Many-Objective Decomposition-based Evolutionary Algorithm with Adaptive Weights

Qing Xu[1], Xiaodan Liang[2,3], Suyu Yang[4], Hongxia Zheng[4], Hanning Chen[2,5,*], Kun Jia[6], Yaxing Yuan[3,7]

[1]*School of Artificial Intelligence, Tiangong University, Tianjin, 300387, China*
[2]*School of Computer Science and Technology, Tiangong University, Tianjin, 300387, China*
[3]*Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education, Beijing, 100039, China*
[4]*School of Software, Tiangong University, Tianjin, 300387, China*
[5]*School of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin, 300457, China*
[6]*Central Research Institute Qingdao Branch, Qingdao, 266000, China*
[7]*The Open University of China, Beijing, 100039 China*
**Corresponding Author.*

Abstract:
Many-objective evolutionary algorithms are difficult to maintain good individual convergence and population diversity. As the number of objectives increases, there are more and more non-dominated solutions, and some existing diversity metrics are no longer useful. In this paper, a many-objective decomposition-based evolutionary algorithm with adaptive weights (MaOEA-AWM) is proposed. The convergence and diversity of individuals in high-dimensional target space are balanced in the proposed MaOEA-AWM through the use of a scaling method known as angle penalty distance. Additionally, a weight vector adaptation approach is proposed to adjust the weight vector distribution. Experiments show that MaOEA-AWM has strong competitiveness in many-objective optimization problems compared with seven advanced algorithms.

Keywords: evolutionary algorithm, adaptive weights, decomposition-based, many-objective problems

INTRODUCTION

Multi-objective optimization problem (MOP) refers to the problem that multiple objective functions need to be optimized simultaneously in the optimization process, but the objectives compete with each other. MOPs can be summarized as [1]

$$\min_{x}\ f(x) = \left(f_1(x), f_2(x), \dots, f_M(x)\right) \\ s.t.\ \ x \in X \tag{1}$$

where $X \in \mathbb{R}^m$ is the decision space with the decision vector $x = (x_1, x_2, \dots, x_n) \in X$. Optimization problems with more than three objectives are called Many-objective problems (MaOPs). Due to the contradictory nature of the objectives, it is uncommon to find a single solution that simultaneously optimizes all of them. Hence, it is possible to achieve a set of Pareto-optimal solutions that illustrate the tradeoffs among various objectives [2].

When MOEAs are used to solve MaOPs, their performance degrades significantly due to dimensionality [3]. In addition, because of the high dimension, it becomes very difficult to maintain good population diversity [4,5]. Therefore, to overcome the above problems, a large number of MaOEAs [6] have been proposed by researchers. The existing MaOEAs can be broadly divided into three categories.

The first category of methods mainly involves modifying or relaxing the Pareto dominance relation to select excellent individuals. Such as ε-dominance [7], L-dominance [8], preference order ranking [9] and fuzzy Pareto dominance [10]. In the literature [11], a grid-based dominance metric is proposed to solve the MaOPs, which is called grid-based EA (GrEA). S-CDAS [12] extends the understanding of the dominant region.

The second category is the decomposition-based MaOEA. These algorithms transform MaOPs into multiple single-objective optimization subproblems with a set of reference vectors [13]. Among these algorithms, the most classical one is MOEA/D [14], which uses a predefined set of weight vectors to search for PS. The adaptive allocation search in MOEA/D-AM2M [15] and the adaptive scalarization method in MOEA/D-PaS [16]. Thus, diverse weights may give rise to different Pareto optimal sets [17].

Indicator-based approaches are the third kind. Including the indicator-based EA [18], the *S*-metric selection-based MOEA [19], a dynamic neighborhood MOEA based on hypervolume (HV) indicator [20], and the fast HV-based EA (HypE) [21]. Hypervolume (HV) refers to a widely used indicator in MOEA. When the objective is larger, the calculation cost of the HV value is very expensive. There are some simple indicators, such as the I(ε)+ indicator in IBEA [22] and the R2 indicator in R2-EMOA [23].

When solving MaOPs, the selection criteria of the enhanced convergence algorithm may lose the effectiveness of pushing the population to PF. Indicator-based algorithms often require high computational time complexity [24]. In the literature [25-27], inspired by the idea of decomposition-based algorithms, MaOEA-AWM is proposed. MaOEA-AWM uses an angle-based method to select individuals, which improves the convergence of individuals and the distribution of populations. The adaptive adjustment method of weight vectors is used to adjust the distribution of vectors to obtain a suitable Pareto Set (PS). The main contributions of MaOEA-AWM are summarized as follows:

(1) Different from most decomposition-based algorithms, the penalty-based boundary intersection (PBI) method is used as the criterion for evaluating individuals. The angle penalty distance (APD) is used to select individuals and to identify the main weight vector and the auxiliary weight vector, which helps to improve the convergence and diversity of individuals.

(2) Different from the principle of population division using general weight vectors, such as the population division principle mentioned in MOEA/D-M2M, MaOEA-AWM uses main-auxiliary weight vectors to divide the population. The main weight vector and its auxiliary weight vector are grouped into a partition. The new method of dividing the population can accelerate the convergence speed of the population and maintain the individuals' number in population.

(3) Different from the adaptive strategy of global weight vector adjustment in RVEA [28], an adaptive adjustment method for weight vectors is proposed in order to obtain the appropriate PS. This method adjusts the weight vector in the partition. The new strategy optimizes the distribution of weight vectors and enhances the diversity of individuals.

RELATED WORK

Weight Vector

According to the literature [29], a method of creating a uniform distribution weight vector is proposed. A set of uniformly distributed points are constructed:

$$w_i = (w_i^1, w_i^2, \ldots, w_i^M), i = 1,2,\ldots,N$$
$$w_i^j \in \left\{\frac{0}{W}, \frac{1}{W}, \ldots, \frac{W}{W}\right\}, \sum_{j=1}^{M} w_i^j = 1 \tag{2}$$

where N is the number of uniformly distributed reference points, M is the number of objectives, and W is a positive integer of the simplex-lattice design [30]. The corresponding unit weight vector v_i is obtained:

$$v_i = \frac{w_i}{\|w_i\|} \tag{3}$$

Fig.1 shows the transformation of the reference point from a hyperplane to a hypersphere.

The spatial relationship between vectors v_1 and v_2 is measured by the cosine value of the acute angle θ between them. The calculation formula is

$$\cos\theta = \frac{v_1 v_2}{\|v_1\|\|v_2\|} \tag{4}$$

where ||·|| represents the 2-norm calculation. Similarly, the space relationship can be measured by the cosine value of the acute angle β between vector a and individual A. The calculation formula is

$$\cos\beta = \frac{aA}{\|a\|\|A\|} \tag{5}$$

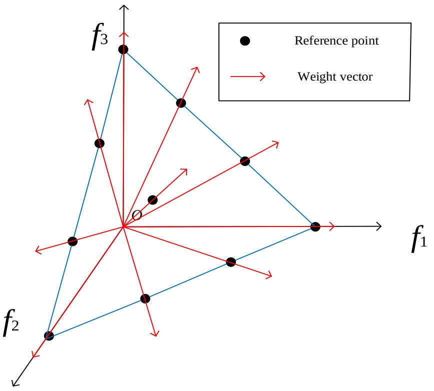


Figure 1. On three-objective, ten uniformly distributed weight vectors are generated

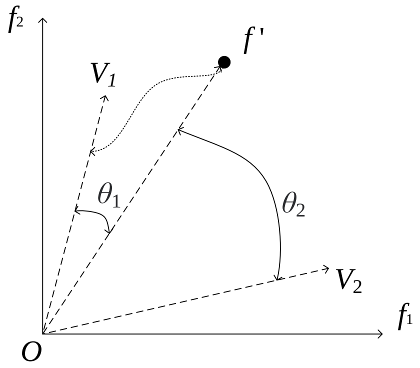


Figure 2. The association of individuals with vectors

Population Partition

The partitioning method adopted by RVEA is introduced. The population P_t is divided into N subpopulations, which is based on the association partition of the example in Fig. 2. The spatial relationship between two vectors (objective vector f'_i and weight vector v_j):

$$cos\ \theta_{i,j} = \frac{f_i^{'} v_j}{\|f_i^{'}\|} \tag{6}$$

Only when the angle between f'_i and v_j is the smallest, individual I_i is assigned to a subpopulation P_j.

The Angle-Penalized Distance APD

Angle-penalized distance (APD) is a distance metric commonly used in MaOEAs. The calculation of APD can be achieved by the following steps.

Firstly, the distance between individuals A and B is calculated, which is expressed as d (A, B).

Then, the angle difference between individual A and B is calculated and expressed as θ (A, B). The cosine similarity can be used to measure the angle difference. The specific calculation Eq. is

$$\theta(A, B) = arccos(\frac{f(A) \times f(B)}{\|f(A)\| \times \|f(B)\|}) \tag{7}$$

Among them, $f(A)$ and $f(B)$ are the objective vectors of individuals A B in the objective space, respectively. Finally, the Euclidean distance d (A, B) is multiplied by an angle penalty factor and expressed as P (θ (A, B)). The angle penalty factor can be defined according to specific problems, and the common forms are linear penalty, exponential penalty, and so on.

$$P(\theta(A, B)) = d(A, B) \times g(\theta(A, B)) \tag{8}$$

Here, g (θ (A, B)) is the angle penalty function, which is defined according to the specific problem. Through the above steps, the APD value between individual A and B can be obtained.

