Performance Metrics Ensemble for Multiobjective Evolutionary Algorithms

Gary G. Yen, *Fellow*, *IEEE* and Zhenan He

***Abstract*—Evolutionary algorithms have been successfully exploited to solve multiobjective optimization problems. In literature, a heuristic approach is often taken. For a chosen benchmark problem with specific problem characteristics, the performance of multiobjective evolutionary algorithms (MOEAs) is evaluated via some heuristic chosen performance metrics. The conclusion is then drawn based on statistical findings given the preferable choices of performance metrics. The conclusion, if any, is often indecisive and reveals no insight pertaining to specific problem characteristics that the underlying MOEA could perform the best. In this paper, we introduce an ensemble method to compare MOEAs by combining a number of performance metrics using double elimination tournament selection. The double elimination design allows characteristically poor performance of a quality algorithm to still be able to win it all. Experimental results show that the proposed metrics ensemble can provide a more comprehensive comparison among various MOEAs than what could be obtained from single performance metric alone. The end result is a ranking order among all chosen MOEAs, but not quantifiable measures pertaining to the underlying MOEAs.**

***Index Terms*—Double elimination design, ensemble method, evolutionary algorithms (EAs), performance metrics.**

I. Introduction

Evolutionary d algorithms have established themselves as *the* approaches for exploring the Pareto-optimal fronts in multiobjective optimization problems. Multiobjective Evolutionary Algorithms (MOEAs) do not guarantee to identify optimal tradeoffs, but attempt to find a good approximation. Although numerous MOEAs are available today, effort is made in the continuing pursuit of more efficient and effective designs to search for Pareto optimal solutions for a given problem. By the No Free Lunch theorem [1], any algorithm's elevated performance over one class of problems is exactly paid for in loss over another class. Therefore, comparative studies are always conducted [2]. They aim at revealing advantages and weaknesses of the underlying MOEAs and at determining the best performance pertaining to specific class of problem characteristics. However, in absence of any established comparison criteria, none of the claims based on heuristically chosen performance metrics for the Pareto-optimal solutions generated can be made convincingly. In literature, when an MOEA is proposed, a number of benchmark problems are often selected to quantify the performance. Since these are artificially crafted benchmark functions, their corresponding Pareto fronts can be made available to measure the performance. Based on a set of heuristically chosen performance metrics, the proposed MOEA and some competitive representatives are evaluated statistically given a large number of independent trials. The conclusion, if any is drawn, is often indecisive and reveals no additional insight pertaining to the specific problem characteristics that the proposed MOEA would perform the best [3-4].

Zitzler *et al.* [2] proposed three optimization goals to be measured: the distance of the resulting non-dominated set to the Pareto-optimal front should be minimized, a good (in most cases uniform) distribution of the solutions found in objective space is desirable, and the extent of the obtained non-dominated front should be maximized. In literature, there are many ***unary*** performance metrics used to compare MOEAs. These metrics can be broadly divided into five categories according to the optimization goals. Each category mainly evaluates the quality of a Pareto-optimal set in one aspect only. The first category involves metrics assessing the number of Pareto optimal solutions in the set: *Ratio of Non-dominated Individuals* (RNI) [5] measures the proportion of the non-dominated solutions found with respect to the population size; *Error Ratio* (ER) [6] checks the proportion of non true Pareto points in the approximation front over the population size; *Overall Non-dominated Vector Generation* (ONVG) [6] simply counts the number of distinct non-dominated individuals found; and the n-ary performance metric, *Pareto Dominance Indicator* (NR) [7], measures the ratio of non-dominated solutions contributed by a particular approximation front to the non-dominated solutions provided collectively by all approximation fronts. Within the second category, metrics measuring the closeness of the solutions to the theoretical Pareto front are given: *Generational Distance* (GD) [6] measures how far the evolved solution set is from the true Pareto front; a complementary metric of GD called *Inverted Generational Distance* (IGD) [8] concerns how well is the Pareto-optimal front represented by the obtained solution set; and *Maximum Pareto Front Error* (MPFE) [6] focuses on the largest distance between the individual in the theoretical Pareto front and the points in the approximation front. In the third category, metrics are relating

Manuscript received January 15, 2012; revised April 30, 2012, July 30, 2012, and October 12, 2012.

The Authors are with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74075 USA (e-mail: gyen@okstate.edu; zhenan@okstate.edu).

on distribution of the solutions: *Uniform Distribution* (UD) [5] quantifies the distribution of an approximation front under a pre-defined parameter; *Spacing* [9] measures how evenly the evolved solutions distribute themselves; and *Number of Distinct Choices* (NDC$_\mu$) [10] identifies solutions that are sufficiently distinct for a special value μ. Fourth, metrics concerning spread of the solutions are included: *Maximum Spread* (MS) [2] measures how well the true Pareto front is covered by the approximation set. In the last category, metrics consider both closeness and diversity at the same time: *Hyperarea and Ratio* (or Hypervolume Indicator) [6, 11] calculates the volume covered by the approximation front with respect to a properly chosen reference point.

Furthermore, there are some ***binary*** performance metrics used to compare a pair of algorithms. I_ε [12] defines an ε-dominant relation between algorithms, enclosing hypercube indicator and coverage difference metrics (*D*-metric) [13]. The *C*-metric, or Set Coverage, considers the domination relations between two algorithms, i.e., how good an approximation front obtained from one algorithm dominates an approximation front obtained by another algorithm and vice versa [14].

However, no single metric alone can faithfully measure MOEA performance. Every metric can provide some specific, but incomplete, quantifications of performance and can only be used effectively under specified conditions. For example, *UD* does a poor job when the Pareto front is discontinuous, while Hypervolume Indicator can be misleading if the Pareto optimal front is non-convex [6]. This implies that one metric alone cannot entirely evaluate MOEAs under various conditions. Every metric focuses on some problem-specific characteristics while neglects information in others. Every *carefully crafted* metric has its unique attribute; no metrics alone can substitute others completely. Therefore, a single metric alone cannot provide a comprehensive measure for MOEAs. For a specific test problem, we cannot ascertain which metrics should be applied in order to faithfully quantify the performance of MOEAs. Common practice is to exploit various metrics to determine which combination is a better choice. Apparently, this process adds a heavy computational cost.

To overcome these deficiencies and arrive at a fair evaluation of MOEAs, performance metrics ensemble is proposed in this research work. The ensemble method uses multiple metrics collectively to obtain a better assessment than what could be obtained from any of single performance metric alone. Metrics ensemble not only can give a comprehensive comparison between different algorithms, but avoid the choosing process and can be directly used to assessing MOEAs.

In literature, the ensemble approaches can be found in statistics and machine learning. Supervised learning algorithms search through a space to find a suitable hypothesis that will make good predictions for a given problem. Ensemble methods combine multiple hypotheses to form a better one than could be obtained from any of the constituent models [15]. It always combines many weak learners in an attempt to produce a strong one. Furthermore, ensembles tend to yield better results when there is a significant diversity among the models [16]. There are some well-regarded designs: bagging [17], boosting [18], Bayesian model averaging [19], stacked generalization [20] and the random subspace method [21]. The application of multiple performance metrics is first introduced by Zitzler *et al.* in [22]. They discuss how to use hierarchies of metrics so that each metric is a refinement of the preference structure detected by a lower-level metric. However, there exists no publication in literature, to our best knowledge, regarding performance metrics ensemble. Without any reference information, MOEAs are evaluated and compared based on a single metric at a time. In this paper, we propose a double elimination tournament selection operator to compare approximation fronts obtained from different MOEAs in a statistically meaningful way. The double elimination design allows characteristically poor performance of a quality algorithm to still be able to win it all. In every competition, one metric is chosen randomly to compare. After the whole process, every metric could be selected multiple times and a final winning algorithm is to be identified. This final winner would have been compared under all the metrics considered so that we can make a fair conclusion based on an overall assessment.

The remaining sections complete the presentation of this paper. Section 2 provides the consolidated literature review on the performance metrics proposed in literature. Section 3 describes the proposed performance metrics ensemble approach in detail, including the double elimination tournament selection operator. In Section 4, we elaborate on the experiment results for selected benchmark problems. Finally, a conclusion is drawn in Section 5 along with pertinent observations.

II. Literature Review on Performance Metrics

Selected performance metrics will be briefly reviewed according to the way how they are classified in this paper.

A. Metrics Assessing the Number of Pareto Optimal Solutions in the Set

1) Ratio of Non-dominated Individuals (*RNI*) [5]:

The performance measure of an approximation front *X* is:

$$RNI = \frac{\left|\overline{X}\right|}{n}, \tag{1}$$

where $\overline{X}$ denotes the set of non-dominated individuals in population X whose size is n. Clearly $RNI \in [0,1]$, the larger is the better. When $RNI = 1$, it implies all the individuals in X are non-dominated. When $RNI = 0$, it implies none of the individuals in X is non-dominated. *RNI* is a significant measure in that it checks the proportion of non-dominated individuals in population, X.

