Metamodeling for Multimodal Selection Functions in Evolutionary Multi-Objective Optimization

Proteek Roy
Michigan State University
East Lansing, Michigan 48824
royprote@egr.msu.edu

Rayan Hussein
Michigan State University
East Lansing, Michigan 48824
husseinr@egr.msu.edu

Kalyanmoy Deb
Michigan State University
East Lansing, Michigan 48824
kdeb@egr.msu.edu

ABSTRACT

Most real-world optimization problems involve computationally expensive simulations for evaluating a solution. Despite significant progress in the use of metamodels for single-objective optimization, metamodeling methods have received a lukewarm attention for multi-objective optimization. A recent study classified various metamodeling approaches, of which one particular method is interesting, challenging, and novel. In this paper, we study this so-called M6 method in detail. In this approach, a selection operator's assignment function, as it is implemented in an evolutionary multi-objective optimization (EMO) algorithm, is directly metamodeled. Thus, this methodology requires only one selection function to be metamodeled irrespective of multitude of objective and constraint functions in a problem. However, the flip side of the methodology is that the resulting function is multimodal having a different optimum for every desired Pareto-optimal solution. We have used two different selection functions based on two recent ideas: (i) KKT proximity measure function and (ii) multimodal based evolutionary multi-objective (MEMO) selection function. The resulting metamodeling methods are applied to a number of standard two and three-objective constraint and unconstrained test problems. Near Pareto-optimal solutions are found using only a fraction of high-fidelity solution evaluations compared to usual EMO applications.

KEYWORDS

Multi-modal, Multi-objective optimization, Expensive, Surrogate, Neural Network

ACM Reference format:
Proteek Roy, Rayan Hussein, and Kalyanmoy Deb. 2017. Metamodeling for Multimodal Selection Functions in Evolutionary Multi-Objective Optimization. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071219

1 INTRODUCTION

Many practical optimization problem are confronted with the difficulty that objective functions and constraints are computationally expensive to evaluate. Objective and constraint values often come from an expensive simulation of a system. Because of that, most of the time, researchers are bound to have a limited number of solution evaluations to get optimum feasible solutions. Although parallel and distributed computing reduce the total simulation time, effective searching is inescapable to get better solutions within limited function evaluation. Researchers have used metamodels or surrogate models to make a low-fidelity approximation of the objective functions and constraints.

A good number of surrogate models or metamodels have been proposed in literature[14][18][21]. Among them, Kriging [14] or Gaussian Process model [10] are popular, as they provide an error estimate in addition to the approximate function value. Although being popular, Kriging doesn't perform well when the number of variables is large, or the target function is complicated with multi-modality and nonlinearity. Knowles [15] extended the EGO method (Efficient Global Optimization [14]) proposed for single-objective problems for evolutionary multi-objective optimization (EMO) and proposed ParEGO algorithm with an aggregated function. The main drawback of the ParEGO algorithm is that it metamodels each objective function separately and the overall algorithm cannot generate multiple candidate points at one iteration. S-Metric selection based EGO (SMS-EGO) [17] optimizes the S-metric, a hypervolume-based measure, by using the covariance matrix adaptation evolution strategy (CMS-ES) algorithm by extending the idea of Emmerich et al. [9]. Similar to ParEGO, SMS-EGO evaluates a single solution at each iteration. MOEA/D-EGO, proposed by Zhang et al. [22], combines a decomposition based multi-objective EA (MOEA/D) with EGO. In this approach, a multi-objective optimization problem (MOP) is decomposed into multiple single objective sub-problems. In each iteration, a Gaussian stochastic process model is built for each sub-problem with high-fidelity solutions which were previously evaluated. Expected improvement of these sub-problems are then optimized simultaneously by using the MOEA/D procedure for generating multiple candidate solutions. The main weakness of this approach is that it needs separate metamodels for each sub-problem. Researchers have tried to classify metamodel-based EMO studies from a variety of angles. Jin [13] proposed a taxonomy based on evolution control that explains how one can alternate among high-fidelity and low fidelity evaluation of solutions. A more recent work [6] classifies the existing methodologies into six categories based on number of models used and complexity of those low-fidelity models. From that classification, one very interesting methodology, namely, the sixth methodology (or M6) turns out to be a promising proposition in terms of its algorithmic novelty, challenges it offers to a metamodeling approach and a new direction for research. This method proposes to have a single metamodel for the complete MOP by considering all objective functions and all

GECCO '17, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3071178.3071219

constraint functions, and by making every desired Pareto-optimal solution as a separate optimal point of the resulting metamodel. As it sounds, the approach is demanding but, if possible to achieve, is extremely appealing for EMO researchers and practioners alike. In this paper, we examine the development and application of M6 using two different EMO concepts.

Given an MOP with M objectives and J constraints, one can model each objective and constraint function separately, thus having $(M + J)$ total metamodels. In the recent taxonomy [6], this methodology is called M1. One can combine all objectives with some scalarization methods e.g. weighted sum, ϵ-constraint, Tchebychev, or achievement scalarization function (ASF) [16, 19] and create $(1 + J)$ metamodels. Moreover, this method requires as many metamodels to be formed as the number of desired Pareto-optimal points (say H). This methodology is called M3. Not to a great surprise, one can also combine all the constraints and reduce required number of metamodels to $(M + 1)$ for M1 and 2 for M3 to achieve two new methodologies M2 and M4, respectively. With H Pareto-optimal solutions, M3 and M4 requires a total of $H(J + 1)$ and $2H$ total metamodels. Objectives and constraints can be grouped together and their combined or independent modeling can be attributed to M1 to M4. However, one challenging way to reduce computational effort is to construct only one low-fidelity model with the selection function of a EMO algorithm directly [6]. As in many decomposition-based EMO methods, the selection function can be either unimodal (to find a single Pareto-optimal solution at a time) or multimodal in which multiple Pareto-optimal solutions are found simultaneously (like in NSGA-II [4]. The former approach is called M5 [12] and the second approach is called M6 and is the focus of this study. Using reference point based scalarization method, authors of M5 [12] have tried to find a single optimum at a time, thereby requiring a total of H metamodels. Here, we attempt to find multiple Pareto-optimal solutions using a single metamodel from start to finish.

