

A Multiple Surrogate Assisted Decomposition-Based Evolutionary Algorithm for Expensive Multi/Many-Objective Optimization

Ahsanul Habib¹, Hemant Kumar Singh¹, Tinkle Chugh¹, Tapabrata Ray¹, and Kaisa Miettinen²

Abstract—Many-objective optimization problems (MaOPs) contain four or more conflicting objectives to be optimized. A number of efficient decomposition-based evolutionary algorithms have been developed in the recent years to solve them. However, computationally expensive MaOPs have been scarcely investigated. Typically, *surrogate-assisted* methods have been used in the literature to tackle computationally expensive problems, but such studies have largely focused on problems with 1–3 objectives. In this paper, we present an approach called hybrid surrogate-assisted many-objective evolutionary algorithm to solve computationally expensive MaOPs. The key features of the approach include: 1) the use of multiple surrogates to effectively approximate a wide range of objective functions; 2) use of two sets of reference vectors for improved performance on irregular Pareto fronts (PFs); 3) effective use of archive solutions during offspring generation; and 4) a local improvement scheme for generating high quality infill solutions. Furthermore, the approach includes constraint handling which is often overlooked in contemporary algorithms. The performance of the approach is benchmarked extensively on a set of unconstrained and constrained problems with regular and irregular PFs. A statistical comparison with the existing techniques highlights the efficacy and potential of the approach.

Index Terms—Computational cost, metamodels, multiobjective optimization, reference vectors.

I. INTRODUCTION

REAL-WORLD problems often require simultaneous optimization of multiple conflicting objectives. A few (of numerous) application areas include control system design [1], automotive design [2], and aviation [3]. Such problems are referred to as multiobjective optimization problems (MOPs). MOPs with four or more objectives can be further subcategorized as many-objective optimization problems (MaOPs) due to additional challenges they pose to optimization methods [4].

Multiobjective evolutionary algorithms (MOEAs) are a common choice to solve MOPs owing to their versatility in handling problems that typically do not possess smooth mathematical properties [5], i.e., they are highly nonlinear, nondifferentiable, discontinuous, or even *black-box* in nature. Being population-based means that they generate a set of tradeoff solutions; referred to as a Pareto front (PF) in the objective space and Pareto set in the decision space. However, MOEAs in general require several generations and consequently excessive number of function evaluations in order to converge to satisfactory solutions. This evidently makes their application untenable for problems where each function evaluation requires a computationally expensive simulation [6] such as computational electro-magnetics, computational fluid dynamics, and finite element analysis. Despite a proliferation of studies on MaOPs in the recent years, very few have considered problems, where the number of functional evaluations are to be severely restricted in order to obtain a solution in a reasonable amount of time.

On the other hand, there have been substantial efforts in the surrogate-assisted optimization domain in handling computationally expensive problems [6]. In a surrogate assisted approach, an expensive function is replaced by a surrogate model (also known as a metamodel or an approximation model) which is cheap to evaluate. Predominantly, such approaches have been developed for single-objective optimization [6]. Recently, there has been an increasing number of studies exploring their use for MOPs [7], but most of the strategies are not efficiently scalable for MaOPs. Only recently have dedicated techniques for dealing with MaOPs emerged [8]–[10] that use surrogates, such as Kriging, neural networks (NNs), or a combination of different types of surrogates. The approaches have typically dealt with so called “regular” problems. Regularity in this case entails that when using decomposition-based optimization techniques, the PFs of

Manuscript received April 27, 2018; revised August 15, 2018, October 9, 2018, and December 13, 2018; accepted February 3, 2019. Date of publication February 12, 2019; date of current version November 27, 2019. This work was supported in part by the Natural Environment Research Council under Grant NE/P017436/1. The work of A. Habib was supported by the Australian Government via “Research Training Program Scholarship” and the travel grant received from the University of Jyväskylä (funding of Prof. K. Miettinen). The work of T. Ray was supported by Australian Research Council under Grant DP190101271. This paper is related to the thematic research area DEMO of the University of Jyväskylä. (Corresponding author: Ahsanul Habib.)

A. Habib, H. K. Singh, and T. Ray are with the School of Engineering and Information Technology, University of New South Wales, Canberra, Australia (e-mail: a.habib@student.adfa.edu.au; h.singh@adfa.edu.au; t.ray@adfa.edu.au).

T. Chugh was with University of Jyväskylä, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland. He is now with the Department of Computer Science, University of Exeter, Exeter EX4 4QF, U.K. (e-mail: t.chugh@exeter.ac.uk).

K. Miettinen is with University of Jyväskylä, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland (e-mail: kaisa.miettinen@jyu.fi).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author.

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.2019.2899030

such problems can be well mapped using a set of uniformly sampled points [and corresponding reference vectors (RVs)] generated through the normal boundary intersection (NBI) method [11]. However, as discussed in recent studies [12], a simple inversion of the PF can deteriorate the performance of this strategy significantly. Hence, there is a further need to design and test algorithms across a range of problems with a greater variety in the nature of PFs.

In order to address the above research gaps, this paper is set out with an aim of developing an efficient decomposition-based algorithm for solving real-world MaOPs where the shape of the PF is unknown using a very limited number of function evaluations. The key developments undertaken include the use of multiple surrogates, two sets of RVs and local improvement for an effective identification of *infill solutions* selected to undergo actual function evaluation. Further, the algorithm efficiently uses an archive of evaluated solutions when generating offspring solutions and includes constraint handling which is lacking in existing algorithms.

Next, the background and related work are discussed in Section II, followed by a description of our algorithm in Section III. Numerical experiments are detailed in Section IV and include benchmarking against existing algorithms on an extensive set of unconstrained (conventional and inverted) and constrained problems as well as three practical MaOPs. The conclusions and future directions are discussed in Section V.

II. BACKGROUND

In this section, we review the works that are relevant to this paper with a focus on three key areas. The first relates to surrogate-assisted optimization, in particular the choice of surrogate model(s), selection of training data, use of multiple surrogate models, and the choice of the performance metric(s) during the evolutionary search. The second relates to the adaptation (of RVs, aggregation functions) within the context of decomposition-based algorithms in order to handle irregular PFs. The last one relates to a local improvement of infill solutions, i.e., measures taken to increase the probability that a new solution selected for evaluation will lead the search in a favorable direction.

A. Surrogate-Assisted MOEAs

As mentioned previously, surrogate-assisted approaches are a popular choice for handling computationally expensive optimization problems [6], [13]. They reduce the runtime of the optimization task significantly by approximating the underlying expensive function(s). The surrogate models can be combined with evolutionary algorithms to yield surrogate-assisted evolutionary algorithms (SAEAs) [14] that have been often employed to solve single objective expensive optimization problems [6]. Recently, there has been a growing trend toward employing surrogate models within MOEA frameworks, as evident from the review papers [7], [15].

Surrogate models can be employed for solving MOPs in various ways. Some prominent approaches include predicting the Pareto-rank of a solution [16], [17], pairwise dominance comparison [18], approximating individual,

or scalarized functions [19], [20]. For approximating individual objective/constraint function(s), Gaussian process regression/Kriging [21] has been a popular choice, despite the computational overhead in its training time with an increasing number of data points and decision variables. The popularity can be attributed to the fact that besides predicting the value, Kriging also provides the associated confidence bounds (or limit of uncertainty) which can, in-turn, be used to calculate the probability of improvement and the expected improvement (EI) metrics [22]. EI maximization-based efficient global optimization (EGO) has been successfully implemented in MOEA/D-EGO [23] and ParEGO [20] for solving MOPs. Both of these methods use a Chebyshev method [24] to compute an aggregated function value along a given RV. The former one runs MOEA/D in the background to maximize the EI of the aggregated objective along a particular RV and the Kriging model is updated to obtain a set of ND solutions from which a predefined number of infill solutions is selected. ParEGO maximizes the EI of the aggregated function along a randomly chosen RV to obtain a single infill solution in each iteration.

Apart from Kriging-based approaches, there have been studies which used other surrogate models like NNs [25], k -nearest neighbor-based NN [26], support vector regression [27], polynomial response surface method (RSM) [28], and radial basis functions (RBFs) [29]. While the above-mentioned studies employ only a single type of surrogate model for function approximation in the multiobjective context, there have also been studies that fit the best among multiple surrogate models for function approximation. For instance, Husain and Kim [30] used Kriging, RSM, and RBF while Montano *et al.* [31] employed RBF with different basis functions as multiple surrogate models. Recently, in [19], a novel approach utilizing Kriging, RBF, multilayer perceptrons, and RSM was proposed which adaptively constructs spatially distributed surrogate models depending on the accuracy of a particular surrogate model within a design neighborhood in order to capture the function landscape accurately. The algorithm has been recently extended to many-objective (MaO) domain [10] with a decomposition-based approach. There have also been studies that used multiple surrogate models as an ensemble to approximate the function, such as [32].

Although in principle, some of the prominent existing approaches, such as MOEA/D-EGO [23], ParEGO [20], CPS-MOEA [26], and SMS-EGO [33] can be applied to solve MaOPs, scarce attention has been paid toward developing dedicated algorithms to handle MaOPs so far. A few works have emerged recently to handle this challenge, such as classification-based approach using a feed-forward NN (CSEA) [9] and Kriging-based K-RVEA [8] (which has a constrained version cK-RVEA [34]). In Table I, we summarize the strengths and limitations of the algorithms and the problems they have been tested with. The key limitations of the above methods include absence of constraint handling, use of Pareto ranking (not efficient for MaOPs), use of computationally expensive metrics, such as HV, use of a single set of RVs (not very successful for irregular problems [35]), and a number of additional user-defined parameters.