2) Error Ratio (*ER*) [6]:

It is defined as the proportion of non true Pareto points:

$$ER = \frac{\sum_{i=1}^{n} e(x_i)}{n}. \tag{2}$$

x_i denotes an individual in an approximation front X and n is the number of individuals in X. $e(x_i)=0$, if $x_i \in PF_{true}$. Otherwise, $e(x_i)=1$. PF_{true} denotes the true Pareto set. This metric does assume the PF_{true} is made available. Lower values of *ER* refer to smaller proportions of non-true Pareto points in *X* and represent better non-dominated sets.

3) Overall Non-dominated Vector Generation (*ONVG*) [6]:

It measures the number of non-dominated individuals found in an approximation front during MOEA evolution. It is defined as:

$$ONVG = |PF_{known}|, \tag{3}$$

where PF_{known} represents the obtained approximation front. From [23], too few individuals in PF_{known} make the front's representation poor and too many vectors may overwhelm the decision maker. Also, [24] proves that algorithm *A* outperforms B on this metric does not necessarily imply algorithm A is clearly better than B.

4) Pareto Dominance Indicator (*NR*) [7]:

Considering the approximation fronts, $A_1, A_2, \cdots, A_m$ obtained by different algorithms, this n-ary metric measures the ratio of non-dominated solutions that is contributed by a particular solution set A_1 to the non-dominated solutions provided by all algorithms:

$$NR(A_1, A_2, \cdots, A_m) = \frac{|A_1 \cap B|}{|B|}, \tag{4}$$

where $B = \{b_i | \forall b_i, \neg\exists a_j \in (A_1 \cup A_2 \cup \cdots \cup A_m) \prec b_i\}$, and $a_j \prec b_i$ implies that a_j dominates b_i. A_1 is the set under evaluation.

B. Metrics Measuring the Closeness of the Solutions to the True Pareto Front

1) Final Generational Distance (*GD*) [6]:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}, \tag{5}$$

where $d_i = \min_j \| f(x_i) - PF_{true}(x_j) \|$ refers to the distance in objective space between individual x_i and the nearest member in the true Pareto front, and n is the number of individuals in the approximation front. This metric, assuming PF_{true} is readily available, is a measure representing how "far" the approximation front is from the true Pareto front. Lower value of *GD* represents a better performance.

2) Inverted Generational Distance (*IGD*) [8]

This metric measures both convergence and diversity. Let PF_{true} is a set of uniformly distributed solutions in true Pareto front. X is the set of non-dominated solutions in the approximation front PF_{known}:

$$IGD = \frac{\sum v \in PF_{true} \, d(v, X)}{|PF_{true}|} \tag{6}$$

$d(v, X)$ denotes the minimum Euclidean distance between *v* and the points in X. To have a low value of *IGD*, the set X should be close to PF_{true} and cannot miss any part of the whole PF_{true}.

3) Maximum Pareto Front Error (*MPFE*) [6]:

It measures a worst case scenario in term of the largest distance in the objective space between any individual in the approximation front and the corresponding closest vector in the true Pareto front.

$$MPFE = \max_i d_i. \tag{7}$$

d_i, defined earlier, is referred to as the distance in objective space between individual x_i and the nearest member in the true Pareto front. From [23], for a non-dominated set, a good performance in *MPFE* does not ensure it is better than another one with a much worse *MPFE*.

C. Metrics Focusing on Distribution of the Solutions

1) Uniform Distribution (*UD*) [5]:

It measures the distribution of non-dominated individuals on the found trade-off surface. For a given set of non-dominated individuals $\overline{X}$ in a population *X*:

$$UD = \frac{1}{1 + S_{nc}}, \tag{8}$$

where $S_{nc} = \sqrt{\frac{\sum_{i=1}^{N_{\overline{X}}} \left(nc(\overline{x}_i) - \overline{nc}(\overline{X})\right)^2}{N_{\overline{X}} - 1}}$ is the standard deviation of niche count of the overall set of non-dominated individuals in $\overline{X}$, $N_{\overline{X}}$ is the size of the set $\overline{X}$, and $\overline{nc}(\overline{X})$ is the mean value of niche counts, $nc(\overline{x}_i), \forall i = 1, 2, \cdots, N_{\overline{X}}$. Specifically, niche count of individual x_i is defined as

$$nc(\overline{x}_i) = \sum_{j=1, j\neq i}^{N_{\overline{X}}} Sh(x_i, x_j),$$

$$Sh(x_i, x_j) = \begin{cases} 1, & \text{if } d(x_i, x_j) < \sigma_{share} \\ 0, & \text{otherwise} \end{cases}.$$

$d(x_i, x_j)$ is the distance between individuals x_i and x_j in the objective space, and σ_{share} is a user-defined parameter to quantify the closeness.

2) Spacing [9]:

This metric is a value measuring how evenly the non-dominated solutions are distributed along the approximation front,

$$S = \sqrt{\frac{1}{\overline{n}} \sum_{i=1}^{\overline{n}} (d_i - \overline{d})^2}, \tag{9}$$

where d_i is the Euclidean distance in objective space between individual x_i and the nearest member in the true Pareto front, and $\overline{n}$ is the number of individuals in the approximation front. This metric requires low computational overhead and can be generalized to more than two dimensions.

3) Number of Distinct Choices (NDC_μ) [10]:

In this metric, only those solutions that are sufficiently distinct from one another should be accounted for as useful design options. Let μ, $(0<\mu<1)$, be a user specified parameter which can be used to divide an m-dimensional objective space into $1/\mu^m$ number of small grids. Each of the grids refers to indifference region $T_\mu(q)$ wherein any two solutions within the same grid are considered similar to one another. The quality $NT_\mu(q,P)$ indicates whether or not there is any individual $p_k \in P$ that falls into the region $T_\mu(q)$. Specifically

$$NT_\mu(q,P)=\begin{cases}1, & \exists p_k \in P, p_k \in T_\mu(q)\\ 0, & \forall p_k \in P, p_k \notin T_\mu(q)\end{cases}.$$

$NDC_\mu(P)$ defines the number of distinct choices for a pre-specified value of μ:

$$NDC_\mu(P)=\sum_{l_m=0}^{(1/\mu)-1}\cdots\sum_{l_2=0}^{(1/\mu)-1}\sum_{l_1=0}^{(1/\mu)-1}NT_\mu(q,P) \tag{10}$$

From [10], for a pre-specified value of μ, an observed Pareto solution set with a higher value of the quantity $NDC_\mu(P)$ is preferred to a set with a lower value.

D. *Metrics Concerning Spread of the Solutions*

1) Maximum Spread (*MS*) [2]:

It addresses the range of objective function values and takes into account the proximity to the true Pareto front, assuming available. This metric is applied to measure how well the PF_{true} is covered by the PF_{known}.

$$MS=\sqrt{\frac{1}{M}\sum_{i=1}^{M}\left[\frac{\min(PF_{known,i}^{\max},PF_{true,i}^{\max})-\max(PF_{known,i}^{\min},PF_{true,i}^{\min})}{PF_{true,i}^{\max}-PF_{true,i}^{\min}}\right]^2} \tag{11}$$

where $PF_{known,i}^{\max}$ and $PF_{known,i}^{\min}$ are the maximum and minimum of the ith objective in PF_{known}, respectively; and $PF_{true,i}^{\max}$ and $PF_{true,i}^{\min}$ are the maximum and minimum of the ith objective in PF_{true}, respectively. M denotes the number of objectives considered. A higher value of *MS* reflects that a larger area of the PF_{true} is covered by the PF_{known}.

E. *Metrics Considering both Closeness and Diversity*

1) Hyperarea and Ratio (*Hypervolume Indicator*) [6, 11]:

It calculates the hypervolume of the multi-dimensional objective space enclosed by approximation front PF_{known} and a reference point. For example, an individual x_i in PF_{known} for a two-dimensional MOP defines a rectangle area, $a(x_i)$, bounded by an origin and $f(x_i)$. The union of such rectangle areas is referred to as Hyperarea of PF_{known},

$$H(PF_{known})=\left\{\bigcup_i a(x_i)\middle|\forall x_i \in PF_{known}\right\} \tag{12}$$

As pointed out in [11], this metric requires defining a reference point of the region and could be misleading if PF_{known} is nonconvex. In [25], suggestion is given as how to properly choose a reference point. In [6], Veldhuizen also propose a Hyperarea Ratio metric defined as:

$$HR=\frac{H(PF_{known})}{H(PF_{true})}. \tag{13}$$

Apparently, PF_{true} is given as a reference. In the proposed performance metrics ensemble to be presented in the next section, we adopt the Hyperarea Ratio metric.