MAOEA-AWM

Motivation

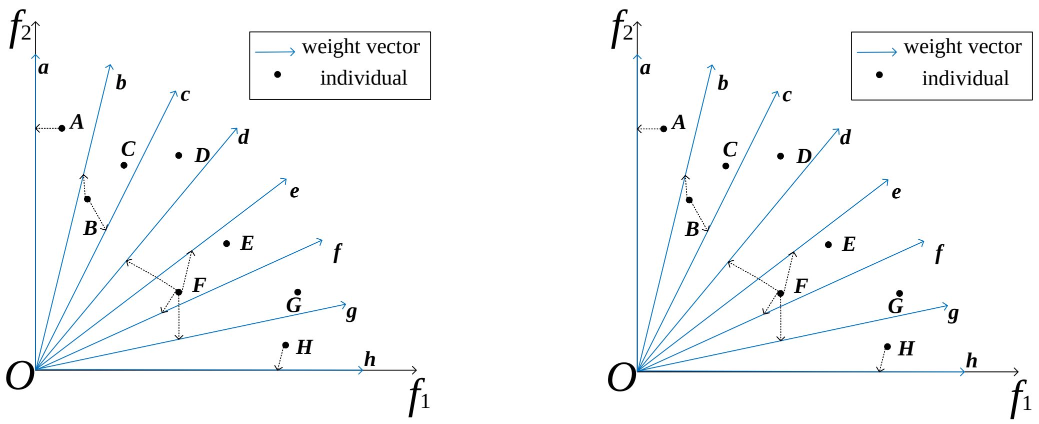
RVEA divides the population into multiple sub-populations depending on weight vectors and selects the best individuals from every sub-population to enter the next generation. However, when the search space becomes large, a small number of high-quality individuals corresponds to a large number of weight vectors, as illustrated in Fig.3(a). There are 8 weight vectors (*a*-*h*) and 8 individuals (*A*-*H*), where vector *a* corresponds to individual *A*, vectors *b* and *c* correspond to individual *B*, vectors *d*, *e*, *f*, and *g* correspond to individual *F*, and vector *h* corresponds to individual *H*. Individuals *C*, *D*, *E*, and *G* are not corresponding to any weight vector, while *B* and *F* are corresponding to more than one weight vector. At this time, individual selection is shown in Fig.3(b). Only individuals *A*, *B*, *F*, and *H* can enter the next generation, and the number of individuals in the population becomes smaller.

In order to solve the population shrinkage and diversity decline caused by the mismatch between individuals and weight vectors, the adjustment of the weight vector should be adopted. Some rules should be modified.

Firstly, based on a certain evaluation index or method, the important weight vectors need to be determined, which remain unchanged in the process of weight vector adjustment. As illustrated in Fig.4(a), the vectors *a*, *b*, *f*, and *h* are determined as the important weight vectors. During the weight adjustment, the vectors *a*, *b*, *f*, and *h* are unchanged.

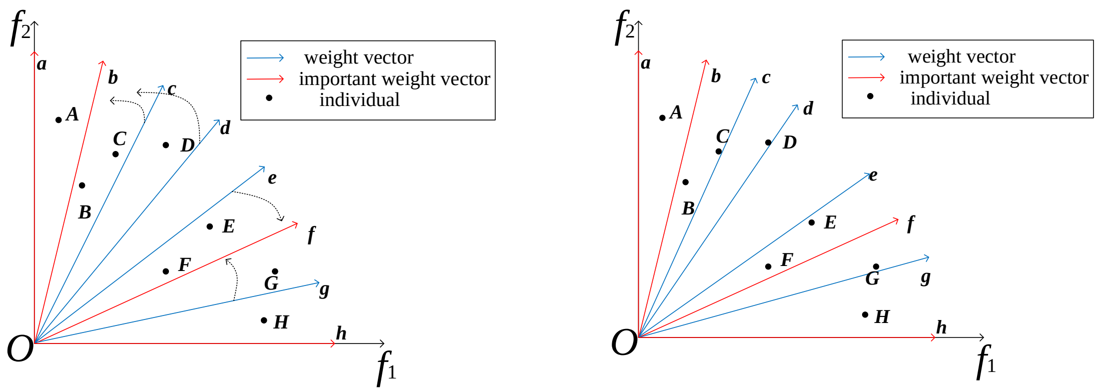
Secondly, the auxiliary weight vectors around the important weight vector should be adjusted to optimize the distribution of weight vectors and increase the diversity of individuals. In Fig.4 (a), vectors *c*, *d* are auxiliary weight vectors of vector *b*, and vectors *e*, *g* are auxiliary weight vectors of vector *f*. In Fig.4(b), after adjusting the weight vector, the auxiliary vectors *c* and *d* converge to the important weight vector *b*, and the auxiliary vectors *e* and *g* converge to the important weight vector *f*.

It is worth noting that the important weight vectors are not always constant, they may be changed according to the weight vector adjustment rules and the index. In Fig. 5, when the previous important weight vector *b* is no longer important, it becomes the auxiliary weight vector of the new important weight vector *c*. Similarly, the previous important weight vector *f* becomes the auxiliary weight vector of the new important weight vector *g*.



(a) Before the individual is selected (b) After the individual is selected

Figure 3. The corresponding situation of the weight vector and individual



(a) Determination of important weight vector and its auxiliary weight vector

(b) After the weight vector is adjusted

Figure 4. Important weight vector and auxiliary weight vector

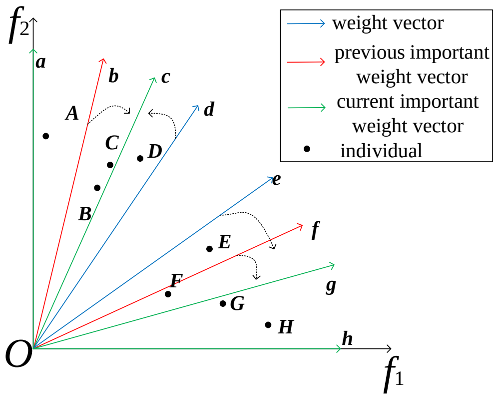


Figure 5. The vectors b and f are no longer important weight vectors, and the vectors a, c, g and h become new important weight vectors.

Main Framework

Algorithm 1 describes the main framework of MaOEA-AWM.

At first, population P_0 with N individuals and a set of unit weight vectors V_0 are generated (lines 1&2). The offspring population O_t was generated by the parent population P_t [32] (line 4). O_t is combined with P_t for elite selection (line 5). The next-generation population P_{t+1} is determined by environmental selection (line 6). The set W_{APD} records the APD value of all weight vectors V_t. The main weight vector V_{main} is determined based on W_{APD} (line 7). V_{main} represents the better of the weight vectors. When the current generation meets the specific condition (line 8), some weight vectors are adaptively adjusted (line 9).

The three components of MaOEA-AWM will be introduced in the following, including environmental selection (*Algorithm* 2), decision of the main weight vector (*Algorithm* 3), and weight vector adaptation (*Algorithm* 4).

Environmental Selection

In MaOEA-AWM, APD is used to evaluate individuals. The small value of APD represents the individual's good performance. *Algorithm* 2 describes the environmental selection of MaOEA-AWM.

Firstly, Normalize population P_t (line 1). Secondly, the APD of each individual I_j in P_t is calculated and stored in the I_{APD} (lines 2-4). Then, for each weight vector, the angle between the vector w and each individual in P_t is calculated (line 6), and the individuals with the smallest angle ($Angle_{min}$) to the weight vector w are found (line 7). Thirdly, the individual with the smallest APD is found and associated with the weight vector w (line 8). $I_{\text{APD-}min}$

stores the smallest APD. The individual index *temp* with the smallest APD is found (line 9). Finally, the APD of the weight vector w is $I_{\text{APD-}min}$, which is stored in W_{APD} (line 10). Individuals with index stored in temporarily are selected into the next generation population P_{t+1} (line 11).

***Algorithm* 1 Main Framework**

Input: The maximal number of generations t_{max}
Output: Final population P_{tmax}.

Create population P_0, the size of N;
Generate a set of unit weight vectors V_0;
while $t < t_{max}$ do
    O_t = Creation Offspring (P_t);
    $P_t = P_t \cup O_t$;
    [P_{t+1}, W_{APD}] = Environmental-Selection (P_t, V_t); (***Algorithm* 2**)
    V_{main} = Decide-Main-Vector (V_t, W_{APD}); (***Algorithm* 3**)
    if $t < t_{max} \times 90\%$ & mod (t, ($t_{max} \times 5\%$)) == 0
        V_{t+1} = Weight-Vector-Adaptation (P_{t+1}, V_t, V_{main}); (***Algorithm* 4**)
    End if
End while

***Algorithm* 2 Environmental Selection**

Input: t_{max}, P_t, weight vector set $V_t = \{V_1, V_2, \ldots, V_N\}$.
Output: Population P_{t+1}, APD of weight vectors W_{APD}.

