

Research Article

A Decomposition-Based Multiobjective Optimization Evolutionary Algorithm with Adaptive Weight Generation Strategy

Guo-Zhong Fu , Tianda Yu , Wei Li , Qiang Deng , and Bo Yang 

Science and Technology on Reactor System Design Technology Laboratory, Nuclear Power Institute of China, Chengdu 610213, China

Correspondence should be addressed to Guo-Zhong Fu; guo-zhongfu@hotmail.com

Received 17 June 2021; Revised 9 August 2021; Accepted 12 August 2021; Published 2 September 2021

Academic Editor: Mohammad Yazdi

Copyright © 2021 Guo-Zhong Fu 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.

Multiobjective evolutionary algorithm based on decomposition (MOEA/D) is the seminal framework of multiobjective evolutionary algorithms (MOEAs). To alleviate the nonuniformly distributed solutions generated by a fixed set of evenly distributed weight vectors in the presence of nonconvex and disconnected problems, an adaptive vector generation mechanism is proposed. A coevolution strategy and a vector generator are synergistically cooperated to remedy the weight vectors. Optimal weight vectors are generated to replace the useless weight vectors to assure that optimal solutions are distributed evenly. Experiment results indicate that this mechanism is efficient in improving the diversity of MOEA/D.

1. Introduction

Multiobjective optimization problems (MOPs), different from the single-objective optimization problems (SOPs), have more than one objective function, and the objective functions conflict with each other. In the optimization process, the relationship between the objective function values is partially ordered. The optimization cannot be selected by simply comparing the values. The optimized solution of MOPs is a set of mutually compromised solutions, i.e., Pareto optimal solution set (PS) [1–5]. The performance metrics of the PS are diversity and convergence. Also, the multiobjective optimization supports simplify the design of products and their several key performances like efficiency, reliability, availability, and lifetime cost-saving [6–11].

Conventional analytical algorithms aggregate multiple objective functions into one, which can be solved by the analytical method. However, these algorithms can only obtain one optimized solution at each iteration. Only by running multiple times and setting different single “composite” objective functions can we obtain enough Pareto optimal solutions. At the same time, since the optimization

process of each iteration is independent of each other, the information in the iteration process cannot be shared, requiring complex computations [12]. Furthermore, these algorithms are problem-dependent. Different aggregation strategies would be only applicable for specific MOP problems. An evolutionary algorithm (EA) is a population-based and problem-independent optimization algorithm. It poses computers the ability to solve complex optimization in the absence of gradient information and obtains a set of optimization solutions in parallel at each iteration. Multiobjective evolutionary algorithms (MOEA) have become an important branch in the field of EA and evolutionary multiobjective optimization (EMO) research. The iconic research on employing EA to solve MOPs refers to Goldberg, by which the notion of nondominated sorting and niching technique are introduced [13]. It established the nondominated based framework of MOEAs, by which many researches are developed [14–17]. However, most objective optimization problems (MaOPs) with more than 3 objectives, which make it is difficult for nondominated based MOEAs to obtain efficient solutions. Because the portion of nondominated solutions in the whole population increase

dramatically, it would be difficult to discriminate optimized solutions from the population. Relaxed dominance based MOEAs [18–21], preference-based MOEAs [22–24], indicator based MOEAs [25–27], and decomposition based MOEAs [28, 29] are introduced to tackle MaOPs. The decomposition based MOEAs decompose the MOPs into a set of single-objective subproblems with reference to prescribed weight vectors and then optimize these subproblems at the same time. Due to its superiority in strong convergence to the Pareto front and wide scope of adaption in complex MOPs and MaOPs, decomposition based MOEAs have been the research hotspot in EMO. Recently most researches on decomposition based MOEAs are based on the seminal framework of MOEA/D proposed by Zhang and Li [28].

In the preliminary algorithm, MOEA/D assumes that a set of uniformly distributed weight vectors could result in well-distributed solutions. However, Li and Zhang found that different weight vectors could result in different distributed solutions with respect to different optimization problems [30]. Referring to a convex problem illustrated in Figure 1, a set of uniformly distributed weight vectors $\mathbf{w} = \{w_1, w_2, w_3, w_4, w_5, w_6\}$ would result in a set of non-uniformly distributed solutions $\mathbf{s} = \{s_1, s_2, s_3, s_4, s_5, s_6\}$.

