10     $\hat{C}_{k+1}^j \leftarrow \text{ctr}_j(\hat{P}_{k+1}), \forall j \in \{1, \dots, J\}$  // high-fidelity evaluations (constraints)
11     $eval \leftarrow eval + \rho$ 
12     $\mathcal{F}_{k+1}^m \leftarrow \text{Create\_Surrogate\_Model}(\hat{F}_{k+1}^m), \forall m \in \{1, \dots, M\}$  // Surrogate independently each objective function
13    if M1 then
14       $\hat{C}_{k+1}^j \leftarrow \text{Create\_Surrogate\_Model}(\hat{C}_{k+1}^j), \forall j \in \{1, \dots, J\}$  // Surrogate independently each constraint
15    else if M2 then
16       $\hat{V}_{k+1} \leftarrow \text{CV}(\hat{C}_{k+1}^j, \forall j \in \{1, \dots, J\})$ 
17       $\hat{C}_{k+1} \leftarrow \text{Create\_Surrogate\_Model}(\hat{V}_{k+1})$ 
18    if  $k = 0$  then
19       $P_t \leftarrow \text{LHS}(\mu, n)$  // Initialize EMO's population
20    else
21       $P_t \leftarrow \text{filter best } \mu \text{ solutions from } \hat{P}_{k+1}$ 
22     $k \leftarrow k + 1$ 
23     $P_{t+1} \leftarrow \text{EMO}(\mathcal{F}_k, \hat{C}_k, P_t, \Gamma)$ ; // Optimize surrogate model
24     $t \leftarrow t + 1$ 
25 return  $P_T \leftarrow \text{Non-dominated solutions } \cup \hat{P}_k$ 

```

metamodeled independently (framework M3) or together as an overall constraint violation (framework M4). The metamodels are then optimized using RGA and the obtained best solution is used as in-fill point and the solution is included in the archive for next metamodeling task. Due to similarities of M3-2 and M4-2 with M6, we defer the discussion on these two frameworks until Section VII.

For frameworks M3-1 and M4-1 (and M5), since every metamodeling effort results in a formulation that is expected to make one specific Pareto-optimal solution (say, $\mathbf{x}^{(*,h)}$) as the target, there is an important aspect about the sequence of the parameterized formulations which we discuss next. In total, there are H different scalarizations each focusing in finding a single Pareto-optimal point $\mathbf{x}^{(*,h)}$. Each scalarization may

construct a new metamodel at every τ generations in order to progressively approach the respective Pareto-optimal solution. However, the sequence of choosing consecutive scalarization will play an important role in the success of the overall procedure. In one approach (neighborhood sweep approach), the first scalarization targets one extreme Pareto-optimal solution. Once new and improved solutions near it are found, the next scalarization will target a neighboring Pareto-optimal solution. Hence, having a number of near-optimal solutions from previous scalarizations will allow the overall procedure to be more efficient and implicitly parallel. Also, getting optimal solutions from one extreme part of the Pareto-optimal front may allow a better normalization of objectives, which is an essential part of any EMO algorithm. To find and use all extreme points from the beginning of a run, in another approach (diverse sweep approach), after every scalarization is applied, the next scalarization may consider a new reference point that is maximally away along the reference plane from all the reference points that have already been considered. Both above approaches have merits of their own and a mixed approach may be better. In this paper, we only adopt the neighborhood sweep approach.

For each scalarization, the RGA is started with a metamodel created with α proportion of sample points close to the reference line passing through the specific reference point in the objective space. This is because points far away from the focal region of metamodeling do not contribute much to the generated metamodeling function and also less points for metamodeling help reduce computational time. Then, the RGA is applied κ times, to take care of the inherent stochasticities of the RGA procedure. The parameters $\alpha = 0.7$ and $\kappa = 5$ are observed to perform well experimentally and kept fixed for all problems of this paper. Again, every RGA solution is included in the archive to make a new metamodel before a new RGA run is performed. After making one pass of consecutive scalarizations involving all reference points, the process is repeated in reverse order one time to make more refined metamodeling of initial scalarizations. The basic structure of M3 and M4 is outlined in Algorithm 2. For M3-2 and M4-2, RGA must be replaced with a multimodal RGA, similar to one described in Section VII.

VI. FRAMEWORK M5

Frameworks M1–M4 are straightforward extensions of single-objective metamodeling frameworks used in the context of evolutionary algorithms. To take care of multitude of objectives and constraints, each is categorized the way the objectives and/or constraints are metamodeled separately or in a combined manner. Frameworks M5 and M6 proposes a more direct metamodeling approach which not only reduces the cardinality of distinct metamodeling efforts, it also algorithm specific. A metamodel of the outcome of an algorithm's selection operation is directly constructed here. We discuss M5 here and M6 will be discussed in the next section.