The paper is organized as follows. Section 2 presents the related past works that are relevant to the development of M6. Section 3 discusses our proposed methods. Experimental setting and results are presented in Section 4. Section 5 concludes our study and proposes future work.

2 BACKGROUND

In this section, we first discuss the concept of metamodeling the selection function of an EMO algorithm and proposes two different ways of formulating the selection function for multi-objective optimization problems.

2.1 Metamodeling the Selection Function

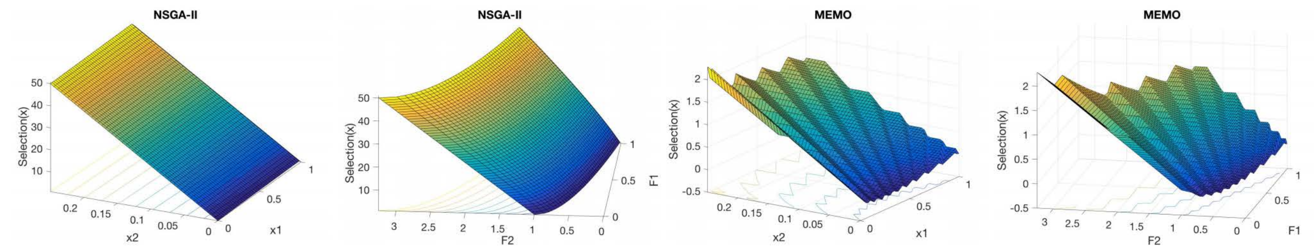
EMO algorithms are mostly different from each other in their way of constructing the selection operator. Having multiple conflicting objectives and multiple constraints to be satisfied, an EMO's selection operator treats two aspects essential for converging near to the Pareto-optimal front and for finding a diverse set of solutions: (i) emphasis for non-dominated solutions and (ii) emphasis for diverse solutions. NSGA-II [4] uses a non-dominated sorting procedure of the entire population at any generation to achieve the first aspect and uses a front-wise crowding distance operator to achieve the second aspect. NSGA-III [7] uses the non-dominated sorting for the first aspect, but uses a more complex niching operator based on a set of given reference directions (W) to achieve the second aspect. No matter what EMO algorithm is considered, it makes a balance between these two aspects to finally provide a ranked list of the population at any generation. Figures 1a and 1b shows how NSGA-II's selection operator ranks 2,500 solutions chosen uniformly from the entire two-dimensional search space for a 2-D ZDT1 problem, in variable and objective spaces, respectively. All first rank solutions are assigned a selection function value within [0,1], with two extreme solutions having a value of zero. Intermediate first rank solutions are assigned a selection function value based on their crowding distance value. All second rank solutions have a selection function value within [1,2], and so on. It is interesting to note that the selection function value gets worse as the solutions are away from the Pareto-optimal front (for which $x_2 = 0$). The same is reflected in the objective-space plot in Figure 1b.

We now argue that instead of metamodeling each of the two objective functions separately and accumulating approximations from two metamodels, if instead a single metamodel is performed to approximate the above selection function, the number of metamodeling effort can be reduced. Theoretically, such a selection function has infinite optima, but if we are interested in finding H ($|W| = H$, a finite size) Pareto-optimal solutions dictated by a set of H pre-specified reference directions, the above selection function will reduce to a H-modal selection function. Such an idea is novel for multi- and many-objective optimization and the procedure shields the number of objectives and constraints from the number of metamodeling efforts that must be performed.

2.2 KKT Proximity Measure Based Approach

The above idea of metamodeling the underlying selection function of an EMO algorithm, instead of metamodeling each and every objective and constraint function, opens the door for trying the idea on successful EMO algorithms. Although this is an interesting direction for research and development, in this paper, we suggest two ideas, developed by using two recently proposed performance metrics for EMO. In this subsection, we discuss the KKT proximity measure approach.

Karush-Kuhn-Tucker (KKT) proximity measure was recently developed [3] to determine the level of convergence of non-dominated solutions in an EMO algorithm. At any point, the KKTPM value can be computed by using (exact or numerical) gradients of objectives and constraint functions. The KKTPM construction is such that if a value within [0,1] is achieved for a point, the point is guaranteed to be a feasible solution and a KKTPM value larger than one means it is infeasible. Moreover, any solution having a KKTPM value equal to zero means that it is guaranteed to be a Pareto-optimal solution. Furthermore, in certain problems it has been observed that KKTPM value of a solution is correlated to its distance from the Pareto-optimal front [3]. This latter property of KKTPM motivates us to use it as a selection function for our M6 metamodeling study. Thus, for a solution $\mathbf{x}$, we simply compute its KKTPM value and set it equal to its Selection($\mathbf{x}$).



(a) NSGA-II selection function on variable space. **(b) NSGA-II selection function in objective space.** **(c) MEMO selection function on variable space.** **(d) MEMO selection function on objectives space.**

Figure 1: Different selection function in variable and objective space.

2.3 MEMO Based Approach

Recently, a multi-modal based EMO or MEMO approach [20] was proposed by constructing a multimodal single-objective function from a multi- or many-objective problem, which possess a finite number of multiple global optima. The location of each optimum is determined by the chosen reference direction. The MEMO approach used the Achievement Scalarization Function (ASF), which is given or h-th reference direction, as follows:

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

A set of H reference directions $\mathbf{w}^{(h)}$ is used to determine the location of H Pareto-optimal solutions. Reference points are equally-angled directions from an ideal point $\mathbf{z}$ and are created using Das and Dennis’s method [2]. Any preference, if necessary, can be incorporated to create the reference vectors. Objective values are normalized by population-minimum and population-maximum before calculating ASF value. In this study, we define the MEMO selection function, as follows:

$$\mathrm{MEMO}(\mathbf{x}) = \min_{h=1}^{H} \mathrm{ASF}^{(h)}(\mathbf{x}). \quad (2)$$

Figures 1c and 1d show the MEMO selection function on the same ZDT1 problem on the variable space and objective space, respectively, for 11 reference directions. It is clear that for each reference direction there is a single Pareto-optimal point which makes the MEMO selection function to have a local minimum there. It is interesting to note how the MEMO selection function has become multi-modal and demonstrates the presence of all 11 minima, each corresponding to a different Pareto-optimal solution.