III. Performance Metrics Ensemble

A. *The Proposed Framework*

Figure 1 shows the process of Performance Metrics Ensemble proposed. The final output from the performance metrics ensemble is a ranking order of all MOEAs considered. Please note the proposed design does not provide a quantifiable performance measure for a given MOEA. Instead it attempts to rank the selected MOEAs comprehensively through a collection of performance metrics. A number of MOEAs are presented as input. Given the same initial population, each of MOEAs considered generates an approximation front. Among these approximation fronts, a winning front is selected according to a randomly chosen performance metric. To arrive at a statistically meaningful conclusion, 50 independent trials are conducted. This process results into 50 approximation fronts deriving from the MOEAs considered. A double elimination tournament selection is applied to these 50 approximation fronts and one ultimate winning approximation front will be identified. The MOEA which is responsible to this approximation front will be assigned with ranking one. This MOEA is regarded as the winning algorithm among all MOEAs participated. The approximation fronts which are generated by this winning MOEA will be removed from 50 approximation fronts. The remaining approximation fronts will then go through another round of double elimination tournament to identify the second winning MOEA with ranking two. The process will repeat until the complete ranking order of all MOEAs considered is assigned.

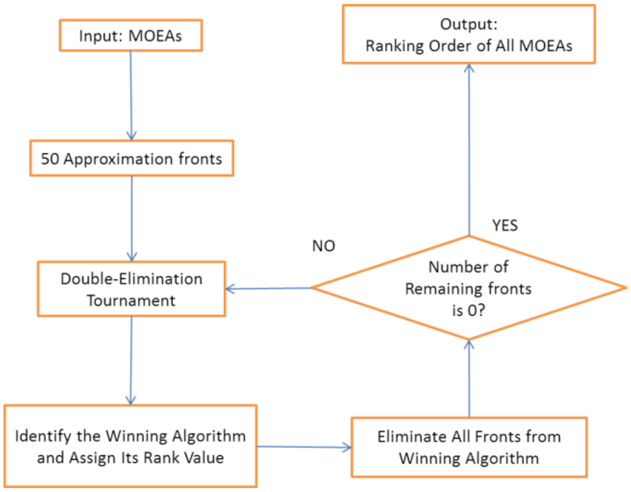

Fig. 1 The proposed framework for Performance Metrics Ensemble

B. Double Elimination Tournament

The proposed Double Elimination Tournament down selects an approximation front (as the winning front) out of all approximation fronts available using a series of binary tournament selections. In each tournament selection, a performance metric from metrics ensemble is randomly chosen for comparison.

Figure 2 depicts the process of double elimination tournament in a general setting. Suppose the tournament has a pool size of N approximation fronts to begin with. The N/2 "qualifier" binary tournaments are held as normal, and the whole pool is divided into two parts: winner bracket contains N/2 winners and loser bracket N/2 losers. Then, in each of the bracket, N/4 binary tournament selections are competed so that each part is further divided again. In both parts, there are N/4 new winners and N/4 new losers. The N/4 losers from loser bracket will lose twice and be eliminated from further consideration. The N/4 winner from winner bracket will be reserved in winner bracket for the next round of competition. Additionally, N/4 losers from winner bracket and N/4 winners from loser bracket will be paired for binary tournaments. Specifically, one approximation front from winner bracket and one from loser bracket will be matched for a binary tournament. Afterward, we obtain N/4 winners which will be placed in the loser bracket for the next round of competition. Those N/4 losers lost twice and will be eliminated from the pool. This process reduces the total number of approximation fronts in the pool from N to N/2 (i.e., N/4 in winner bracket and N/4 in loser bracket). Repeat the same process; the number of candidate approximation fronts will be trimmed down from N/2 to N4, N/4 to N/8, and eventually down to 2. The remaining two will then compete given a randomly chosen performance metric. If the one from winner bracket wins, it will be declared as the final winner. If the one from loser bracket wins, one more round of competition will be held to decide the ultimate winner. Please note if N is an odd number to begin with the double elimination tournament process, one approximation front randomly chosen will be held back and (N-1)/2 binary tournaments will be called. After competitions, the one that was held back will be added into both the winner bracket and loser bracket to assure it will be fully considered in the competition process.

The motivation for applying the double elimination tournament is that it gives every individual approximation front at least two chances to take part in the competition. This design would be helpful to preserve good approximation fronts. Because of the stochastic process, one quality approximation front may lose the competition if a biased performance metric is chosen. For example, for a benchmark problem with discontinuous Pareto front, performance metric UD will not offer a fair assessment. If this occurs in the single elimination tournament, a quality front could be lost forever. However, in the double elimination tournament, even an approximation front loses once; it still has an opportunity to compete and to win it all. Double elimination design allows a characteristically poor performance of a quality MOEA under the special environment still be able to win it all.

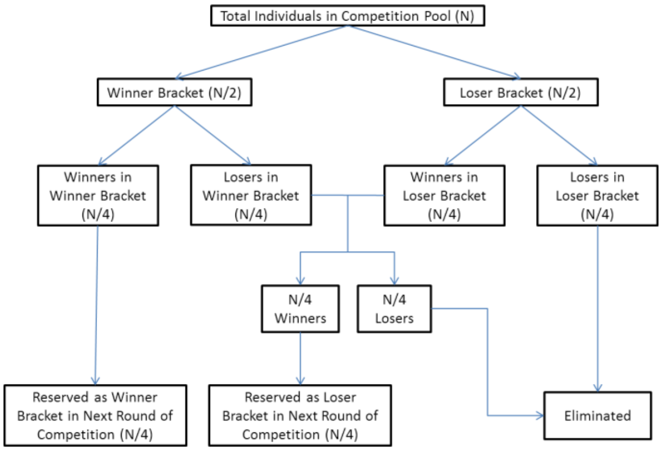
Total Individuals in Competition Pool (N)
Winner Bracket (N/2)
Loser Bracket (N/2)
Winners in Winner Bracket (N/4)
Losers in Winner Bracket (N/4)
Winners in Loser Bracket (N/4)
Losers in Loser Bracket (N/4)
N/4 Winners
N/4 Losers
Reserved as Winner Bracket in Next Round of Competition (N/4)
Reserved as Loser Bracket in Next Round of Competition (N/4)
Eliminated

Fig. 2 The process of Double Elimination Tournament

Specifically, each competing MOEA will produce an approximation front given the same initial population. One will be donned as the winner using a randomly chosen performance metric. Out of 50 independent runs, 50 approximation fronts will be resulted. Some may come from the same MOEA. It is also possible that an MOEA has no representation in the 50 approximation fronts. Out of such a large number of competitions, most likely every performance metric will be chosen multiple times to compete.

In Figure 3(a), 25 pairs of binary tournaments will be held to result 25 winners in winner bracket and 25 losers in loser bracket. In every competition, a randomly chosen performance metric from metric ensemble will be used. In each bracket, one approximation front will be randomly chosen. The remaining 24 approximation fronts will form 12 pairs of binary tournaments. The one that was held back will join both winner bracket and loser bracket to result into 13 winners and 13 losers in each bracket. Those 13 winners from winner bracket will be reserved as winners for the next round of double elimination tournament. Those 13 losers from loser bracket, which lost twice already, will be eliminated from the candidate pool. 13 losers from winner bracket and 13 winners from loser bracket are then paired to compete. 13 winners will be reserved as losers in the next round of double elimination tournament, while 13 losers which lost twice each and will be eliminated from further consideration.

In Figure 3(b), 13 remaining approximation fronts are in the winner bracket while 13 are in the loser bracket. A similar process of double elimination tournament continues to trim down the number of approximation fronts to 7 winners and 7 losers. 14 more approximation fronts will be eliminated from the pool. In Figure 3(c), 7 remaining approximation fronts are in the winner bracket and 7 are in the loser bracket. A similar process repeats to cut down the number of approximation fronts to 4 winners and 4 losers. 8 more approximation fronts will be eliminated from the pool. In Figure 3(d), the process takes one more step to down select the number of approximation fronts to

2 winners and 2 losers, while in Figure 3(e), one more step results into one winner and one loser. In Figure 3(f), the remaining two will compete based on a randomly chosen performance metric. If the one from winner bracket wins, it will be declared as the final winner. If the one from loser bracket wins, one more round of competition will be held to decide the ultimate winner. The MOEA which is responsible to the ultimate winning approximation front will be honored as the winning MOEA with ranking order one. Please note 101 binary tournaments will be called to decide the overall winning MOEA beginning with 50 approximation fronts.

