Research Article

Evaluating Multiobjective Evolutionary Algorithms Using MCDM Methods

Xiaobing Yu ^{1,2,3,4}, YiQun Lu,⁴ and Xianrui Yu⁴

¹Collaborative Innovation Center on Forecast and Evaluation of Meteorological Disasters,
Nanjing University of Information Science & Technology, Nanjing 210044, China

²Research Center for Prospering Jiangsu Province with Talents, Nanjing University of Information Science & Technology,
Nanjing 210044, China

³China Institute for Manufacture Developing, Nanjing University of Information Science & Technology, Nanjing 210044, China

⁴School of Management Science and Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China

Correspondence should be addressed to Xiaobing Yu; yuxb111@163.com

Received 15 November 2017; Accepted 10 February 2018; Published 19 March 2018

Academic Editor: David Bigaud

Copyright © 2018 Xiaobing Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The evaluation of multiobjective evolutionary algorithms (MOEAs) involves many metrics, it can be considered as a multiple-criteria decision making (MCDM) problem. A framework is proposed to estimate MOEAs, in which six MOEAs, five performance metrics, and two MCDM methods are used. An experimental study is designed and thirteen benchmark functions are selected to validate the proposed framework. The experimental results have indicated that the framework is effective in evaluating MOEAs.

1. Introduction

Without a loss of generality, the mathematical formula of multiobjective problems (MOPs) can be expressed as follows:

$$\min F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \quad x \in \Omega, \quad (1)$$

where $\vec{x} = (x_1, x_2, \dots, x_n)$ is the decision vector in the decision space Ω . $F(x)$ is the objective function. Generally speaking, the objectives contradict each other. We cannot find a single solution to optimize all the objectives. Optimizing one objective often leads to deterioration in at least one objective.

Over the past two decades, many multiobjective evolutionary algorithms (MOEAs) have been proposed, such as vector evaluated genetic algorithm (VEGA) [1], Pareto archived evolution strategy (PAES) [2], strength Pareto evolutionary algorithm (SPEA) [3], SPEA2 [4], Pareto envelope-based selection algorithm (PESA) [5, 6], nondominated sorting algorithm (NSGA) [7], NSGAI [8], multi-objective evolutionary algorithm based on decomposition (MOEAD) [9, 10], indicator-based evolutionary algorithm (IBEA) [11], epsilon-multiobjective evolutionary algorithm

(epsilon-MOEA) [12], multiobjective particle swarm optimizer (MOPSO) [13], speed-constrained multiobjective particle swarm optimizer (SMPSO) [14], generalized differential evolution (GDE3) [15], ABYSS [16], multiobjective symbiotic organism search (MOSOS) [17], multiobjective differential evolution algorithm (MODEA) [18], grid-based adaptive MODE (GAMODE) [19]. These algorithms make great contributions to the development of evolutionary algorithms and optimization approaches. These methods try to make population move towards the optimal Pareto front region.

In the single optimization, the algorithm performance can be evaluated by the difference between $f(x)$ and function optimal value. However, the method cannot be adopted in MOPs. In order to solve the problem, many criteria are proposed to evaluate the performance of MOEAs. In fact, the experiment results of almost every algorithm indicate that the proposed algorithm is competitive compared with the state-of-the-art algorithms. Nondominated objective space and box plot are adopted in SPEA2 [4]. NSGAI employs convergence and diversity metrics to compare with SPEA and PAES [8]. The set convergence and inverted generation distance (IGD) are used to evaluate the performance of MOEAD [9].

FIGURE 1: Evaluation framework.

Epsilon indicator is used in IBEA [11]. Convergence measurement, spread, hypervolume and computational time are selected as performance metrics in epsilon-MOEA [12]. To validate the proposed MOPSO, four quantitative performance indexes (success counting IGD, set coverage, two-set difference hypervolume) and qualitative performance index (plotting the Pareto fronts) are adopted [13]. Three quality indicators, additive unary epsilon indicator, spread, and hypervolume, are considered in SMPSO [14]. Spacing, binary metrics C and V are used in GD3 [15]. Three metrics, generation distance (GD), spread, and hypervolume, are used to estimate ABYSS [16]. GD, diversity, computational time, and box plot are considered as measurement in MOSOS [17]. GD and diversity metrics are adopted in MOEDA [18]. There are three metrics, GD, IGD, and hypervolume, in GAMODE [19].