The focus of M5 is to find a single Pareto-optimal solution at a time by using a parameterized scalarization of all objective functions. However, instead of constructing metamodels

Algorithm 2: Frameworks M3 and M4

Input : Objectives: $(f_1, \dots, f_M)^T$, constraints: $(g_1, \dots, g_J)^T$, n (variables), ρ (sample size), SE_{\max} (total high-fidelity evaluations), RGA (real-parameter genetic algorithm), Γ (parameters of RGA), R (reference direction set), κ (number of points created for each reference direction), α (fraction of samples used for metamodel), **ASF** (scalarization function), CV (constrained violation function)

Output: P_T

```

1  $P \leftarrow \text{LHS}(\rho, n)$  // initialization with Latin Hypercube Sampling
2  $F \leftarrow f_m(P), \forall m \in \{1, \dots, M\}$  // high-fidelity evaluations (functions)
3  $C \leftarrow g_j(P), \forall j \in \{1, \dots, J\}$  // high-fidelity evaluations (constraints)
4  $eval \leftarrow \rho$  // number of function evaluations
5 while  $eval < SE_{\max}$  do
6   for  $r \in R$  do
7     // for each reference direction  $r$ 
8      $P_r \leftarrow \text{Sort } P$  according to distance from  $r$  and select nearest  $\alpha\rho$  solutions
9      $\mathcal{F}_r \leftarrow \text{Create\_Surrogate\_Model}(\text{ASF}(r, P_r))$ 
10    if M3 then
11       $C_r^j \leftarrow \text{Create\_Surrogate\_Model}(C^j(P_r), \forall j \in \{1, \dots, J\})$ 
12    else if M4 then
13       $V_r \leftarrow \text{CV}(C^j(P_r), \forall j \in \{1, \dots, J\})$ 
14       $C_r \leftarrow \text{Create\_Surrogate\_Model}(V_r)$ 
15    for  $i = 1$  to  $\kappa$  do
16       $\mathbf{x}_r \leftarrow \text{RGA}(\mathcal{F}_r, C_r, \Gamma)$  // returns the best found solution
17       $f_{\mathbf{x}_r}^m \leftarrow f_m(\mathbf{x}_r), \forall m \in \{1, \dots, M\}$  // Evaluate objectives of  $\mathbf{x}_r$ 
18       $c_{\mathbf{x}_r}^j \leftarrow g_j(\mathbf{x}_r), \forall j \in \{1, \dots, J\}$  // Evaluate constraints of  $\mathbf{x}_r$ 
19       $P \leftarrow P \cup \{\mathbf{x}_r\}$ 
20       $F \leftarrow F \cup [f_{\mathbf{x}_r}^1, \dots, f_{\mathbf{x}_r}^M]$ 
21       $C \leftarrow C \cup [c_{\mathbf{x}_r}^1, \dots, c_{\mathbf{x}_r}^J]$ 
22       $eval \leftarrow eval + 1$ 
23      if  $eval \geq SE_{\max}$  then
24        Break out of all loops
25 return  $P_T \leftarrow$  Non-dominated solutions of  $P$ 

```

for the scalarized objective function and constraint functions separately, as was done in M3 and M4, here, the combined effect of an EMO's selection operation is metamodeled. For example, while comparing two solutions A and B in an EMO for a particular scalarized problem, say with ASF having a specific \mathbf{z} and \mathbf{w} , the operator computes ASF values for both A and B and then the winner is selected using the constraint-domination principle [15]. We can then directly construct a metamodel of the resulting ASF and use as a single-objective optimization problem. In this approach, we formulate the following combined selection function ($S(\mathbf{x})$) by considering all objective functions and all constraint functions together, but the final focus is to make the global optimum of S -function as one of the targeted Pareto-optimal solution

$$S(\mathbf{x}) = \begin{cases} \text{ASF}(\mathbf{x}), & \text{if } \mathbf{x} \text{ is feasible} \\ \text{ASF}_{\max} + \text{CV}(\mathbf{x}), & \text{otherwise.} \end{cases} \quad (7)$$

Here, the parameter ASF_{\max} is the worst ASF function value of all feasible solutions of the archive. After metamodeling the above selection function S , we formulate the expected improvement (EI) function [14] and optimize the EI function using an RGA. The best solution is then used as an in-fill point and a new metamodel of S is created using the new archive.

Clearly, other scalarization approaches, such as weighted-sum function or epsilon-constraint function or a generic Tchebyshev function [9] can also be used. This methodology is a generative multiobjective optimization procedure in which one Pareto-optimal point is determined to be found at a time [24]. By changing the reference point \mathbf{z} one at a time and keeping the reference direction \mathbf{w} fixed and by using the neighborhood sweep method of scalarization, as described in the previous section, M5 can generate a number Pareto-optimal solutions. The algorithm is outlined in Algorithm 3.

VII. FRAMEWORK M6

Framework M6 culminates into an ambitious metamodeling procedure in which only one metamodel is required to find multiple Pareto-optimal solutions in a single run. On the face of it, this may sound an approach which is too good to be true, a little thought will reveal that if by any procedure we are able to construct a multimodal selection function $M_S(x)$ having H Pareto-optimal solutions as multiple distinct global optimum of it, then we can employ a multimodal (niching-based [27]), single-objective evolutionary algorithm to find and capture multiple optima in a single simulation run. In this paper, we suggest one such approach based on a recently developed theoretical performance metric for multiobjective optimization [28], but certainly other approaches (such as recently proposed multimodal selection function in MEMO [29]) are possible.

Based on Karush–Kuhn–Tucker (KKT) optimality conditions, the first author and his students have recently developed a performance metric called the KKT proximity measure (KKTTPM) [30] which makes a monotonous increase in KKTTPM values for an increase in domination level of solutions. An interesting aspect of KKTTPM is that its value is zero for all Pareto-optimal solutions and it takes a positive value as a solution gets more and more dominated. Such a property motivates us to use KKTTPM as a potential multimodal selection function for the purpose of developing an M6 framework. However, we employ a discretization procedure by which we convert theoretical KKTTPM function having infinitely many Pareto-optimal solutions to a discrete function having a finite set of solutions as distinct optima of the resulting multimodal problem, so that a multimodal EA can be used to find them. Also, KKTTPM considers all objectives and constraint satisfaction into its computational procedure, thereby making it an ideal M_S -function for our purpose.