The original MEMO implementation does not combine constraints into the selection function, rather constraints (of type $g_j(\mathbf{x}) \leq 0$) are handled separately by the constrained tournament selection method [4]. Here, we modify the above MEMO selection function slightly for constrained problems by combining all normalized constraint violations ($\bar{g}_j(\mathbf{x})$) as a function [5], as follows, and making an adjustment described next.

$$\mathrm{CV}(\mathbf{x}) = \sum_{j=1}^{J} \langle \bar{g}_j(\mathbf{x}) \rangle, \quad (3)$$

where $\langle \alpha \rangle = \alpha$, if $\alpha > 0$; zero, otherwise. A constrained MEMO selection function is defined as follows:

$$\mathcal{S}(x) = \begin{cases} \mathrm{MEMO}(\mathbf{x}), & \text{if } \mathbf{x} \text{ is feasible}, \\ \mathrm{MEMO}(\mathbf{x})_{\max} + \mathrm{CV}(\mathbf{x}), & \text{otherwise}. \end{cases} \quad (4)$$

3 PROPOSED METHOD

The overall M6 algorithm for solving computationally expensive MOP is described here. First, we create an archive population P_0 of size ρ with initial random solutions based on Latin Hypercube Sampling (LHS) [1] in entire variable space. Then we evaluate the objective and constraint functions with “high-fidelity” solution evaluations. Thereafter, we sort P_0 according to distances from all reference directions $w \in W$ in the objective space. Population P_0 is then classified into different clusters according to the shortest distance from r. We pick the best solution L_w to be the leader of each direction. We then compute a suitable selection function $\mathcal{S}(.)$ with objectives and constraints for each solution. With these selection function values, we then build a surrogate model of the selection function. We then employ a niching based real-parameter genetic algorithm (N-RGA) to perform an optimization run on this model. N-RGA is designed to return at most $|W|$ points X, where each solution is the best for one niche (or, cluster). In the beginning, some of the clusters may be empty and thus $|X|$ is expected to be smaller than $|W|$. When the size of the predicted set X exceeds the allowed number of solution evaluations $SE_{\max}$ we pick only $(SE_{\max} - |X|)$ solutions to restrict our high-fidelity evaluation to $SE_{\max}$. We then evaluate objectives and constraints of those H best-niched solutions (X) with high-fidelity solution evaluation and report. The whole algorithm is presented in Algorithm 1.

3.1 Initialization

Due to the multimodal structure of the M6 surrogate model, it is expected that an adequate number of initial points are required to have a reasonable starting metamodel of the problem. In order to conduct an effective search, we require a good representation of solutions over different parts of the objective space. Although simple Latin Hypercube Sampling is enough for creating good representative solutions, some variable density problems e.g. ZDT6, DTLZ4 etc. require a more sophisticated initialization procedure to get relatively uniform distribution in the objective space. Here we propose a methodology for initialization which is solely based on diversity. First, we create η solutions which is a fraction of the initial population of size ρ using the LHS sampling and evaluate them with

Algorithm 1: Multimodal-Selection Based Algorithm

Input : Objectives: $[f_1, \ldots, f_M]^\top$, constraints: $[g_1, \ldots, g_J]^\top$, n (variables), ρ (sample size), $SE_{\max}$ (total high fidelity evaluations), N-RGA (multi-modal real-parameter genetic algorithm), Γ (parameters of RGA), W (reference direction set), S (multi-modal constrained selection function)

Output: P_T

$P \leftarrow LHS(\rho, n)$ `// initialization with Latin Hypercube Sampling`
$F \leftarrow f_m(P), \forall m \in \{1, \ldots, M\}$ `// high fidelity evaluations (functions)`
$C \leftarrow g_j(P), \forall j \in \{1, \ldots, J\}$ `// high fidelity evaluations (constraints)`
$eval \leftarrow \rho$ `// number of function evaluations`
while $eval < SE_{\max}$ **do**
  for $w \in W$ **do**
    `// for each reference direction` w
    $L_w \leftarrow$ Sort P according to distance from w and pick the best solution
  end
  $L = \{L_1, \ldots, L_{|W|}\}$ `// vector of` $|W|$ `leaders`
  Fitness $\leftarrow S(F, C)$ `// Compute selection function`
  $\mathcal{F} \leftarrow$ *Create_Surrogate_Model*(Fitness) `// Surrogate model for selection function`
  $X \leftarrow$ N-RGA$(\mathcal{F}, L, \Gamma)$ `// returns multiple optimized solutions, one for each reference line; niching is performed in x-space with L`
  if $|X| + |P| > SE_{\max}$ **then**
    $X \leftarrow X(1 : (SE_{\max} - |P|))$ `// Choose best` $(SE_{\max} - |P|)$ `metamodeled solutions`
  end
  $F_X^m \leftarrow f_m(X), \forall m \in \{1, \ldots, M\}$ `// Evaluate objectives of X`
  $C_X^j \leftarrow g_j(X), \forall j \in \{1, \ldots, J\}$ `// Evaluate constraints of X`
  $P \leftarrow P \cup X$;
  $F \leftarrow F \cup F_X$;
  $C \leftarrow C \cup C_X$;
  $eval \leftarrow eval + |X|$;
end
return $P_T \leftarrow P(1 : |W|)$

high-fidelity evaluations. We then calculate the average distance of each solution from its τ nearest neighbors in the objective space. This average distance is then used to locate new sample point in the search space in a non-uniform manner. New points are added in slabs of η% at a time to fill up the whole initial population. Crossover and mutation probability is set as 1.0 and $1/n$ (where n is the number of variables), respectively. We create η solutions each time and fill up P with ρ solutions in total.