TABLE I
SUMMARY OF NOTABLE EXISTING WORKS INVOLVING SURROGATE-ASSISTED MULTI/MANY-OBJECTIVE OPTIMIZATION APPROACH

Algorithm	Strengths	Limitations	Problems studied
ParEGO [20]	The MOP is scalarized using Chebyshev method into a single-objective problem along a randomly selected RV in each generation. Kriging surrogate model is trained on the scalarized function. Since the surrogate models are not built for each individual objective function, the algorithm is less memory intensive, quick to run and extendable to MaOPs	Slow convergence due to a steady-state nature, random selection of RV in each generation may not assure required diversity and use of a fixed number of training points to limit computational cost in model building is problem dependent and more so when the overall budget is limited, no constraint handling method is discussed	KNO1 (2-variables, 2 objectives), OKA1-2 (3 variables, 2 objectives), VLMOP2-3 (2 variables, 2 and 3 objectives), DTLZ1a (modified, 6 variables, 2 objectives), DTLZ2a, DTLZ4a and DTLZ7a (modified, 8 variables, 3-objectives)
SMS-EGO [33]	The use of \mathcal{S} -metric is somewhat beneficial in enhancing convergence by promoting the offspring with the higher HV contribution calculated from the predicted objective values from Kriging	HV calculation is itself computationally prohibitive for MaOPs, no constraint handling method is discussed	OKA2 (2 variables, 3 objectives), R_ZDT1 (6 variables, 2 objectives), R_ZDT4 _{relax} (3 variables, 2 objectives) and R_DTLZ2 (6 variables, 3 and 5 objectives)
MOEA/D-EGO [23]	Scalable to MaOPs, clustering is done in the decision space to reduce training samples in model building	Depends on various user-defined parameters, with a limited computing budget, training surrogate models in different overlapping clusters may introduce unnecessary computational overhead with no enhanced prediction accuracy, no constraint handling method is discussed	KNO1 (2 variables, 2 objectives), ZDT1-4.6 (8 variables, 2 objectives), F1-F4 [36] (8 variables, 2 objectives) and DTLZ2 (6 variables, 3 objectives)
CPS-MOEA [26]	Scalable to MaOPs, classifier based pre-selection excludes evaluation of potential worse offspring solutions	Inherits the drawbacks of ND sorting when extended to MaOPs, classifier performance depends on the number of neighboring solutions during prediction which is a user defined parameter, no infill selection mechanism is adopted to limit the number of function evaluations, no constraint handling method is discussed	F1-10 [37] (30 variables, 2 objectives)
K-RVEA [8]/ cK-RVEA [34]	Less memory intensive due to limited training set which is equal to the initial population, novel and efficient infill selection technique which utilizes the Kriging's prediction error confidence bound to select infill solutions from less explored regions	Relies on several user defined parameters, fixed number of training points is arguable with a limited computational budget, limitations in solving problems with inverted fronts due to employment of a conventional RV guided approach	DTLZ1-7 (10 variables, 3-10 objectives), WFG1-9 (9-11 variables, 3-10 objectives), free-radical polymerization of vinyl acetate [38] (4 variables, 3 objectives)
CSEA [9]	Less memory intensive due to training only a single classifier to predict good or bad (closer or away from PF) solutions instead of training surrogate models for each objective or constraint function	Several user-defined parameters, lower accuracy in solving MaOPs due to radial projection in lower objective space, shares the similar drawbacks of ND sorting based approaches for MaOPs, no constraint handling method is discussed	DTLZ1-7 (10 variables, 3-10 objectives), WFG1-9 (9-11 variables, 3-10 objectives), MaF1-5 (10 variables, 3-10 objectives) and a car cab design problem [39] (11 variables, 9 objectives)

B. Adaptation for Dealing With Irregular PFs

Recently, there has been a growing trend toward adapting RVs during the course of search in order to closely approximate different shapes of the PF, such as “regular,” disconnected, degenerated, inverted, or strongly convex/concave PFs. Adaptation can be on the aggregation function or the RVs itself. A notable implementation of an adaptation of aggregation functions can be found in [40], while studies on RV adaptation appear in [2] and [41]–[43]. However, limited work has been done regarding the effect of the *reference point(s)* on the search performance. The term “reference point” has been used in different ways in the literature. For example, in [39] it refers to the points generated on the hyperplane through NBI, whereas in [44] it refers to the point from which the RVs originate. Here, we refer to the latter meaning. While the common practice is to use a set of RVs originating from the ideal point (formed by best objective function values available), it is mostly effective when the solution diversity is relatively easy to maintain [12]. The effect of reference point specification in the context of MOEA/D was studied in [44]. Some attempts have also been made to use the nadir point (formed by worst objective function values in the current ND set) as a reference point for generating RVs [45] which proved to be effective for solving problems with “inverted” PFs. Recently, some studies have suggested to use two sets of RVs, one originating from the ideal and the other one emerging from the nadir point [35], [46]–[48]. These studies

demonstrate the usefulness of a dual set of RVs compared to some of the most popular algorithms (having only a set of RVs originated from the ideal point) without RV adaptation, e.g., MOEA/D [49], NSGA-III [39], θ -DEA [50], among others and with RV adaptation, e.g., MEAD/D-AWA [41] and RVEA [51].

It is also important to note that all such schemes have their own considerations and additional parameters; for example, how often to adapt the directions, whether to consider deletion of one RV at a time or more, whether to have a mechanism to bring back/reinstate original RVs if they are incorrectly deleted, etc. Most importantly, it takes a significant amount of time to adapt the RVs to resemble the true shape of the PF, since the adaptation typically progresses by insertion/deletion of only up to a few RVs at a time.

Beside the adaptation of the RVs, there is another class of algorithms which focuses on complicated PF shapes with degenerated and disconnected PFs. These algorithms focus on identifying redundant objectives via principal component analysis or linear/nonlinear correlation between objectives [52], Pareto corner sorting [53], clustering [54], or a probability model-based estimation of distribution algorithms which are able to discover regularity models in the solution space besides the objective space [37], [55], [56]. However, to be able to discover the correlation among different objectives or for discovering the regularity model in the decision space, the models need to be trained with a large number of samples, i.e.,

actually evaluated Pareto optimal solutions. Additionally, the works [52], [53], [57] are more focused on objective reduction (instead of directly solving the problem in the original M -objective space), whereas the studies in [37], [55], and [58] focus on only up to three-objective problems while only [56] recently extended the study of [37] into an MaO domain.

None of the above studies, however, consider the problem to be computationally expensive, which subsumes that a reasonable amount of time (i.e., function evaluations) is generally available for learning and adaptation. Hence, employing RV adaptation strategies or employing objective reduction/estimation of distribution algorithms for solving computationally expensive problems is not straightforward as they need to be substantially customized to suit the low computation budget paradigm.

C. Local Search

Within the general framework of evolutionary algorithms, there are some studies which capitalize on a local improvement/search mechanisms for achieving faster convergence. The prominent choices/challenges include whether to improve all solutions or some of the solutions, how to select the solutions, and the local search strategy itself. Hybridization for decomposition-based MOPs are aimed at improving solutions of a particular subproblem with the help of a certain aggregation method.

Local search was used to improve all generated offspring solutions in [59] and [60]. Some studies have attempted to improve a few promising solutions with the choice guided by probability in [61] and objective space clustering in [62]. Other approaches include use of local search at initial stages and use of EA-based search when the solutions are sufficiently closer to the PF [63], use of stand-alone state-of-the-art multiobjective nonlinear simplex search [64], [65], and use of local search guided by various forms of fitness approximations [32], [66], [67]. However, a majority of these studies only focuses on MOPs, except [10], [62], and [63] which discuss problems with up to four objectives [62], [63] and ten objectives [10].

III. PROPOSED ALGORITHM

In this paper, we propose an algorithm called hybrid surrogate-assisted MaO evolutionary algorithm (HSMEA). The key driving factor in the design of HSMEA is the assumption that each function evaluation is computationally expensive. The implications of this assumption are that the algorithmic “overheads,” such as recombination/selection operators, training of surrogates, local search on surrogates, etc. (which could take, e.g., a few minutes) are considered negligible compared to a true function evaluation (which could take, e.g., hours or days).

Consequently, the number of true function evaluations that can be done during the search is very limited¹ and is the predominant indicator of the optimization runtime. Therefore, the

components of HSMEA are designed to use the archive of already existing solutions efficiently and employ a number of mechanisms (discussed shortly) to improve the possibility that the next solution(s) selected for evaluation bring significant improvements in the objective functions. At the same time, HSMEA is also intended to deal with problems with a range of PF shapes instead of being specialized to regular PFs, as well as deal with both unconstrained and constrained MaOPs. Therefore, it also includes strategies to deal with these features of MaOPs efficiently. In achieving these goals, some inspirations are taken from existing works that have addressed parts of the problem; such as K-RVEA [8], MOEA/D-SQA [60], and MOEA/D-MR [35]. The notable features of HSMEA include the following.

- 1) Multiple types of surrogates are used to approximate the objectives in order to strengthen the ability to approximate a wide(r) range of functions, compared to the existing MaOP algorithms where only a single type of a surrogate is used. At any given point in the search, the surrogate that most closely approximates the current data set is used for prediction.
- 2) Two sets of RVs are used. Solutions are assigned to each set of RVs separately and the set which results in a better s -energy (a measure of diversity) is chosen. This is done in order to make the algorithm more flexible in dealing with irregular PFs compared to existing algorithms which use only a single set of RVs.
- 3) A local improvement scheme (utilizing local search subject to angle constraints) is employed to improve the infill solutions, based on the Euclidean distance (ED) metric. This is done in order to improve the likelihood of the offspring exhibiting an improved performance when truly evaluated (since each evaluation is expensive).
- 4) An ϵ -constraint handling is embedded in the algorithm to deal with constraints.
- 5) Lastly, the algorithm uses the solutions from an archive \mathcal{A} of truly evaluated solutions effectively to generate offspring solutions.

Now, we can present the problem formulation considered and describe the details of the approach. Formally, a generic MOP can be defined as

$$\begin{aligned} & \underset{(x)}{\text{minimize:}} \quad (f_1(x), f_2(x), \dots, f_M(x)) \\ & \text{subject to:} \quad x_L \leq x \leq x_U \\ & \quad \quad \quad g_a(x) \leq 0, a = 1, \dots, p \\ & \quad \quad \quad h_b(x) = 0, b = 1, \dots, q \end{aligned} \quad (1)$$

where $x \in \mathbb{R}^D$, where D is the number of variables of the problem, x_L and x_U are upper and lower bounds of the variables, and f_1, f_2, \dots, f_M are M objective functions to be optimized subject to p inequality and q equality constraints.