Much research has tried to redesign the generation method of weight vectors so that the optimized solutions can be evenly distributed in the target space, thereby enhancing their diversity. So far, representative weight vector generation methods include simplex-lattice [31], double-layer simplex lattice [32], and uniform design [33]. To obtain an optimal solution with uniform distribution, it demands that the Pareto optimal frontier is not of discrete, degenerate, and singular shapes. In view of this, the current research on weight vectors is mainly focused on the adaptive strategy. According to the sparse degree of the optimized solution, the weight vectors are automatically added or deleted to meet the needs of improving diversity. Relevant researchers have developed effective algorithms regarding the adaptive strategy. Jain and Deb proposed an adaptive reference point reduction strategy based on NSGA-III [32] and A-NSGA-III [34]. A-NSGA-III constructs $(m-1)$ simplex lattices of reference points and deletes the reference points around crowded reference points that have no correlation with the optimized solution. Qi et al. [35] proposed an adaptive weight adjustment strategy (MOEAD-AWA). In the optimization process, first assume that the algorithm converges to the real Pareto optimal frontier, periodically delete the crowded weight vector, and add a new weight vector to the sparse area. Cheng et al. [36] adopted two sets of reference vectors to deal with different problems. However, these methods have certain limitations. These algorithms are specifically designed to solve certain problems. Giagkiozis et al. [37] pointed out that the use of adaptive strategies to generate new weight vectors may cause insufficient convergence for the entire evolution process. In view of this, Wang et al. [38] proposed a method PICEA-w based on the coevolution strategy of weight vectors and optimized solutions. By calculating the Chebyshev function values of the optimized solutions with respect to the weight vectors, the optimal solution and the dominant weight vector are selected into the next iteration together. However, the

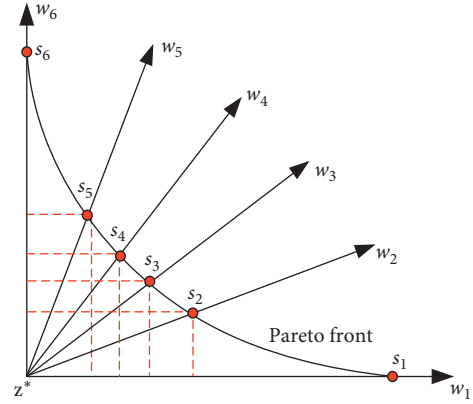


FIGURE 1: Nonuniformly distributed solutions with respect to uniformly distributed weight vectors.

process of adding weight vectors adopts a randomly generated method and does not make full use of the current solution information. Therefore, based on the coevolution strategy of PICEA-w, this paper introduces an adaptive weight generation mechanism based on MOEA/D (MOEA/D-AW), which adopts the weight vector generation mechanism [35, 39] to make full use of the distribution of the generated solution in the objective space, and reversely guide the addition and deletion of the weight vector.

The remainder of this paper is organized as follows: Section 2 describes the mechanism of PICEA-w. Section 3 introduces the optimized weight generation mechanism and proposes a novel adaptive weight generation based on the coevolution strategy of PICEA-w. Experiment settings are illustrated in Section 4 and experimental results are illustrated in Section 5. Finally, Section 6 concludes this paper.

2. Description of PICEA-W

2.1. Basic Procedure of PICEA-W. The pseudocode of PICEA-w is illustrated in Algorithm 1. 5 functions involved in PICEA-w are illustrated in Table 1.

Brief introductions of the functions involved in Table 1 are listed below (refer to the original research [38] for details),

2.1.1. weightGenerator. Function *weightGenerator* is used to generate N_w weight vectors $\mathbf{w}_j = \{w_1, \dots, w_i, w_M\}$, $j = 1, 2, \dots, N_w$. We define the weight vector \mathbf{w}_j in the following form:

$$w_1 = 1 - \sqrt[M-1]{\text{rand}}$$

$$w_i = \left(1 - \sum_{k=1}^{i-1} w_k\right) \left(1 - \sqrt[M-1]{\text{rand}}\right) ; w_M = 1 - \sum_{j=1}^{i-1} w_k, \quad (1)$$

where N_w is the number of weight vectors, *rand* is a random number between 0 and 1, and M is the number of objectives.