Figure 2 shows the effect of incremental initialization procedure on ZDT6 problem which has a bias of solutions on the larger f_1 values.

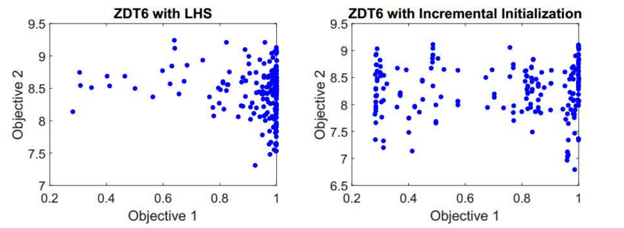


Figure 2: Distribution of 200 samples using (a) LHS and (b) incremental initialization.

3.2 Assigning Leader Solutions

Leader solutions L_w for each cluster w are selected based on the high-fidelity fitness function. First, we sort the population according to orthogonal distances from each of the reference directions. Each solution is assigned to the nearest reference direction. Thus, there are $|W|$ possible clusters, although some clusters may not contain any member. We assign the best solution for each cluster to be the leader of that cluster. This leader is then used for niching in N-RGA by using the nearest leader in the variable space.

3.3 Creating Surrogate Model

As described before, we used two multi-modal selection function $\mathcal{S}(.)$ that can constitute multiple optima, each for a specific Pareto-optimal solution. Each reference vector $w \in W$ targets one global optimum in general. To make the model more accurate, only solutions which are evaluated exactly are used for model construction. In this study, we use either Kriging or Artificial Neural Network method to perform this step. We discuss specific parameters for the surrogate modeling methods in the next section. The choice of our models is two-fold. First, we want to investigate the performance with a structured modeling procedure (Kriging). Knowing the fact that a Kriging method may be difficult to model a multi-modal landscape, we include an ANN method to model such a complex landscape, particularly motivated by the recent progress on deep learning [11] method's ability to model any arbitrarily complex relationship. The comparison between Kriging and ANN in the context of M6 metamodeling approach is an important part of this study.

3.4 Niched Real-parameter Genetic Algorithm (N-RGA)

To solve a multimodal problem for finding multiple optimal solutions, we also need an optimization algorithm which is capable of finding multiple solutions. We devise a niching based genetic algorithm which preserves solutions from each niche with the help of leader solutions, described above. Objective functions and constraints are provided by Kriging or ANN, whichever is being used. A mating selection is used to restrict parents to be chosen from the same cluster. We use other parameters similar to a standard real-parameter genetic algorithm [8]. At the end of N-RGA run, we pick at most one solution from each cluster for further high-fidelity evaluation. After we finish this cycle, we re-evaluate the leader of each cluster for the next iteration.

4 EXPERIMENTAL RESULTS

In this section, we compare two proposed selection function approaches with two different models – Kriging and ANN. We denote these four combinations as MEMO-NN, MEMO-KR, KKT-NN and KKT-KR, where KR stands for Kriging and NN stands for ANN. Following parameter settings are used for N-RGA: 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. For ANN, we use two hidden layers with 10 and 7 neurons each. These parameters are found by limited trial-and-error runs. We train the model until we get a validation error of 10^{-6} or 150 epochs have elapsed. We use ReLU activation and initialize the weights randomly each time we train the model. For each methodology, we perform 10 runs on all problems. We show the obtained solutions of the median IGD run in each case in order to have a graphical comparison.

4.1 Two-Objective Constrained and Unconstrained Problems

We apply four methods to two-objective unconstrained problems ZDT1, ZDT2, ZDT3 and ZDT6 with ten ($n = 10$) variables and a maximum of only $SE_{\max} = 500$ high-fidelity solution evaluations. For each problem, we have used an initial sample size of $\rho = 200$. The obtained non-dominated solutions for ZDT1, ZDT3 and ZDT6 are shown in Figures 3, 4 and 5. It is clear from the figures that KKT-NN and MEMO-NN are able to solve ZDT1 problems. The obtained points are very close to the respective true Pareto-optimal fronts and have a good distribution of points on the entire front. Table 1 shows the average and standard deviation of IGD metric values for all methodologies. In ZDT2 problem, KKT-NN perform the best. For ZDT3 problem, which has a disconnected Pareto optimal front, MEMO-NN perform the best, followed by MEMO-KR. The multi-modal KKTPM-based NN or KR approach is not able to get quite close to the true Pareto-optimal front with only 500 evaluations. This is due to the added complexity the multi-modal KKTPM surface offers to the metamodeling approach when dealing with a disconnected Pareto-optimal front. As mentioned before, ZDT6 is a variable density problem and all the methods find it hard to get to optimal front within limited function evaluation. MEMO-NN shows best performance comparing with other approaches. It is clear that methodology MEMO-NN performs the best on the unconstrained two-objective test problems. Methodologies having statistically insignificant performance from the best performing method in each problem are also marked in bold with the respective p-value in Wilcoxon signed-ranked test.

Next, we apply our methods to two-objective constrained problems: BNH, SRN, TNK, and OSY [8], For each problem an initial sample of size $\rho = 300$ with $SE_{\max} = 800$. The obtained non-dominated solutions of BHN, SRN and TNK are shown in Figures 6, 7 and 8 respectively. All four methods are able to find a close and well-distributed set of trade-off points to true Pareto-optimal front for BNH and SRN problems except MEMO-NN. It gets worse performance in BNH problem while performs the best for SRN.

Table 1: Computed IGD values for test and real world problems.