Among these metrics, some focus on the convergence of MOEAs, while some pay attention to the diversity of MOEAs. Convergence is to measure the ability to attain global Pareto front and diversity is to measure the distribution along the Pareto front. It is observed that every proposed algorithm often introduces few metrics to estimate the performance based on the results from benchmarks. The conclusions of these MOEAs are that they are the best and competitive. However, it is unfaithfully to measure MOEAs performance by one or two metrics. Every metric can just demonstrate some specific qualification of performance while neglecting other information. For instance, the metric GD can provide information about the convergence of MOEAs, but it cannot evaluate the diversity of MOEAs. Therefore, these evaluations are not comprehensive. It cannot entirely estimate the whole performance of MOEAs. As evaluation of MOEAs involves many metrics, it can be regarded as a multiple-criteria decision making (MCDE) problem. MCDE techniques can be used to cope with the problem. In order

to overcome the problem and make fair comparisons, a framework using MCDE methods is proposed. In the framework, comprehensive performance metrics are established, in which both convergence and diversity are considered. Two MCDE methods are employed to evaluate six MOEAs. The efforts can give more fair and faithful comparisons than single metric.

The rest of this paper is organized as follows: Section 2 proposes the framework, in which six algorithms, five performance metrics, and two MCDE methods are briefly introduced. Experiments are presented in Section 3 and conclusions are illustrated in Section 4.

2. Evaluation Framework

A framework is proposed to evaluate multiobjective algorithms in Figure 1. Six MOEAs, five performance metrics, and two MCDE methods are employed in the framework.

2.1. Six MOEAs

(1) *NSGAI* [8]. NSGAI was proposed to solve the high computational complexity, lack of elitism, and specifying of the sharing parameter of NSGA. In NSGAI, a selection operator is designed by creating a mating pool to combine the parent population and offspring population. Nondominated sort and crowding distance ranking are also implemented in the algorithm.

(2) *PAES* [2]. The Pareto archived evolution strategy (PAES) is a simple evolutionary algorithm. The algorithm is considered as a $(1 + 1)$ evolution strategy, employing local search from a population of one but using a reference archive of previously found solutions in order to identify the approximate dominance ranking of the current and candidate solutions vectors.

(3) *SPEA2* [4]. The strength Pareto evolutionary algorithm (SPEA) was proposed in 1999 by Zitzler. Based on the SPEA, an improved version, namely, SPEA2, was proposed, which incorporated a fine-grained fitness assignment, a density estimation technique, and an enhanced archive truncation method.

(4) *MOEAD* [9]. Multiobjective evolutionary algorithm based on decomposition (MOEAD) was proposed by Li and Zhang. It decomposes a multiobjective optimization problem into a number of scalar optimization subproblems and optimizes them simultaneously. Each subproblem is optimized by only using information from neighboring subproblems, which makes the algorithm effective and efficient. It won the outstanding paper award of IEEE Transactions on evolutionary computation.

(5) *MOPSO* [13]. Multiobjective particle swarm optimizer (MOPSO) is based on Pareto dominance and the use of a crowding factor to filter out the list of available leaders. Different mutation operators act on different subdivisions of the swarm. The epsilon-dominance concept is also incorporated in the algorithm.

(6) *SMPSO* [14]. Speed-constrained multiobjective PSO (SMPSO) was proposed in 2009. It allows generating new particle position in which velocity is extremely high. The turbulence factor and an external archive are designed to store the nondominated solutions found during search.

2.2. Performance Metrics. Nowadays, there are many metrics to measure performance of MOEAs. Among them, the following five metrics are widely employed. They can reveal the convergence and diversity of MOEAs very well. However, many researches just employ a few of them to evaluate algorithms and argue that their proposed algorithms are the best. In fact, it is unfair to give the conclusion without comprehensive metrics and evaluations. Therefore, the five metrics are selected to make the comprehensive comparisons.

(1) *GD*

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}, \quad (2)$$

where $d_i = \min_j \|f(x_i) - PF_{\text{true}}(x_j)\|$ is the distance between nondominated solution $f(x_i)$ and the nearest Pareto front solution in objective space. It is to measure the closeness of the solutions to the real Pareto front. If GD is equal to zero, this reveals that all the nondominated solutions generated are located in the real Pareto front. Therefore, the lower value of GD indicates that the algorithm has better performance [20].