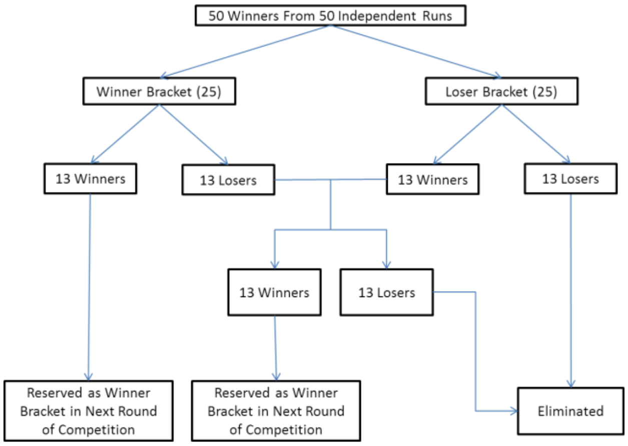
50 Winners From 50 Independent Runs
Winner Bracket (25)
Loser Bracket (25)
13 Winners
13 Losers
13 Winners
13 Losers
13 Winners
13 Losers
Reserved as Winner Bracket in Next Round of Competition
Reserved as Winner Bracket in Next Round of Competition
Eliminated

Fig. 3(a). From 50 individuals down to 26 individuals

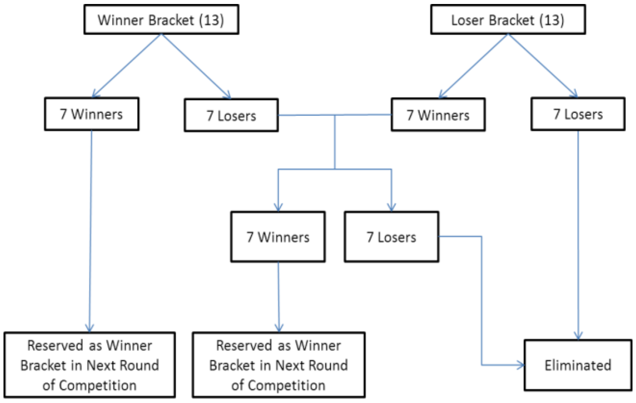
Winner Bracket (13)
Loser Bracket (13)
7 Winners
7 Losers
7 Winners
7 Losers
7 Winners
7 Losers
Reserved as Winner Bracket in Next Round of Competition
Reserved as Winner Bracket in Next Round of Competition
Eliminated

Fig. 3(b). From 26 individuals down to 14 individuals

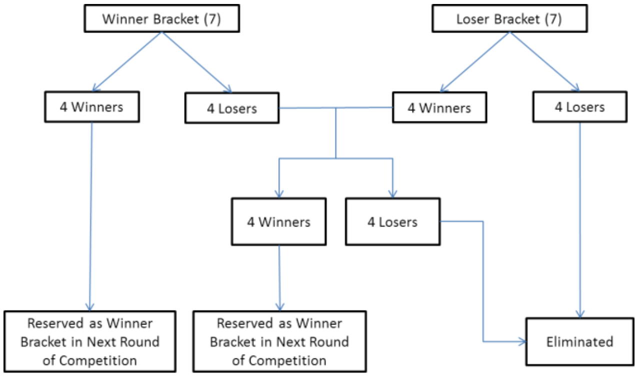
Winner Bracket (7)
Loser Bracket (7)
4 Winners
4 Losers
4 Winners
4 Losers
4 Winners
4 Losers
Reserved as Winner Bracket in Next Round of Competition
Reserved as Winner Bracket in Next Round of Competition
Eliminated

Fig. 3(c). From 14 individuals down to 8 individuals

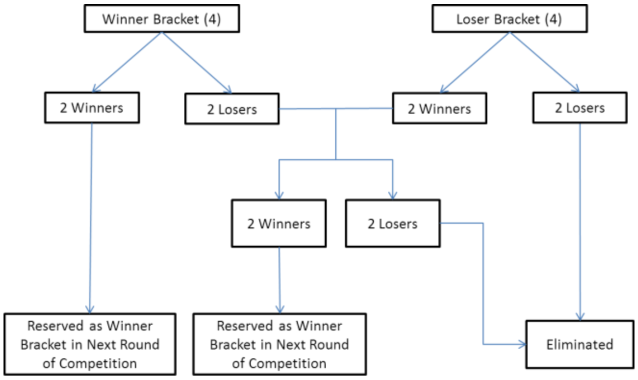
Winner Bracket (4)
Loser Bracket (4)
2 Winners
2 Losers
2 Winners
2 Losers
2 Winners
2 Losers
Reserved as Winner Bracket in Next Round of Competition
Reserved as Winner Bracket in Next Round of Competition
Eliminated

Fig. 3(d). From 8 individuals down to 4 individuals

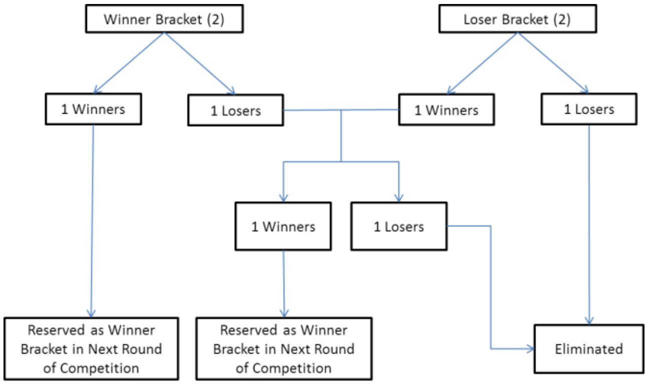
Winner Bracket (2)
Loser Bracket (2)
1 Winners
1 Losers
1 Winners
1 Losers
1 Winners
1 Losers
Reserved as Winner Bracket in Next Round of Competition
Reserved as Winner Bracket in Next Round of Competition
Eliminated

Fig. 3(e). From 4 individuals down to 2 individuals

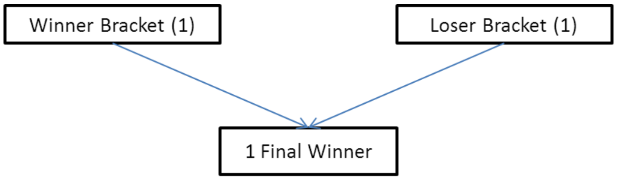

Fig. 3(f). From 2 individuals down to 1 winner

Those approximation fronts out of original 50, if generated from the winning MOEA, will be removed from the candidate pool. The double elimination tournament process repeats until a winning approximation front is found and the MOEA which is responsible to this winning approximation front will be declared as the second winning MOEA with ranking order two. The process repeats until all MOEAs are ranked.

IV. Experimental Results

A. *Selected MOEAs for Comparison*

In the experiment, five state-of-the-art MOEAs are chosen for competition. They are SPEA 2 [26], NSGA-II [27], IBEA [28], PESA-II [29], and MOEA/D [30]. In the proposed framework, no restriction is placed upon any MOEAs ever developed. Indeed, even multiobjective particle swarm optimization algorithms [31-32] could be considered, as long as a population based heuristic is adopted to solve a multiobjective optimization problem. A brief overview of each chosen MOEA is given below.

SPEA 2 [26] assigns a strength value to each individual in both main population and elitist archive which incorporates both dominated and density information. To avoid individuals dominated by the same archive members having identical fitness values, both dominating and dominated relationships are taken into account. The final rank value of a current individual is generated by the summation of the strengths of the individuals that dominate it. The density value of each individual is obtained by the nearest neighbor density estimation. The final fitness value is the sum of rank and density values. In addition, the number of elitists in elitist archive is maintained to be constant.

NSGA-II [27] proposes a non-dominated sorting approach to assign Pareto ranking and a crowding distance assignment method to implement density estimation for each individual. In a tournament selection design between two individuals, the one with a lower rank value or the one located in a less crowded region when both belong to the same front will be selected. A fast non-dominated sorting approach, an elitism scheme, and a parameter-less niching sharing method are combined to produce a better spread of solutions in some testing problems.

The main idea of IBEA [28] is to apply a binary performance measure directly to the selection process. IBEA is combined with arbitrary indicators which are first defined by the optimization goal and can be adapted to the preferences of the user without any additional diversity preservation mechanism such as fitness sharing.

In PESA-II [29], the unit of selection is a hyperbox in objective space. PESA-II assigns selective fitness to the hyperboxes in objective space which are occupied by at least one individual in the current approximation to the Pareto front. The resulting selected individual is randomly chosen from the hyperbox. This method is more effective to obtain a good spread in the front than selection based on individuals.

MOEA/D [30] decomposes a multiobjective optimization problem into a number of scalar optimization subproblems and optimizes them simultaneously. MOEA/D has lower computational complexity at each generation because each subproblem is optimized by only using information from its several neighboring subproblems.

B. Selected Benchmark Test Problems

We utilize five widely used bi-objective ZDT test instances (i.e., ZDT 1, ZDT 2, ZDT 3, ZDT 4, and ZDT 6) [14], a 3-objective DTLZ 2 [28, 33], two 5-objective WFG 1 and WFG 2 [34], and a 10-objective DTLZ 1 [28, 33] in comparing all MOEAs. These benchmark functions are carefully crafted to exploit specific problem characteristics to challenge the underlying MOEAs at hand.

C. Selected Performance Metrics

In Section 1, we have broadly classified performance metrics into five groups. In this experiment, five metrics from each of all five different groups are chosen. They are *Pareto Dominance Indicator* (NR), *Inverted Generational Distance* (IGD), *Spacing*, *Maximum Spread* (MS), and *Hypervolume Indicator* (also called *S*-metric in [13]). The less the IGD and Spacing values, the better the algorithm's performance; the more the NR, MS, and S values, the better the algorithm's performance. Please note binary performance metrics, such as ε -indicator or C-metric, can be easily adopted into the proposed design since binary tournament is used here as a baseline.