Algorithm	MEMO-NN		MEMO-KR		KKT-NN		KKT-KR	
Problem	Average	SD	Average	SD	Average	SD	Average	SD
ZDT 1	**0.0192**	0.0165	0.0244	0.0057	**0.0098**	0.0040	0.0249	0.0153
	p=0.2123		p=0.000182		-		p=0.0211	
ZDT 2	0.0112	0.0141	0.0389	0.0310	**0.0057**	0.0013	0.0103	0.0051
	p=0.0376		p=0.0001865		-		p=0.0312	
ZDT 3	**0.0558**	0.0323	**0.0385**	0.0202	0.3211	0.2310	0.7111	0.2054
	p=0.1859		-		p=0.000439		p=0.000182	
ZDT 6	**0.1411**	0.0379	0.6290	0.1931	5.2591	0.9046	5.9401	0.7896
	-		p=0.000182		p=0.000182		p=0.000182	
BNH	1.4410	0.7354	**0.5899**	0.3524	**0.4668**	0.1538	**0.4177**	0.1852
	p=0.0010		p=0.1620		p=0.2730		-	
SRN	**1.1180**	0.1992	2.1175	0.3092	1.5061	0.5474	1.5824	0.2557
	-		p=0.000182		p=0.0173		p=0.0017	
TNK	**0.0398**	0.0050	**0.0403**	0.0056	0.0591	0.0156	**0.0388**	0.0038
	p=0.7337		p=0.4274		p=0.0010		-	
OSY	**35.5800**	8.0531	**30.8750**	8.2780	50.8310	13.8701	39.7462	8.7765
	p=0.1403		-		p=0.0022		p=0.0257	
Welded Beam	**1.0850**	0.3502	**1.589**	0.7212	**1.3020**	0.3771	2.5987	1.1322
	-		p=0.1405		p=0.2730		p=0.0010	
C2DTLZ2	**0.0650**	0.0105	**0.0607**	0.0117	0.1907	0.0108	0.1037	0.0268
	p=0.2413		-		p=0.000182		p=0.000582	
DTLZ2	0.0517	0.0120	0.0656	0.0100	0.1316	0.0489	**0.0345**	0.0043
	p=0.000182		p=0.000182		p=0.000182		-	
DTLZ4	**0.0775**	0.0406	0.1675	0.0453	0.2651	0.0577	0.3828	0.0277
	-		p=0.0013		p=0.000246		p=0.000182	
DTLZ5	0.0208	0.0053	0.0271	0.0041	0.1149	0.0115	**0.0057**	0.0034
	p=0.000246		p=0.000182		p=0.000182		-	
Car Side	0.4796	0.0722	0.4806	0.0959	**0.3953**	0.1852	**0.3915**	0.08211
	p= 0.0257		p=0.0376		p=0.4274		-	

Although TNK has provided difficulties to all four methods due to discontinuities in its Pareto-optimal front, all methods come close to the true front. In this problem, KKT-KR performs the best, followed by MEMO-NN, and MEMO-KR. In OSY, none of all methods perform well because of the increased number of constraints.

Next, a two-objective welded-beam design problem [8] is solved using four methods using $SE_{\max} = 1,000$. Table 1 shows that, MEMO-NN performs the best followed by MEMO-KR and KKT-NN while KKT-KR finds it difficult to find minimum-cost (f_1) solutions. This real-world problem requires more high-fidelity solution evaluations to find the near Pareto-optimal solutions.

4.2 Three-Objective Constrained and Unconstrained Problems

We apply four methods to three-objective unconstrained (DTLZ2, DTLZ4, and DTLZ5) and constrained (C2DTLZ2) test problems. Each of these problems has seven variables. We fix $SE_{\max} = 1,000$ for DTLZ2 and DTLZ5, and $SE_{\max} = 2,000$ for DTLZ4 due to multi-modality in its landscape. For C2DTLZ2, we have used $SE_{\max} = 1,500$. Figures 9, 10, and 11 show the Pareto-optimal surface of DTLZ2, DTLZ4 and C2DTLZ2 problems with obtained non-dominated solutions. First, KKT-KR performs the best on DTLZ2 then MEMO-NN and MEMO-RK, while KKT-NN finds it difficult to solve this problem. For DTLZ4, MEMO-NN performed the best while other methods did not perform well compare to MEMO-NN. DTLZ4 is a variable density and multi-modal problem which is very hard to solve with lower budget. But MEMO-NN is able to create a well distributed set of solutions with only 2,000 solution evaluations. On DTLZ5, KKT-KR algorithm performs the best followed by MEMO-NN and MEMO-KR. Pareto-optimal solutions of DTLZ5 lies on a curve and every method performs well on this problem.

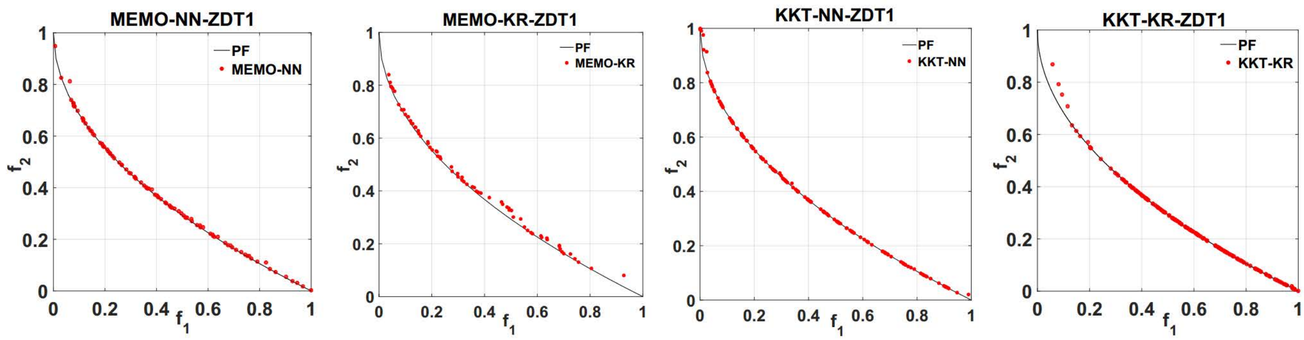


Figure 3: Non-dominated solutions for problem ZDT1 using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