(2) *IGD*. PF_{true} is a set of uniformly distributed points in the objective space. P_A is the nondominated solution set obtained by an algorithm and the distance from PF_{true} to P_A is defined as

$$IGD(P_A, PF_{\text{true}}) = \frac{\sum_{v \in PF_{\text{true}}} d(v, P_A)}{|PF_{\text{true}}|}, \quad (3)$$

where $d(v, P_A)$ is the minimum Euclidean distance between v and the points in P_A . Algorithms with smaller IGD values are desirable [21, 22].

(3) *Hypervolume*. This hypervolume metric calculates the volume (in the objective space) covered by members of nondominated solutions sets obtained by MOEAs where all objectives are to be minimized [16]. A hypervolume can be calculated as follows:

$$HV = \text{volume} \left(\bigcup_{i=1}^{|P_A|} v_i \right). \quad (4)$$

The larger the HV value is, the better the algorithm is.

(4) *Spacing*. The metric spacing is to measure how uniformly the nondominated set is distributed. It can be formulated as follows:

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \bar{d})^2}, \quad (5)$$

where d_i is the same as the d_i in GD metric, $\bar{d}$ is the average value of d_i , and n is the number of individuals in nondominated set. The smaller the spacing is, the better the algorithm performs [23, 24].

(5) *Maximum Pareto Front Error*. It is to measure the worst case and can be formulated as follows:

$$MPFE = \max d_i, \quad (6)$$

where d_i is the same as d_i employed in GD. MPFE is the largest distance among these d_i . The lower the value of MPFE is, the better the algorithm is [25].

In order to elaborate the five metrics, Figure 2(a) reveals the distance used in GD, space, and MPFE. Figure 2(b) presents the distance in IGD metric, and Figure 2(c) depicts the HV metric.

2.3. TOPSIS. TOPSIS is one of MCDM methods to evaluate alternatives. In TOPSIS, the best alternative should have two characteristics: one is the farthest from the negative-ideal solution and the other one is the nearest to positive-ideal solution. The negative-ideal solution is a solution that maximizes the cost criteria and minimizes the benefit criteria, which has all the worst values attainable from the criteria. The positive-ideal solution minimizes the cost criteria and maximizes the benefit criteria. It is consisted of all the best values attainable from the criteria [26, 27]. TOPSIS consists of the following steps.

Step 1 (obtain decision matrix). If the number of alternatives is J and the number of criteria is n , decision matrix with n rows and J columns will be obtained as in Table 1.

In Table 1, f_{ij} ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, J$) is a value indicating the performance rating of each algorithm j th with respect to each criterion i th.

TABLE 1: The multiple attribute decision matrixes.

	Algorithm 1	Algorithm 2	...	Algorithm J
Criterion 1	f_{11}	f_{12}	...	f_{1J}
Criterion 2	f_{21}	f_{22}	...	f_{2J}
...	...	...	...	...
Criterion i	f_{i1}	f_{i2}	...	f_{iJ}
...	...	...	...	...
Criterion n	f_{n1}	f_{n2}	...	f_{nJ}

FIGURE 2: The distance and nondominated solutions used in above metrics.

Step 2 (normalize decision matrix). According to (7), the normalized value f_{ij} is calculated as follows:

$$r_{ij} = \frac{f_{ij}}{\sqrt{\sum_{j=1}^J f_{ij}^2}}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, J. \quad (7)$$

Step 3 (calculate the weighted normalized decision matrix). The matrix is calculated by multiplying the normalized decision matrix and its weights are presented as

$$v_{ij} = w_i \times r_{ij}, \quad (8)$$

where w_i is the weight of the i th criterion, $\sum_{i=1}^n w_i = 1$.

Step 4 (find the negative-ideal and positive-ideal solutions).

$$\begin{aligned} A^- &= \{v_1^-, v_2^-, \dots, v_n^-\} \\ &= \left\{ \left(\min_j v_{ij} \mid i \in I' \right), \left(\max_j v_{ij} \mid i \in I'' \right) \right\} \\ A^* &= \{v_1^*, v_2^*, \dots, v_n^*\} \\ &= \left\{ \left(\max_j v_{ij} \mid i \in I' \right), \left(\min_j v_{ij} \mid i \in I'' \right) \right\}, \end{aligned} \quad (9)$$

where I' is associated with cost criteria and I'' is associated with benefit criteria.