1 Normalization (P_t);
2 **for** j = 1 to $|P_t|$ do
3   Calculate the I_{APD} of each individual I_j; /* refer to Eq. (8) *
4 **End for**
5 **for** i = 1 to N do
    /*Calculate the smallest angle value between vector w and individuals. */
6   *Angle* = pdist2 (P_t, w, *'cosine'*);
7     $Angle_{min}$ = min (*Angle*);
8     $I_{\text{APD-}min}$ = min ($I_{\text{APD}}(Angle_{min})$);
9   *temp* = find ($I_{\text{APD}} == I_{\text{APD-}min}$);
10     $W_{\text{APD}}(i) = I_{\text{APD-}min}$;
11     $P_{t+1} = (P_{t+1}, I_{temp})$;
12 **End for**

***Algorithm* 3 Decide Main Vector**

Input: $V_t = \{V_1, V_2, \ldots, V_N\}$, APD of weight vectors W_{APD}, pre-defined threshold *zeta*.
Output: Main vectors V_{main}.

for i = 1 to $|V_t|$ do
    $V_{main}(i) = V$ (find (W_{APD} == min (W_{APD})));
    for j = 1 to $|V_t|$ do
        if j ~= i & cosine ($V_{main}(i)$, V_j) <= *zeta*
            V_j is the auxiliary vector of $V_{main}(i)$;
            Remove the APD of V_j in W_{APD};
        End if
    End for
    Remove the APD of $V_{main}(i)$ in W_{APD};
End for

The angle penalty distance (APD) is used to select individuals rather than the widely used penalty-based boundary intersection (PBI) method, mainly because APD has higher target number scalability than PBI for solving MaOPs. PBI is based on the Euclidean distance calculation method, and APD is based on the angle distance calculation. The angle is always constant.

Decision of the Main Vectors

Algorithm 3 describes how to determine the main weight vector. The target space is divided into several sub-spaces by weight vectors. Every sub-space is the corresponding main weight vector space. The main weight vector selection strategy includes two steps: 1) Determining the main vector; 2) Population division.

a) Determining the main vector: All weight vectors are divided into main weight vectors and auxiliary weight vectors.

***Algorithm* 4 Weight Vector Adaptation**	
Input: Generation index t, weight vector set $V_t = \{V_1, V_2, \ldots, V_N\}$, main weight vector set V_{main}. **Output:** Weight vector set V_{t+1}.	
1	The individual $P_t^{\text{APD-max}}$ associated with the worst-performing vector and the individual $P_t^{\text{APD-min}}$ associated with the best-performing vector are calculated respectively;
2	$V_{t+1} = \frac{V_t \circ (P_t^{APD-max} - P_t^{APD-min})}{\|V_t \circ (P_t^{APD-max} - P_t^{APD-min})\|}$; /* refer to Eq. (9) */

The main weight vector is defined as the weight vector with good performance among all weight vectors, namely the weight vector with a small APD value. The weight vector whose angle difference from main weight vectors is smaller than the pre-defined threshold cosine ($V_{main}(i)$, V_j) <= *zeta* is defined as an auxiliary weight vector. The vector with the smallest APD value is selected as the main weight vector $V_{main}(i)$ (line 2). When the angle between $V_{main}(i)$ and other vectors V_j in V_t is less than the predetermined threshold *zeta*, the other vectors become $V_{main}(i)$'s auxiliary vectors (lines 4&5). APDs for $V_{main}(i)$ and V_j are removed in W_{APD} (lines 6&9).

b) Population division: A sub-space is defined by the main weight vector and all of its auxiliary weight vectors.

Weight Vector Adaption

When dealing with more complex PF, the final Pareto solution set may have bad results [31]. In order to deal with these defects, some studies have adopted weight vector adaptation. For example, in literature [32], each weight is periodically adjusted. Unlike other previous papers, some changes should be made:

(1) Weight adjustment occurs only within the partitions generated in Algorithm 3. Within a partition, the auxiliary weight vector will gather together with the main weight vector.

(2) Weight adjustment method: The main weight vector does not move, and the auxiliary weight vectors are adjusted by formulas:

$$V_{t+1} = \frac{V_t \circ (P_t^{APD-max} - P_t^{APD-min})}{\|V_t \circ (P_t^{APD-max} - P_t^{APD-min})\|} \tag{9}$$

where Vt+1 represents the next-generation partition vector and Vt represents the current partition vector. PtAPD-max represents the individual associated with the worst-performing vector in the partition, and PtAPD-min represents the individual associated with the best-performing vector in the partition. The ∘ operator denotes the Hadamard product.

Algorithm 4 introduces the whole process of weight vector adaptive. First of all, the weight adjustment adaptation condition is determined. The individual PtAPD-max associated with the worst-performing vector and the individual PtAPD-min associated with the best-performing vector are calculated respectively (line 2). Finally, the weight vector adaptation is performed with Eq. (9) (line 3).

EXPERIMENTAL

Experimental Design

In this chapter, MaOEA-AWM is compared with seven state-of-the-art MaOEAs on the DTLZ, SDTLZ1, and WFG.

Experimental design

To evaluate the effectiveness of MaOEA-AWM, this study selected 16 well-known testing problems from three benchmark test suites, DTLZ1-DTLZ6 [33], SDTLZ1, and WFG1-WFG9. These test problems possess diverse characteristics and challenge the performance of the MOEAs. Table 1 lists the specific characteristics of these testing problems.

Table 1. The characteristics of DTLZ1-6, SDTLZ1, and WFG1-9

Problem	Characteristic
DTLZ1	PF is a linear hyperplane.
DTLZ2	A relatively simple test example.
DTLZ3	PF presents a conical shape.
DTLZ4	It introduces an objective function that maps parametric variables to DTLZ2.
DTLZ5	It tests whether the algorithm can converge to a degenerate curve.
DTLZ6	It tests whether the algorithm can maintain sub-populations in different Pareto optimal regions.
SDTLZ1	Compared to the original DTLZ1, SDTLZ1 is challenging due to the strong scaling.
WFG1	WFG1 adopts a flat bias and PF hybrid structure design.
WFG2	PF consists of several disconnected convex segments whose variables are inseparable.
WFG3	The decision variable is indivisible and degenerate.
WFG4	WFG4 is characterized by multi-modal, large "mountains".
WFG5	WFG5 is a deceptive problem.
WFG6	WFG6 is an inseparable reduction problem.
WFG7	A separable single-peak problem, but it is parametrically dependent.
WFG8	WFG8 is indivisible and parameter-dependent.
WFG9	WFG9 is inseparable, and WFG9 will cause parameter dependence.

Comparative algorithm

In order to comprehensively validate the performance of MaOEA/D-AWM, this paper selected several representative MOEAs for comparison, including RVEA, NSGA-III [34], MOEA/DD [35], MOEA/D-AWA, PREA [36], KnEA and MOEA/D-UR [37]. The fundamentals of these seven state-of-the-art algorithms are briefly described below:

1) RVEA applies a framework similar to that of the NSGA-II, from which RVEA adopts an elitism strategy, where the offspring population is generated using traditional genetic operators.

2) MOEA/DD is a decomposition-based MOEA. The distributed deployment and collaborative update mechanism is adopted by MOEA/DD to realize the combination of global search and local optimization.

3) NSGA-III is an extension of NSGA-II and is characterized by the combination of non-dominated sorting with a decomposition-based niche strategy. This strategy replaces the traditional crowded distance and maintains the diversity of the population.

4) MOEA/D-AWA has an adaptive scheme to dynamically change the weights at the late stage of the optimization. Starting at 80% of the evolutionary process, weight vectors in the dense regions are periodically removed and new weight vectors are generated in the sparse regions.

5) PREA is a region-based many-objective evolutionary algorithm with a diversity maintenance mechanism using parallel distance to handle MOPs and MaOPs.

6) KnEA is a knee point-driven evolutionary algorithm for solving many-objective problems. Solutions for the next generation are first chosen based on the non-dominance selection criterion, and then knee points are used as the secondary selection criteria.

7) MOEA/D-UR is an improved algorithm based on the MOEA/D framework. It adopts a unified representation to deal with different types of decision variables.

Performance metric

Performance metrics are often employed to assess algorithm performance on a test suite. Hypervolume (HV) and inverted generational distance (IGD) [38] are applied in this experiment. These two performance indicators, which are very popular in academia, are adopted to measure the convergence and diversity of eight algorithms.

IGD measures the distance between the PF generated by the algorithm and the true PF. HV measures the range of the PF generated by the algorithm in the objective space. The small IGD value and the large HV value mean that the convergence and diversity of the algorithm are good.

Performance settings

a) Population Size: For MaOEA-AWM, RVEA, MOEA/DD, and NSGA-III, the population size is obtained by referring to Eq. (2). For the problem of $M \geq 8$, the method proposed in literature [39] can be used to apply the two-layer vector generation strategy. The population size settings are shown in Table 2. The population sizes of other algorithms are determined according to the literature [6], [14], and [40].

b) Parameter Settings: Referring to the literature [35], the values of parameters T, δ, and θ unique to MOEA/DD are set. Referring to literature [36], the values of parameter I^{r}_{∞} unique to PREA is set. Referring to the literature [12], the values of parameter T unique to KnEA are set. Referring to the literature [28], the values of parameters α and fr unique to RVEA are set. Referring to the literature [37], the values of parameters T, δ, and nr unique to MOEA/D-UR are set.