2.1.2. updateA. Function *updateA* is used to upgrade the nondominated solutions in A . When the size of A surpasses A size, it will execute function *trimA* to make sure that the size of A remains A size.

<p>Inputs: Number of population N, initial population S, N_w weight vectors $W = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{N_w}\}$, maximum iteration number $iter_{max}$, number of objective functions M, size of archive A A size</p> <p>Outputs S, W, A, Best F</p> <p>(1) Best $F \leftarrow \phi$</p> <p>(2) $S \leftarrow$ initialize $S(N)$</p> <p>(3) $F_S \leftarrow$ obj(S)</p> <p>(4) Best $F \leftarrow$ update A (Best F, F_S)</p> <p>(5) $W \leftarrow$ weightGenerator (N_w)</p> <p>(6) while $iter < iter_{max}$ do</p> <p>(7) $Sc \leftarrow$ genetic Operator (S)</p> <p>(8) $F_{Sc} \leftarrow$ obj(Sc)</p> <p>(9) (Joint S, Joint F) \leftarrow Union (S, Sc, F_S, F_{Sc})</p> <p>(10) $Wc \leftarrow$ weight Generator (N_w)</p> <p>(11) $\theta \leftarrow$ theta Configuration ($iter$, $\pi/2$)</p> <p>(12) (S, F_S, W) \leftarrow coEvolve (Joint F, Joint S, Joint W, θ)</p> <p>(13) Best $F \leftarrow$ update A (Best F, F_S, A size)</p> <p>(14) end while</p>

ALGORITHM 1: PICEA-w.

TABLE 1: Functions involved in Algorithm 1.

Name of functions	Description
weightGenerator	Generate random weight vectors
update A	Update archive in accordance with the obtained nondominated solutions
thetaConfiguration	Obtain the value of angle intended for the function <i>coEvolve</i>
coEvolve	Obtain new solutions and new weight vectors
trim A	Confine the size of the archive

2.1.3. *trimA*. Function *trimA* is used to confine the size of A to remain initialized value with reference to the clustering technique in SPEA2 [15].

2.1.4. *thetaConfiguration*. Function *thetaConfiguration* is used to obtain an angle θ as one of the input parameters of function *coEvolve*, and the θ is generated as follows:

$$\theta = \frac{\pi}{2} \frac{iter}{iter_{max}}, \quad (2)$$

where $iter$ is the current number of iterations and $iter_{max}$ is the maximum number of iterations.

2.1.5. *coEvolve*. Function *coevolve* generates optimized solution S , optimized objective values F_S , and optimized weight vector W by coevolution algorithm with all input parameters which are θ , *JointS*, *JointF* and *JointW*. *JointS* is the joint of parent and offspring solutions at $iter$, *JointF* is the mapping of *JointS* in the objective space, and *JointW* is the joint of optimized weight vectors and randomly generated weight vectors.

The basic procedure of coevolution in function *coevolve* is listed as follows:

- (1) Calculate the angles between each weight vector \mathbf{w}_j , $j = 1, 2, \dots, 2N_w$ in *JointW* and each objective vector \mathbf{F}_i , $i = 1, 2, \dots, 2N$ in *JointF* to obtain a relevant angle vector Φ in the scale of $2N \times 2N_w$ when

the angle between \mathbf{F}_i and \mathbf{w}_j is less than θ , \mathbf{F}_i , $i = 1, 2, \dots, 2N$ is regarded as the neighbor of \mathbf{w}_j . Any elements in Φ surpassing θ will be treated as inf. The performance of \mathbf{F}_i is evaluated by calculating the Chebyshev scalarizing function of \mathbf{w}_j and \mathbf{F}_i . Replace the elements that manifest the neighboring relations in Φ with the Chebyshev scalarizing values to obtain a performance vector \mathbf{P} in the scale of $2N \times 2N_w$. Sort the elements of \mathbf{P} in descending order by \mathbf{w}_j to obtain a sequence vector \mathbf{R} in the scale of $2N \times 2N_w$ of which 1 represents the best performance.