Step 5 (calculate the n -dimensional Euclidean distance). The separation of each algorithm from the ideal solution is presented as follows:

$$D_j^+ = \sum_{i=1}^n d(v_{ij}, v_j^*). \quad (10)$$

The separation of each algorithm from the negative-ideal solution is defined as follows:

$$D_j^- = \sum_{i=1}^n d(v_{ij}, v_j^-). \quad (11)$$

Step 6 (calculate the relative closeness to the ideal solution). The relative closeness of the algorithm j th is defined as

$$CC_j^* = \frac{D_j^-}{D_j^+ + D_j^-} \quad i = 1, 2, \dots, J. \quad (12)$$

Step 7 (rank algorithms order). The CC is between 0 and 1. The larger the CC is, the better the algorithm j is.

2.4. VIKOR Method. The VIKOR was proposed by Opricovic and Tzeng [28–31]. The method is developed to rank and select from a set of alternatives. The multicriteria ranking index is introduced based on the idea of closeness to the ideal solutions. The VIKOR requires the following steps.

Step 1. Determine the worst f_i^- values of criterion and the best f_i^* value of criterion as follows ($i = 1, 2, \dots, n$):

$$f_i^- = \begin{cases} \min_j f_{ij}, & \text{for benefit criteria} \\ \max_j f_{ij}, & \text{for cost criteria,} \end{cases} \quad j = 1, 2, \dots, J \quad (13)$$

$$f_i^* = \begin{cases} \max_j f_{ij}, & \text{for benefit criteria} \\ \min_j f_{ij}, & \text{for cost criteria,} \end{cases} \quad j = 1, 2, \dots, J,$$

where f_{ij} is the value of i th criterion for alternative a_j , n is the number of criteria, and J is the number of alternatives.

Step 2. S_j and R_j ($j = 1, 2, \dots, J$) can be formulated as follows:

$$S_j = \frac{\sum_{i=1}^n w_i (f_i^* - f_{ij})}{(f_i^* - f_i^-)} \quad (14)$$

$$R_j = \max_i \left[\frac{w_i (f_i^* - f_{ij})}{(f_i^* - f_i^-)} \right],$$

where w_i is the weight of i th criteria. S_j and R_j are employed to measure ranking.

Step 3. Compute the values Q_j ($j = 1, 2, \dots, J$) as follows:

$$Q_j = \frac{v(S_j - S^*)}{(S^- - S^*)} + \frac{(1-v)(R_j - R^*)}{(R^- - R^*)}, \quad (15)$$

$$S^* = \min_j S_j, \quad S^- = \max_j S_j, \quad R^* = \min_j R_j, \quad R^- = \max_j R_j,$$

where the alternative obtained by S^* is with a maximum utility, the alternative acquired by R^* is with a minimum individual regret of the opponent, and v is the weight of the strategy of the majority of criteria and is often set to 0.5.

Step 4. Rank the alternatives in decreasing order. Rank the three measurements, respectively: Q , S , and R .

Step 5. The alternative a is considered as the best if the following two conditions are met:

C1: $Q(a') - Q(a) \geq 1/(J-1)$, where a' is the alternative with second position in the ranking list by Q and J is the number of alternatives;

C2: alternative a should be the best ranked by S or R .

3. Experiments

The experiments are designed to evaluate the above six algorithms. In order to make fair comparisons, thirteen test benchmark functions are widely used in MOPs and employed in the experiments. They can be divided into two groups: ZDT suites and WFG suites. All of these test suites are minimization of the objective. The detailed information is given in Table 2 [32, 33].

The mathematical forms of WFG can be obtained in [32] and ZDT suites are presented as follows:

$$\text{ZDT1: } \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x) \left[1 - \sqrt{\frac{f_1(x)}{g(x)}} \right], \quad g(x) = 1 + \frac{9(\sum_{i=2}^n x_i)}{(n-1)}, \quad x \in [0, 1] \end{cases} \quad (16)$$

$$\text{ZDT2: } \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x) \left[1 - \left(\frac{f_1(x)}{g(x)} \right)^2 \right], \quad g(x) = 1 + \frac{9(\sum_{i=2}^n x_i)}{(n-1)}, \quad x \in [0, 1] \end{cases} \quad (17)$$