1) The maximum generations for different test instances are set as follows. The maximum generation gen_{max} of ZDT 3 is 250, gen_{max} of the other four test instances is 1000.

2) All algorithms are realized in MATLAB R2021b.

Table 2. Setting population size

M	Size
3	91
5	132
8	156
10	275
15	135

Results and Discussion

The purpose of this section is to investigate the validity of MaOEA-AWM for various types of MaOPs. Wilcoxon rank-sum test was used to compare the results obtained by MaOEA-AWM with the results obtained by seven comparative algorithms.

Performance on DTLZ1-DTLZ6

Table 3 and 4 show the HV and IGD values of MaOEA-AWM and the other seven algorithms in DTLZ1-6, and highlight the best data in bold.

For DTLZ1, MaOEA-AWM obtains the smallest IGD and the largest HV in a high-dimensional environment among all algorithms. For 3 objectives, the HV value of MaOEA-AWM is not much different from that of MOEA/D-UR. For 5 objectives, the IGD value of MaOEA-AWM is not greater than that of MOEA/D-AWA. For 8 objectives, the IGD value of MaOEA-AWM is not greater than that of MOEA/D-UR.

For DTLZ2, MaOEA-AWM performs well on objectives 5, 8, and 10, but poorly on objectives 3 and 15. The HV and IGD performance of MOEA/D-UR on objective 3 are better than those of MaOEA-AWM. The HV value of PREA is larger than that of MaOEA-AWM on objective 15.

For DTLZ4, the MaOEA-AWM has the best overall performance on 5, 8, and 10 objectives of this test problem, but the HV value on 3 objectives is slightly lower than that of PREA. The HV value on 15 objectives shows that MaOEA-AWM is not as good as PREA, NSGAIII, KnEA, and MOEA/D-UR.

For DTLZ5, MaOEA-AWM has excellent performance on HV value and IGD value. It is not idle that the performance of MaOEA-AWM on 5 objectives is slightly worse than that of MOEA/D-AWA and MOEA/DD.

Table 3. The statistical results of HV values measured by DTLZ1-DTLZ6

Problem	M	RVEA	PREA	MOEA/D-AWA	MOEA/DD	NSGAIII	KnEA	MOEA/D-UR	MaOEA-AWM
DTLZ1	3	1.3616e-1=	4.5663e-1=	3.5104e-1=	6.6758e-2=	4.3495e-1=	1.4143e-1=	4.5853e-1+	4.2855e-1
	5	3.7858e-1-	4.6305e-1-	4.6525e-1=	2.8005e-1-	2.0274e-1-	5.0025e-1=	4.7623e-1=	4.3314e-1
	8	4.7865e-2-	5.0178e-1-	4.9601e-1=	4.3808e-1-	4.5572e-3-	0.0000e+0-	5.0203e-1=	5.5103e-1
	10	5.4250e-1=	4.9031e-1=	4.3733e-1=	5.5558e-1=	3.3980e-2-	0.0000e+0-	3.9253e-1-	5.9253e-1
	15	1.8272e-1-	4.0167e-1-	1.6967e-1-	4.3301e-1-	6.7389e-4=	0.0000e+0=	4.6543e-1-	6.7393e-1
DTLZ2	3	5.5237e-1=	5.5753e-1+	5.4915e-1=	5.5473e-1+	5.5611e-1+	5.5195e-1=	5.5839e-1+	5.5301e-1
	5	7.6517e-1-	7.2689e-1-	7.0191e-1-	7.6040e-1-	7.5359e-1-	7.5197e-1-	8.2347e-1-	8.4987e-1
	8	8.5462e-1=	8.4432e-1=	6.7157e-1-	7.2589e-1-	8.0688e-1-	8.7286e-1=	8.7062e-1=	8.7327e-1
	10	9.2457e-1=	7.2322e-1=	6.7824e-1-	9.0049e-1=	8.4786e-1=	9.2539e-1=	9.0265e-1=	9.2898e-1
	15	4.4902e-1=	7.2264e-1+	2.8370e-1-	5.1048e-1=	6.5906e-1=	8.6694e-1+	7.1629e-1=	5.6519e-1
DTLZ3	3	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0
	5	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0
	8	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0
	10	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0
	15	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0
DTLZ4	3	5.5165e-1=	5.9816e-1+	4.1796e-1=	5.0842e-1=	5.1183e-1=	5.5154e-1=	5.6307e-1+	5.5312e-1
	5	7.6955e-1-	7.2476e-1-	7.3446e-1-	7.2106e-1-	7.2481e-1-	7.4056e-1-	7.5632e-1-	8.2786e-1
	8	8.4097e-1=	8.1027e-1=	7.9374e-1-	8.4034e-1=	8.5159e-1=	8.4856e-1=	8.2301e-1=	8.5814e-1
	10	9.1211e-1-	8.5318e-1=	8.4017e-1-	8.8785e-1-	8.7450e-1-	9.1145e-1=	7.4025e-1-	9.6719e-1
	15	5.8506e-1-	7.3665e-1=	4.2104e-1-	6.4224e-1=	8.1435e-1+	9.5256e-1+	7.5865e-1+	7.3915e-1
DTLZ5	3	1.4306e-1-	1.5612e-1=	1.9199e-1=	1.8242e-1=	1.9337e-1=	1.5573e-1-	1.7652e-1-	1.9346e-1
	5	9.4508e-2=	9.2924e-2-	1.1486e-1+	1.0791e-1+	8.4327e-2=	5.9089e-2-	1.0235e-1+	9.5188e-2
	8	9.0797e-2=	8.6156e-2-	9.9125e-2=	8.5281e-2-	7.3733e-2-	1.1614e-3-	8.5681e-2-	9.9363e-2
	10	9.1778e-2-	9.4235e-2=	9.3287e-2=	9.3911e-2=	4.8591e-2-	1.0008e-2-	9.2237e-2=	9.4871e-2
	15	9.1207e-2-	9.1854e-2=	9.2055e-2=	9.2040e-2=	8.2382e-2-	1.7639e-2-	9.2139e-2=	9.2613e-2
DTLZ6	3	1.4030e-1-	1.9475e-1-	1.9214e-1=	1.7904e-1-	1.9076e-1=	1.5736e-1-	1.9236e-1=	1.9425e-1
	5	6.7092e-2+	1.3538e-1=	1.1355e-1+	8.2179e-3-	0.0000e+0=	0.0000e+0=	1.2565e-1+	3.8043e-2
	8	4.6787e-2=	9.2446e-2=	9.9486e-2=	8.3998e-3=	0.0000e+0=	0.0000e+0=	0.0000e+0=	9.9882e-2
	10	4.7464e-2=	6.9426e-2=	7.3440e-2=	3.3213e-2=	0.0000e+0=	0.0000e+0=	7.2360e-2=	7.8218e-2
	15	9.0994e-2+	0.0000e+0=	9.1740e-2+	7.3372e-2+	0.0000e+0=	0.0000e+0=	9.8624e-2=	0.0000e+0
+/-/=		2/11/17	3/8/18	3/9/18	3/10/17	2/10/18	2/10/18	6/7/17	