First we create a set of sample points from the variable space by using the Latin hypercube approach and then compute their objective values by high-fidelity computations. We then compute KKTTPM value of each sample point. KKTTPM function on the entire search space is then metamodeled using the Kriging procedure. Thereafter, a multimodal RGA (M-RGA)

TABLE I
NUMBER OF METAMODELS NEEDED FOR EACH OF THE SIX METAMODELING APPROACHES OF THIS PAPER

Framework	M1-1	M1-2	M2-1	M2-2	M3	M4	M5	M6
Objective Handling	Independent	Independent	Independent	Independent	Combined	Combined	Combined	Combined
Constrained Handling	Independent	Independent	Combined	Combined	Independent	Combined	Combined	Combined
# of Selection Function (s)	M+J	M+J	M+1	M+1	1+J	2	1	1
# of Metamodel (s)	$H\kappa(M+J)$	$\kappa(M+J)$	$H\kappa(M+1)$	$\kappa(M+1)$	$H\kappa(1+J)$	$H\kappa(2)$	$H\kappa$	κ
# of Solution (s)	One	Many	One	Many	One	One	One	Many

Algorithm 3: Framework M5

Input : Objectives: $(f_1, \dots, f_M)^T$, constraints: $(g_1, \dots, g_J)^T$, n (variables), ρ (sample size), SE_{\max} (total high-fidelity evaluations), RGA (real-parameter genetic algorithm), Γ (parameters of RGA), EI (efficient global optimization), R (reference direction set), κ (number of points created for each reference direction), α (fraction of samples used for metamodel), ASF (scalarization function), CV (constrained violation function)

Output: P_T

```

1 P ← LHS( $\rho, n$ ) // initialization with Latin Hypercube Sampling
2 F ←  $f_m(P), \forall m \in \{1, \dots, M\}$  // high-fidelity evaluations (functions)
3 C ←  $g_j(P), \forall j \in \{1, \dots, J\}$  // high-fidelity evaluations (constraints)
4 eval ←  $\rho$  // number of function evaluations
5 while eval <  $SE_{\max}$  do
6   for  $r \in R$  do
7     // for each reference direction  $r$ 
8      $P_r \leftarrow$  Sort P according to distance from  $r$  and select nearest  $\alpha\rho$  solutions
9      $P_r^s \leftarrow$  Feasible( $P_r$ ) // find feasible solutions of  $P_r$ 
10     $P_r^u \leftarrow P_r \setminus P_r^s$  // locate infeasible solutions of  $P_r$ 
11     $Fitness_r^s \leftarrow$  ASF( $r, P_r^s$ ) // fitness of feasible solutions
12     $V_r^u \leftarrow$  CV( $C^j(P_r^u), \forall j \in \{1, \dots, J\}$ ) // constraint violation function
13     $Fitness_r^u \leftarrow V_r^u + (\max_{\forall P_r^u} Fitness_r^s)$  // fitness of infeasible solutions
14     $\mathcal{F}_r \leftarrow$  Create_Surrogate_Model( $Fitness_r$ ) // Surrogate model for constrained ASF function model
15    for  $i = 1$  to  $\kappa$  do
16       $\mathbf{x}_r \leftarrow$  RGA( $\mathcal{F}_r, EI, \Gamma$ ) // returns the best found solution
17       $f_{\mathbf{x}_r}^m \leftarrow f_m(\mathbf{x}_r), \forall m \in \{1, \dots, M\}$  // Evaluate objectives of  $\mathbf{x}_r$ 
18       $c_{\mathbf{x}_r}^j \leftarrow g_j(\mathbf{x}_r), \forall j \in \{1, \dots, J\}$  // Evaluate constraints of  $\mathbf{x}_r$ 
19       $P \leftarrow P \cup \{\mathbf{x}_r\}$ 
20       $F \leftarrow F \cup [f_{\mathbf{x}_r}^1, \dots, f_{\mathbf{x}_r}^M]$ 
21       $C \leftarrow C \cup [c_{\mathbf{x}_r}^1, \dots, c_{\mathbf{x}_r}^J]$ 
22      eval ← eval + 1
23      if eval ≥  $SE_{\max}$  then
24        Break out of all loops
25 return  $P_T \leftarrow$  Non-dominated solutions of P

```

possible solution for each reference direction. While in M6, a M-RGA is employed to search the metamodeled KKTPM function for finding new and multiple multimodal solutions. In our proposed approach, we set H reference directions like in NSGA-III approaches [17] or in MOEA/D [18]. Thereafter, each obtained multimodal point which is associated with a specific reference line based on its closeness (orthogonal normalized distance) to the line in the objective space is used as an in-fill point. These closest high-fidelity solutions are called leaders. In each iteration of M-RGA, for each offspring population member, the nearest leader (computed using normalized Euclidean distance in the *variable* space) is identified. Then, the offspring population member is associated with the same reference line as its nearest leader. All members associated with a reference line are declared to lie on the same cluster. Then, the selection operation is restricted within the same cluster solutions in M-RGA. The first parent is chosen at random, but care is given to ensure every cluster is considered one at a time. Thereafter, the second parent choice is not at random, rather a solution from the same cluster is chosen at random for comparison using their KKTPM values. If no other solution is found in the same cluster, the first parent becomes an automatic winner of the selection process. This restriction of selection operator among similar population members in variable space will eventually form multiple niches for different reference lines within the population. At the end of a M-RGA run, the best metamodeled KKTPM value in the cluster of each reference line is saved to the archive. Thus, the M-RGA run sends multiple well-diversified solutions having small KKTPM values for high-fidelity evaluations. This process is continued until the termination of the overall M6 algorithm. The algorithm is outlined in Algorithm 4. For M3-2 and M4-2, the above multimodal RGA can be used with metamodeled combined objective function and single or multiple metamodeled constraint functions, instead of a single combined selection function used in M6.

Based on the above detail description of six different meta-modeling frameworks, we now summarize the number of metamodels needed for each framework in Table I. Recall that M , J , H , and κ are the number of objectives, constraints, desired number of Pareto-optimal solutions, and number of solutions created from a single metamodel, respectively. Clearly, M1-1 requires maximum number of meta-modeling constructions ($H\kappa(M+J)$); M2-1 requires $H\kappa(1+J)$; M3 requires $H\kappa(M+1)$; M4 needs $H\kappa(2)$; M5 requires $H\kappa$; and M6 requires κ metamodels. While M1-1 needs the maximum number of meta-modeling efforts, M6 requires the least.

While the cardinality reduces from M1 to M6, in general, the relative complexity of the resulting landscapes is likely to increase from M1 to M6. For a given set of objective

is employed to search the metamodeled KKTPM function for finding new and multiple multimodal solutions. In M5, RGA optimizes a single-objective metamodel to find the best

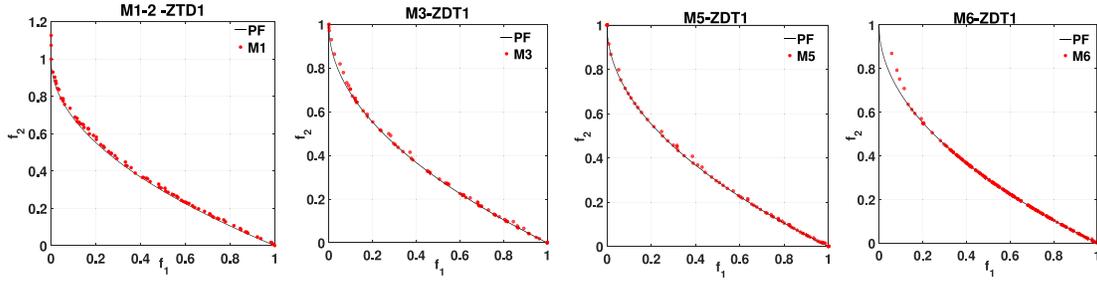


Fig. 2. Obtained nondominated solutions for problem ZDT1 using frameworks M1, M3, M5, and M6 from left to right.