For solving the above defined problems, an overview of HSMEA is shown in Algorithm 1, followed by the details of its key components. The pseudocodes for different components of HSMEA are presented in Section V of the supplementary file provided.

¹The exact quantification of “limited” can vary upon the time taken for a function evaluation and the total time available for optimization for a given problem.

Algorithm 1 HSMEA

Input: FE_{max} = Maximum number of function evaluations, N_W = Number of RVs in the two sets W_0^{min} and W_0^{max} , N_{init} = Number of initial solutions, N_{is} = Maximum number of infill solutions, N_{sgt} = Number of surrogate models.

Output: Non-dominated solutions from Archive \mathcal{A} .

- 1: **Generate** two sets of RVs, W_0^{min} and W_0^{max} using NBI method.
- 2: **Initialize** population P_{init} where, $|P_{init}| = N_{init}$.
- 3: $FE = |P_{init}|$.
- 4: **Update** Archive \mathcal{A} of the actually evaluated solutions and allowable constraint violation for epsilon level comparison, ϵ_{CV} .
- 5: **Construct** N_{sgt} surrogate models for each objective (and constraint function if expensive) based on \mathcal{A} .
- 6: **Update** the minimum and maximum objective values z_{min} and z_{max} based on the feasible solutions in the current population. Retain the previous values of z_{min} and z_{max} if there is no feasible solution.
- 7: **Adapt** W_0^{min} and W_0^{max} using z_{min} and z_{max} resulting in W_0^{min} and W_0^{max} .
- 8: Parent solutions $P = P_{init}$.
- 9: **while** ($FE \leq FE_{max}$) **do**
- 10: **Generate** offsprings C from P , where, $|C| = N_W$.
- 11: **Predict** objective values and compute constraint violations (CV) for C .
- 12: Current population $R = P + C$.
- 13: **Update** z_{min} , z_{max} and ϵ_{CV} .
- 14: **Adapt** W_0^{min} and W_0^{max} using z_{min} and z_{max} .
- 15: $(I, R) = \text{InfillSelection}(N_{is}, R, W_0^{min}, W_0^{max}, z_{min}, z_{max}, \epsilon_{CV})$
 /* I is the set of infill solutions */
- 16: **Evaluate** solutions in I .
- 17: $R = R + I$.
- 18: $FE = FE + |I|$.
- 19: **Update** Archive \mathcal{A} .
- 20: **Update** Surrogate of all objectives and predictions of all solutions in R .
- 21: **Update** z_{min} , z_{max} and ϵ_{CV} .
- 22: **Adapt** W_0^{min} and W_0^{max} using z_{min} and z_{max} .
- 23: $P = \text{EnvironmentalSelection}(N_W, R, \mathcal{A}, W_0^{min}, W_0^{max}, \epsilon_{CV})$
- 24: **end while**

A. Generation of Reference Vectors and Initialization of Population

As mentioned before, this algorithm uses a dual set of RVs. The first set of RVs W_0^{min} is generated using the NBI method [11] with the origin as the ideal point (best objective values of the population, denoted as z_{min} throughout the paper) with minimum value of each objective for all feasible solutions in the population (0^M in the normalized objective space). The set W_0^{min} consists of N_W points on the hyperplane with a uniform spacing $d = 1/H$ for any number of objectives M with H unique sampling locations along each objective axis. Similarly, the second set of RVs, W_0^{max} is constructed from the point with co-ordinates as the maximum value (worst objective value of the population, denoted as z_{max} throughout the paper) of each objective for all feasible solutions in the population (1^M in the normalized objective space). Please take note that the objective space is normalized based on the minimum and maximum objective values of the population and not the estimated nadir point.

For a three-objective problem, the two sets of RVs are shown in Fig. 1. For problems with more than six objectives, a two-layered approach is followed for both W_0^{min} and W_0^{max}

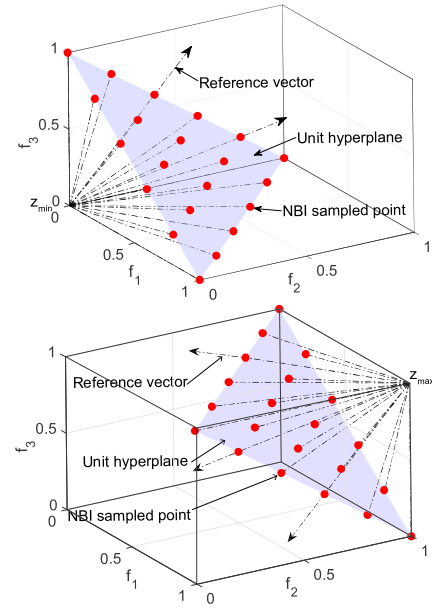


Fig. 1. Sets of RVs originating from z_{min} and z_{max} .

as proposed in [39]. We use the abbreviation DR to refer to a dual reference set (from z_{min} and z_{max}) and SR to refer to a single reference set (from z_{min}).

The size of the initial population is predefined by the user (N_{init}). Solutions are initialized within the variable bounds employing Latin hypercube sampling with a “maximin” criterion [68].

B. Constraint Handling

An epsilon level comparison as introduced in [69] is used in this paper for constraint handling as its performance is often better than that of the feasibility first scheme [70]. A feasibility ratio (FR) is first calculated as the proportion of feasible solutions in the current population. Then, the constraint violation (CV) of each solution in the population is calculated considering all the constraints. Next, the average CV of the population is calculated by taking the mean of all individual CV. Finally, the allowable CV, i.e., ϵ_{CV} is computed with the help of the mean CV and the FR of the population. The solutions having no more than the allowable CV are compared based on their objective values. The CV and the ϵ_{CV} are calculated as follows:

$$CV_i = \sum_{a=1}^p \max(g_{i,a}(x), 0) + \sum_{b=1}^q \max(|h_{i,b}(x) - \delta|, 0)$$

$$CV_{mean} = 1/N_W \sum_{i=1}^{N_W} CV_i$$

$$FR = (\text{no. of feasible solutions in population of size } N_W) / N_W$$

$$\epsilon_{CV} = CV_{mean} \times FR \quad (2)$$

where $10^{-3} \leq \delta \leq 10^{-6}$ (we have used $\delta = 10^{-5}$ in this paper).

C. Construction/Update of the Surrogate Models

Upon evaluating the initial population, surrogate models are constructed for each objective function. In this paper, we have used Kriging, RSM with polynomial degree of 1 and 2 (RSM1 and RSM2, respectively) and RBF. However, other surrogate models can be considered too. To construct the surrogate models, 80% of the samples are used for training and the remaining 20% of the samples are used for validation, in-line with the previous studies using multiple surrogates [19], [71]. As we are dealing with a very small archive (maximum of 300 solutions), we are only performing training and testing here. We have arbitrarily chosen the percentage training and testing, as there is no rule of thumb. The surrogate model with the minimum root mean-squared error (RMSE) value is chosen as the representative surrogate model for a particular objective function. The RMSE is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} (y_n(x) - \hat{y}_n(x))^2} \quad (3)$$

where N_{test} is the number of test points for RMSE calculation, and $y_n(x)$ and $\hat{y}_n(x)$ are the actual and predicted values of the n th test point, respectively.

The surrogate models are retrained and all predictions are updated whenever the archive is updated with new truly evaluated solutions. If constraint functions are expensive to evaluate, surrogate models are needed for them as well. However, in the computational experiments in this paper, we consider constraints cheap to evaluate for a fair comparison with the other compared algorithm.²

D. Adaptation of Reference Vectors

RVs are adapted following the scheme suggested in [51]. However, unlike [51], they are adapted in every generation, thus eliminating the need for an user defined parameter, i.e., the update frequency. The update scheme for the i th RV is presented in the following equation:

$$W_i^{\min} = \frac{W_{0,i}^{\min} \odot (z_{\max} - z_{\min})}{\|W_{0,i}^{\min} \odot (z_{\max} - z_{\min})\|}; \quad i = 1, \dots, N_W \quad (4)$$

where W_i^{\min} and $W_{0,i}^{\min}$ are the i th adapted and initially generated (refer to Algorithm 1) RV of the RV set W^{\min} , respectively. Besides, $(z_{\max} - z_{\min})$ is the difference between the best and worst feasible objective values of the current population and \odot is the Hadamard product [72] for element-wise multiplication of two vectors of equal size. Furthermore, W^{\max} is updated similarly, for which, z_{\min} and z_{\max} are interchanged.

E. Offspring Generation

In each generation, N_W offspring solutions are generated using simulated binary crossover (SBX) [5] and differential evolution operator [73] with an equal probability. For the former, two random parents are chosen, while for the latter three

²Note that this is not the limitation of the presented approach itself. Provisions have been kept within the proposed framework for the consideration of expensive constraints in future studies.

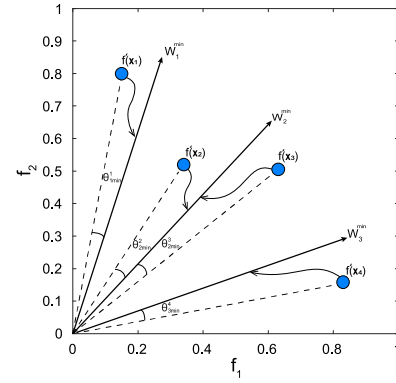


Fig. 2. Assignment of solutions to RVs (W^{\min}).

parents are randomly chosen. The resulting offspring solution³ undergoes polynomial mutation (PM) [5] and the process is repeated until N_W offsprings are generated.