Table 4. The statistical results of IGD values measured by DTLZ1-DTLZ6

Problem	M	RVEA	PREA	MOEA/D-AWA	MOEA/DD	NSGAIII	KnEA	MOEA/D-UR	MaOEA-AWM
DTLZ1	3	3.9193e-1 -	2.2330e-1 -	2.1075e-1 -	5.0209e-1 -	1.7313e-1 -	5.2266e-1 -	1.8506e-1 -	1.5990e-1
	5	4.1310e-1 =	2.8156e-1 =	2.3575e-1 =	3.6896e-1 =	4.5460e-1 =	3.9441e-1 =	2.9665e-1 =	3.0565e-1
	8	5.1984e-1 =	2.1815e-1 =	2.1630e-1 =	2.5785e-1 =	1.8827e+0 -	6.1258e+0 -	2.0431e-1 +	3.4204e-1
	10	2.2384e-1 =	2.1658e-1 =	3.7821e-1 =	3.5820e-1 =	6.3982e-1 =	8.2383e+0 -	2.0358e-1 =	2.1518e-1
	15	5.6046e-1 -	3.9323e-1 -	4.4299e-1 -	2.0217e-1 =	1.8464e+0 -	2.8158e+1 -	2.3254e-1 -	2.0054e-1
DTLZ2	3	5.5893e-2 -	5.0146e-2+	6.0621e-2 -	5.5154e-2 -	5.4838e-2 =	5.5853e-2 -	5.0353e-2 -	5.3813e-2
	5	2.1405e-1 =	2.3464e-1=	2.6820e-1 -	2.1404e-1 =	2.1653e-1 -	2.1437e-1 -	2.2035e-1=	2.1400e-1
	8	3.8885e-1 =	4.8900e-1 -	4.7548e-1 -	3.9250e-1 -	4.4847e-1 -	4.1791e-1 -	3.9032e-1 =	3.7353e-1
	10	5.2426e-1 -	6.7433e-1 -	6.1329e-1 -	5.0980e-1 -	6.1880e-1 -	5.1897e-1 -	5.7638e-1 -	5.0505e-1
	15	9.0912e-1 -	6.9536e-1 +	9.3980e-1 -	6.8581e-1 +	7.4742e-1 +	6.5836e-1 +	6.7639e-1 +	8.7378e-1
DTLZ3	3	1.8602e+1 -	6.4122e-2-	1.1537e+1 -	1.7998e+1 -	1.0703e+1 -	1.5790e+1 -	6.8122e+0=	7.0121e+0
	5	1.3708e+1=	1.0839e+1+	6.8183e+0=	1.7386e+1=	2.3032e+1=	1.6078e+1=	1.3659e+1=	1.4667e+1
	8	1.3745e+1=	1.1297e+1 =	8.1837e+0=	1.4197e+1=	2.7509e+1 -	2.2371e+2 -	2.0471e+2 -	1.3823e+1
	10	1.3993e+1=	1.0265e-1 -	1.0676e+1=	1.9673e+1=	2.6637e+1=	2.4293e+2 -	1.3658e+1=	8.0418e+0
	15	2.3263e+1 -	1.074e-1 =	1.5498e+1=	1.6673e+1-	2.6314e+1 -	2.7898e+2 -	1.5623e+1=	1.5317e+1
DTLZ4	3	5.5792e-2 =	4.0254e-1 -	3.5124e-1 -	1.5314e-1 -	1.5245e-1 -	5.6163e-2 =	3.3684e-1 -	5.5015e-2
	5	2.1411e-1 =	2.7695e-1 =	2.9185e-1 -	3.0245e-1 -	2.6894e-1 -	2.2156e-1 -	2.1561e-1 =	2.1266e-1
	8	4.6568e-1-	4.8429e-1 -	5.6258e-1 -	4.4634e-1 -	4.2531e-1 -	4.1902e-1 -	4.3649e-1 -	3.8084e-1
	10	5.7552e-1 -	5.3685e-1 =	6.7059e-1-	6.1743e-1 -	6.0815e-1 -	5.2281e-1 =	5.3256e-1 =	5.0906e-1
	15	8.7496e-1 -	6.1713e-1 =	1.0128e+0 -	8.5620e-1 -	7.6876e-1 -	6.5255e-1 =	6.1907e-1 =	6.1801e-1
DTLZ5	3	8.1874e-2 -	1.9211e-2 -	1.9419e-2 -	3.0259e-2 -	1.1967e-2 =	7.0668e-2 -	1.0687e-2 =	1.0206e-2
	5	2.3419e-1 -	1.2091e-1 -	9.0632e-2 =	9.0469e-2 =	1.4494e-1 -	1.6816e-1 -	8.9203e-2 +	9.0450e-2
	8	3.5822e-1 -	8.7263e-2 =	8.8640e-2 =	2.1541e-1 -	2.2235e-1 -	3.3724e-1 -	1.5822e-1 -	8.4922e-2
	10	6.7458e-1 -	2.6132e-1 -	1.4029e-1 =	1.4146e-1 =	2.9622e-1 -	3.4153e-1 -	1.2339e-1 =	1.0729e-1
	15	6.6643e-1 -	3.4116e-1 -	2.4241e-1 =	2.3939e-1 =	3.4558e-1 -	4.5870e-1 -	2.4259e-1 =	2.3702e-1
DTLZ6	3	8.6687e-2 -	2.5165e-2 -	1.8537e-2 =	3.5816e-2 -	1.8047e-2 =	7.3092e-2 -	7.2362e-2 -	1.0103e-2
	5	2.4215e-1 -	9.2341e-1 -	9.0149e-2 =	5.7827e-1 -	1.3768e+0 -	5.2595e-1 -	8.7569e-2 +	9.0069e-2
	8	6.3858e-1 -	1.6545e+0 =	1.8887e-1 =	5.5689e-1 -	4.0573e+0 -	3.1298e+0 -	33678e+0 -	1.8555e-1
	10	5.6017e-1 -	2.0525e+0 -	3.1335e-1 =	4.4493e-1 -	4.1339e+0 -	2.7178e+0 -	5.1168e-1 -	2.9385e-1
	15	5.4782e-1 +	2.6351e+0 -	2.7869e-1 +	4.1923e-1 +	4.5719e+0 -	3.2352e+0 -	2.0355e+0 =	1.8664e+0
+/-/=		1/19/10	3/16/11	1/14/15	2/17/11	1/22/7	1/24/5	4/11/15	

For DTLZ6, MaOEA-AWM performed significantly better than others on this problem, except for 5 objectives and 15 objectives. MOEA/D-AWA was better than MaOEA-AWM on 15 objectives, and MOEA/D-UR was better than MaOEA-AWM on 5 objectives.

On the whole, the MaOEA-AWM has the best overall performance on DTLZ test cases, although it performs poorly on a few individual issues.

Performance on SDTLZ1

Table 5. The statistical results of HV values measured by SDTLZ1

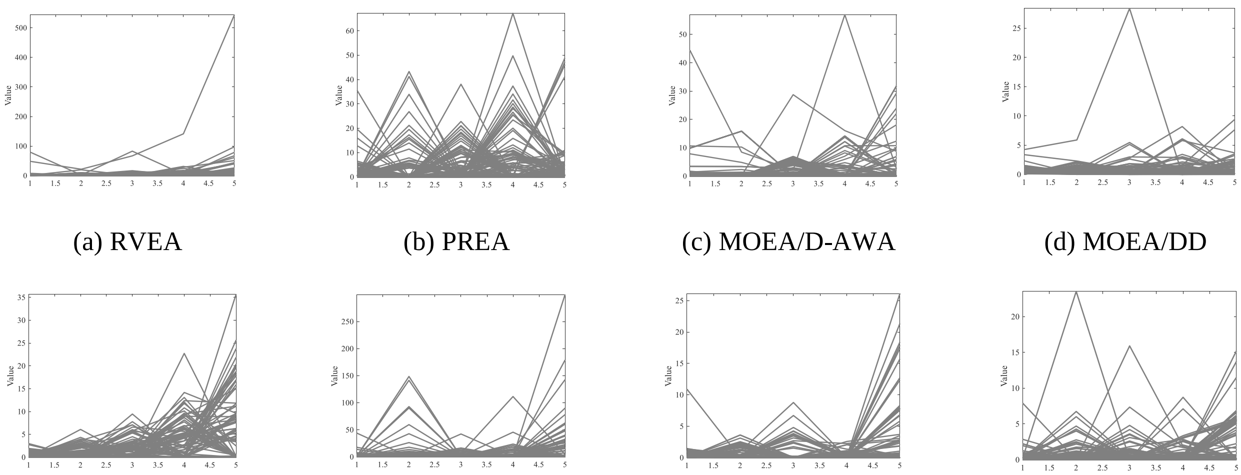
Problem	M	RVEA	PREA	MOEA/D-AWA	MOEA/DD	NSGAIII	KnEA	MOEA/D-UR	MaOEA-AWM
SDTLZ1	3	3.9133e-1 =	6.0235e-1 +	2.4444e-1 =	6.9107e-3 =	5.7207e-1 =	1.7038e-1 =	7.1657e-1 +	5.9687e-1
	5	5.6381e-2 -	5.6351e-2 -	2.5584e-1 -	1.5995e-2 -	4.5979e-2 -	7.6681e-3 -	3.9687e-1 =	4.1984e-1
	8	1.3372e-1 -	1.9679e-1 -	1.8839e-1 -	5.5885e-2 -	2.0220e-1 -	0.0000e+0=	2.3263e-1 =	2.3376e-1
	10	4.4399e-2 -	2.9848e-1 =	4.7045e-2 -	2.7429e-1 =	1.6454e-3 -	0.0000e+0=	2.2981e-1 -	2.7582e-1
	15	3.5927e-2 -	5.0339e-2 -	1.5584e-1 =	5.1637e-2 -	1.0504e-2 -	0.0000e+0=	1.6037e-1 =	2.3713e-1
+/-/=		0/4/1	1/3/1	0/3/2	0/3/2	0/4/1	0/1/4	1/1/3	

Table 6. The statistical results of IGD values measured by SDTLZ1

Problem	M	RVEA	PREA	MOEA/D-AWA	MOEA/DD	NSGAIII	KnEA	MOEA/D-UR	MaOEA-AWM
SDTLZ1	3	3.9731e-1 -	1.0368e-1 +	8.5339e-1 -	8.2459e-1 -	2.1858e-1 -	5.9141e-1-	1.2346e-1 =	1.2883e-1
	5	2.1781e+0 -	1.6357e+0 =	1.7311e+0=	1.4306e+0=	2.0963e+0 -	1.4381e+0=	1.4203e+0=	9.6631e-1
	8	7.3043e+0-	7.0336e+1-	7.7144e+0 -	5.8799e+0=	7.6202e+0 -	2.8696e+1 -	5.9697e+0=	5.7496e+0
	10	1.7055e+1=	2.3964e+1 -	3.6843e+1 -	2.9560e+1 -	2.5355e+1 -	1.8514e+2 -	3.4893e+1 -	1.8897e+1
	15	7.0984e+2=	7.0369e+3 -	8.2470e+2=	7.6961e+2=	1.2776e+3 -	9.0821e+3 -	1.3646e+3 -	6.2462e+2
+/-/=		0/3/2	1/3/1	0/3/2	0/2/3	0/5/0	0/4/1	0/2/3	

Tables 5 and 6 show the HV and IGD values of MaOEA-AWM and the other seven algorithms in SDTLZ1, and highlight the best data in bold. For the five-objective SDTLZ1, the final solutions obtained by all the algorithms are shown in Fig.6.