Algorithm 4: Framework M6

Input : Objectives: $(f_1, \dots, f_M)^T$, constraints: $(g_1, \dots, g_J)^T$, n (variables), ρ (sample size), SE_{\max} (total high-fidelity evaluations), M-RGA (multi-modal real-parameter genetic algorithm), Γ (parameters of M-RGA), R (reference direction set), $KKTPM$ (multi-modal constraint function)

Output: P_T

```

1 P ← LHS( $\rho, n$ ) // initialization with Latin
  Hypercube Sampling
2 F ←  $f_m(P), \forall m \in \{1, \dots, M\}$  // high-fidelity
  evaluations (functions)
3 C ←  $g_j(P), \forall j \in \{1, \dots, J\}$  // high-fidelity
  evaluations (constraints)
4 eval ←  $\rho$  // number of function evaluations
5 while eval <  $SE_{\max}$  do
6   Lp ← Cluster ID of each solution  $\mathbf{x} \in P$  according to
     distance from  $r$ 
7   Fitness ←  $KKTPM(F, C)$  //  $KKTPM$  for each
     solution
8    $\mathcal{F}$  ← Create_Surrogate_Model(Fitness) // Surrogate
     model for constrained  $KKTPM$  model
9   X ← M-RGA( $\mathcal{F}, Lp, \Gamma$ ) // returns multiple
     optimized solutions, one for each
     reference line; niching is performed
     in  $\mathbf{x}$ -space with Lp
10  if  $|X| + eval > SE_{\max}$  then
11    X ← X(1:( $SE_{\max} - eval$ )) // Choose best
      ( $SE_{\max} - eval$ ) metamodeled  $KKTPM$ 
      solutions
12   $F_X^m \leftarrow f_m(X), \forall m \in \{1, \dots, M\}$  // Evaluate
     objectives of X
13   $C_X^j \leftarrow g_j(X), \forall j \in \{1, \dots, J\}$  // Evaluate
     constraints of X
14  P ← P  $\cup$  X;
15  F ← F  $\cup$   $F_X^m$ ;
16  C ← C  $\cup$   $C_X^j$ ;
17  eval ← eval +  $|X|$ ;
18 return  $P_T \leftarrow$  Non-dominated solutions of P

```

and constraint functions, M1 approximates the original functions as they are. However, M6 attempts to approximate the most complex selection function which is multimodal and discontinuous at the constraint boundaries. Understandably, more sample points are required for modeling more complex landscapes from M1 to M6, in general. Although we recognize this effect, in this first study, we use an identical number of sample points for constructing metamodels for all six frameworks to mainly show the working of them.

VIII. RESULTS

In this section, we compare the results obtained by all six proposed metamodeling frameworks. We use the same parameter settings for RGA or M-RGA, each of which uses the binary tournament selection operator, simulated binary crossover (SBX), and polynomial mutation, with parameters as follows: population size = $10n$, where n is a number of variables, number of generations = 100, crossover probability = 0.95, mutation probability = $1/n$, distribution index for SBX operator = 1, and distribution index for polynomial mutation operator = 10. The NSGA-II procedure, wherever used, is also applied with the same parameter values as above. For repeating a metamodeling formulation for each formulation, we have used $\kappa = 5$ and $\tau = 50$ generations. We performed ten runs of all frameworks on all test problems. As mentioned earlier, for unconstrained problems (ZDT) with no constraints and three-objective problem C2DTLZ2 having a single constraint, the behaviors of both M1 and M2 are identical. In such a case, we only show the results for M1. The same situation occurs for M3 and M4 and we only show the results of M3. In order to have a graphical comparison, we show the obtained tradeoff solutions for the median inverted generational distance (IGD) run in each case. It is worth mentioning here that in this first paper, we do not make any effort to choose optimal parameter setting for each algorithm, rather identical parameters are used to provide a representative performance of each framework.

A. Two-Objective Unconstrained Problems

First, we apply our proposed frameworks to two-objective unconstrained problems ZDT1–ZDT3 and ZDT6 with ten ($n = 10$) variables, $H = 21$ reference directions, and with a maximum of only $SE_{\max} = 500$ high-fidelity solution evaluations. For each problem, we have used an initial sample size of $\rho = 300$. The obtained nondominated solutions are shown in Figs. 2 and 3, for frameworks M1, M3, M5, and M6, respectively. Figures for ZDT3 and ZDT6 are put in the supplementary material. As mentioned above, M2 becomes identical to M1 and M4 becomes identical to M3 for unconstrained problems. For brevity, we also do not show results from M1-1 framework. For brevity, we denote M1-2 and M2-2 by M1 and M2, respectively, in the rest of this paper.

It is clear from the figures that all four frameworks are able to solve ZDT1 problem fairly well in only 500 high-fidelity SEs, but frameworks M5 and M1 perform the best. The

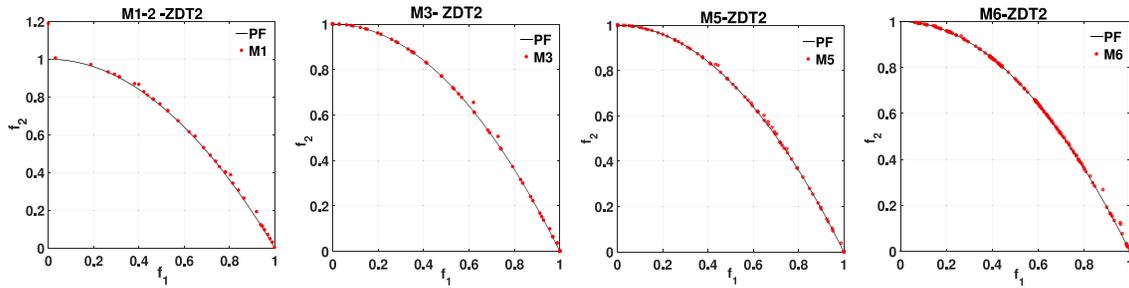


Fig. 3. Obtained nondominated solutions for problem ZDT2 using frameworks M1, M3, M5, and M6 from left to right.