F. Assignment

The assignment of solutions to RVs is done for feasible solutions only. For assigning the feasible solutions of the current population to W^{\min} , the objective values of the solutions of the current population are translated, i.e., $f'_j(x_i) = f_j(x_i) - z_{\min_j}$, where $f_j(x_i)$ is the value of the j th objective of the i th solution in the population and z_{\min_j} is the minimum value of the j th objective in the current population. Thereafter, the acute angle between a solution and all RVs is calculated [51]. A solution is assigned to the RV which has the smallest acute angle with that solution. This process divides the population into different subpopulations. The assignment process is illustrated in Fig. 2 for a two-objective case for three RVs originating from z_{\min} . The same principle is applied for assigning solutions to RVs originating from z_{\max} . Each solution is thus, assigned twice, i.e., to an RV in W^{\min} and W^{\max} .

G. Selection of Infill Solutions (InfillSelection)

For computationally expensive MOPs with a limited budget, it is imperative that solutions be carefully selected for evaluation. With surrogates in place, it is in principle possible to improve each offspring through a local search [60]. In this paper, however, instead of attempting to improve every offspring solution, the attempts are limited to improve at most N_{is} solutions corresponding to both RV sets (W^{\min} and W^{\max}). Such an approach is adopted to reduce the overhead of local searches. Moreover, to limit the number of actual function evaluations, the local search operation is done using the predicted values from the surrogate models. The entire process involves three key steps presented below and further elaborated subsequently.

- 1) For both RV sets, identify at most N_{is} solutions as candidates for further improvement via local search.
- 2) For each of the above selected solutions, use a local search to improve its performance, i.e., minimize or

³DE+PM generates a single offspring solution, but SBX+PM generates two offspring solutions and one of them is chosen with an equal probability.

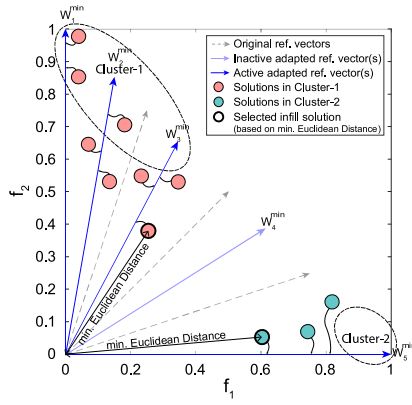


Fig. 3. Selecting infill solutions with the help of W^{\min} .

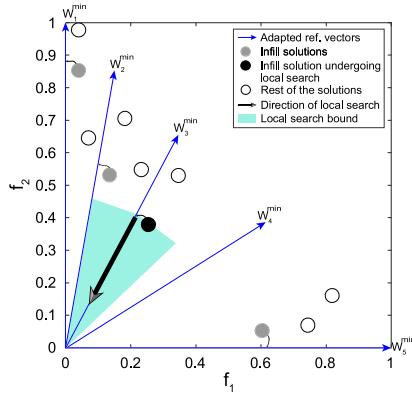


Fig. 4. Local improvement of an initial infill solution along W_3^{\min} . The shaded region is the feasible region of the angle constraint.

maximize the selection metric based on the RV set under consideration. In this paper, ED is used as the selection metric. For W^{\min} , the ED is minimized toward z_{\min} while for W^{\max} , the ED is maximized away from z_{\max} .

- 3) From the solutions identified in steps above for both set of RVs, select at most N_{is} solutions for actual evaluation with expensive functions.

In the first step, for any RV set, if all the members of the population (parents and offspring combined) are infeasible, N_{is} solutions are selected based on CV. If the number of feasible solutions (whose CV is not more than ϵ_{CV}) is less or equal to N_{is} , these solutions are selected as candidates for local search. If the above selection results in fewer than N_{is} solutions, the remaining solutions are selected based on CV. If the number of feasible solutions is greater than N_{is} , the feasible solutions among them are assigned to both RV sets. After the assignment, some RVs might end up with more than one solution while some RVs might remain empty with no solutions assigned to them. The nonempty (with at least one solution assigned to it) RVs are clustered to a maximum of N_{is} clusters and the best solution is selected from each cluster based on the ED. A solution from each cluster is selected which has the minimum ED from z_{\min} for W^{\min} and maximum ED from z_{\max} for W^{\max} . The widely used K -means method is employed for clustering [74]. Fig. 3 shows the RV clustering process for five RVs for the W^{\min} RV set. Here, RVs W_1^{\min} to W_3^{\min} form cluster-1 while W_5^{\min} belongs to cluster-2 with

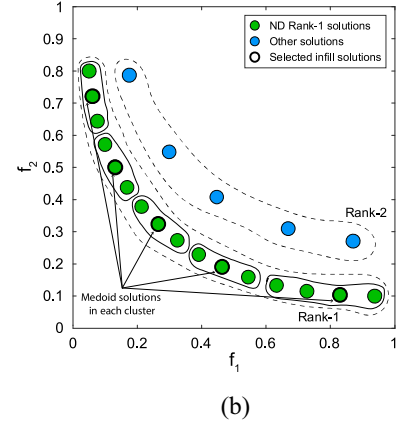
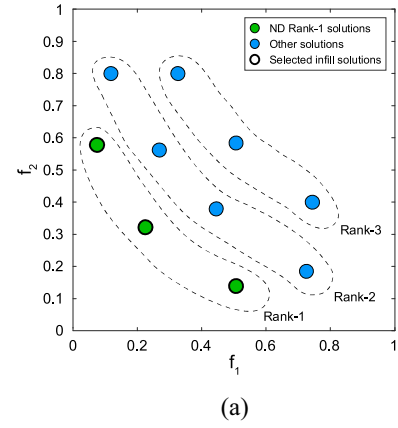


Fig. 5. Two different infill selection scenarios are presented here for $N_{is} = 5$. (a) Scenario-1: no of ND rank-1 solutions is 3, hence, all solutions are selected as infill solutions. (b) Scenario-2: no. of ND rank-1 solutions is 16 and the indicated cluster medoids are selected to achieve diversity.

a single member. The fourth RV, W_4^{\min} is empty and hence excluded from the clustering process.

The above step will result in the identification of a maximum of N_{is} solutions for each RV set. These solutions are the initial infill solutions. For all initial infill solutions that are feasible, a local search is conducted with ED optimization (we have used interior-point algorithm based on [75] and [76] in MATLAB built-in function `fmincon` in the numerical experiments, but other suitable optimizers could also be used). As mentioned above, every solution has an assigned RV based on the acute angle. The angle between the assigned RV and its closest RV is used as an angle constraint during local search to restrict the improved solution to be in the vicinity of the RV its initial solution was assigned to. If some of the selected solutions are infeasible, they are improved via a local search minimizing their CV. Fig. 4 illustrates an example where the direction of local search and the angle constraint is highlighted for W^{\min} . The similar process is applied for W^{\max} as well. The only difference is that for the RV set W^{\min} , ED is minimized toward z_{\min} while for the RV set W^{\max} , the ED is maximized away from z_{\max} as mentioned above.

The above-mentioned steps will lead to a maximum of N_{is} solutions for each RV set. Solutions identified for improvement and the improved solutions obtained from both the RV sets

TABLE II
PARAMETER SETTINGS FOR THE COMPARED ALGORITHMS

Global parameters	Parameters for HSMEA	Parameters for CSEA (as suggested in [9])	Parameters for K-RVEA/cK-RVEA1 (as suggested in [8] and [34])
a) No. of initial solns., N_{init} : $11D-1$ (taken from literature [8, 20, 23] and for a fair comparison with other algorithms) for unconstrained and 50 for constrained problems (for fair comparison with [34]). b) Maximum no. of function evaluations, FE_{max} : 300 for all problems. c) No. of RVs, N_W generated using a single layer NBI (for < 6 objectives) or double layered approach (≥ 6 objectives). d) For SBX: $p_{crossover} = 0.9$, $\eta_{crossover} = 30$; For DE: $CR = 1$, $F = 0.5$; For PM: $p_{mutation} = 1/D$, $\eta_{mutation} = 20$.	a) No. of infill solns. in each generation, $N_{is} = 5$ (at most, for a fair comparison with compared algorithms). b) Stopping criterion for local search using surrogate models is 1000 evaluations (in-line with MATLAB's default setting of 10D). As most of the problems in this study use 10 decision variables, we have chosen a fixed value here. However, during the experiments, we have observed that most of the local searches converged well within this limit).	a) No. of reference solns., $k = 2$. b) Maximum epochs for training the FNN, $T = 500$. c) No. of hidden neurons, $H = 2D$ (at most). d) No. of iterations for each learning is 800.	a) RVEA parameter, $\alpha = 2$. b) No. of infill solns. in each generation, $ u = 5$. c) Maximum no. of solutions for training Kriging models, $N_I = 11D-1$. d) Parameter for updating the Kriging models, $\delta = 0.05N_W$. e) No. of gens before updating the Kriging models, $w_{max} = 20$.
	Parameters for MOEA/D-EGO (as suggested in [23])	Parameters for ParEGO (as suggested in [20])	Parameters for CPS-MOEA (as suggested in [26])
	a) MOEA/D parameters: probability of choosing parents locally, $\delta = 0.9$ and maximum no. of solns. replaced by each of the offsprings, $n_r = 2$. b) The no. of function evaluations at each generation, $K_E = 5$. c) Maximum no. of surrogate-assisted fitness approximations before the surrogate update is $20 \times (11D-1)$. d) The maximum no. of points used for building a local model, $L_1 = 80$. e) The minimum no. of points used for building a local model, $L_2 = 20$.	a) No. of infill solns. in each generation is 1. b) Augmentation coefficient in Chebyshev function, $\rho = 0.05$. c) Maximum no. of solns. for updating Kriging models is $11D-1+25$. d) Maximum no. of surrogate-assisted fitness approximations before the surrogate update is 10^4 .	a) Population size is 50. b) Number of nearest neighbors, $k = 5$.

are combined resulting in a set of a maximum size of $4N_{is}$ solutions. If the number of feasible solutions is no more than N_{is} , they are all selected as infill solutions for evaluating with the expensive functions. If there are more than N_{is} feasible solutions, ND sorting is applied on them. The solutions in the first front (ND rank-1) are first selected. If there are more than N_{is} such solutions, they are partitioned into N_{is} clusters and the ones closest to the cluster medoid are selected as infill solutions for evaluation with the expensive functions. If there are fewer than N_{is} solutions in the first front, they are directly selected (no clustering required) for actual evaluation.