$$\text{ZDT3: } \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x) \left[1 - \sqrt{\frac{f_1(x)}{g(x)}} - \left(\frac{f_1(x)}{g(x)} \right) \sin(10\pi x_1) \right], \quad g(x) = 1 + \frac{9(\sum_{i=2}^n x_i)}{(n-1)}, \quad x \in [0, 1] \end{cases} \quad (18)$$

$$\text{ZDT6: } \begin{cases} f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ f_2(x) = g(x) \left[1 - \left(\frac{f_1(x)}{g(x)} \right)^2 \right], \quad g(x) = 1 + 9 \left[\frac{(\sum_{i=2}^n x_i)}{(n-2)} \right]^{0.25}, \quad x \in [0, 1]. \end{cases} \quad (19)$$

The parameters settings of these algorithms are the same as the original paper. The maximum function evaluations

are set to 25,000. Each algorithm runs thirty times and the average values of performance metrics are obtained.

TABLE 2: Benchmark test functions information.

Problem	Separable	Modality	Bias	Geometry
ZDT1	Yes	No	No	Convex
ZDT2	Yes	No	No	Concave
ZDT3	Yes	Yes	No	Disconnected
ZDT6	Yes	Yes	Yes	Concave
WFG1	Yes	No	Polynomial flat	Convex, mixed
WFG2	No	No	No	Convex, disconnected
WFG3	No	No	No	Linear, degenerate
WFG4	Yes	Yes	No	Concave
WFG5	Yes	Yes	No	Concave
WFG6	No	No	No	Concave
WFG7	Yes	No	Parameter dependent	Concave
WFG8	No	No	Parameter dependent	Concave
WFG9	No	Yes	Parameter dependent	Concave

TABLE 3: Five metric results of ZDT1 results.

	NSGAII	PAES	SPEA2	MOEAD	MOPSO	SMPSO
GD	$2.27E-4$	$4.85E-3$	$1.73E-4$	$4.32E-4$	$7.95E-5$	$1.17E-4$
IGD	$4.89E-3$	$6.60E-2$	$3.92E-3$	$6.41E-3$	$5.22E-3$	$3.70E-3$
MPFE	$1.21E-2$	$3.66E-1$	$1.36E-2$	$9.32E-3$	$1.81E-3$	$8.78E-3$
Spacing	$6.72E-3$	$3.11E-2$	$3.17E-3$	$1.01E-2$	$6.15E-3$	$1.40E-3$
Hypervolume	$6.59E-1$	$6.07E-1$	$6.62E-1$	$6.56E-1$	$6.60E-1$	$6.62E-1$

TABLE 4: Normalized decision matrix of five performance metrics.

	NSGAII	PAES	SPEA2	MOEAD	MOPSO	SMPSO
GD	0.0465	0.9939	0.0355	0.0885	0.0163	0.0240
IGD	0.0731	0.9864	0.0586	0.0958	0.0780	0.0553
MPFE	0.0330	0.9981	0.0371	0.0254	0.0049	0.0239
Spacing	0.1969	0.9115	0.0929	0.2960	0.1802	0.0410
Hypervolume	0.4131	0.3805	0.4150	0.4112	0.4137	0.4150

3.1. Results. In order to elaborate the whole calculation process, the ZDT1 results of five metrics are presented in Table 3. The four metrics GD, IGD, MPFE, and spacing of SMPSO are the smallest, and hypervolume is the biggest. PAES is the worst because the performances of five metrics are the worst among the six algorithms. Normalized decision matrix of five performances metrics is presented in Table 4. Suppose that the weight is equal to $1/5$. Thus, according to Table 4, positive-ideal and negative-ideal solutions can be defined as follows:

$$\begin{aligned}
 Z^+ &= \{0.0163, 0.0553, 0.0239, 0.0410, 0.4150\} \times \frac{1}{5}; \\
 Z^- &= \{0.9939, 0.9864, 0.9981, 0.9115, 0.3805\} \times \frac{1}{5};
 \end{aligned} \quad (20)$$

then, the distances D^+ and D^- are calculated according to (10) and (11), demonstrated in Table 5. The global performance of each algorithm is determined by CC^* , calculated in (12) and

presented in Table 5. Therefore, the ranking of six algorithms is as follows: SMPSO > SPEA2 > MOPSO > NSGAII > MOEAD > PAES. SMPSO is the best algorithm and PAES is the worst one for ZDT1.