For SDTLZ1, the results show the comprehensive performance of MaOEA-AWM is the best, but for some objectives, such as the IGD value of 10 objectives, MaOEA-AWM is not smaller than other algorithms, and for the HV value of 3 objectives, MOEA/D-UR is greater than MaOEA-AWM. These observations show the MaOEA-AWM can handle this scaled DTLZ problem.



(a) RVEA (b) PREA (c) MOEA/D-AWA (d) MOEA/DD

(e) NSGAIII (f) KnEA (g) MOEA/D-UR (h) MaOEA-AWM

Figure 6. SDTLZ1 5 objectives

Performance on WFG1-WFG9

Table 7. The statistical results of HV values measured by WFG1-WFG9

Problem	M	RVEA	PREA	MOEA/D-AWA	MOEA/DD	NSGAIII	KnEA	MOEA/D-UR	MaOEA-AWM
WFG1	3	5.6890e-1-	7.3156e-1=	7.4869e-1=	3.7941e-1-	6.7646e-1=	4.3142e-1-	7.4657e-1=	7.5450e-1
	5	5.4936e-1-	6.5916e-1-	7.8084e-1=	3.4032e-1-	5.4401e-1-	6.2878e-1-	6.9626e-1-	7.8532e-1
	8	4.5855e-1-	5.9122e-1-	6.0415e-1=	2.9005e-1-	5.3232e-1-	5.5754e-1-	62655e-1=	6.6235e-1
	10	6.0234e-1=	5.3175e-1=	6.8718e-1=	3.4993e-1-	5.8261e-1-	5.4369e-1-	6.3179e-1=	7.0445e-1
	15	7.0277e-1=	5.9525e-1=	6.8397e-1=	5.8146e-1=	8.3339e-1+	3.9337e-1-	6.9525e-1=	7.1703e-1
WFG2	3	8.8481e-1-	9.1385e-1=	8.8182e-1-	8.8825e-1-	9.1154e-1=	8.4961e-1-	9.0325e-1=	9.1473e-1
	5	9.1366e-1=	9.4794e-1=	9.4414e-1=	9.1734e-1=	9.5046e-1=	9.6387e-1+	9.3649e-1=	9.5440e-1
	8	8.9126e-1=	9.4648e-1=	9.2196e-1=	8.7857e-1=	9.6725e-1=	9.6193e-1=	9.3748e-1=	9.6832e-1
	10	8.8200e-1-	9.3967e-1=	9.4660e-1-	8.6920e-1-	9.6170e-1=	9.5944e-1=	9.4469e-1=	9.6346e-1
	15	8.3998e-1=	9.3360e-1+	8.5740e-1=	7.9439e-1=	8.9315e-1=	8.7157e-1+	9.3633e-1+	8.0219e-1
WFG3	3	3.1829e-1-	3.7623e-1=	3.7322e-1=	2.3650e-1-	3.6011e-1=	2.7778e-1-	3.6921e-1=	3.8592e-1
	5	4.5776e-2=	9.2817e-2=	8.6098e-2=	4.1560e-2=	6.8023e-2=	6.1795e-2=	8.9634e-2=	9.6106e-2
	8	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0
	10	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0
	15	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0=	0.0000e+0
WFG4	3	5.1811e-1-	5.0697e-1=	5.2238e-1=	5.2077e-1=	5.2157e-1=	4.6893e-1-	5.4286e-1=	5.7426e-1
	5	6.9541e-1=	6.9527e-1+	5.9826e-1=	6.7156e-1=	7.0366e-1+	7.1747e-1+	6.7366e-1=	6.6438e-1
	8	7.1524e-1+	7.6702e-1+	6.9044e-1+	6.4805e-1+	7.6585e-1+	8.2725e-1+	7.9350e-1+	6.0499e-1
	10	7.2638e-1+	8.2554e-1+	6.2768e-1=	6.8574e-1=	8.3298e-1+	8.5867e-1+	7.0854e-1+	6.5084e-1
	15	5.0387e-1=	7.5864e-1+	2.8911e-1-	4.4105e-1=	5.7178e-1=	7.4120e-1+	8.5368e-1+	5.3259e-1
WFG5	3	4.9624e-1-	5.0993e-1=	4.7713e-1-	4.9602e-1=	5.0478e-1=	4.5066e-1-	5.0794e-1=	5.1037e-1
	5	6.7983e-1+	6.1278e-1+	5.4304e-1=	6.2760e-1+	6.8078e-1+	6.9371e-1+	6.9467e-1+	5.4876e-1
	8	7.2612e-1=	7.3608e-1=	6.1367e-1=	5.5993e-1-	6.9034e-1=	7.8905e-1=	7.7306e-1=	7.9205e-1
	10	7.7078e-1=	7.5938e-1=	6.3603e-1-	5.9555e-1-	7.9629e-1=	8.2510e-1=	7.2908e-1=	8.3455e-1
	15	5.8129e-1=	7.3683e-1=	2.4972e-1-	3.6617e-1-	5.7089e-1=	8.0910e-1+	7.6533e-1=	6.5904e-1
WFG6	3	4.7584e-1=	4.9094e-1-	4.4913e-1=	4.5922e-1-	4.7585e-1=	4.0660e-1-	5.0165e-1=	5.0676e-1
	5	6.5327e-1=	6.5972e-1=	5.1945e-1-	6.2569e-1=	6.4588e-1=	6.6144e-1=	6.6978e-1=	6.7950e-1
	8	6.3429e-1=	6.8913e-1=	5.8280e-1-	4.7319e-1-	6.7532e-1=	7.4085e-1=	6.9887e-1=	7.8576e-1
	10	4.9546e-1-	7.4735e-1=	5.8496e-1-	6.5246e-1-	7.6585e-1=	7.8698e-1=	7.5235e-1=	7.9133e-1
	15	3.0812e-1-	6.7863e-1+	2.5173e-1-	2.6155e-1-	5.3919e-1=	8.0033e-1+	7.5623e-1+	5.2662e-1
WFG7	3	5.1519e-1=	5.1277e-1=	5.1178e-1=	5.1786e-1=	5.3193e-1=	4.7441e-1-	5.5310e-1=	5.5123e-1
	5	7.1381e-1=	5.5329e-1-	5.6728e-1-	6.4974e-1-	6.9038e-1-	7.3299e-1=	7.5146e-1=	7.5597e-1
	8	7.2991e-1=	6.7624e-1-	6.1613e-1-	5.1069e-1-	7.1560e-1=	8.4700e-1+	7.7242e-1=	7.7968e-1
	10	7.3697e-1-	7.3392e-1-	5.9310e-1-	7.3962e-1-	8.2312e-1=	8.7198e-1=	8.5852e-1=	8.7593e-1
	15	3.7709e-1-	6.2682e-1=	2.5071e-1-	1.8742e-1-	6.2272e-1=	7.0435e-1+	8.6836e-1+	6.5524e-1
WFG8	3	4.2305e-1-	4.2349e-1-	4.2480e-1-	4.3447e-1=	4.4238e-1=	4.0254e-1-	4.6559e-1=	4.6244e-1
	5	5.6740e-1=	5.7337e-1=	4.3221e-1-	5.2739e-1-	5.7337e-1=	5.7317e-1=	5.9817e-1=	6.3627e-1
	8	4.3146e-1-	5.9338e-1=	4.1340e-1-	3.1340e-1-	5.9428e-1=	6.0390e-1=	5.8641e-1=	6.4550e-1
	10	3.5351e-1-	7.3264e-1+	4.7492e-1-	6.6023e-1+	7.3049e-1+	6.4574e-1+	7.6831e-1+	6.0182e-1
	15	3.1315e-1=	8.0908e-1+	2.6031e-1=	1.3803e-1-	5.0370e-1=	5.9803e-1+	4.7872e-1=	4.5016e-1
WFG9	3	4.7174e-1-	5.2406e-1=	4.1875e-1-	4.6518e-1-	5.0118e-1=	4.4953e-1-	5.1217e-1=	5.3036e-1
	5	6.2958e-1=	6.8870e-1+	4.7412e-1-	5.0087e-1-	5.9691e-1=	6.5333e-1+	5.8794e-1=	6.0455e-1
	8	5.8945e-1=	7.1263e-1+	5.1005e-1-	4.5449e-1-	5.9353e-1=	7.5702e-1+	4.6348e-1-	6.8456e-1
	10	5.9791e-1-	7.5742e-1+	4.5978e-1-	5.4090e-1-	6.5373e-1=	7.1634e-1+	4.6035e-1-	6.8245e-1
	15	4.3079e-1-	4.9298e-1-	2.2505e-1-	2.5606e-1-	4.6802e-1-	5.2202e-1-	4.6465e-1-	5.3129e-1
+/-/=		3/18/24	5/14/26	1/23/21	3/26/16	6/5/34	10/20/15	9/16/20	