To visually illustrate this, in Fig. 5, two different scenarios are presented for different numbers of ND rank-1 solutions for $N_{is} = 5$. In scenario-1, we have ten solutions divided into three ND fronts. There are three rank-1 solutions which is less than N_{is} . Hence, these solutions are selected as infill solutions.

On the other hand, in scenario-2, there are 20 solutions among which 16 are rank-1 solutions and four solutions are rank-2. The 16 ND rank-1 solutions are divided into five clusters by the K-means algorithm and later, the five medoids are selected as infill solutions.

H. Selecting Parent Solutions for the Next Generation (Environmental Selection)

Environmental selection identifies solutions to be carried to the next generation as parents. Besides considering the parent and offspring solutions, the solutions in the archive \mathcal{A} are considered in environmental selection in our algorithm. If all solutions in the combined set (i.e., the parents, offspring and archive \mathcal{A}) of solutions are infeasible, they are sorted based on their CV. If the number of feasible solutions is no greater than

N_W , they are chosen as parents and the rest of the solutions are chosen based on their CV to make N_W parents (as the population size is fixed in our algorithm equal to the number of RVs). On the other hand, if the number of feasible solutions is more than N_W , we need to limit the number of parents to N_W . For this, we need to select one solution for each RV. As we have two RV sets W^{\min} and W^{\max} , first, the combined set of solutions is assigned to one of the updated RV sets (say, W^{\min}) as mentioned in Section III-F. The assignment process may lead to some of the RVs being empty (as discussed before). For the nonempty RVs, one solution is selected from each of them based on the selection metric (ED). Next, the nonempty RVs are removed and the combined set of solutions are assigned to the previously empty RVs and the same process of selection is followed to select a total of N_W solutions as parents with the help of N_W RVs. A similar process can be followed in case of W^{\max} to obtain another set of N_W solutions.

Now, there are two sets of potential parent populations (obtained from W^{\min} and W^{\max} , respectively), one of which must be selected as the parent population for the next generation. The choice between these two sets is based on the s -energy [77] metric defined as follows:

$$E(F, s) = \sum_{1 \leq i \leq k \leq N_W} \|f(x_i) - f(x_k)\|^{-s}, \quad s > 0$$

$$f \in \mathbb{R}^M, \quad F = \{f(x_l), l = 1, 2, \dots, N_W\} \quad (5)$$

where F is the set of objective values of the solutions in the population, $\|\cdot\|$ is the l^2 -norm of the difference between the objective values of i th and k th solutions, and the parameter s is set to be $M - 1$, where M is the number of objectives.

Here, s -energy is chosen for its simplicity of calculation and most importantly, it is not dependent on a reference set (preferably, uniformly distributed points on the true or best known PF for mathematical and real-world problems, respectively) unlike some other metrics such as IGD and R2-indicator. A lower value of s -energy indicates a better diversity. Therefore, the set of solutions (out of the two discussed above) that results with the lowest s -energy is selected as the parent population of the next generation.

IV. NUMERICAL EXPERIMENTS

The performance of HSMEA is quantitatively assessed using the well studied unconstrained DTLZ [78] and WFG [79] problems and their inverted (“minus”) versions proposed in [12], as well as constrained test problems (C1_DTLZ1, C2_DTLZ2, and C3_DTLZ4 [2]) with 3, 4, 6, 8, and 10 objectives. Furthermore, three real-world engineering design optimization problems, i.e., a three-objective car side impact problem [2], a five-objective water resource management problem [80], and a ten-objective general aviation aircraft [3] design optimization problems are solved. The number of variables for all unconstrained and constrained DTLZ problems is set to 10. For WFG problems, the numbers of variables are set as 10, 10, 9, 9, and 11 for 3, 4, 6, 8, and 10-objective problems, respectively. The performance of the proposed algorithm is compared with the contemporary state-of-the-art SAEAs for MaOPs

TABLE III
 H_1 AND H_2 VALUES FOR THE NUMBER OF RVs AND N_W FOR DIFFERENT NUMBERS OF OBJECTIVES M

M	(H_1, H_2)	N_W
3	(13,0)	105
4	(7,0)	120
6	(4,1)	132
8	(3,2)	156
10	(3,2)	275

CSEA [9], K-RVEA [8], MOEA/D-EGO [23], ParEGO [20], and CPS-MOEA [26] for all unconstrained problems and with cK-RVEA1 (the best performing approach reported in [34]) for all constrained/engineering design optimization problems. The parameter settings for numerical experiments are listed in Table II.

A. Performance Metrics

Inverted generational distance (IGD) [81] and HV [82] are used as the metrics for benchmarking. The reference sets for IGD calculation are obtained from the PlatEMO framework [83] for the DTLZ and WFG problems as well as constrained C1_DTLZ1, C2_DTLZ2, and C3_DTLZ4 problems while for the minus DTLZ and WFG problems they were derived by inverting these sets as suggested in [12]. For the engineering design problems, the reference sets have been taken from [84]. The spacings in different layers of the RVs, i.e., H_1 and H_2 values are presented in Table III for different objectives while the numbers of points in the sets are listed in Table IV. To statistically assess the performance of the algorithms, a Wilcoxon rank-sum (WRS) test [85] is performed on the results obtained from the median runs (out of 25 independent runs) of all problems across all objectives for all algorithms with a 5% confidence level.

The overall performance is also visually presented using performance profile [86] plots. A performance profile is generally used as a statistical tool for observing the performance of different algorithms, i.e., to assess how well/how fast and what percentage of problems were solved by the given algorithms relative to each other. In this paper, the performance profiles are plotted on the *median* of the IGD values obtained for the set of all problems for all objectives studied. The x -axis of a performance profile plot represents the goal value, τ (which in this case represents the ratio of the median best IGD of a particular algorithm compared to the best performing algorithm for a specific problem), while the y -axis ($\rho_s(\tau)$) denotes the cumulative distribution of the median best IGD (i.e., the percentage of problems an algorithm is able to solve within a factor τ with respect to the best algorithm). Hence, different algorithms can be compared on a given level of goal value τ and the winner is the one which reaches a value of $\tau = 1$ first.

B. Influence of Multiple Surrogate Models, Dual Set of Reference Vectors, and Local Search

We first investigate the effect of using multiple types of surrogates over a single surrogate. Kriging is chosen as a representative single surrogate model given its popularity in the

TABLE IV

NUMBER OF POINTS IN REFERENCE SETS FOR IGD CALCULATION OVER DIFFERENT VALUES OF M FOR DIFFERENT PROBLEMS UNDER STUDY

M	Number of points in reference sets		
	DTLZ7/ DTLZ7 ⁻¹	WFG2/ WFG2 ⁻¹	C2_DTLZ2
3	6084	4101	2932
4	10648	10708	6466
6	59049	32191	10623
8	78125	66342	8934
10	19683	115610	13451
Other problems			
3	5050		
4	10660		
6	33649		
8	50388		
10	92378		

literature. For this analysis, we use the proposed HSMEA algorithm with only a single set of RVs (from z_{\min}) for simplicity; and compare the two versions, one with multiple surrogates and another with just Kriging. A summary of the results of WRS tests for IGD metric based on 25 independent runs across all unconstrained (DTLZ and WFG standard and minus problems) and constrained problems with different objectives is presented in Table V. Detailed statistical results are available in Tables VI–VIII of the supplementary file for unconstrained standard and minus DTLZ and WFG problems as well as constrained DTLZ problems. In Table V, we list the total numbers of problem instances (n), wins (w), losses (l), and ties (t) of HSMEA with multiple surrogate models against HSMEA with Kriging.

Looking at the values of wins and losses from Table V, one can conclude that the performance of the multiple surrogate approach is similar for standard (19 wins and 18 losses) and constrained problems (three wins and one loss) but significantly better for minus problems (24 wins and 8 losses). Thus in light of offering greater flexibility of representation, the choice of multiple surrogates over a single surrogate (Kriging in this case) is beneficial. Although training multiple surrogates incurs additional computational cost, as discussed in Section III, it is considered negligible in relation to the computational cost of the actual function evaluations.

Having established the benefits of using multiple surrogates, we investigate whether DR (dual set of RVs) offers any benefit over SR (single set of RVs). A summary of results of WRS tests for IGD metric based on 25 independent runs across all unconstrained (DTLZ and WFG standard and minus problems) and constrained problems with different objectives is presented in Table VI comparing baseline HSMEA with DR against baseline HSMEA with SR (detailed results are available in Tables IX–XI of the supplementary file). The symbols n , w , l , and t are the same as above indicating number of instances, wins, losses, and ties for HSMEA with DR.

Looking at the values of wins and losses overall, one can conclude that the performance of DR on standard problems is worse (15 wins and 39 losses), but once again for minus problems, the performance is significantly better (40 wins and

TABLE V

WRS TEST RESULTS FOR IGD METRIC FOR BASELINE HSMEA WITH MULTIPLE AND SINGLE SURROGATE FOR DIFFERENT NUMBERS OF OBJECTIVES

M	Standard Problems (n/w/l/t)	Minus Problems (n/w/l/t)	Constrained Problems (n/w/l/t)
3	16/4/3/9	13/4/1/8	3/0/0/3
4	16/7/2/7	13/6/7/0	3/0/0/3
6	16/2/2/12	13/4/0/9	3/1/0/2
8	16/2/4/10	13/5/0/8	3/1/0/2
10	16/4/7/5	13/5/0/8	3/1/1/1
Total=	80/19/18/43	65/24/8/33	15/3/1/11

10 losses). For the constrained problems, DR offers a similar performance to SR. Hence, for black-box problems with unknown PF shapes, using DR offers marginally improved performance (a total of 55 wins and 49 losses). The effectiveness of the use of a dual set of adaptive RVs is further substantiated by implementing two existing popular RV adaptation strategies RVEA* [51] and A-NSGA-III [2] within the HSMEA framework and comparing their performances with the proposed HSMEA. The results comprehensively demonstrate the usefulness of the proposed approach. The study is included in Section VI of the supplementary file due to space limitations.