For VIKOR method, the Q , S , and R are calculated and presented in Table 6. According to the feature of Q , S , and R , SMPSO is the best one while PAES is the worst one. SPEA2 is better than MOEAD. However, as the condition $Q(a') - Q(a) > 1/(6-1) = 0.2$ cannot be satisfied, S value is used to determine the ranking for NSGAII, SPEA2, MOEAD, MOPSO. Therefore, the ranking among six algorithms is SMPSO > SPEA2 > MOPSO > NSGAII > MOEAD > PAES.

Tables 7 and 8 give the complete rankings of TOPSIS and VIKOR methods for all benchmark functions. For the thirteen test functions, NSGAII wins in one problem, WFG1, and performs better in ZDT3, WFG2, WFG4, and WFG8. SPEA2 wins in five problems: ZDT3, WFG2, WFG4, WFG7, and WFG8. SPEA2 provides better performance in ZDT1, ZDT2, WFG1, WFG3, and WFG9. However, it achieves worse

TABLE 5: The results of DD^+ , DD^- , and CC from TOPSIS.

	NSGAII	PAES	SPEA2	MOEAD	MOPSO	SMPSO
DD^+	0.1623	1.8889	0.0641	0.2689	0.1411	0.0205
DD^-	1.7818	0	1.8369	1.7143	1.8175	1.8750
CC	0.9165	0	0.9663	0.8644	0.9280	0.9892

TABLE 6: The results of Q , S , and R from VIKOR.

	NSGAII	PAES	SPEA2	MOEAD	MOPSO	SMPSO
Q	0.2204	2.0000	0.0590	0.3823	0.1825	0
S	0.0624	1.0000	0.0230	0.1080	0.0441	0.0054
R	0.0358	0.2000	0.0119	0.0586	0.0320	0.0038

TABLE 7: The TOPSIS rankings.

Functions	NSGAII	PAES	SPEA2	MOEAD	MOPSO	SMPSO
ZDT1	4	6	2	5	3	1
ZDT2	4	6	2	3	5	1
ZDT3	2	6	1	3	4	5
ZDT6	4	6	5	1	2	3
WFG1	2	4	1	3	6	5
WFG2	2	6	1	3	4	5
WFG3	3	6	2	5	1	4
WFG4	2	4	1	3	6	5
WFG5	5	6	4	2	3	1
WFG6	2	6	4	5	3	1
WFG7	5	6	1	2	4	3
WFG8	2	6	1	4	5	3
WFG9	3	6	2	5	4	1

TABLE 8: The VIKOR rankings.

Functions	NSGAII	PAES	SPEA2	MOEAD	MOPSO	SMPSO
ZDT1	4	6	2	5	3	1
ZDT2	4	6	2	3	5	1
ZDT3	2	6	1	3	4	5
ZDT6	4	6	5	1	2	3
WFG1	1	4	2	3	6	5
WFG2	2	6	1	3	4	5
WFG3	3	6	2	5	1	4
WFG4	2	4	1	3	6	5
WFG5	5	6	4	2	3	1
WFG6	3	6	5	4	2	1
WFG7	4	6	1	3	5	2
WFG8	2	6	1	4	5	3
WFG9	3	6	2	5	4	1

result in WFG6. MOEAD obtains the best result in ZDT6 and better performance in WFG5. MOPSO wins in one problem, WFG3, and gets better results in ZDT6 and WFG6. SMPSO provides the best performance in five problems: ZDT1, ZDT2, WFG5, WFG6, and WFG9.

TOPSIS and VKIOR methods achieve same rankings for ZDT1, ZDT2, ZDT3, ZDT6, WFG2, WFG3, WFG4, WFG5, WFG8, and WFG9.

However, TOPSIS and VKIOR methods have different rankings for WFG1, WFG6, and WFG7. Take the WFG1 as

an instance. The final values of TOPSIS and VKIOR are presented in Table 9. As there are six algorithms, J is set to six, $1/(J - 1) = 1/(6 - 1) = 0.2$. It indicates that the Q value difference between two algorithms should be more than 0.2. Otherwise, the rank between the two algorithms is determined by S or R .