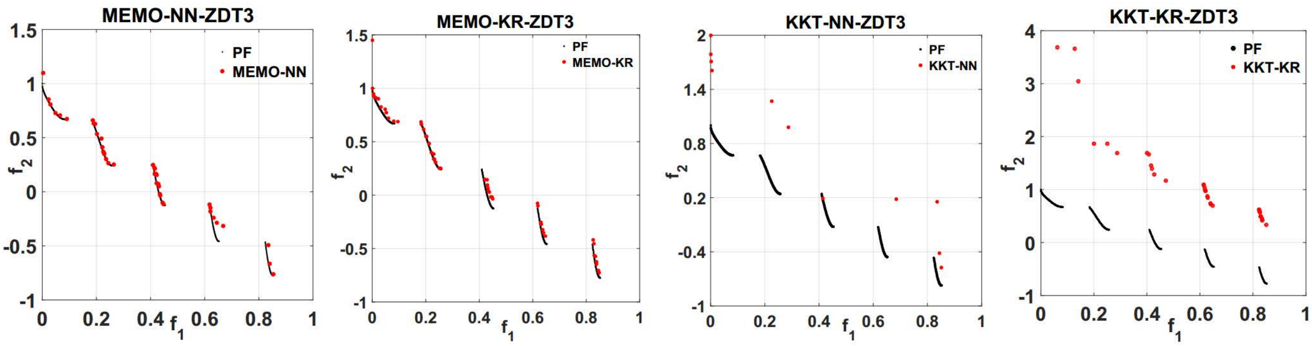


Figure 4: Non-dominated solutions for problem ZDT3 using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

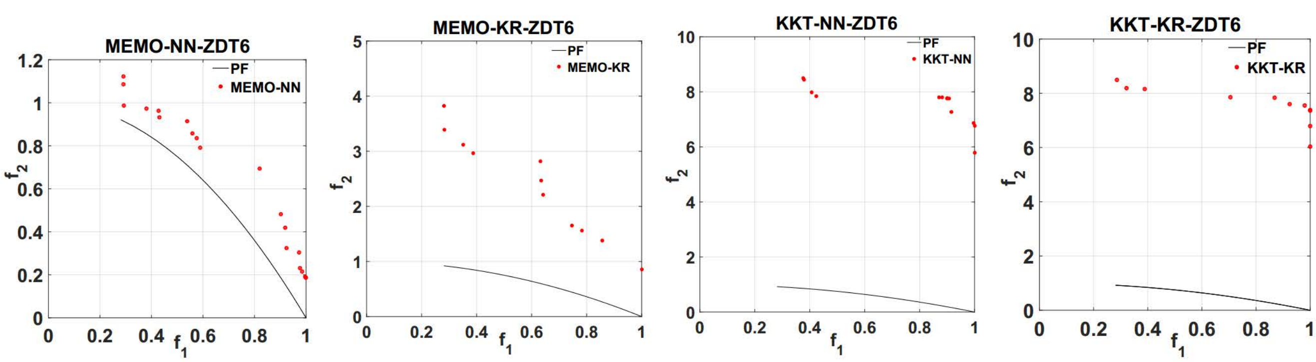


Figure 5: Non-dominated solutions for problem ZDT6 using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

On C2DTLZ2 problem, Algorithm MEMO-KR and MEMO-NN perform well and finds a close and well-distributed set of trade-off points near true Pareto-optimal front while other algorithms did not converge close enough. Finally, we apply four methods to a car side-impact problem having three objectives and 10 constraints with $SE_{\max} = 2,000$. Algorithm KKT-KR and KKT-NN performs the best on this problem, while MEMO-NN and MEMO-KR methods are also able to find a widely distributed and well-converged set of solutions.

From the above results, it is clear that the use of MEMO selection method and ANN as a metamodeling technique makes a good combination and it performs well on most of the test and real world multi-objective problems. KKTPM-based method performs well if constraints are linear. Neural network tends to perform relatively better when number of objective increases from 2 to 3. MEMO approach provides a better distribution of solutions in objective space than KKTPM based approach.

5 CONCLUSIONS

In this paper, we have proposed and evaluated a selection function based metamodeling methodology for multiobjective optimization. Recently proposed KKTPM and MEMO-based selection functions have been compared against each other for this purpose. Also, the efficacy of two metamodeling techniques (ANN and Kriging) on the above selection function approaches has been investigated. On a number of two and three-objective, constrained and unconstrained

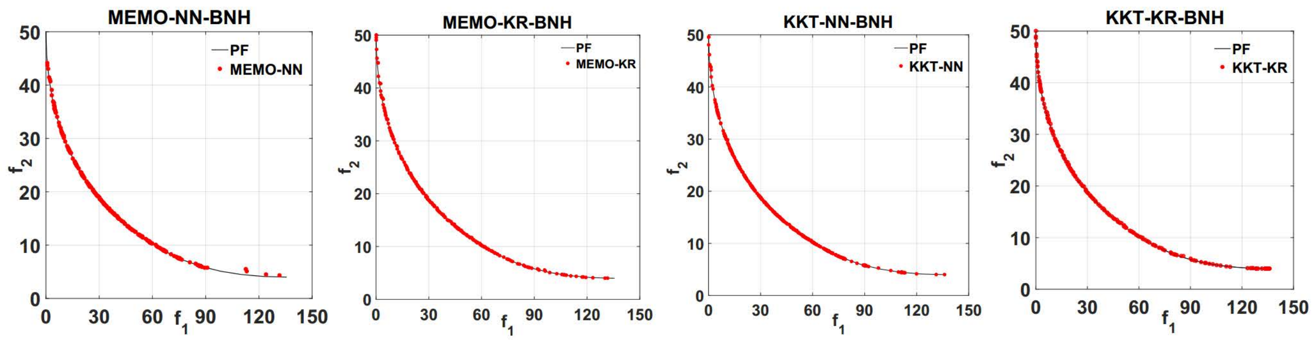


Figure 6: Non-dominated solutions for problem BNH using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