Finally, we investigate the effectiveness of local search. For this, we have compared the performance of baseline HSMEA having multiple surrogate models, DR and local search with the variant of baseline HSMEA with multiple surrogate models and DR only. The summary of WRS test results on IGD is presented in Table VII (detailed IGD statistics are presented in Tables XII–XIV of the supplementary file for standard, minus, and constrained problems, respectively).

From Table VII, it can be observed that the local search scheme is beneficial for all problem types under study. The usefulness of local search is reflected in the WRS test results for the standard unconstrained problems as the HSMEA variant with local search scores 56 wins, 13 losses, and 11 ties compared to the HSMEA variant without local search among 80 problem instances. For minus problems it yields 31 wins, 19 losses, and 15 ties among 65 problem instances, while for constrained problems, it scores 12 wins, 3 losses, and 1 tie compared to its counterpart without local search.

C. Results and Benchmarking

The results reported so far establish the potential benefits of using multiple surrogates, dual set of RVs, and local search. Subsequently, we compare the performance of HSMEA with the state-of-the-art approaches, i.e., CSEA, K-RVEA, MOEA/D-EGO, ParEGO, and CPS-MOEA. The summary of WRS tests is presented in Table VIII for unconstrained standard DTLZ and WFG test problems and corresponding minus problems. More detailed results are available in Tables XII and XIII of the supplementary file. In the tables, the symbols n , w , l , and t are as before indicating now HSMEA performed compared to each of the other algorithms.

The results clearly support the view that HSMEA (with a dual set of RVs, multiple surrogates, and local improvement

TABLE VI
WRS TEST RESULTS FOR IGD METRIC FOR BASELINE HSMEA WITH
DUAL AND SINGLE SET OF RVs FOR DIFFERENT
NUMBERS OF OBJECTIVES

M	Standard Problems (n/w/l/t)	Minus Problems (n/w/l/t)	Constrained Problems (n/w/l/t)
3	16/4/6/6	13/5/3/5	3/0/2/1
4	16/4/7/5	13/8/2/3	3/0/2/1
6	16/2/12/2	13/9/1/3	3/1/1/1
8	16/4/6/6	13/9/2/2	3/2/0/1
10	16/1/8/7	13/9/2/2	3/2/0/1
Total=	80/15/39/26	65/40/10/15	15/5/5/5

TABLE VII
WRS TEST RESULTS FOR IGD METRIC FOR BASELINE HSMEA (DR)
WITH LOCAL SEARCH AND WITHOUT LOCAL SEARCH FOR
DIFFERENT NUMBERS OF OBJECTIVES

M	Standard Problems (n/w/l/t)	Minus Problems (n/w/l/t)	Constrained Problems (n/w/l/t)
3	16/9/3/4	13/4/7/2	3/2/1/0
4	16/10/3/3	13/2/5/6	3/2/1/0
6	16/12/2/2	13/7/2/4	3/2/1/0
8	16/12/2/2	13/9/2/2	3/3/0/0
10	16/13/3/0	13/9/3/1	3/3/0/1
Total=	80/56/13/11	65/31/19/15	15/12/3/1

of infill solutions) shows significantly better performance for standard and at-par performance on minus problems while being compared with the recently proposed algorithm CSEA. The WRS test results for IGD metric show 50 wins, 18 losses, and 12 ties (on a total of 80 problem instances) for HSMEA in standard problems while in minus problems it scores 26 wins and losses with 13 ties (on a total of 65 problem instances). When HSMEA and K-RVEA are compared, the observations are quite opposite. While HSMEA shows significantly better performance for minus problems, it shows similar performance for standard problems. More specifically, HSMEA achieves 35 wins and losses and 20 ties in standard problems while for minus problems it scored 43 wins and 22 losses with no ties. These results indicate that none of the studied algorithms is capable of fully handling all problem types. To further clarify, the HSMEA loss percentages from Table VIII (i.e., the win percentages of the compared algorithms) on the total number of problem instances (for the standard and minus problems, respectively) is presented in Table IX. From the table, it is apparent that none of the peer algorithms is able to successfully handle all types of PFs. The best performing peer algorithms in each type of problems is not able to outperform HSMEA and the best performing peer algorithm in one type of problem is not the best performing peer algorithm in another problem type. For example, K-RVEA is the best performing peer algorithm for solving standard problems which shares 43.75% wins with HSMEA while solving minus problems, HSMEA is no worse than K-RVEA in 66.16% of the problem instances. On the other hand, CSEA is the best performing algorithm for handling minus problems which shares 40% wins with HSMEA, however, if we observe its performance in standard problems, it wins in only 22.50% of the problem instances. Hence, based on the problems studied, the proposed algorithm HSMEA is preferable for its robustness of handling

TABLE VIII
TEST RESULTS WITH THE IGD METRIC BASED ON 25 RUNS ACROSS
STANDARD DTLZ AND WFG PROBLEMS OVER DIFFERENT
NUMBERS OF OBJECTIVES

Problem Types	M	CSEA (n/w/l/t)	K-RVEA (n/w/l/t)	MOEA/D-EGO (n/w/l/t)	ParEGO (n/w/l/t)	CPS-MOEA (n/w/l/t)
Standard Problems	3	16/12/3/1	16/11/4/1	16/16/0/0	16/16/0/0	16/16/0/0
	4	16/8/5/3	16/8/3/5	16/14/0/2	16/16/0/0	16/15/0/1
	6	16/10/4/2	16/5/9/2	16/13/0/3	16/16/0/0	16/13/0/3
	8	16/10/3/3	16/6/10/0	16/14/1/1	16/16/0/0	16/13/0/3
	10	16/10/3/2	16/5/9/2	16/13/1/2	16/16/0/0	16/12/2/2
Total =		80/50/18/12	80/35/35/10	80/70/2/8	80/80/0/0	80/69/2/9
Minus Problems	3	13/5/5/3	13/8/5/0	13/4/5/4	13/10/1/2	13/5/5/3
	4	13/4/8/1	13/8/5/0	13/7/6/0	13/13/0/0	13/5/7/1
	6	13/2/7/4	13/9/4/0	13/10/1/2	13/13/0/0	13/6/4/3
	8	13/7/3/3	13/9/4/0	13/10/0/3	13/13/0/0	13/11/2/0
	10	13/8/3/2	13/9/4/0	13/12/0/1	13/13/0/0	13/11/2/0
Total =		65/26/26/13	65/43/22/0	65/43/12/10	65/62/1/3	65/38/20/7

TABLE IX
SUMMARY OF THE COMBINED TEST RESULTS FOR ALL OBJECTIVES
WITH THE IGD METRIC BASED ON 25 RUNS ACROSS STANDARD
AND MINUS DTLZ AND WFG PROBLEMS

Problem Type		CSEA	K-RVEA	MOEA/D-EGO	ParEGO	CPS-MOEA
Standard Problems	n/w/l/t	80/50/18/12	80/35/35/20	80/70/2/8	80/80/0/0	80/69/2/9
	HSMEA Loss	22.50%	43.75%	2.50%	0.00%	2.50%
Minus Problems	n/w/l/t	65/26/26/13	65/43/22/0	65/43/12/10	65/62/1/3	65/38/20/7
	HSMEA Loss	40.00%	33.84%	18.46%	1.50%	30.77%

problems with standard or inverted PFs. Another observation from Table VIII is that in comparison with K-RVEA, the performance of HSMEA is inferior for the standard 6, 8, and 10 objective problems and needs further algorithmic improvements to address this gap.

Next, the performance of HSMEA is compared with cK-RVEA1 on constrained DTLZ and constrained engineering design problems. The statistics of the IGD metric across 25 independent runs are presented in Table X. The table also includes WRS test results for the algorithms presented. Here, \uparrow , \downarrow , and \approx indicates whether HSMEA is statistically significantly better, worse, or equivalent to cK-RVEA1. As before, n , w , l , and t are indicating the number of instances, wins, losses, and ties of HSMEA according to the WRS test compared to cK-RVEA1.

From Table X, it can be observed that HSMEA comprehensively outperforms cK-RVEA1 in most of the problem instances. For constrained DTLZ problems C1_DTLZ1, C2_DTLZ2, and C3_DTLZ4, HSMEA scores 10 wins, 2 losses, and 3 ties among 15 problems instances while, it was the clear winner (three wins and no losses or ties in three problem instances) in all engineering design problems.

For completeness, the performance profile plots on median IGD values (of all problems with different numbers of objectives) are presented in Fig. 6. Here, it can be observed that in the standard unconstrained problems, HSMEA solves all problems before any other compared approach [reaching to $\rho_s(\tau)$ first] except K-RVEA. HSMEA dominates K-RVEA for the first $\approx 50\%$ of the problems after which, K-RVEA shows a slightly improved performance for the next $\approx 30\%$ of the problems and thereafter, HSMEA again dominates for the next $\approx 20\%$ of the problems reaching $\rho_s(\tau) = 1$ first. CSEA, MOEA/D-EGO, CPS-MOEA, and ParEGO follow next. On the other hand, although according to the WRS test, HSMEA

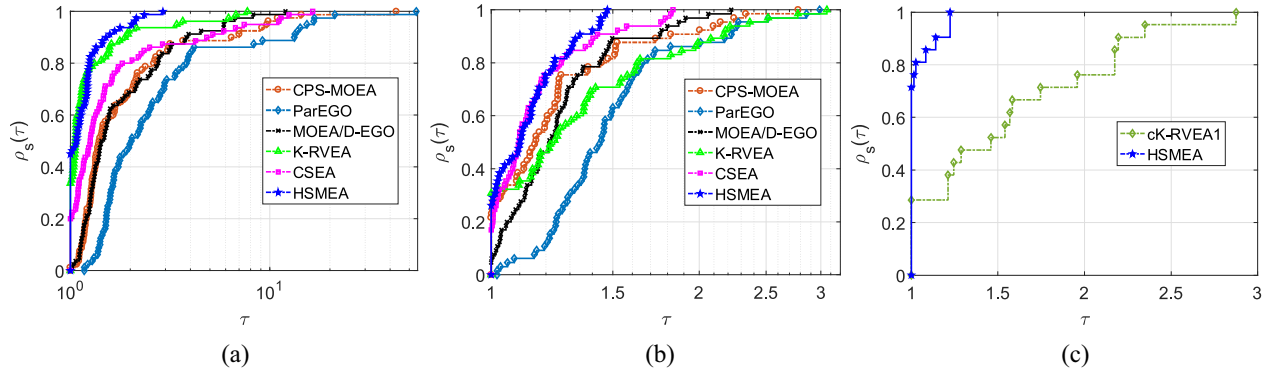


Fig. 6. Performance profile plots considering the median IGD values of 25 independent runs. (a) Standard unconstrained problems. (b) Minus unconstrained problems. (c) Constrained and engineering design optimization problems.