TABLE II
COMPUTED IGD VALUES FOR TEST PROBLEMS. BEST PERFORMING FRAMEWORK AND
OTHER STATISTICALLY SIMILAR FRAMEWORKS ARE MARKED IN BOLD

	M1-2		M2-2		M3		M4		M5		M6	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
ZDT 1	0.0115	0.0006	-	-	0.0185	0.0046	-	-	0.0103	0.0028	0.0249	0.0153
	$p = 0.12120$		-	-	$p = 0.00170$		-	-	-		$p = 0.02570$	
ZDT 2	0.0105	0.0027	-	-	0.0130	0.0013	-	-	0.0078	0.0011	0.0103	0.0051
	$p = 0.00170$		-	-	$p = 0.00019$		-	-	-		$p = 0.73370$	
ZDT 3	0.0136	0.0019	-	-	0.0254	0.0154	-	-	0.0246	0.0027	0.7111	0.2054
	-		-	-	$p = 0.06390$		-	-	$p = 0.00018$		$p = 0.00018$	
ZDT 6	1.6144	0.1965	-	-	0.3774	0.1099	-	-	0.1208	0.0311	5.9401	0.7896
	$p = 0.00018$		-	-	$p = 0.00024$		-	-	-		$p = 0.00018$	
BNH	0.5322	0.2164	0.4605	0.1324	0.3854	0.0724	0.3539	0.1543	0.3727	0.0798	0.4177	0.1852
	$p = 0.02070$		$p = 0.00880$		$p = 0.05290$		-		$p = 0.30530$		$p = 0.27120$	
SRN	0.7452	0.2473	1.0477	0.1765	1.5564	0.1742	1.9072	0.3237	2.8532	0.5795	1.5824	0.2557
	-		$p = 0.00910$		$p = 0.00024$		$p = 0.00016$		$p = 0.00018$		$p = 0.00025$	
TNK	0.0173	0.0059	0.0748	0.0208	0.0107	0.0010	0.0300	0.0075	0.0654	0.0353	0.0388	0.0038
	$p = 0.00033$		$p = 0.00018$		-		$p = 0.00018$		$p = 0.00018$		$p = 0.00018$	
OSY	6.9534	6.3585	31.0684	10.8709	5.8386	1.1182	39.4488	13.2070	23.8865	7.9269	39.7462	8.7765
	$p = 0.47250$		$p = 0.00018$		-		$p = 0.00018$		$p = 0.00018$		$p = 0.00018$	
Welded Beam	0.8428	0.3655	2.0504	0.7854	1.5036	0.5551	2.1265	0.7054	1.0268	0.2396	2.5987	1.1322
	-		$p = 0.00058$		$p = 0.01400$		$p = 0.00044$		$p = 0.16200$		$p = 0.00033$	
DTLZ2	0.0387	0.0045	-	-	0.0634	0.0038	-	-	0.0318	0.0029	0.0345	0.0043
	$p = 0.00077$		-	-	$p = 0.00018$		-	-	-		$p = 0.30750$	
DTLZ4	0.1208	0.0380	-	-	0.2614	0.0485	-	-	0.0856	0.0113	0.3828	0.0277
	$p = 0.03760$		-	-	$p = 0.00018$		-	-	-		$p = 0.00018$	
DTLZ5	0.0039	0.0005	-	-	0.0176	0.0017	-	-	0.0102	0.0005	0.0057	0.0034
	-		-	-	$p = 0.00018$		-	-	$p = 0.00018$		$p = 0.00280$	
C2DTLZ2	0.0485	0.0046	-	-	0.05902	0.0020	-	-	0.0490	0.0017	0.1037	0.0268
	-		-	-	$p = 0.00025$		-	-	$p = 0.21210$		$p = 0.00018$	
Car Side	0.2148	0.0252	0.2128	0.0192	0.1992	0.0241	0.2691	0.0425	0.1530	0.0113	0.3915	0.08211
	$p = 0.00044$		$p = 0.00018$		$p = 0.00044$		$p = 0.00018$		-		$p = 0.00018$	
DTLZ2-5D	0.0416	0.0030	-	-	0.0929	0.0029	-	-	0.0840	0.0029	0.2512	0.0202
	-		-	-	$p = 0.00018$		-	-	$p = 0.00018$		$p = 0.00018$	
C2DTLZ2-5D	0.1510	0.0266	-	-	0.0720	0.0190	-	-	0.1792	0.0302	0.2311	0.0140
	$p = 0.00018$		-	-	-		-	-	$p = 0.00018$		$p = 0.00018$	

obtained points are very close to the respective true Pareto-optimal fronts and have a good distribution of points on the entire front. Most past studies [16] have used tens of thousands of high-fidelity SEs to have a similar performance, while here using the proposed metamodelling frameworks, we are able to find a similar set of points in only 500 high-fidelity SEs. Table II shows the IGD metric value for all runs perform for all frameworks. The GD metric values are kept in the supplementary material for brevity. The IGD metric is computed using 21 true Pareto-optimal points obtained for each reference point, one at a time. Since each framework is expected to find the respective Pareto-optimal point, this IGD computation is able to distinguish a set of nondominated points from another set depending on their convergence level to the desired set of points. However, to compute the GD metric, we use a large number of Pareto-optimal fronts so as to get a clear

idea of the convergence level to the Pareto-optimal front. In ZDT2 problem, two selection function-based metamodelling frameworks (M5 and M6) perform the best, followed by M1 and then M3. It is interesting to note that despite multimodal nature of the selection function landscape with M6, it is able to find multiple near Pareto-optimal points with only 500 high-fidelity solution evaluations. The convex or nonconvex nature of the Pareto-optimal front does not seem to matter to all frameworks. For ZDT3 problem in terms of IGD measure, M1 and M3 perform the best, followed by M5 then M6. For ZDT6 problem, framework M5 performs the best, while other frameworks are not able to make a good performance in 500 high-fidelity solution evaluations. Nevertheless, the results on the first three problems indicate that all frameworks have performed well with a fraction of solution evaluations than they are usually used in standard studies [16]. Overall, it is clear

TABLE III
COMPUTED IGD VALUES FOR ZDT PROBLEMS. BEST PERFORMING FRAMEWORK AND OTHER STATISTICALLY SIMILAR FRAMEWORKS ARE MARKED IN BOLD

	M1-2		M3		M5		M6		NSGA-II	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
ZDT 1	0.0115	0.0006	0.0185	0.0046	0.0103	0.0028	0.0249	0.0153	0.3567	0.0935
ZDT 2	0.0105	0.0027	0.013	0.0013	0.0078	0.0011	0.0103	0.0051	0.7063	0.2306
ZDT 3	0.0136	0.0019	0.0254	0.0154	0.0246	0.0027	0.7111	0.2054	0.2801	0.0731
ZDT 6	1.6144	0.1965	0.3774	0.1099	0.1208	0.0311	5.9401	0.7896	5.0010	0.2883

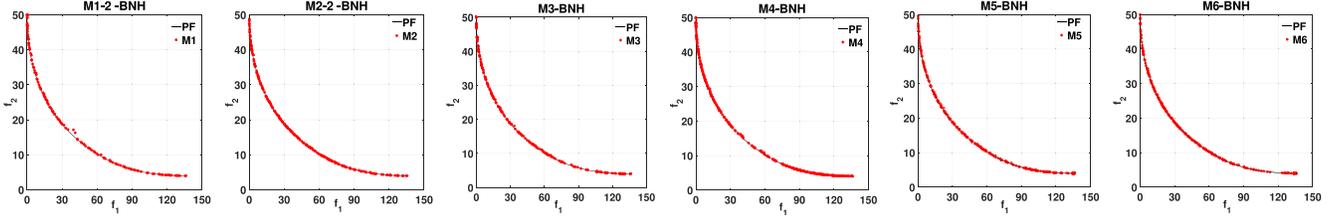


Fig. 4. Obtained nondominated solutions for BNH using M1–M6 from left to right and top to bottom.