In S-metric, we define the reference point according to [25] for different benchmark problems. That is, for ZDT1 and ZDT4, we choose the reference point to be (3, 100). The choices of reference points for ZDT 2, ZDT 3, ZDT 6 and 3-objective DTLZ 2 are (3/2, 4/3), (100, 5.446), (1.497, 4/3), (1.180, 1.180, 1.180), respectively. For 5-objective WFG 1 and WFG 2, based on the multiple experiments results in [34], the reference point is set to be (20, 20, 20, 20, 20). For 10-objective DTLZ 1, each dimension of the reference point is set at 100.

D. Parameter Setting in Experiment

According to [26-30], the population size in all five MOEAs is set to be 100 for all of the 2-objective test instances, 300 for the 3-objective test instance, and 500 for the more than three objectives test instances. The stopping criterion is set at 250 generations. Initial populations are generated by uniformly, randomly sampling from the search space in all the algorithms.

The simulated binary crossover (SBX) and polynomial mutation are used in SPEA 2 [26], NSGA-II [27], IBEA [28], and MOEA/D [30]. The crossover operator generates one offspring, which is then modified by the mutation operator. Following the practice in [27], the distribution indexes in SBX and the polynomial mutation are set to be 20. The crossover rate is 1.00, while the mutation rate is $1/l$ and l is the number of decision variables.

In IBEA [28], the Hypervolume indicator is selected as the comparison indicator. In PESA-II [29], the crossover rate is chosen to be 0.7 and uniform crossover is used. Hyper-grid size is 32×32. Other control parameters remain identical. In MOEA/D [30], the number of the weight vectors in the neighborhood of each weight vector T is set to be 20.

E. Experiment Results

i. <u>2-objective ZDT 1</u>

1) Preliminary Iteration

This step generates 50 approximation fronts as the initial population for double elimination tournament. In these 50 winning fronts, SPEA 2 wins 19 times, NSGA-II wins 11 times, IBEA wins 3 times, PESA-II wins 5 times, and MOEA/D wins 12 times. During the competitions, performance metric is randomly chosen from five available metrics. In summary, IGD is used 11 times, NR 10 times, Spacing 12 times, S-metric 10 times, and MS 7 times.

2) Iteration 1

This is the first step in the double elimination tournament that 50 fronts are competed to survive 26 in the candidate pool. In these 26 fronts, SPEA 2 wins 9 times, NSGA-II wins 8 times, IBEA wins 0 time, PESA-II wins 1 time, and MOEA/D wins 8 times. Note that IBEA is completely eliminated with any hope to win the overall competition.

The 50 approximation fronts are first paired into 25 binary tournaments. In this process, IGD is chosen 4 times, NR 7 times, Spacing 4 times, S-metric 6 times, and MS 4 times, respectively.

The remaining process involves 12 binary tournaments to generate 13 winners reserved in a winner bracket for the next iteration, 12 binary tournaments to generate 13 losers to be eliminated from further consideration, and 13 binary tournaments to generate 13 fronts reserved in a loser bracket for the next iteration. Altogether 37 binary tournaments are held using 37 randomly chosen performance metrics: IGD is used 9 times, NR 6 times, Spacing 7 times, S-metric 7 times, and MS 8 times.

3) Iteration 2

In the second step of double elimination tournament, 26 remaining approximation fronts are trimmed down to 14 in the candidate pool. In these remaining 14 fronts, SPEA 2 accounts for 6, NSGA-II 2, PESA-II 1, and MOEA/D 5 times.

Out of 19 binary tournaments held in this iteration, performance metric of IGD is used 5 times, NR 4 times, Spacing 5 times, S-metric twice, and MS 3 times.

4) Iteration 3

In the third step of double elimination tournament, 14 approximation fronts remained in the pool are once again down select to 8 fronts. In these 8 fronts, SPEA 2 wins 3 times, NSGA-II wins twice, and MOEA/D wins 3 times. Note that PESA-II is also eliminated in this iteration.

Out of 10 binary tournaments called in this iteration, IGD is chosen three times, NR 1 time, Spacing 3 times, S-metric 3 times, while MS is not used at all.

5) Iteration 4

It is the fourth step of double elimination tournament, 8 approximation fronts will be further reduced down to 4 survivors. In these 4 fronts, SPEA 2 accounts for two wins and MOEA/D 2 times. Note that NSGA-II is eliminated in this iteration to be the ultimate winner.

Out of 6 binary tournaments held in this iteration, IGD is chosen once, Spacing twice, S-metric once, and MS twice.

6) Iteration 5

In the fifth step of double elimination tournament, 4 approximation fronts are trimmed one more time to two in the candidate pool. In these two remaining fronts, SPEA 2 wins once and MOEA/D wins once.

Out of three competitions, performance metrics IGD is selected once, S-metric once, and MS once.

7) Iteration 6

In the final step of double elimination tournament, the ultimate winner is to be identified. The final winner is SPEA 2 and performance metric, S-metric, is chosen to compete. Note SPEA 2 is coming from winner bracket.

8) Iteration 7

Removing all the fronts (i.e., 18) generated by SPEA 2 out of 50 approximation fronts, the remaining 32 fronts continue through double elimination tournaments. MOEA/D is identified as the second winner with respect to the benchmark function ZDT 1. NSGA-II is the third winner. When the proposed framework complete, the resulting ranking order shows: rank 1-SPEA 2, rank 2- MOEA/D, rank 3-NSGA-II, rank 4-PESA-II, and rank 5-IBEA for the benchmark function ZDT 1.

Please note if S-metric is been used to compare SPEA 2 and NSGA-II, NSGA-II would be declared as the winner. However, if any of the remaining four performance metrics is been used, SPEA 2 will win the competitions.

The experiment result in [30] also confirmed that MOEA/D performs better than NSGA-II in ZDT1.

35 repeated and independent experiments (given different initial populations to begin with) on ZDT 1 show consistent finding in the ranking order. This implies the robustness of the proposed performance metric ensemble approach in ranking the selected MOEAs.

ii. <u>2-objective ZDT 2</u>

Due to the similarity in the process, no detail is given here. The final ranking order for benchmark function ZDT 2 is: rank 1-SPEA 2, rank 2-MOEA/D, rank 3-NSGA-II, rank 4-IBEA, and rank 5-PESA-II.

During the process of generating the ultimate winner, SPEA 2, came from loser bracket, takes two rounds of competitions to beat NSGA-II to be donned the winner with rank one. In these two rounds of competitions, S-metric is first chosen as performance metric, while IGD is then used in the final binary tournament. Apparently, even though SPEA 2 is regarded as the winner from the ensemble, MOEA/D comes in as the close second.

Also, in [30], MOEA/D shows better performance than NSGA-II in ZDT2.

iii. <u>2-objective ZDT 3</u>

The final ranking order for benchmark function ZDT 3 is: rank 1-NSGA-II, rank 2-MOEA/D, rank 3-IBEA, rank 4- SPEA 2, and rank 5-PESA-II.

The experiment result in [30] also confirmed that MOEA/D performs worse than NSGA-II in this benchmark function.

It is interesting to observe that if S-metric is been used to compare NSGA-II and MOEA/D, MOEA/D would be acknowledged as the winner. However, if any of the remaining four performance metrics is been used, NSGA-II will win the competitions.

iv. <u>2-objective ZDT 4</u>

The final ranking order for benchmark function ZDT 4 is: rank 1-MOEA/D, rank 2-NSGA-II, rank 3-PESA-II, rank 4-IBEA, and rank 5-SPEA 2.

It is interesting to observe that even SPEA 2 is the ultimate winner in ZDT 1 and ZDT 2 test functions; it is eliminated from the process in the early stage during every independent competition for ZDT4. It clearly implies that SPEA 2 has difficulties handling problems with many local Pareto-optimal fronts.

Please note if IGD or NR was used to compare NSGA-II and MOEA/D, NSGA-II would win the close competition. However, if other metrics are involved, MOEA/D will survive to be a close winner.

v. 2-objective ZDT 6

The final ranking order for benchmark function ZDT 6 is: rank 1-MOEA/D, rank 2-IBEA, rank 3-NSGA-II, rank 4-SPEA 2, and rank 5-PESA-II.

Reference [28] shows IBEA performs better than NSGA-II and SPEA 2, and reference [30] acknowledges the same result that MOEA/D performs better than NSGA-II in the test function ZDT 6.

Again, SPEA 2 and PESA-II are eliminated at very early stage during every round of competitions. It appears SPEA 2 is having difficulties in handling problems with Pareto-optimal solutions non-uniformly distributed over the global Pareto front.

vi. 3-objective DTLZ 2

The final ranking order for benchmark function DTLZ 2 is: rank 1-IBEA, rank 2-MOEA/D, rank 3-SPEA 2, rank 4-NSGA-II, and rank 5-PESA-II.