TABLE X
TEST RESULTS FOR IGD METRIC OBTAINED BY HSMEA AND
CK-RVEA1 FOR CONSTRAINED PROBLEMS. THE BEST
MEAN RESULTS ARE HIGHLIGHTED IN BOLD

Prob.	M	HSMEA			cK-RVEA1		
		Min	Mean	Max	Min	Mean	Max
C1_DTLZ1	3	0.0378	0.0430	0.0504	↑	0.0431	0.1094
	4	0.0602	0.0739	0.0938	↑	0.0911	0.1844
	6	0.1131	0.1362	0.1657	↑	0.1314	0.2173
	8	0.1647	0.1878	0.2142	≈	0.1488	0.1851
	10	0.1857	0.2214	0.2474	↓	0.1629	0.1855
C2_DTLZ2	3	0.0725	0.0853	0.1098	↑	0.1481	0.2161
	4	0.1729	0.2040	0.2560	↑	0.2622	0.4104
	6	0.2640	0.3025	0.4003	↑	0.3596	0.5999
	8	0.2885	0.3319	0.4538	↑	0.4422	0.7897
	10	0.3125	0.3466	0.5463	↑	0.6117	0.8962
C3_DTLZ4	3	0.2361	0.4410	0.8810	≈	0.2971	0.4018
	4	0.4873	0.6257	0.9289	↓	0.4492	0.8156
	6	0.5991	0.7302	0.8615	≈	0.6273	0.7249
	8	0.6289	0.6934	0.7780	↑	0.7827	0.9498
	10	0.6903	0.7261	0.7698	↑	0.8292	0.9383
CSI	3	0.2744	0.2932	0.3232	↑	0.3337	0.4725
WRM	5	24973.8499	28969.2517	39621.3171	↑	47806.5140	83425.4210
GAA	10	25.4959	49.9513	121.5041	↑	53.9196	113.8283
n/w/l/t		--/--				18/13/2/3	

and CSEA are comparable in minus problems, the performance profile plot shows that HSMEA is better than CSEA for first $\approx 45\%$ of the problems. After this both the algorithms show competitive performance and for the last $\approx 15\%$ of problems, HSMEA dominates CSEA reaching $\rho_s(\tau) = 1$ first, followed by MOEA/D-EGO, CPS-MOEA, K-RVEA, and ParEGO.

For completeness, all algorithms have also been compared using HV. Due to space limitations, the results are included in Section I of the supplementary file, where the summary of the WRS test results are presented in Tables I and II for standard and minus unconstrained problems, respectively. The best, mean and worst values and WRS test results are presented in Table III for constrained and engineering design problems. Subsequently, the statistical results for both types of unconstrained problems are shown in Tables IV and V of the supplementary file. The relative performances are largely consistent with the observations based on IGD above. Moreover, analysis of the performance of the algorithms under study with fewer numbers function evaluations have also been included in Section III of the supplementary file. An investigation is done to observe if there are considerable differences in

the performance of HSMEA across different types of problems when compared with the five peer algorithms mentioned above. Once again, the observations are in-line with the results presented here with a maximum of 300 function evaluations.

V. CONCLUSION

In this paper, an HSMEA is proposed to solve computationally expensive MaOPs with a very limited function evaluation budget. In this algorithm, objective functions are approximated using a number of surrogate models (Kriging, RSM1, RSM2, and RBF) and the best one based on a minimum RMSE is chosen in order to approximate different types of functions closely. Furthermore, the algorithm employs RVs for decomposition. Two sets of RVs are used in order to provide flexibility of dealing with different PF shapes. A local improvement mechanism is incorporated to identify better infill solutions for faster convergence. Furthermore, the information of the complete archive of evaluated solutions is utilized to generate better offspring.

The performance of HSMEA is tested on a wide range of unconstrained test problems including the standard DTLZ and WFG test suites as well as their minus variants, in addition to constrained problems and engineering design problems. From the results, it is observed that the proposed algorithm is able to perform significantly better than two recently proposed approaches CSEA and K-RVEA in either of the problem types (i.e., standard or minus) and shows similar performance in the other type of problems. This indicates its reliability in dealing with a wide range of problems. HSMEA outperforms MOEA/D-EGO, ParEGO, and CPS-MOEA for both types of problems. Thus, overall, HSMEA shows significant promise in solving computationally expensive MaOPs of different types which indicates the potential suitability of this algorithm for real-world problems for which the shape of the PFs are unknown.

Some future research directions include improving the performance for problems with higher number of variables/objectives, extending the study to consider the expensive constraints, adaptation of RVs/introduction of RV assisted objective reduction or estimation of distribution methods to deal with generic irregular PFs (such as degenerate, disconnected, highly convex/concave, etc.) and developing methods

to incorporate preferences of an expert or a decision maker in the solution process.

ACKNOWLEDGMENT

A. Habib would like to thank Dr. K. S. Bhattacharjee and Dr. R. Cheng for discussions and sharing insights regarding this paper and Dr. C. He for sharing CSEA and MATLAB compatible MOEA/D-EGO codes.

REFERENCES

- [1] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *Proc. Evol. Multi Criterion Optim.*, 2005, pp. 14–32.
- [2] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
- [3] T. Ray, K. Tai, and K. C. Seow, "An evolutionary algorithm for multi-objective optimization," *Eng. Optim.*, vol. 33, no. 3, pp. 399–424, 2001.
- [4] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 2419–2426.
- [5] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
- [6] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61–70, 2011.
- [7] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Comput.*, to be published. doi: 10.1007/s00500-017-2965-0.
- [8] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, Feb. 2018.
- [9] L. Pan *et al.*, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 74–88, Feb. 2019.
- [10] K. S. Bhattacharjee, H. K. Singh, and T. Ray, "Multiple surrogate-assisted many-objective optimization for computationally expensive engineering design," *J. Mech. Design*, vol. 140, no. 5, pp. 1–10, 2018.
- [11] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating Pareto optimal points in multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
- [12] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, Apr. 2017.
- [13] L. V. Santana-Quintero, A. A. Montaña, and C. A. Coello 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.
- [14] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005.
- [15] G. Montemayor-García and G. Toscano-Pulido, "A study of surrogate models for their use in multiobjective evolutionary algorithms," in *Proc. Int. Conf. Elect. Eng. Comput. Sci. Autom. Control (CCE)*, 2011, pp. 1–6.
- [16] I. Loshchilov, M. Schoenauer, and M. Sebag, "Dominance-based Pareto-surrogate for multi-objective optimization," in *Proc. Asia-Pac. Conf. Simulat. Evol. Learn.*, 2010, pp. 230–239.
- [17] C.-W. Seah, Y.-S. Ong, I. W. Tsang, and S. Jiang, "Pareto rank learning in multi-objective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 1–8.
- [18] S. Bandaru, A. H. C. Ng, and K. Deb, "On the performance of classification algorithms for learning Pareto-dominance relations," in *Proc. IEEE Congr. Evol. Comput.*, 2014, pp. 1139–1146.
- [19] K. S. Bhattacharjee, H. K. Singh, and T. Ray, "Multi-objective optimization with multiple spatially distributed surrogates," *J. Mech. Design*, vol. 138, no. 9, pp. 1–10, 2016.
- [20] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multi-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2006.
- [21] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Stat. Sci.*, vol. 4, no. 4, pp. 409–423, 1989.
- [22] A. I. Forrester, A. Söbester, and A. J. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*. Chichester, U.K.: Wiley, 2008.
- [23] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multi-objective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.
- [24] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer, 1999.
- [25] G. Kourakos and A. Mantoglou, "Development of a multi-objective optimization algorithm using surrogate models for coastal aquifer management," *J. Hydrol.*, vol. 479, pp. 13–23, Feb. 2013.
- [26] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput.*, 2015, pp. 2883–2890.
- [27] M. Herrera, A. Guglielmetti, M. Xiao, and R. F. Coelho, "Metamodel-assisted optimization based on multiple kernel regression for mixed variables," *Struct. Multidiscipl. Optim.*, vol. 49, no. 6, pp. 979–991, 2014.
- [28] G. P. Liu, X. Han, and C. Jiang, "A novel multi-objective optimization method based on an approximation model management technique," *Comput. Methods Appl. Mech. Eng.*, vol. 197, nos. 33–40, pp. 2719–2731, 2008.
- [29] T. Akhtar and C. A. Shoemaker, "Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection," *J. Glob. Optim.*, vol. 64, no. 1, pp. 17–32, 2016.
- [30] A. Husain and K.-Y. Kim, "Enhanced multi-objective optimization of a microchannel heat sink through evolutionary algorithm coupled with multiple surrogate models," *Appl. Therm. Eng.*, vol. 30, no. 13, pp. 1683–1691, 2010.
- [31] A. A. Montano, C. A. Coello Coello, and E. Mezura-Montes, "MODE-LD+SS: A novel differential evolution algorithm incorporating local dominance and scalar selection mechanisms for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Barcelona, Spain, 2010, pp. 1–8.
- [32] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.
- [33] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted *S* selection," in *Proc. Parallel Problem Solving Nat.*, 2008, pp. 784–794.
- [34] T. Chugh, K. Sindhya, K. Miettinen, J. Hakanen, and Y. Jin, "On constraint handling in surrogate-assisted evolutionary many-objective optimization," in *Proc. Parallel Problem Solving Nat.*, 2016, pp. 214–224.
- [35] Z. Wang, Q. Zhang, H. Li, H. Ishibuchi, and L. Jiao, "On the use of two reference points in decomposition based multiobjective evolutionary algorithms," *Swarm Evol. Comput.*, vol. 34, pp. 89–102, Jun. 2017.
- [36] H. Li and Q. Zhang, "Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.
- [37] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.
- [38] A. Mogilicharla, T. Chugh, S. Majumdar, and K. Mitra, "Multi-objective optimization of bulk vinyl acetate polymerization with branching," *Mater. Manuf. Processes*, vol. 29, no. 2, pp. 210–217, 2014.
- [39] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [40] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. Annu. Conf. Genet. Evol. Comput.*, 2010, pp. 519–526.
- [41] Y. Qi *et al.*, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.
- [42] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.
- [43] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2321–2334, Aug. 2018.
- [44] R. Wang, J. Xiong, H. Ishibuchi, G. Wu, and T. Zhang, "On the effect of reference point in MOEA/D for multi-objective optimization," *Appl. Soft Comput.*, vol. 58, pp. 25–34, Sep. 2017.