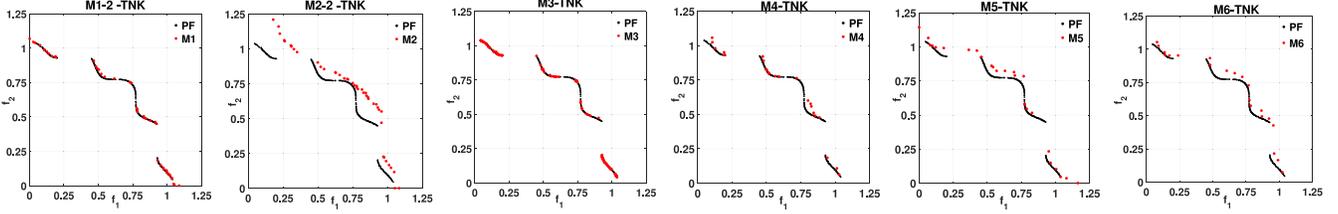


Fig. 5. Obtained nondominated solutions for the TNK problem using frameworks M1–M6 from left to right and top to bottom.

that framework M5 performs the best on the unconstrained two-objective test problems. Frameworks having statistically insignificant performance from the best performing framework in each problem are also marked in bold with the respective p -value in Wilcoxon signed-ranked test.

For a further analysis, we compare IGD and GD performance metric values of the proposed metamodelling frameworks with NSGA-II without any metamodelling framework for solving the ZDT problems with only 500 evaluations. It is clear from Table III that the proposed metamodelling frameworks show superior performance, thereby suggesting a clear advantage of the use of metamodelling frameworks. Similar superior convergence is also observed through the GD metric, results of which can be found in the supplementary material.

B. Two-Objective Constrained Problems

Next, we apply our proposed frameworks to two-objective constrained problems: Binh’s problem (BNH), Srinivas and Deb’s problem (SRN), Tanaka’s problem (TNK), and Osyczka’s problem (OSY) [15]. For each problem, $H = 21$ reference directions, an initial sample size of $\rho = 400$ (for OSY $\rho = 500$), and a maximum of $SE_{\max} = 800$ high-fidelity SEs are fixed. The obtained nondominated solutions are shown in Figs. 4 and 5 for M1–M6, respectively. Figures for SRN and OSY are put in the supplementary document. All six frameworks are able to find a close and well-distributed set of tradeoff points to true Pareto-optimal front (shown by

a solid line in each case) for BNH and SRN problems. With only $SE_{\max} = 800$ high-fidelity SEs used in this paper, the performance of these frameworks is noteworthy. However, the problem TNK has provided difficulties to all six frameworks, due to discontinuities in its Pareto-optimal front. Although all frameworks come close to the true Pareto-optimal front, M3 performs the best, but as presented in Table II, M1, M4, and M6 also perform well. In OSY, frameworks M3 and M1 perform the best. Although metamodelling are constructed progressively, the independent approximation of constraints adopted by M3 and M1 produced more accurate results than the use of a combined constraint function in this difficult problem. Another aspect which has become clear from these results is that modeling of a combined constraint violation function (as in M2 and M4) is not better than modeling each constraint independently (as in M1 and M3). However, an integrated constraint handling using the aggregated ASF approach or the KKTPM approach is better.

C. Three-Objective Constrained and Unconstrained Problems

Next, we apply all six frameworks to three-objective optimization problems (DTLZ2, DTLZ4, and DTLZ5) and also to a three-objective constrained problem (C2DTLZ2). Each of these problems are considered for seven variables and considered for $H = 91$ reference directions. We fix $SE_{\max} = 1000$ high-fidelity SEs and $\rho = 500$ for DTLZ2 and DTLZ5, and $SE_{\max} = 2000$ and $\rho = 700$ for DTLZ4 due to

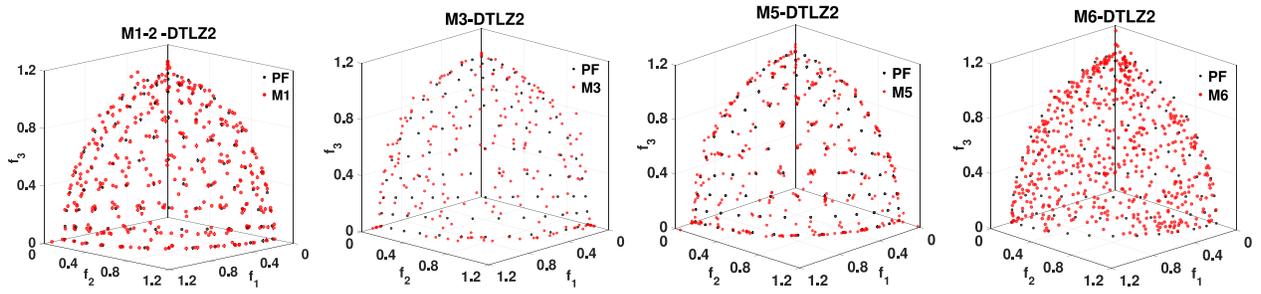


Fig. 6. Obtained nondominated solutions for problem DTLZ2 using frameworks M1, M3, M5, and M6 from left to right.

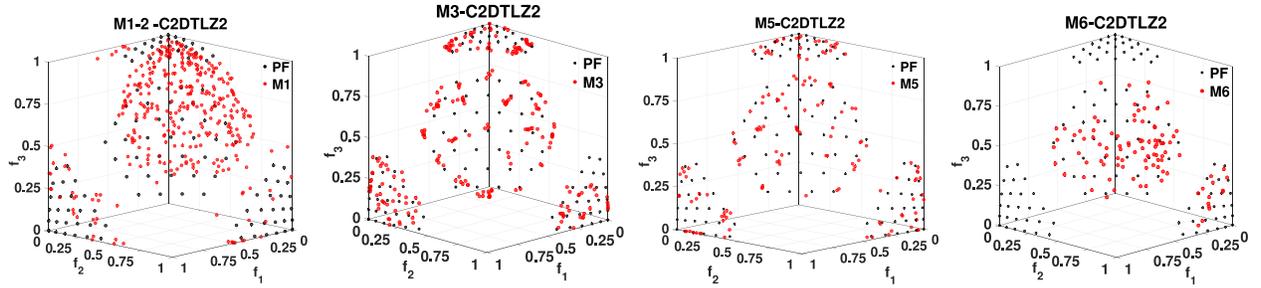


Fig. 7. Graphical results for problem C2DTLZ2 using frameworks M1, M3, M5, and M6 from left to right.