Reference [27] has suggested that SPEA 2 seems to possess advantages over NSGA-II in higher dimensional problems. In [28], IBEA is shown to be better than SPEA 2 and NSGA-II in this benchmark function. The experiment result in [30] has also identified that MOEA/D generates a better result than NSGA-II does. These findings throughout literature have been consistent to what we have observed through the proposed performance metrics ensemble.

It is interesting to observe that IBEA which does not perform very well on ZDT1-ZDT4 benchmark problems is the clear winner of this three-dimensional test function. Please note if Spacing is been used to compare MOEA/D and IBEA, MOEA/D would be declared as the winner. However, if any of the remaining four performance metrics is used, IBEA win the competitions.

vii. 5-objective WFG 1

The final ranking order for benchmark function WFG1 is: rank 1-IBEA, rank 2-MOEA/D, rank 3-NSGA-II, rank 4- SPEA 2, and rank 5-PESA-II.

It is interesting to observe that if S-metric is been used to compare IBEA and MOEA/D, MOEA/D would be acknowledged as the winner. Actually, in the problem WFG1, these two algorithms have nearly equal performance. Reference [35] also confirms the same result that IBEA is much better than NSGA-II in many-objective optimization problems.

viii. 5-objective WFG 2

The final ranking order for benchmark function WFG 2 is: rank 1-IBEA, rank 2-NSGA-II, rank 3-MOEA/D, rank 4- SPEA 2, and rank 5-PESA-II.

It is interesting to observe that if Spacing metric is used SPEA 2 receives a much better performance than others. That means, SPEA 2 has a good distribution between each individuals in the approximation front. However, it achieves a very low final ranking. This is because both convergence and spread measures of its approximation front are very poor.

According to [34], NSGA-II is claimed to have a better performance than MOEA/D and SPEA 2 in WFG 2.

IBEA is favored in both high-dimension objectives optimization problems (e.g., DTLZ2) and many-objectives optimization problem (e.g., WFG1 and WFG 2). This seems to imply that IBEA bears advantages over other algorithms to deal with challenges inherited through increasing the number of objectives.

ix. 10-objective DTLZ 1

The final ranking order for benchmark function DTLZ 1 is: rank 1-IBEA, rank 2-MOEA/D, rank 3-NSGA-II, rank 4- SPEA 2, and rank 5-PESA-II.

If Spacing metric is used to compare NSGA-II and SPEA 2, SPEA 2 would be acknowledged as the winner. Actually, in this problem, these two algorithms have nearly equal performance. Therefore, this experimental result does not contradict the conclusion in [27].

Again, IBEA and MOEA/D perform much better than other MOEAs in this many-objectives optimization problem.

F. Computational Complexity Analysis

Assuming in the first 50 approximation fronts, each of five MOEAs is responsible for 10 fronts. The total number of metric evaluations needed to complete the ranking order for a given benchmark function is 804, including the 250 metric evaluations made to generate the 50 approximation fronts before the performance metrics ensemble process. Every time one metric from ensemble is randomly chosen to evaluate one approximation front. 804 is considered an average measure for computational time of the proposed performance metric ensemble process. This is with respect to 1,250 metric evaluations if each of five metrics is used to quantify the performance of five MOEAs out of 50 independent trials.

Given a desktop computer with Intel(R) 2.20 GHZ processor and 4GB of RAM, each experiment will be run for 35 independent times. For a 2-objective benchmark problem, ZDT 1, assuming each approximation front having 100 solutions and true Pareto-front having 15 solutions, the average CPU time for the proposed performance metrics ensemble process is 2.1723s, while the maximum time needed is 3.7680s and minimum 1.2500s. For a 3-objective benchmark problem, DTLZ 2, assuming each approximation front having 300 solutions and true Pareto-front having 20 solutions, the average CPU time for performance metrics ensemble process is 7.6662s, while the maximum time needed is 13.9060s and minimum 4.6830s. For a 5-objective benchmark problem, WFG 1, assuming each approximation front having 500 solutions and true Pareto-front having 100 solutions, the average CPU time for performance metrics ensemble process is 17.0892s, while the maximum time needed is 20.9790s and minimum 12.2570s. For a 10-objective benchmark problem, DTLZ 1, assuming each approximation front having 500 solutions and true Pareto-front having 200 solutions, the average CPU time for performance metrics ensemble process is 25.7763s, while the maximum time needed is 34.5440s and minimum 20.0700s. Table I shows the computational time (in second) of the performance metrics ensemble process for each test problem.

Table I:
Computational time for each test problem

Problem	Max. Time	Ave. Time	Min. Time
2-obj ZDT 1	3.7680	2.1723	1.2500
2-obj ZDT 2	3.6510	2.2104	1.2303
2-obj ZDT 3	3.9230	2.1938	1.5091
2-obj ZDT 4	4.2315	2.3019	1.2803
2-obj ZDT 6	4.0129	2.2801	1.4671
3-obj DTLZ 2	13.9060	7.6662	4.6830
5-obj WFG 1	20.9790	17.0892	12.2570
5-obj WFG 2	22.1061	17.4671	12.7629
10-obj DTLZ 1	34.5440	25.7763	20.0700

As expectedly, the computation time needed for performance metrics ensemble process grows in a polynomial order when the number of objectives increases. Meanwhile, different test problems with the same number of objectives call for nearly the same computational time, e.g., ZDT 1-ZDT 6. Therefore, the computational time of performance metrics ensemble process is mainly determined by the number of objectives involved in the test problem.

G. Comparison Results between Individual Metric Alone and Metrics Ensemble

The focus paid here is to compare the final ranking orders by each performance metric individually and by the proposed metric ensemble. In tables below, “A1” represents SPEA 2, “A2” NSGA-II, “A3” MOEA/D, “A4” PESA-II, and “A5” IBEA. From Tables II-III, in simple problems such as ZDT 1 and ZDT 2, ranking orders generated by different metric individually are slightly different from each other. Metrics ensemble, more or less performing a majority vote, achieves a good compromise from all results collectively.

Table II:
Comparison Results of ZDT 1

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A1	A2	A1	A1	A1	A1
2	A3	A1	A3	A3	A2	A3
3	A2	A3	A2	A2	A3	A2
4	A4	A4	A5	A4	A4	A4
5	A5	A5	A4	A5	A5	A5

Table III:
Comparison Results of ZDT 2

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A1	A1	A1	A3	A1	A1
2	A3	A2	A2	A1	A2	A3
3	A2	A3	A3	A2	A3	A2
4	A4	A4	A5	A4	A4	A5
5	A5	A5	A4	A5	A5	A4

Tables IV-VI shows comparison results in ZDT 3, ZDT 4, and ZDT 6, respectively. In each of the problem, there are some unique characteristics in the Pareto-optimal fronts that make the problem difficult to solve. Ranking orders generated by each metric individually contradict appreciably with each other. Metrics ensemble provides a comprehensive evaluation for all algorithms.

Table IV:
Comparison Results of ZDT 3

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A2	A2	A3	A5	A3	A2
2	A3	A5	A2	A3	A2	A3
3	A1	A3	A1	A2	A5	A5
4	A4	A1	A5	A1	A1	A1
5	A5	A4	A4	A4	A4	A4

Table V:
Comparison Results of ZDT 4

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A2	A2	A3	A3	A3	A3
2	A3	A3	A4	A2	A2	A2
3	A5	A4	A2	A1	A5	A4
4	A4	A5	A5	A4	A4	A5
5	A1	A1	A1	A5	A1	A1

Table VI:
Comparison Results of ZDT 6

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A3	A5	A4	A3	A3	A3
2	A2	A3	A3	A2	A5	A5
3	A5	A2	A5	A1	A2	A2
4	A4	A1	A2	A4	A4	A1
5	A1	A4	A1	A5	A1	A4

Table VII:
Comparison Results of 3-objective DTLZ 2

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A5	A2	A1	A3	A5	A5
2	A3	A5	A3	A1	A1	A3
3	A2	A1	A5	A5	A3	A1
4	A1	A3	A2	A2	A4	A2
5	A4	A4	A4	A4	A2	A4

Table VIII:
Comparison Results of 5-objective WFG 1

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A5	A2	A5	A1	A3	A5
2	A2	A3	A3	A3	A5	A3
3	A4	A5	A1	A2	A2	A2
4	A3	A1	A2	A4	A1	A1
5	A1	A4	A4	A5	A4	A4

Table IX:
Comparison Results of 5-objective WFG 2

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A5	A2	A1	A3	A2	A5
2	A2	A5	A3	A2	A5	A2
3	A3	A3	A2	A5	A3	A3
4	A1	A1	A5	A4	A4	A1
5	A4	A4	A4	A1	A1	A4

Table X:
Comparison Results of 10-objective DTLZ 1

Rank	IGD	NR	Spacing	MS	S	Ensemble
1	A5	A1	A2	A3	A3	A5
2	A3	A5	A1	A5	A5	A3
3	A1	A3	A3	A1	A1	A2
4	A4	A2	A5	A4	A2	A1
5	A2	A4	A4	A2	A4	A4

Tables VII-X shows the results in high-dimensional optimization problems: 3-objective DTLZ 2, 5-objective WFG 1, 5-objective WFG 2, and 10-objective DTLZ 1. In each problem, ranking orders generated by each metric individually contradict severely with each other. We cannot draw any conclusion based on each single metric solely. Instead, metrics ensemble offers a rational evaluation for all MOEAs considered.