Table 8. The statistical results of IGD values measured by WFG1-WFG9

Problem	M	RVEA	PREA	MOEA/D-AWA	MOEA/DD	NSGAIII	KnEA	MOEA/D-UR	MaOEA-AWM
WFG1	3	8.1631e-1-	5.5569e-1=	5.2513e-1=	1.2166e+0-	5.8627e-1=	1.0257e+0-	4.9986e-1+	5.0850e-1
	5	1.2307e+0-	9.6938e-1+	9.3826e-1+	1.7597e+0-	1.2859e+0=	1.3485e+0=	9.6927e-1+	1.0055e+0
	8	1.8460e+0-	1.5965e+0=	1.7867e+0=	2.3683e+0-	1.9113e+0-	1.6228e+0=	1.8628e+0-	1.5699e+0
	10	1.7456e+0=	1.7985e+0-	1.9967e+0-	2.4395e+0-	2.0281e+0-	1.9438e+0-	1.6962e+0=	1.6359e+0
	15	2.3958e+0=	2.5368e+0-	2.4711e+0-	2.6234e+0-	2.5918e+0-	3.1500e+0-	2.4891e+0 -	2.3677e+0
WFG2	3	2.2399e-1-	2.1379e-1-	2.8227e-1-	1.9767e-1=	1.8056e-1=	2.5333e-1-	2.1360e-1-	1.7776e-1

	5	5.1530e-1=	4.9819e-1=	8.3334e-1-	5.4438e-1=	5.1160e-1=	7.4515e-1-	5.1694e-1=	5.0006e-1
	8	1.2085e+0-	1.0936e+0=	1.3771e+0-	1.6861e+0-	1.3115e+0-	1.1255e+0=	1.0977e+0=	1.0819e+0
	10	1.3293e+0-	1.3428e+0-	1.8294e+0-	1.9555e+0-	1.5038e+0-	1.3368e+0-	1.3237e+0-	1.2782e+0
	15	2.1764e+0-	1.9562e+0=	3.1470e+0-	2.2847e+0-	3.5516e+0-	2.9613e+0-	1.9232e+0=	1.8429e+0
WFG3	3	2.4013e-1-	1.0517e-1=	1.2786e-1=	4.0442e-1-	1.4584e-1=	3.0496e-1-	1.1141e-1=	1.0107e-1
	5	7.2474e-1=	5.2399e-1+	9.2763e-1=	9.6458e-1=	8.2867e-1=	8.4866e-1=	7.0396e-1 =	7.1406e-1
	8	2.7478e+0=	1.5684e+0+	2.3606e+0=	2.7994e+0=	1.3813e+0+	1.3493e+0+	2.6997e+0-	2.6410e+0
	10	5.1248e+0=	5.9651e-1+	3.5469e+0=	3.4370e+0=	2.1150e+0+	2.0139e+0+	3.3670e+0-	3.0682e+0
	15	7.3722e+0=	9.3073e-1=	6.9486e+0=	6.9188e+0=	5.2147e+0=	5.0551e+0+	8.6320e-1+	6.3920e+0
WFG4	3	2.6966e-1=	2.6937e-1=	2.9278e-1-	2.4768e-1=	2.2922e-1=	3.2408e-1-	2.4738e-1=	2.2759e-1
	5	1.2247e+0=	1.1917e+0=	1.8478e+0-	1.3739e+0=	1.2285e+0=	1.8257e+0-	1.3693e+0-	1.2145e+0
	8	3.6798e+0=	3.4917e+0+	3.8268e+0=	3.9381e+0=	3.5854e+0=	3.4895e+0+	3.1813e+0+	3.6241e+0
	10	6.0253e+0=	6.9341e+0=	7.3056e+0-	6.6127e+0=	5.8368e+0=	5.2553e+0+	6.3447e+0=	6.0065e+0
	15	1.2743e+1=	1.9681e+1=	1.8201e+1-	1.2114e+1=	1.1436e+1=	8.6924e+0+	1.1385e+1=	1.2014e+1
WFG5	3	2.6141e-1-	2.8387e-1-	2.9281e-1-	2.5137e-1=	2.3644e-1=	3.2069e-1-	2.6977e-1=	2.1462e-1
	5	1.2142e+0=	1.3671e+0=	1.7530e+0-	1.3691e+0=	1.2052e+0=	1.9331e+0-	1.2035e+0=	1.1879e+0
	8	3.6832e+0-	3.5493e+0-	3.6493e+0-	4.1008e+0-	3.5446e+0-	3.4870e+0-	3.0809e+0=	3.0526e+0
	10	6.1066e+0-	5.4666e+0-	7.0031e+0-	7.1182e+0-	5.8886e+0-	5.1921e+0=	6.3912e+0-	5.1013e+0
	15	1.1362e+1=	1.2137e+1=	1.3535e+1=	1.2122e+1=	1.1874e+1=	8.7065e+0+	1.1814e+1=	1.2064e+1
WFG6	3	3.0452e-1-	2.6388e-1=	3.4885e-1-	3.0491e-1-	2.7511e-1=	3.9093e-1-	2.7469e-1=	2.7130e-1
	5	1.2550e+0=	1.3575e+0=	1.8355e+0-	1.3964e+0=	1.2443e+0=	1.9153e+0-	1.2476e+0=	1.2395e+0
	8	3.9119e+0-	3.3696e+0=	3.7645e+0=	4.1766e+0-	3.6136e+0=	3.8017e+0-	3.2388e+0=	3.1363e+0
	10	6.7426e+0-	5.9576e+0=	7.1591e+0-	6.3529e+0-	6.0464e+0=	6.0736e+0=	5.8782e+0=	5.7979e+0
	15	1.6853e+1=	1.7199e+1-	1.6699e+1-	1.3740e+1=	1.2249e+1=	9.5692e+0+	9.9260e+0+	1.2261e+1
WFG7	3	2.7892e-1=	2.6358e-1=	3.0100e-1-	2.6046e-1=	2.3305e-1=	3.2774e-1-	2.5963e-1=	2.3245e-1
	5	1.2423e+0=	1.4635e+0=	1.9227e+0-	1.3873e+0-	1.2535e+0=	1.9029e+0-	1.2353e+0=	1.2212e+0
	8	3.6068e+0-	3.4637e+0=	3.8936e+0-	3.8983e+0-	3.5394e+0-	3.4956e+0=	3.1946e+0=	3.1863e+0
	10	5.8006e+0=	5.8013e+0-	7.3707e+0-	6.0095e+0-	5.8384e+0=	5.2583e+0=	5.6136e+0=	5.1467e+0
	15	1.4453e+1=	1.6311e+1+	1.5845e+1-	1.1970e+1=	1.1273e+1+	8.8075e+0+	1.0507e+1+	1.2751e+1
WFG8	3	3.8803e-1=	3.8617e-1=	3.7588e-1=	3.3315e-1=	3.2295e-1=	4.1605e-1-	3.3657e-1=	3.1133e-1
	5	1.3101e+0=	1.5249e+0=	1.8828e+0-	1.4145e+0=	1.3000e+0=	1.8645e+0-	1.3605e+0=	1.1652e+0
	8	3.8339e+0-	4.3325e+0-	4.3111e+0-	4.0627e+0-	3.9836e+0-	3.9269e+0-	3.6733e+0=	3.3056e+0
	10	6.4470e+0-	5.9512e+0-	7.7952e+0-	5.8113e+0=	6.2080e+0=	5.6750e+0=	6.6839e+0=	5.5133e+0
	15	1.3730e+1-	1.6357e+1-	1.8726e+1-	1.3255e+1-	1.2428e+1=	1.0099e+1=	1.4658e+1=	1.0076e+1
WFG9	3	2.8604e-1=	2.6538e-1=	3.8444e-1-	2.9472e-1-	2.3937e-1=	3.2706e-1-	2.4618e-1=	2.2687e-1
	5	1.2081e+0=	1.3288e+0=	1.8635e+0-	1.3944e+0=	1.2277e+0=	1.8459e+0-	1.1582e+0=	1.1030e+0
	8	3.5447e+0=	3.2073e+0+	3.7854e+0=	4.1515e+0=	3.6247e+0=	3.2950e+0+	3.5953e+0-	4.2771e+0
	10	5.8631e+0=	7.0176e+0=	6.7909e+0-	6.2099e+0=	5.6230e+0+	4.9467e+0+	6.1165e+0=	6.0998e+0
	15	1.3449e+1=	1.1699e+1=	1.4175e+1-	1.2186e+1=	1.1397e+1=	8.7407e+0+	1.0687e+1+	1.1773e+1
+/-/=		0/18/27	7/11/27	1/32/12	0/20/25	3/10/32	12/23/10	6/9/30	

Tables 7 and 8 show the HV and IGD values of MaOEA-AWM and the other seven algorithms in WFG1-9, and highlight the best data in bold.

For WFG1, although the IGD performance of MaOEA-AWM is slightly better than other algorithms when there are more than 5 objectives, its performance is worse than that of PREA on 3 and MOEA/D-UR on 5 objectives. NSGAIII has the largest HV value on 15 objectives, and MaOEA-AWM has the largest HV value on other objectives.

For WFG2, MaOEA-AWM has the best overall performance. On 5 objectives, there is no obvious difference in the performance of other algorithms.

For WFG3, the performance of MaOEA-AWM is not idle. Its performance is similar to RVEA, MOEA/D-AWA, and MOEA/DD, and worse than PREA, KnEA, and MOEA/D-UR. And MOEA/D-UR obtained the smallest IGD value.

For WFG4, MaOEA-AWM performed poorly in most cases and only performed well on three objectives.