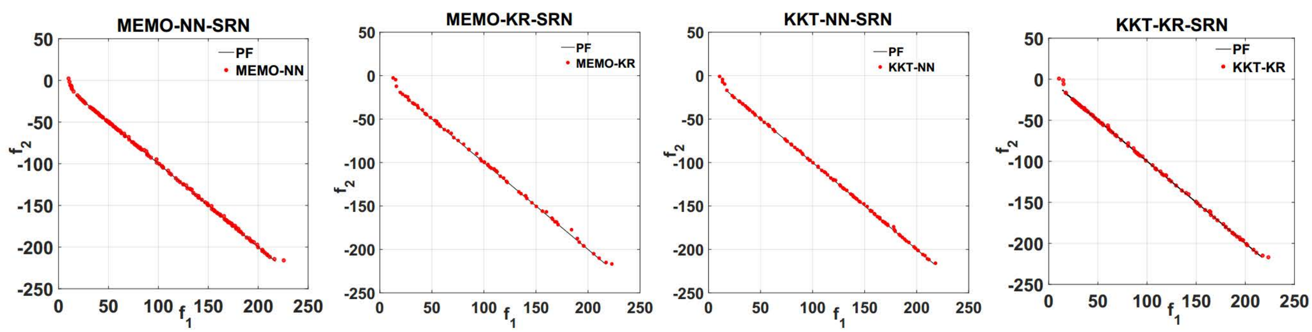


Figure 7: Non-dominated solutions for problem SRN using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

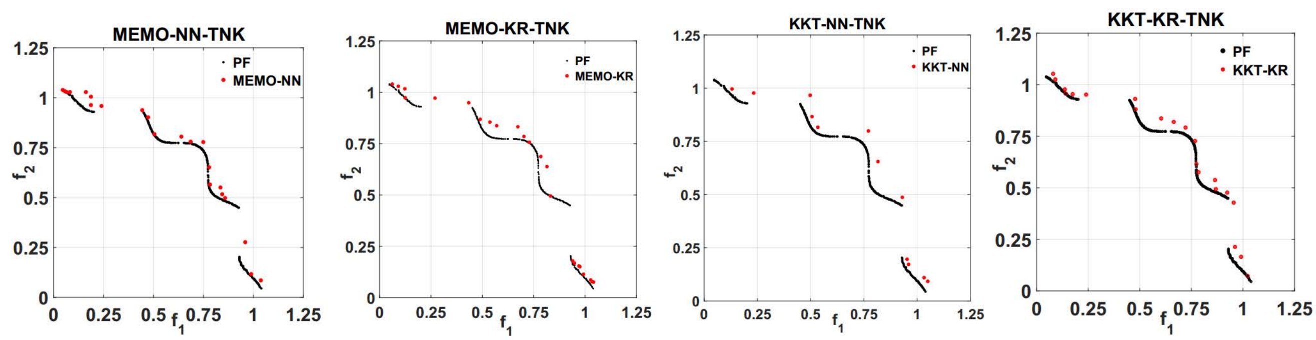


Figure 8: Non-dominated solutions for problem TNK using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

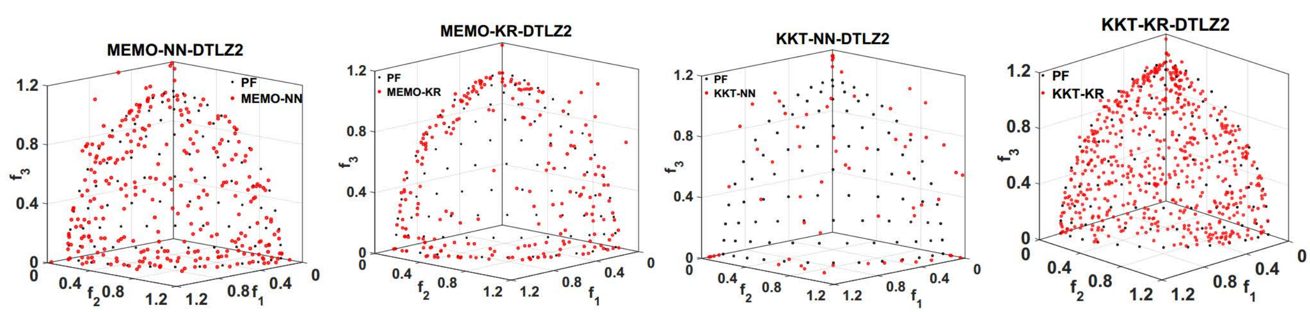


Figure 9: Non-dominated solutions for problem DTLZ2 using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

optimization problems including two engineering design problems, our extensive results has clearly shown that (i) MEMO-based selection function approach is better and (ii) ANN metamodeling

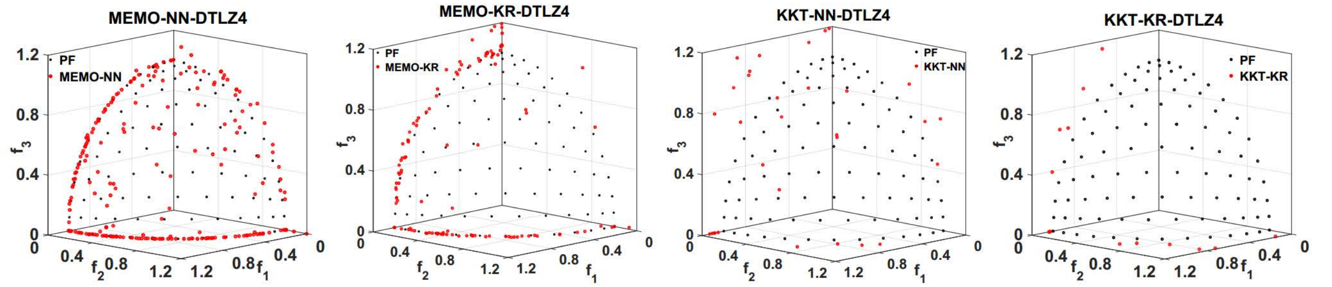


Figure 10: Non-dominated solutions for problem DTLZ4 using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