Therefore, from above discussions, when the test function is simple, each individual metric alone and metrics ensemble share similar outcomes in performance evaluation. However, when the test function becomes more complicated, metrics ensemble provides a much more rational result than that of any metric alone. This is due to the drawback that any individual metric can only quantify one aspect of performance measure. When it comes to a highly complicated and challenging problem, it is better to use metrics ensemble than individual metric. Metrics ensemble can provide a more comprehensive comparison among various MOEAs than what could be obtained from any single performance metric alone.

H. Blind Test

An experiment is designed to blind test the performance through the proposed metrics ensemble with two of the same MOEAs. NSGA-II is chosen here for convenience. They are labeled as MOEA 1 and MOEA 2. The two algorithms go through the performance metrics ensemble process. One winner, either MOEA 1 or MOEA 2, will be chosen. The whole process is repeated 20 times. The approximation fronts from MOEA 1 wins 11 times, while the approximation fronts from MOEA 2 wins 9 times. It is observed that both algorithms perform relatively the same, as rightfully expected so.

I. Choices of Performance Metrics

In order to examine the correlation between the base performance metrics selected in ensemble and the final ranking orders, we construct an experiment and apply it to compare the same five MOEAs in benchmark functions ZDT 1 and ZDT 6. In the original ensemble, five metrics from each of five groups are chosen: NR, IGD, Spacing, MS, and S-metric. In the new ensemble, three new performance metrics from the respective same groups are chosen, instead of those from the original ensemble: RNI replaces NR, GD replaces IGD, and UD replaces Spacing. The remaining two metrics, MS and S-metric, are kept the same. To be exact, the five performance metrics used, one from each of five groups, in the new ensemble are RNI, GD, UD, MS, and S-metric.

Follow the same process, in ZDT 1, the resulting ranking order shows: rank 1-SPEA 2, rank 2- MOEA/D, rank 3-NSGA-II, rank 4-PESA-II, and rank 5-IBEA. This result is exactly the same as the one using the original ensemble metrics. Using the new metrics ensemble, the final ranking order for ZDT 6 is: rank 1-MOEA/D, rank 2-IBEA, rank 3-NSGA-II, rank 4-PESA-II, and rank 5-SPEA 2. There is only slight difference in ranks 4 and 5 between the new and the original ensembles.

From above results, in both benchmark problems, final ranking orders from the new ensemble are very similar to, if not exactly the same as, those by the original ensemble. Properties of performance metrics appear to have no significant effect in the final rank result. It again supports the robust finding in ranking order among competing MOEAs through the proposed ensemble approach.

In the new ensemble, metrics RNI may give contradictory indication of performance to the Pareto dominance in some extreme condition. For example, given two approximation fronts a and b, all solutions in b are dominated by solutions in a but when use RNI metric solely to evaluate these two fronts, fronts b will win. This happens because RNI only considers non-domination ratio within a front, not between competing fronts. The proposed metrics ensemble with double elimination scheme and stochastic mechanism, will not allow a single metric to dominate the final ranking order.

J. Test on a Set of Similarly Structured Problems

Instead of making a bold judgment in terms of the performance ranking for a given test function alone, it is more conclusive if the observation is made based on a collection of similarly structured benchmark functions. In this subsection, we conduct the experiment on test problem F8 [36], which shares similarly problem characteristics with ZDT 4 [34].

Both ZDT 4 and F8 introduce multimodality and have many local Pareto-optimal fronts. By the experiment result in ZDT 4, we have drawn a conclusion that MOEA/D does best when it encounters a test problem with a lot of local Pareto-optimal fronts. In order to substantiate this conclusion, we use the similar problem F8 to test selected MOEAs again.

After applying the proposed performance metrics ensemble, the final ranking order for benchmark function F8 is: rank 1-MOEA/D, rank 2-NSGA-II, rank 3-SPEA 2, rank 4-PESA-II, and rank 5- IBEA, while the ranking order for ZDT 4 is: rank 1-MOEA/D, rank 2-NSGA-II, rank 3-PESA-II, rank 4- IBEA, and rank 5-SPEA 2. The overall winner remains unchanged in these two problems.

Because the same result is observed based on a collective of similarly structured benchmark functions, we can confidently draw a conclusion that MOEA/D is good at handling problems with multimodality and lots of local Pareto fronts.

Therefore, performance metrics ensemble allows a comprehensive measure and more importantly reveals additional insight pertaining to specific problem characteristics that the underlying MOEA could perform the best.

K. Observations and Insights

SPEA 2 is regarded as the ultimate winner in test problems ZDT 1 and ZDT 2 among five MOEAs selected. Although ZDT 1 has a convex Pareto-optimal front while ZDT 2 has a non-convex counterpart to ZDT1, both ZDT1 and ZDT2 share some common characteristics. They do not have local Pareto-optimal fronts and their global Pareto-optimal fronts are continuous. From the above observation, we can safely state that, if the test problem has continues global Pareto-optimal fronts and do not have local Pareto-optimal fronts, SPEA 2 will perform well in this type of MOPs in relevant to other MOEAs chosen for competitions.

NSGA-II has the best performance in ZDT 3, which represents the discreteness feature and has a Pareto-optimal front consisting of several non-contiguous convex parts. Therefore, if there is a test problem with discrete Pareto-optimal front, NSGA-II could be considered as a preferable choice of MOEAs to solve the problem.

MOEA/D wins in both ZDT4 and ZDT6. ZDT4 is difficult to solve because it has many local Pareto-optimal fronts. A large number of local Pareto-optimal fronts make the global Pareto front difficult to find and MOEAs need to exploit their ability to deal with multimodality. MOEA/D also wins in a similarly structured problem in F8. ZDT6's Pareto-optimal solutions are non-uniformly distributed along the global Pareto front. The front is biased for solutions which have a large $f_1(x)$ value. Therefore, MOEA/D will exhibit a good performance when it encounters a test problem which has a lot of local Pareto-optimal fronts or Pareto-optimal solutions are not uniformly distributed.

IBEA wins in 3-objective DTLZ 2, 5-objective WFG 1 and WFG2, and 10-objective DTLZ 1. It appears to embrace advantages over other MOEAs to deal with high-dimension objectives problems.

Once again, our observations confirmed with the findings from "No Free Lunch Theorem" for optimization [1]: any algorithm's elevated performance over one class of problems is exactly paid for in loss over another class. Table XI provides a summary of specific problem characteristics that one MOEA (out of five chosen) would perform the best.

TABLE XI:
Summary of Specific Problem Characteristics that specific MOEAs would Perform the Best

MOEA	Characteristic of Problems it can solve effectively
SPEA 2	Not have local Pareto-optimal fronts and global Pareto-optimal fronts are continuous.
NSGA-II	Pareto-optimal front represents the discreteness feature and consists of several noncontiguous convex parts.
IBEA	Have high-dimension objective
MOEA/D	Have a lot of local Pareto-optimal fronts; Pareto-optimal solutions are not uniformly distributed in its global Pareto front.

V. Conclusion

In literature, we have witnessed a growing number of studies devoted for MOEA. When an MOEA was proposed, a number of benchmark problems are often chosen ad hoc to quantify the performance. Given a set of heuristically chosen performance metrics, the proposed MOEA and some state-of-the-art competitors are evaluated statistically given a large number of independent trials. The conclusion, if any been drawn, is often indecisive and reveals no additional insight pertaining to the specific problem characteristics that the proposed MOEA would perform the best. On the other end, when an MOP application with real-world complications arises, we often have no clue which MOEA should be chosen to attain the best opportunity to be successful. When an MOEA was proposed in literature, no insight in this regard has even been offered.

To address this concern, an ensemble method on performance metrics is proposed in this paper, knowing no single metric alone can faithfully quantify the performance of a given design under real-world scenarios. A collection of performance metrics, measuring the spread across the Pareto-optimal front and the ability to attain the global trade-off surface closeness, could be incorporated into the ensemble approach.

A double elimination tournament selection operator is proposed to compare approximation fronts obtained from different MOEAs in a statistically meaningful way. The double elimination design avoids the risk of a quality MOEA from been easily eliminated due to an unfair assessment from a performance metric chosen. This design allows a comprehensive measure and more importantly reveals additional insight pertaining to specific problem characteristics that the underlying MOEA could perform the best. For a given real-world problem, if we know its problem characteristics (e.g., a Pareto front with a number of disconnected segments and a large number of local optima), we may make an educated judgment to choose the specific MOEA for its superior performance given the problem characteristics. Please note the proposed design does not provide an independent, quantifiable performance measure for a given problem. Instead, it attempts to rank the selected MOEAs comprehensively through a collective performance metrics.