According to the performance of MaOEA-AWM on WFG3 and WFG4, it can be found MaOEA-AWM is easy to fall into the local optimum, which may be due to the fact that MaOEA-AWM does not change the main weight vector quickly.

For WFG5, RVEA, MOEA/D-AWA, MOEADD, NSGAIII, and KnEA performed much worse than MaOEA-AWM on 3-10 objectives. KnEA performed better on 15 objectives.

For WFG6, the performance of MOEA/D-UR and MaOEA-AWM is similar and better than other algorithms on 3-10 objectives, but MaOEA-AWM is slightly worse than MOEA/D-UR on 15 objectives.

For WFG7, the performance of MOEA/D-UR and MaOEA-AWM is nearly the same and better than other algorithms.

For WFG8, MaOEA-AWM has shown significant performance. However, on 10 objectives, the HV value of MOEA/D-UR is larger than that of other algorithms, and PREA has the largest HV value on 15 objectives.

For WFG9, in most objectives' cases, the performance of PREA and KnEA is better than that of other algorithms, while the performance of MaOEA-AWM is better than that of other algorithms on 3 and 5 objectives.

On the whole, in most cases, the performance of MaOEA-AWM is outstanding or very competitive among the seven algorithms. MaOEA-AWM outperforms the other seven compared algorithms for WFG2, WFG5-WFG8, while PREA, KnEA, and MOEA/D-UR offer better performance than MaOEA-AWM for WFG3, WFG4, and WFG9.

CONCLUSION

This paper proposed a many-objective evolutionary algorithm (called MaOEA-AWM) for handling many-objective problems. MaOEA-AWM has a simple decomposition-based structure with few variable parameters. MaOEA-AWM's main feature is to adopt an angle-based criteria (APD) to select outstanding individuals in the environmental selection process. Based on APD, all outstanding weight vectors are selected as main weight vectors, and vectors around main weight vectors are their auxiliary weight vectors. The main weight vector and its auxiliary weight vectors are grouped into a partition. In the partition, the weight vector is adjusted; that is, the main weight vector is not moved, and auxiliary weight vectors gather around it. The purpose of vector adjustment is to maintain the diversity of search directions.

MaOEA-AWM is contrasted with seven advanced MaOEAs against 16 problems from DTLZ, SDTLZ, and WFG to validate the effectiveness of APD selection and vector adaption in MaOEA-AWM. MaOEA-AWM performs well in the majority of cases, according to the results.

However, MaOEA-AWM is easy to fall into the local optimum, which may be caused by the single evaluation indicator of the weight vector. The performance of MaOEA-AWM on MOPs in WFG is not very good. It may be further modified by increasing the weight vector evaluation indicator or changing the predetermined angle between the main and auxiliary weight vectors.

REFRENCES

[1] K. Deb, "Multi-objective optimization using evolutionary algorithms." John Wiley & Sons, Inc (2001).
[2] R. Cheng, Y. Jin, M. Olhofer and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization." IEEE Transactions on Evolutionary Computation 20.5 (2016).
[3] H. Ishibuchi, N. Akedo, Y. Nojima. "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems. " IEEE Transactions on Evolutionary Computation 19 (2) (2015):264-283.
[4] Q. Zhang, A. Zhou and Y. Jin, "RM-MEDA: A Regularity Model-Based Multi-objective Estimation of Distribution Algorithm." Evolutionary Computation IEEE Transactions on (2008).
[5] Y. Jin, B. Sendhoff,"Connectedness, regularity and the success of local search in evolutionary multi-objective optimization." Congress on Evolutionary Computation IEEE (2003).
[6] L. Li, G. Li, L. Chang, "A many-objective particle swarm optimization with grid dominance ranking and clustering." Applied Soft Computing 96 (2020): 106661.
[7] M. Laumanns, L. Thiele, K. Deb, E. Zitzler. "Combining convergence and diversity in evolutionary multi-objective optimization." Evolutionary Computation 10 (3) (2002):263-282.
[8] 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) 38 (5) (2008):1402-1412.

[9] F. D. Pierro, S. T. Khu, D. A. Savic, "An Investigation on Preference Order Ranking Scheme for Multi-objective Evolutionary Optimization." IEEE Transactions on Evolutionary Computation, 2007, 11(1):17-45.

[10] G. P. Wang, H. Jiang, "Fuzzy-Dominance and Its Application in Evolutionary Many Objective Optimization." 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007) IEEE, 2008.

[11] L. Li, G. Li, L. Chang, "A many-objective particle swarm optimization with grid dominance ranking and clustering." Applied Soft Computing 96 (2020): 106661.

[12] Q. Zhang, H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition." IEEE Transactions on Evolutionary Computation 11 (6) (2007):712-731.

[13] M. Wagner and F. Neumann, "A fast approximation-guided evolutionary multi-objective algorithm." Conference on Genetic and Evolutionary Computation (2013):687-694.

[14] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun and J. Wu, "MOEA/D with Adaptive Weight Adjustment." Evolutionary Computation 2014; 22 (2): 231-264.

[15] H. -L. Liu, F. Gu and Q. Zhang, "Decomposition of a Multi-objective Optimization Problem into a Number of Simple Multi-objective Subproblems." IEEE Transactions on Evolutionary Computation 18.3(2014):450-455.

[16] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods." IEEE Transactions on Evolutionary Computation 20(6) (2016):821-837.

[17] M. Li, X. Yao, "What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multi-objective optimisation." Evolutionary Computation 28.2 (2020): 227-253.

[18] E. Zitzler and L. Thiele, "Multi-objective Optimization Using Evolutionary Algorithms - A Comparative Case Study." 1998.

[19] E. Zitzler, K. Simon, "Indicator-Based Selection in Multi-objective Search." 8th International Conference on Parallel Problem Solving from Nature Springer Berlin Heidelberg (2004):832-842.

[20] N. Beume, B. Naujoks and M. Emmerich, "SMS-EMOA: Multi-objective selection based on dominated hypervolume." European Journal of Operational Research 181.3(2007):1653-1669.

[21] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization." Evolutionary Computation 19 (1) (2011):45-76.

[22] E. Zitzler, K. Simon, "Indicator-Based Selection in Multi-objective Search." 8th International Conference on Parallel Problem Solving from Nature Springer Berlin Heidelberg (2004):832-842.

[23] H. Trautmann, T. Wagner and D. Brockhoff, "R2-EMOA: Focused Multi-objective Search Using R2-Indicator-Based Selection." Springer Berlin Heidelberg (2013):70-74.

[24] M. Wagner and F. Neumann, "A fast approximation-guided evolutionary multi-objective algorithm." Conference on Genetic and Evolutionary Computation (2013):687-694.

[25] H. -L. Liu, L. Chen, Q. Zhang, and K. Deb, "Adaptively Allocating Search Effort in Challenging Many-Objective Optimization Problems." IEEE 3(2018).

[26] C. Bao, D. Gao, W. Gu, L. Xu, "A new adaptive decomposition-based evolutionary algorithm for multi-and many-objective optimization." Expert Systems with Applications 213 (2023): 119080.

[27] M. Luque, S. Gonzalez-Gallardo and R. Saborido, "Adaptive global WASF-GA to handle many-objective optimization problems." Swarm and Evolutionary Computation 54 (2020): 100644.

[28] R. Cheng, Y. Jin, M. Olhofer and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization." IEEE Transactions on Evolutionary Computation 20.5 (2016).

[29] R. Cheng, Y. Jin, and K. Narukawa, "Adaptive Reference Vector Generation for Inverse Model-Based Evolutionary Multi-Objective Optimization with Degenerate and Disconnected Pareto Fronts." Evolutionary Multi-Criterion Optimization (2015):127-140.

[30] L. B. Hare, "Experiments with mixtures: designs, models, and the analysis of mixture data /-2nd ed." Wiley (1991).

[31] R. B. Agrawal, K. Deb, "Simulated Binary Crossover for Continuous Search Space." Complex Systems 9.3(1994):115-148.

[32] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design, " Comput. Sci. Informat., vol. 26, no. 4, (1996):30-45.

[33] K. Deb, "Multi-Objective Optimization Using Evolutionary Algorithms. " New York, NY, USA: Wiley, 2001.
[34] K. Deb, and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints." IEEE Transactions on Evolutionary Computation 18.4(2014):577-601.
[35] K. Li, K. Deb, Q. Zhang and S. Kwong, "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition." IEEE Transactions on Evolutionary Computation (2014):694-716.
[36] J. Yuan, H. L. Liu, F. Gu, Q. Zhang, and Z. He, "Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions." IEEE Transactions on Evolutionary Computation 25.1 (2020): 75-86.
[37] De Farias, R. C. Lucas, A. F. R. Araújo, "A decomposition-based many-objective evolutionary algorithm updating weights when required." Swarm and Evolutionary Computation 68 (2022): 100980.
[38] B. Peter AN, D. Thierens, "The balance between proximity and diversity in multi-objective evolutionary algorithms." IEEE transactions on evolutionary computation 7.2 (2003): 174-188.
[39] K. Deb, and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints." IEEE Transactions on Evolutionary Computation 18.4(2014):577-601.
[40] X. Zhang, Y. Tian and Y. Jin, "A knee point-drivenvolutionary algorithm for many-objective optimization." IEEE Transactions on Evolutionary Computation 19 (6) (2015):761-776.