multimodality in its landscape. For C2DTLZ2, we have used $SE_{\max} = 1500$ and $\rho = 700$.

The Pareto-optimal surface of DTLZ2 and C2DTLZ2 problems and the respective nondominated solutions (all nondominated solutions of the high-fidelity solutions) are shown in Figs. 6 and 7 for M1, M3, M5, and M6 frameworks, respectively. Figures for DTLZ4 and DTLZ5 are put in the supplementary material. First, all four frameworks perform well, in general, on the DTLZ2 problem. Frameworks M5 and M6 perform the best on this problem. Then, on problem DTLZ4, framework M5 performs the best. On DTLZ5, all four frameworks performs well, with M1 performing the best, followed by M6. Thus, on unconstrained three-objective problems, frameworks M5 and M1 perform the best, followed by M6 and then M3.

On C2DTLZ2 problem, frameworks M1 and M5 perform the best, followed by M3. As noted before, since this problem has a single constraint, M1 and M2 will produce identical results, M3 and M4 will also produce identical results. Interestingly, framework M6 is able to find multiple near Pareto-optimal solutions through the multimodal RGA approach; however, further developments are needed to understand full potential of M6.

Finally, we apply all six frameworks to a car side-impact problem having three objectives and 10 constraints. This problem allows us to apply each of our six frameworks on the same problem. More details for this problem are mentioned in the supplementary material.

D. Five-Objective Constrained and Unconstrained Problems

Finally, we apply all six frameworks to five-objective unconstrained DTLZ2 and to five-objective constrained C2DTLZ2 problem. Each of these problems are considered for seven

variables and considered for $H = 210$ reference directions. We fix $SE_{\max} = 2,500$ high-fidelity SEs and $\rho = 900$ for DTLZ2 and C2DTLZ2, respectively. According to the IGD performance metric value in Table II, M1-2 performs the best on DTLZ2, followed by M5 and then M3. For C2DTLZ2, M3 performs the best, followed by M1 and M5. As mentioned earlier, more high-fidelity points are needed for M6 to work better, but these initial results demonstrate the relative performances of proposed six metamodeling frameworks in two to five-objective constrained and unconstrained problems.

IX. CONCLUSION

In this paper, we have presented a taxonomy for the use of different metamodeling frameworks for multiobjective optimization. The taxonomy also extends to cover single-objective optimization problems with and without constraints; however, handling constraints has been put under the same platform as the handling of objectives—a matter which has been ignored in many past such studies. This paper has focused outside the popular strategy of extending metamodeling frameworks for single-objective optimization to multiobjective optimization. The straightforward extension suggests that for every objective function and constraint function, an independent metamodel is needed. Although this is one viable strategy (M1-2), we have proposed five more broad potential strategies and classified many existing studies into the proposed six broad categories. Thereafter, as an initial detailed systematic study, we have implemented a representative algorithm for each of the six proposed frameworks and applied them to solve a number of constrained and unconstrained multiobjective optimization problems. Results are compared against each other and important conclusions about the behavior of each of the frameworks have been revealed. It is worth noting that

although parameters of all six proposed frameworks have not been fine-tuned for their best performance, our obtained results are able to bring out generic properties of different possible multiobjective metamodeling frameworks applied to distinct problem classes.

First, it is interesting to note that compared to standard EMO studies in which tens of thousands of solution evaluations are usually devoted to solve these test problems, here we present results that take only a fraction (a few hundreds to a maximum of two thousands) of solution evaluations in each case to find a near Pareto-optimal set of solutions using a metamodeling-based EMO approach. This is remarkable and the proposed metamodeling frameworks hold promise to the successful applications of metamodeling methods in practical EMO studies.

Second, there is a tradeoff between number of metamodeling efforts and resulting performance. It is intuitive that as the number of objectives increase, the straightforward implementation of metamodeling frameworks for each objective and constraint function independently may not be a computationally efficient approach. The flip side is that the resulting integrated functions become complex to metamodel, thereby requiring more high-fidelity points. Although further studies are needed to make more confident conclusions, we have observed that a metamodeling of an aggregate function to find a single Pareto-optimal solution at a time is a better strategy, as evident from the better performance of frameworks M3 and M5. Both these frameworks constitute an implicit parallel approach, as intermediate points obtained for one reference line may lie in the vicinity of the Pareto-optimal solution of another neighboring reference line, thereby helping to create better metamodels for subsequent reference lines. This and the use of the efficient global optimization (EGO) approach are the key reasons for the superior performance of M5.

Third, it is also clear that a metamodeling of a combined constraint violation function (M2 and M4) is not, in general, a better strategy compared to independent-constraint metamodeling frameworks M1 and M3 both in terms of convergence and diversity issues. However, when constraints are integrated with objective functions by a scalarization function (in M5), better performance has been achieved. The superiority of M5 in most problems of this paper suggests that metamodeling of an implicit selection function involving all objective and constraint functions together either for finding a single solution at a time (or multiple solutions together) as an integrated approach is a better strategy. Framework M6 is interesting, but further investigation is needed with a different parameter setting to completely evaluate its merit.

In general, this paper has clearly indicated the need for metamodeling aggregated approaches compared to their simplistic and independent multiple metamodeling frameworks. This is extremely motivating to pursue development and application of further such frameworks, some of which we are currently pursuing, such as M3-2 and M4-2. Although identical parameter values are used for each framework here to get an overall and initial assessment of simplistic and aggregated approaches, it is likely that each framework performs its best for a different set of parameter values than what

has been used here. It will be interesting to investigate the effect of diverse sweep approach with frameworks M3–M5. Different in-filling methods, such as ParEGO, MOEA/D-EGO, and other approaches, may also be tried with each framework. A similar metamodeling study for preference-based EMO would be another practical extension of this paper. As an immediate future study, we plan to investigate other metamodeling approaches (such as radial basis function, support vector machine, and response surface, to name a few) and acquisition functions, such as EI approach to eventually identify optimized algorithms and determine the scope for each of the six possible metamodeling frameworks for solving various types of problems ranging from two to many objectives. A progressive switch between different frameworks with generations is another possibility, which must also be investigated.

REFERENCES

- [1] A. Cassioli and F. Schoen, "Global optimization of expensive black box problems with a known lower bound," *J. Glob. Optim.*, vol. 57, no. 1, pp. 177–190, 2013.
- [2] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *J. Glob. Optim.*, vol. 21, no. 4, pp. 345–383, 2001.
- [3] L. V. Santana-Quintero, A. A. Montañó, and C. A. C. Coello, "A review of techniques for handling expensive functions in evolutionary multi-objective optimization," in *Computational Intelligence in Expensive Optimization Problems*. Heidelberg, Germany: Springer, 2010, pp. 29–59.