As a next step for this preliminary study, we plan to build a repository to comprehensive quantify a majority of state-of-the-art MOEAs considering a large number of benchmark functions and a large collection of performance metrics. In particular, the focus will be placed toward many-objective optimization problems. Recently, multiple comprehensive comparisons between latest improvements on NSGA-II and MOEA/D for many objectives problems have been made in [37-39], but only single performance metric is used therein. This is in hope to be able to identify specific problem characteristics pertaining to a particular MOEA. The efforts will also be extended into constrained MOPs [40-41] and dynamic MOPs [42]. Additionally, it also makes perfect sense to exploit ways to quantify the degree of betterment when one MOEA is ranked higher than another (i.e., how much better?).

In summary, this study is based on the observations that "indecisive" or "inconclusive" findings are often produced when MOEAs are compared on specific test functions and those insights into how to match MOEAs to problems for which they are most suitable are thus lacking from the literature. The authors understand that appreciable progresses have been made in recent years to better understand some of the properties of performance metrics qualitatively and quantitatively. However, it is also commonly agreed upon that this process of fundamental works will take years to mature. Meanwhile, a compromising and empirical strategy is proposed here to gain additional insights and to move forward in solving difficult problems with real-world complications.

REFERENCES

[1] D.H. Wolpert and W.G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, 1997.

[2] E. Zitzler, L. Thiele, and K. Deb, "Comparison of multiobjective evolutionary algorithms: empirical results," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.

[3] G.G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation," *IEEE Transactions on Evolutionary Computations*, vol. 7, no. 3, pp. 253-274, 2003.

[4] B. Tessema and G.G. Yen, "An adaptive penalty formulation for constrained evolutionary optimization," *IEEE Transactions on Systems, Man and Cybernetics*, *Part A*: *Systems and Humans*, vol. 39, no. 3, pp. 565-578, 2009.

[5] K.C. Tan, T.H. Lee, and E.F. Khor, "Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons," *Artificial Intelligence Review*, vol. 17, no. 4, pp. 253-290, 2002.

[6] D.A. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," *Ph.D. dissertation*, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999.

[7] C.K. Goh and K.C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103-127, 2009.

[8] P. Czyzzak and A. Jaszkiewicz, "Pareto simulated annealing– a metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, pp. 34-47, 1998.

[9] J.R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," *M.S. thesis*, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.

[10] J. Wu and S. Azarm, "Metrics for quality assessment of a multiobjective design optimization solution set," *Journal of Mechanical Design*, vol. 123, no. 1, pp. 18-25, 2001.

[11] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms- a comparative case study," *Proceedings of the International Conference on Parallel Problem Solving from Nature*, Amsterdam, The Netherlands, pp. 292-301, 1998.

[12] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117-132, 2003.

[13] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: methods and applications," *Ph.D. dissertation*, Swiss Federal Institute of Technology, Zurich, Germany, 1999.

[14] M.P. Hansen and A. Jaszkiewicz, "Evaluating the quality of approximations to the nondominated set," *Technical Report IMM-REP-1998-7*, Technical University of Denmark, 1998.

[15] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21-45, 2006.

[16] L. Kuncheva and C. Whitaker, "Measures of diversity in classifier ensembles," *Machine Learning*, vol. 51, no. 2, pp. 181-207, 2003.

[17] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.

[18] R.E. Schapire, Y. Freund, P. Barlett, and W.S. Lee, "Boosting the margin: a new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651-1686, 1998.

[19] J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical Science*, vol. 14, no. 4, pp. 382-401, 1999.

[20] D.H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992.

[21] T.K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832-844, 1998.

[22] E. Zitzler, L. Thiele, and J. Bader, "On set-based multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 58-79, 2010.

[23] D.A. van Veldhuizen and G.B. Lamont, "On measuring multiobjective evolutionary algorithm performance," *Proceedings of IEEE Congress on Evolutionary Computation*, La Jolla, California, pp. 204-211, 2000.

[24] J. Knowles and D. Corne, "On metrics for comparing nondominated sets," *Proceedings of IEEE Congress on Evolutionary Computation*, Honolulu, Hawaii, pp.711-716, 2002.

[25] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point," *Proceedings of ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, Orlando, FL, pp. 87-102, 2009.

[26] E. Zitzler, M. Laumanns, and L. Thiele, "*SPEA2: improving the strength Pareto evolutionary algorithm*," *Technical Report TIK-Report 103*, Swiss Federal Institute of Technology, Zurich, Germany, 2001.

[27] K. Deb, A. Pratab, S. Agrawal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[28] E. Zitzler and K. Simon, "Indicator-based selection in multiobjective search," *Proceedings of International Conference on Parallel Problem Solving form Nature*, Birmingham, UK, 2004.

[29] D.W. Corne, N.R. Jerram, J.D. Knowles, and M.J. Oates, "PESA-II: region-based selection in evolutionary multiobjective optimization," *Proceedings of the Genetic and Evolutionary* Computation Conference, San Francisco, California, pp. 124-130, 2001.

[30] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

[31] W.F. Leong and G.G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Transactions on Systems, Man and Cybernetics*, *Part B*: *Cybernetics*, vol. 38, no. 5, pp. 1270-1293, 2008.

[32] G.G. Yen and W.F. Leong, "Dynamic multiple swarms in multiobjective particle swarm optimization," *IEEE Transactions on Systems, Man and Cybernetics*, *Part A*: *Systems and Human*, vol. 39, no. 4, pp. 890-911, 2009.

[33] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multiobjective optimization test problems," *Proceedings of Congress on Evolutionary Computation*, Honolulu, Hawaii, pp. 825-830, 2002.

[34] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477-506, 2006.

[35] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics*, *Part B*: *Cybernetics*, vol. 38, no. 5, pp. 1402-1412, 2008.

[36] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284-302, 2009.

[37] K. Deb and H. Jain, "An improved NSGA-II procedure for many-objective optimization, part I: solving problems with box constraints," *Technical Report KanGAL-Report 2012009*, Indian Institute of Technology, Kanpur, India, 2012.

[38] K. Deb and H. Jain, "*An improved NSGA-II procedure for many-objective optimization, part II: handling constraints and extending to an adaptive approach*," *Technical Report KanGAL-Report 2012010*, Indian Institute of Technology, Kanpur, India, 2012.

[39] M. Daneshyari and G.G. Yen, "Cultural-based multiobjective particle swarm optimization," *IEEE Transactions on Systems, Man and Cybernetics*, *Part B*: *Cybernetics*, vol. 41, no. 2, pp. 553-567, 2011.

[40] S. Venkatraman and G.G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, col. 9, no. 4, pp. 424-435, 2005.

[41] Y.G. Woldesenbet, G.G. Yen, and B.G. Tessema, "Constraint handling in multi-objective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514-525, 2009.

[42] Y.G. Woldesenbet and G.G. Yen, "Dynamic evolutionary algorithm with variable relocation," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 500-513, 2009.

Gary G. Yen (S'87–M'88-SM'97–F'09) received the Ph.D. degree in electrical and computer engineering from the University of Notre Dame in 1992.

He is currently a Professor in the School of Electrical and Computer Engineering, Oklahoma State University (OSU). Before joined OSU in 1997, he was with the U.S. Air Force Research Laboratory in Albuquerque. His research is supported by the DoD, DoE, EPA, NASA, NSF, and Process Industry. His research interest includes intelligent control, computational intelligence, conditional health monitoring, signal processing, and their industrial/defense applications.

Dr. Yen was an associate editor of the *IEEE Control Systems Magazine*, *IEEE Transactions on Control Systems Technology*, *Automatica*, *Mechantronics*, *IEEE Transactions on Systems, Man and Cybernetics, Part A and Part B* and *IEEE Transactions on Neural Networks*. He is currently serving as an associate editor for the *IEEE Transactions on Evolutionary Computation* and *International Journal of Swarm Intelligence Research*. He served as the General Chair for the *2003 IEEE International Symposium on Intelligent Control* held in Houston, TX and *2006 IEEE World Congress on Computational Intelligence* held in Vancouver, Canada. Dr. Yen served as Vice President for the Technical Activities in 2005-2006 and President in 2010-2011 of the *IEEE Computational intelligence Society* and is the founding editor-in-chief of the *IEEE Computational Intelligence Magazine* 2006-2009. In 2011, he received the Andrew P Sage Best Transactions Paper award from *IEEE Systems, Man and Cybernetics Society*. In 2013, he received Meritorious Service award from *IEEE Computational Intelligence Society*. He is a fellow of IEEE and IET.

Zhenan He received the B.E. degree in automation from the University of Science and Technology Beijing, Beijing, China, in 2008 and M.S. degree in electrical and computer engineering from the Oklahoma State University, Stillwater, in 2011.

He is currently with the School of Electrical and Computer Engineering, Oklahoma State University. His current research interests include multiobjective optimization using evolutionary algorithms, neural networks, and machine learning.