From Table 9, it can be noticed that this condition cannot be met between NSGAII and SPEA2, so the values S are used to compare the two algorithms. The value of NSGAII is smaller than that of SPEA2, so NSGAII is better than

TABLE 9: The results of CC, Q, S, and R value from TOPSIS and VIKOR.

Method	NSGAI	PAES	SPEA2	MOEAD	MOPSO	SMPSO
CC	0.9406	0.5677	0.9686	0.6693	0.0334	0.2336
Q	0.0290	1.3579	0.0062	0.8686	2.0000	1.8874
S	0.0178	0.4095	0.0227	0.2680	0.8142	0.7246
R	0.0178	0.1461	0.0134	0.0984	0.1667	0.1667

FIGURE 3: The Pareto front and nondominated solutions of ZDT1, ZDT2, and ZDT3.

SPEA2, NSGAI is the first, and SPEA2 is the second. However, TOPSIS directly uses CC as the ranking criteria. The CC value of SPEA2 is bigger than NSGAI, so it can be ranked the first and NSGAI is the second in TOPSIS method.

3.2. Discussion. In order to further make comparisons, the best and worst performances of the above six algorithms are selected. The nondominated solutions obtained by the two kinds of algorithms are depicted in Figures 3–5. WFG1, NSGAI, and SPEA2 achieve the best ranking according to

FIGURE 4: Continued.

FIGURE 4: The Pareto front and nondominated solutions of ZDT6, WFG1-WFG4.

VIKOR and TOPSIS methods, respectively, so the nondominated solutions of two algorithms are both shown in Figure 4.

It can be clearly observed that the better algorithm can achieve nondominated solutions uniformly distributed along the Pareto front, and the worse algorithm obtains nondominated solutions, in which both convergence and diversity are poor.

ZDT1, ZDT2, WFG5, WFG6, and WFG9 have common features. They do not have local Pareto front and their Pareto fronts are continuous. SMPSO allows new particle position in which velocity is very high. The turbulence factor and an external archive are designed to store the nondominated solutions found during search. These mechanisms can help population move quickly towards the Pareto front for this type of problems, in which local Pareto front does not exist and Pareto front is continuous.

ZDT 3 and WFG 4 have discrete Pareto front. SPEA2 has achieved performance on the two problems. Thus, if the problem has the discrete Pareto front, SPEA2 is a good choice.

ZDT6 is a biased function as the first objective function value is larger compared to the second one. MOEAD obtains superior results, so, if the problem has the feature, MOEAD should be chosen.

From Tables 7 and 8, the no-free-lunch theorem can also be observed: any optimization algorithm improves performance over one class of problems exactly paid for in loss over

another class. No algorithm can achieve the best or worst performance for all test functions.

4. Conclusions

There are many MOEAs. When a multiobjective optimization algorithm is proposed, the experiment results often indicate that the algorithm is competitive based on one or two performance metrics. Generally, these comparisons are unfair and the results are unfaithful. In order to make fair comparisons and rank these MOEAs, a framework is proposed to evaluate MOEAs. The framework employs six well-known MOEAs, five performance metrics, and two MCDM methods. The six MOEAs are NSGAII, PAES, SPEA2, MOEAD, MOPSO, and SMPSO. The five performance metrics GD, IGD, MPFE, spacing, and hypervolume are selected, in which both convergence and diversity of nondominated solutions are fully considered. Two methods are TOPSIS and VIKOR.

The results have indicated that SPEA2 is the best algorithm and PAES is the worst one. However, SPEA2 cannot perform well on all test functions and PAES also does not achieve the worst performance on all test functions. The experiment results are consistent with the no-free-lunch theorem. What is more, the observation of experiment results shows that the ability of MOEAs to solve MOPs depends on both MOEAs and the features of MOPs.

FIGURE 5: The Pareto front and nondominated solutions of WFG5-WFG9.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank the Natural Science Foundation of China (Grants nos. 71503134, 91546117, 71373131), Key Project of National Social and Scientific Fund Program (16ZDA047), Philosophy and Social Sciences in Universities of Jiangsu (Grant no. 2016SJB630016).