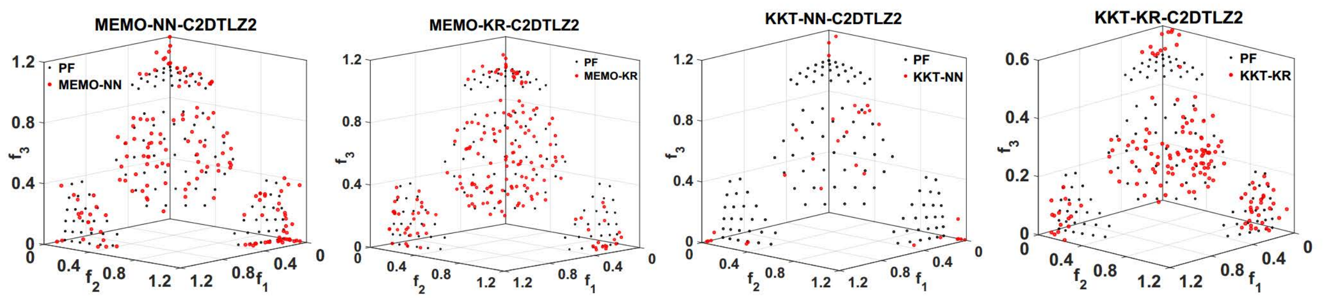


Figure 11: Non-dominated solutions for problem C2DTLZ2 using MEMO-NN, MEMO-KR, KKT-NN, and KKT-KR.

technique is, in general, better able to approximate multimodal selection function landscape portrayed by the overall approach. It is important to highlight that in solving the above problems, only a fraction of solutions evaluations (limited to 500 to 2,000) were allowed, compared to hundreds of thousands of solution evaluations usually set for non-metamodeling based EMO studies. Based on these encouraging results, we plan to pursue a number of further studies: (i) use of different architectures and deeper ANNs for better accuracy models, (ii) use of other selection functions modeling after successful EMO methods and (iii) extend the approach to many-objective optimization problems.

REFERENCES

[1] B. Beachkofski and R. Grandhi. 2002. Improved Distributed Hypercube Sampling. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA*. ARC, Denver, Colorado, 1–7.

[2] I. Das and J.E. Dennis. 1998. Normal-Boundary Intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal of Optimization* 8, 3 (1998), 631–657.

[3] Kalyanmoy Deb, Mohamed Abouhawwash, and Joydeep Dutta. 2015. An optimality theory based proximity measure for evolutionary multi-objective and many-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, Springer International Publishing, Cham, 18–33.

[4] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. 2002. A fast and Elitist multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[5] K. Deb and R. Datta. 2010. A fast and accurate solution of constrained optimization problems using a hybrid bi-objective and penalty function approach. In *IEEE Congress on Evolutionary Computation*. IEEE, Barcelona, Spain, 1–8.

[6] Kalyanmoy Deb, Rayan Hussein, Proteek Roy, and Gregorio Toscano. 2017. *Classifying Metamodeling Methods for Evolutionary Multi-objective Optimization: First Results.* Springer International Publishing, Cham, 160–175.

[7] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (2014), 577–601.

[8] Kalyanmoy Deb and Deb Kalyanmoy. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc., New York, NY, USA.

[9] M.T.M. Emmerich, K.C. Giannakoglou, and B. Naujoks. 2006. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation* 10 (2006), 421–439.

[10] M. Emmerich, K. Giannakoglou, and B. Naujoks. 2006. Single and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels. *IEEE Transactions on Evolutionary Computation* 10, 4 (aug. 2006), 421–439.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT Press, Cambridge, MA, USA. http://www.deeplearningbook.org.

[12] Rayan Hussein and Kalyanmoy Deb. 2016. A Generative Kriging Surrogate Model for Constrained and Unconstrained Multi-objective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16)*. ACM, New York, NY, USA, Article -, 8 pages.

[13] Yaochu Jin. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1, 2 (2011), 61–70.

[14] Donald R Jones. 2001. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization* 21, 4 (2001), 345–383.

[15] J. Knowles. 2006. ParEGO: A Hybrid Algorithm with On-line Landscape Approximation for Expensive Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation* 10, 1 (January 2006), 50–66.

[16] K. Miettinen. 1999. *Nonlinear Multiobjective Optimization.* Kluwer, Boston, New York, NY, USA.

[17] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. 2008. *Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted S-Metric Selection.* Springer Berlin Heidelberg, Berlin, Heidelberg, 784–794.

[18] P. C. Roy and K. Deb. 2016. High dimensional model representation for solving expensive multi-objective optimization problems. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, Vancouver, Canada, 2490–2497.

[19] R. E. Steuer. 1986. *Multiple Criteria Optimization: Theory, Computation and Applica- tion.* New York: Wiley, New York, NY, USA.

[20] Cem C. Tutum and Kalyanmoy Deb. 2015. *A Multimodal Approach for Evolutionary Multi-objective Optimization (MEMO): Proof-of-Principle Results.* Springer International Publishing, Cham, 3–18.

[21] L. Willmes, T. Back, Yaochu Jin, and B. Sendhoff. 2003. Comparing neural networks and Kriging for fitness approximation in evolutionary optimization. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, Vol. 1. IEEE, Canberra, Australia, 663–670 Vol.1.

[22] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. 2010. Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model. *IEEE Transactions on Evolutionary Computation* 14, 3 (2010), 456–474.