- [4] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005.
- [5] L. Shi and K. Rasheed, "A survey of fitness approximation methods applied in evolutionary algorithms," in *Computational Intelligence in Expensive Optimization Problems*. Heidelberg, Germany: Springer, 2010, pp. 3–28.
- [6] A. Díaz-Manríquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal, "A review of surrogate assisted multiobjective evolutionary algorithms," *Comput. Intell. Neurosci.*, vol. 2016, Jun. 2016, Art. no. 9420460.
- [7] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.
- [8] K. Deb and R. Datta, "Hybrid evolutionary multi-objective optimization and analysis of machining operations," *Eng. Optim.*, vol. 44, no. 6, pp. 685–706, 2012.
- [9] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer, 1999.
- [10] D. Horn, T. Wagner, D. Biermann, C. Weihs, and B. Bischl, "Model-based multi-objective optimization: Taxonomy, multi-point proposal, toolbox and benchmark," in *Evolutionary Multi-Criterion Optimization*. Cham, Switzerland: Springer, 2015, pp. 64–78.
- [11] 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, Feb. 2006.
- [12] 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*. Heidelberg, Germany: Springer, 2008, pp. 784–794.
- [13] B. Bischl, S. Wessing, N. Bauer, K. Friedrichs, and C. Weihs, "MOI-MBO: Multiobjective infill for parallel model-based optimization," in *Learning and Intelligent Optimization*. Cham, Switzerland: Springer, 2014, pp. 173–186.
- [14] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, 1998.
- [15] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
- [16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

- [17] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [18] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [19] S. Z. Martínez and C. A. C. Coello, "MOEA/D assisted by RBF networks for expensive multi-objective optimization problems," in *Proc. ACM Genetic Evol. Comput. Conf.*, Amsterdam, The Netherlands, 2013, pp. 1405–1412.
- [20] A. Arias-Montañó, C. A. C. Coello, and E. Mezura-Montes, "Multi-objective airfoil shape optimization using a multiple-surrogate approach," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 1–8.
- [21] Y. Zhang, S. Hu, J. Wu, Y. Zhang, and L. Chen, "Multi-objective optimization of double suction centrifugal pump using Kriging meta-models," *Adv. Eng. Softw.*, vol. 74, pp. 16–26, Aug. 2014.
- [22] M. N. Le, Y. S. Ong, S. Menzel, C.-W. Seah, and B. Sendhoff, "Multi co-objective evolutionary optimization: Cross surrogate augmentation for computationally expensive problems," in *Proc. CEC*, Brisbane, QLD, Australia, 2012, pp. 2871–2878.
- [23] H. K. Singh, T. Ray, and W. Smith, "Surrogate assisted simulated annealing (SASA) for constrained multi-objective optimization," in *Proc. CEC*, Barcelona, Spain, 2010, pp. 1–8.
- [24] R. Hussein and K. Deb, "A generative Kriging surrogate model for constrained and unconstrained multi-objective optimization," in *Proc. ACM GECCO*, Denver, CO, USA, 2016, pp. 573–580.
- [25] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, 2000.
- [26] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple Criteria Decision Making Theory and Application*. Heidelberg, Germany: Springer, 1980, pp. 468–486.
- [27] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proc. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 42–50.
- [28] K. Deb, M. Abouhawwash, and J. Dutta, "An optimality theory based proximity measure for evolutionary multi-objective and many-objective optimization," in *Evolutionary Multi-Criterion Optimization*. Cham, Switzerland: Springer, 2015, pp. 18–33.
- [29] C. C. Tutum and K. Deb, "A multimodal approach for evolutionary multi-objective optimization (MEMO): Proof-of-principle results," in *Evolutionary Multi-Criterion Optimization*. Cham, Switzerland: Springer, 2015, pp. 3–18.
- [30] K. Deb and M. Abouhawwash, "An optimality theory-based proximity measure for set-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 515–528, Aug. 2016.



Kalyanmoy Deb (F'02) received the bachelor's degree from IIT Kharagpur, Kharagpur, India, and the master's and Ph.D. degrees from the University of Alabama, Tuscaloosa, AL, USA, in 1989 and 1991, respectively.

He is the Koenig Endowed Chair Professor with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA. He is largely known for his seminal research in evolutionary multicriterion optimization. He has authored two text books

on optimization and published over 480 international journal and conference research papers. His current research interests include evolutionary optimization algorithms and their application in optimization and machine learning.

Dr. Deb was a recipient of the IEEE CIS EC Pioneer Award in 2018, the Lifetime Achievement Award from Clarivate Analytics in 2017, the Infosys Prize in 2012, the TWAS Prize in Engineering Sciences in 2012, the CajAstur Mamdani Prize in 2011, the JC Bose National Fellowship in 2011, the Distinguished Alumni Award from IIT Kharagpur, in 2011, the Edgeworth-Pareto Award in 2008, the Shanti Swarup Bhatnagar Prize in Engineering Sciences in 2005, and the Thomson Citation Laureate Award from Thomson Reuters. He is a fellow of ASME.



Rayan Hussein (S'18) received the B.Sc. and M.Sc. degrees in electrical and communication engineering from the University of Mosul, Mosul, Iraq, in 2002 and 2005, respectively. He is currently pursuing the doctoral degree with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA.

He was with Asiacell, Sulaymaniyah, Iraq, a mobile telecom company, from 2005 to 2013. His current research interests include evolutionary computation with a focus in metamodeling for constrained multiobjective optimization problems.

Mr. Hussein was a recipient of the Scholarship Award from Higher Committee for Education Development in Iraq, in 2013.



Proteek Chandan Roy (S'16) received the B.Sc. degree in computer science and engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2011, and the master's degree in computer science from Mississippi State University, Starkville, MS, USA, in 2015.

His current research interests include multiobjective optimization, algorithm design, evolutionary computation, surrogate-assisted optimization, and machine learning.



Gregorio Toscano-Pulido (M'11) received the Ph.D. degree from the Center for Research and Advanced Studies, National Polytechnic Institute (CINVESTAV-IPN), Mexico City, Mexico.

In 2015, he took a sabbatical year at the COIN Laboratory, Michigan State University, East Lansing, MI, USA. He is currently an Associate Professor with CINVESTAV-IPN-Tamaulipas, Victoria 87130, Mexico. His current research interests include increasing the state of the art of evolutionary algorithms, computational intelligence related to the

design of efficient multiobjective algorithms, decision making, and the development of solutions for engineering, data-science, and real-world problems.