References

- [1] D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, 1985.
- [2] J. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, vol. 1, pp. 98–105, July 1999.
- [3] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: methods and applications [Ph.D. Doctoral dissertation]*, Swiss Federal Institute of Technology, Zurich, Switzerland, 1999, ETH 13398.
- [4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," TIK-Report 103, 2001.
- [5] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '01)*, pp. 283–290, 2001.
- [6] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto-envelope based selection algorithm for multiobjective optimization," *Parallel Problem Solving from Nature PPSN VI*, pp. 869–878, 2000.
- [7] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1995.
- [8] K. Deb, A. Pratap, S. Agarwal, 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.
- [9] 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.
- [10] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 203–208, Trondheim, Norway, May 2009.
- [11] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature (PPSN VIII)*, pp. 832–842, 2004.
- [12] K. Deb, M. Mohan, and S. Mishra, "A Fast Multi-Objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions," KanGAL Report 2003002, 2003.
- [13] M. R. Sierra and C. A. C. Coello, "Improving PSO-based Multi-Objective Optimization using Crowding, Mutation and epsilon-Dominance," in *Evolutionary Multi-Criterion Optimization*, vol. 3410, pp. 505–519, 2005.
- [14] A. J. Nebro, J. J. Durillo, G. Nieto, C. A. C. Coello, F. Luna, and E. Alba, "SMPSO: a new pso-based metaheuristic for multi-objective optimization," in *Proceedings of the IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM '09)*, pp. 66–73, Nashville, Tenn, USA, April 2009.
- [15] S. Kukkonen and J. Lampinen, "GDE3: The Third Evolution Step of Generalized Differential Evolution," KanGAL Report 2005013, 2005.
- [16] A. J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J. J. Durillo, and A. Beham, "AbYSS: adapting scatter search to multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 4, pp. 439–457, 2008.
- [17] A. Panda and S. Pani, "A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems," *Applied Soft Computing*, vol. 46, pp. 344–360, 2016.
- [18] M. Ali, P. Siarry, and M. Pant, "An efficient Differential Evolution based algorithm for solving multi-objective optimization problems," *European Journal of Operational Research*, vol. 217, no. 2, pp. 404–416, 2012.
- [19] J. Cheng, G. G. Yen, and G. Zhang, "A grid-based adaptive multi-objective differential evolution algorithm," *Information Sciences*, vol. 367–368, pp. 890–908, 2016.
- [20] D. A. Van Veldhuizen, *Multiobjective Evolutionary Algorithms: Classification, Analysis, and New Innovations [Ph.D. dissertation]*, Wright-Patterson Air Force Base, Ohio, Ohio, USA, 1999.
- [21] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, University of Essex, UK, 2008.
- [22] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A Many-Objective Evolutionary Algorithm Using A One-by-One Selection Strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2689–2702, 2017.
- [23] X. Yu, "Disaster prediction model based on support vector machine for regression and improved differential evolution," *Natural Hazards*, vol. 85, no. 2, pp. 959–976, 2017.
- [24] J. R. Schoot, *Fault tolerant design using single and multicriteria genetic algorithm optimization*, Department of Aeronautics and Astronautics [M.S. thesis], Massachusetts Institute of Technology, Cambridge, UK, 1995.
- [25] G. G. Yen and Z. He, "Performance metric ensemble for multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 131–144, 2014.
- [26] Z. Pei, "A note on the TOPSIS method in MADM problems with linguistic evaluations," *Applied Soft Computing*, vol. 36, article no. 3048, pp. 24–35, 2015.
- [27] X. Yu, S. Guo, J. Guo, and X. Huang, "Rank B2C e-commerce websites in e-alliance based on AHP and fuzzy TOPSIS," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3550–3557, 2011.
- [28] S. Opricovic, *Multi criteria optimization of Civil Engineering Systems*, Faculty of Civil Engineering, Belgrade, Serbia, 1998.
- [29] S. Opricovic and G. Tzeng, "Multicriteria planning of post-earthquake sustainable reconstruction," *Computer-Aided Civil and Infrastructure Engineering*, vol. 17, no. 3, pp. 211–220, 2002.
- [30] S. Opricovic and G. H. Tzeng, "Compromise solution by MCDM methods: a comparative analysis of VIKOR and TOPSIS," *European Journal of Operational Research*, vol. 156, no. 2, pp. 445–455, 2004.

- [31] S. Opricovic and G. Tzeng, "Extended VIKOR method in comparison with outranking methods," *European Journal of Operational Research*, vol. 178, no. 2, pp. 514–529, 2007.
- [32] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [33] 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.