- [45] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, Feb. 2016.
- [46] R. Saborido, A. B. Ruiz, and M. Luque, "Global WASF-GA: An evolutionary algorithm in multiobjective optimization to approximate the whole Pareto optimal front," *Evol. Comput.*, vol. 25, no. 2, pp. 309–349, 2017.
- [47] K. S. Bhattacharjee, H. K. Singh, and T. Ray, "Decomposition based evolutionary algorithm with a dual set of reference vectors," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 105–112.
- [48] M. Wu, K. Li, S. Kwong, and Q. Zhang, "Evolutionary many-objective optimization based on adversarial decomposition," *IEEE Trans. Cybern.*, to be published. doi: [10.1109/TCYB.2018.2872803](https://doi.org/10.1109/TCYB.2018.2872803).
- [49] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [50] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, Feb. 2016.
- [51] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
- [52] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, Feb. 2013.
- [53] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, Aug. 2011.
- [54] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts," *IEEE Trans. Cybern.*, to be published. doi: [10.1109/TCYB.2018.2834466](https://doi.org/10.1109/TCYB.2018.2834466).
- [55] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 1167–1189, Oct. 2009.
- [56] Y. Sun, G. G. Yen, and Z. Yi, "Improved regularity model-based EDA for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 662–678, Oct. 2018.
- [57] D. K. Saxena and K. Deb, "Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2007, pp. 772–787.
- [58] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, Jan. 2014.
- [59] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Use of heuristic local search for single-objective optimization in multiobjective memetic algorithms," in *Proc. Conf. Parallel Problem Solving Nat.*, 2008, pp. 743–752.
- [60] Y.-Y. Tan, Y.-C. Jiao, H. Li, and X.-K. Wang, "MOEA/D-SQA: A multi-objective memetic algorithm based on decomposition," *Eng. Optim.*, vol. 44, no. 9, pp. 1095–1115, 2012.
- [61] H. Zhang, A. Zhou, G. Zhang, and H. K. Singh, "Accelerating MOEA/D by Nelder–Mead method," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 976–983.
- [62] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, Aug. 2013.
- [63] J. Shi, Q. Zhang, and J. Sun, "PPLS/D: Parallel Pareto local search based on decomposition," *arXiv preprint arXiv:1709.09785*, 2017.
- [64] L. Prestes, C. Almeida, and R. Gonçalves, "Implementation of quasi-simplex local search on MOEA/D," in *Proc. Lat. America Congr. Comput. Intell.*, 2015, pp. 1–6.
- [65] S. Zapotecas-Martínez and C. A. Coello Coello, "MONSS: A multi-objective nonlinear simplex search approach," *Eng. Optim.*, vol. 48, no. 1, pp. 16–38, 2016.
- [66] P. Koch, O. Kramer, G. Rudolph, and N. Beume, "On the hybridization of SMS-EMOA and local search for continuous multiobjective optimization," in *Proc. Conf. Genet. Evol. Comput.*, 2009, pp. 603–610.
- [67] A. Turco, "MetaHybrid: Combining metamodelling and gradient-based techniques in a hybrid multi-objective genetic algorithm," in *Learning and Intelligent Optimization*. Heidelberg, Germany: Springer, 2011, pp. 293–307.
- [68] M. D. McKay, R. J. Beckman, and W. J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.
- [69] M. Asafuddoula, T. Ray, R. Sarker, and K. Alam, "An adaptive constraint handling approach embedded MOEA/D," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 1–8.
- [70] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, 2000.
- [71] A. Isaacs, T. Ray, and W. Smith, "Multi-objective design optimisation using multiple adaptive spatially distributed surrogates," *Int. J. Prod. Develop.*, vol. 9, nos. 1–3, pp. 188–217, 2009.
- [72] G. P. H. Styan, "Hadamard products and multivariate statistical analysis," *Linear Algebra Appl.*, vol. 6, pp. 217–240, 1973.
- [73] R. Storn, "On the usage of differential evolution for function optimization," in *Proc. Biennial Conf. North Amer. Fuzzy Inf. Process. Soc.*, 1996, pp. 519–523.
- [74] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. Berkeley Symp. Math. Stat. Probab.*, Oakland, CA, USA, 1967, pp. 281–297.
- [75] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *J. Optim.*, vol. 9, no. 4, pp. 877–900, 1999.
- [76] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Math. Program.*, vol. 107, no. 3, pp. 391–408, 2006.
- [77] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Inf. Sci.*, vol. 282, pp. 363–387, Oct. 2014.
- [78] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, USA, 2002, pp. 825–830.
- [79] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2005, pp. 280–295.
- [80] T. W. Simpson, W. Chen, J. K. Allen, and F. Mistree, "Conceptual design of a family of products through the use of the robust concept exploration method," in *Proc. AIAA/USAF/NASA/ISSMO Symp. Multidiscipl. Anal. Optim.*, 1996, pp. 1535–1545.
- [81] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.
- [82] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, Feb. 2012.
- [83] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.
- [84] K. S. Bhattacharjee, H. K. Singh, and T. Ray, "A novel decomposition-based evolutionary algorithm for engineering design optimization," *J. Mech. Design*, vol. 139, no. 4, pp. 1–13, 2017.
- [85] F. Wilcoxon, S. K. Katti, and R. A. Wilcox, *Critical Values and Probability Levels for the Wilcoxon Rank Sum Test and the Wilcoxon Signed Rank Test*. New York, NY, USA: Amer. Cyanamid Company, 1963.
- [86] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Math. Program.*, vol. 91, no. 2, pp. 201–213, 2002.



Ahsanul Habib received the B.Sc. degree in Electrical and Electronic Engineering from the Chittagong University of Engineering and Technology, Chittagong, Bangladesh, in 2010. He is currently pursuing his Ph.D. with the School of Engineering and Information Technology, University of New South Wales (UNSW), Canberra, Australia.

His current research interests include computationally expensive evolutionary optimization, constraint handling, many-objective optimization, multi-fidelity optimization, data-driven decision making, and machine learning.



Hemant Kumar Singh received the B.Tech. degree in Mechanical Engineering from the Indian Institute of Technology Kanpur, Kanpur, India in 2007, and the Ph.D. degree from the University of New South Wales (UNSW), Canberra, Australia in 2011.

He is currently a Senior Lecturer with the School of Engineering and Information Technology, UNSW, Canberra, Australia. His current research interests include evolutionary computation, and design optimization.



Tapabrata Ray received the B.Tech. (Hons.), M.Tech., and Ph.D. degrees from the Indian Institute of Technology Kharagpur, Kharagpur, India.

He is a Professor with the School of Engineering and Information Technology, University of New South Wales (UNSW), Canberra, Australia where he leads the Multidisciplinary Design Optimization (MDO) Group. His current research interests include surrogate assisted/multi-fidelity optimization, multi/many-objective optimization, robust design, and decision-making.



Tinkle Chugh received the Ph.D. degree in mathematical information technology from the University of Jyväskylä (JYU), Jyväskylä, Finland, in 2017.

He is a Post-Doctoral Research Fellow with the Department of Computer Science, University of Exeter, Exeter, U.K. His thesis was a part of the “Decision Support for Complex Multiobjective Optimization Problems (DeCoMo)” project, where he collaborated with “Finland Distinguished Professor (FiDiPro)” Yaochu Jin from the University

of Surrey, Guildford, U.K. He received the best student paper award at IEEE Congress on Evolutionary Computation 2017.

His research interests are machine learning, data-driven optimization, evolutionary computation, and decision making.



Kaisa Miettinen received the Ph.D. degree in mathematical information technology from University of Jyväskylä (JYU), Jyväskylä, Finland.

She is a Professor of Industrial Optimization with JYU, where she is the Head of the Research Group on Industrial Optimization, and the Director of the Thematic Research Area Decision Analytics utilizing Causal Models and Multiobjective Optimization (DEMO). She was with the International Institute for Applied Systems Analysis, Laxenburg, Austria, the KTH Royal

Institute of Technology, Stockholm, Sweden, and the Helsinki School of Economics, Helsinki, Finland. She has authored almost 170 refereed journal, proceedings, and collection papers, edited 14 proceedings, collections, and special issues, and written a monograph entitled *Nonlinear Multiobjective Optimization*. Her current research interests include theory, methods, applications, and software of nonlinear multiobjective optimization including interactive and evolutionary approaches as well as data-driven decision support.

Dr. Miettinen is a recipient of the Georg Cantor Award of the International Society on Multiple Criteria Decision Making (MCDM) for independent inquiry in developing innovative ideas in the theory and methodology of MCDM. She belongs to the editorial boards of seven international journals and the Steering Committee of Evolutionary Multi-Criterion Optimization. She is a member of the Finnish Academy of Science and Letters, Section of Science, and the Immediate-Past President of the International Society on MCDM.