- (2) Choose N \mathbf{F}_i as new F_S based on the sequence vector \mathbf{R} . Sort \mathbf{R} in ascending order. The 1st column is marked as the best performance, followed by the 2nd column, and so on. Then, choose the first N \mathbf{F}_i to form a new F_S .
- (3) Choose N_w \mathbf{w}_j as new W with reference to the newly obtained \mathbf{F}_i of F_S . If \mathbf{F}_i performs the same with multiple weight vectors, choose the farthest \mathbf{w}_j away from \mathbf{F}_i to be its most optimal weight vector.

2.2. *Flaws of PICEA-W*. At each iteration, weight vectors are randomly generated in accordance with “(1).” In the evolution, as the distribution of the mapping of obtained solutions in the objective is not fully used, which induces that the weight vector generated in step 10 of Algorithm 1 is very

likely to be the same as the last generation and eventually has a lack of diversity.

3. A Novel Decomposition-Based MOEA with Adaptive Weight Generation (MOEA/D-AW)

3.1. Optimal Weight Generation Strategy. Given a reference point $\mathbf{z}_i^* = (\mathbf{z}_1^*, \mathbf{z}_2^*, \dots, \mathbf{z}_M^*)$, the optimal weight vector $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M\}$ can be obtained by calculating the Chebyshev scalarizing functions with respect to \mathbf{z}_i^* . The basic procedure is listed as follows:

$$\frac{f_1(s) - \mathbf{z}_1^*}{\mathbf{w}_1} = \frac{f_2(s) - \mathbf{z}_2^*}{\mathbf{w}_2} = \dots = \frac{f_M(s) - \mathbf{z}_M^*}{\mathbf{w}_M}. \quad (3)$$

Meanwhile, as $\mathbf{w}_1 + \mathbf{w}_2 + \dots + \mathbf{w}_M = 1$, the optimal weight vector can be expressed as follows:

$$\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_M) = \left(\frac{f_1(s) - \mathbf{z}_1^*}{\sum_{i=1}^M f_i(s) - \mathbf{z}_i^*}, \dots, \frac{f_M(s) - \mathbf{z}_M^*}{\sum_{i=1}^M f_i(s) - \mathbf{z}_i^*} \right). \quad (4)$$

3.2. Basic Idea of the Proposed Algorithm. MOEA/D-AW makes full use of the current solutions to update the weight vectors instead of a set of fixed weight vectors to improve the diversity of the solutions. Different from PICEA-w, the coevolution strategy is amended by replacing the weight generation mechanism shown in “(1)” with that of “(4).”

3.3. Detailed Description. The pseudocode of MOEA/D is illustrated in Algorithm 2. Firstly, an evenly distributed weight vector is generated by means of that proposed by Das and Dennis [31], $\mathbf{W} \leftarrow \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N\}$. N subproblems $p_i, i = 1, 2, \dots, N$ are determined by $\mathbf{w}_i, i = 1, 2, \dots, N$ to form the initialized population $P^0 = \{p_1^0, p_2^0, \dots, p_N^0\}$. The ideal point \mathbf{z}_0^* in step 3 of Algorithm 2 is determined by minimizing all populations in each objective.

The adaptive weight generation mechanism is manifested in Step 7 of Algorithm 2 with reference to a combination and revision of Step 10 to 12 in Algorithm 1. The weight generation is embedded in function *coEvolve*, once the weight vector with respect to P_i^{iter} could not be its neighbor, the optimal weight vector generated by “(4)” would replace the current one, and then the subproblems in the next iteration would also be updated. The diversity of the solutions is ensured by updating the weight vectors (Algorithms 3 and 4).

As Differential Evolution (DE) operator outstands in convergence with respect to most cases [40], it is used to generate offspring.

4. Experiment Settings

4.1. Test Problems. In this paper, 8 problems from the Walking Fish Group (WFG) test suite [41] invoked in 2-, 4-, 7-objective instances are introduced to test our proposed algorithm and other competitors. The WFG parameters k and l are set to be 18 and 14. The attributes of these problems

include nonseparable/separable, unimodal/multimodal, biased (polynomial, flat, parameter dependent)/unbiased, convex/concave, connected/disconnected, etc. Note that the decision variables in WFGs are $z_i = [0, 2i], i = 1, 2, \dots, n$. Hereinafter, WFGm-n refers to the test problem WFGm with n objectives.

4.2. Performance Metrics. In this paper, 1 qualitative performance metric (Median Attainment Surface [42]) and 4 quantitative performance metrics are included, which are Hypervolume (HV) [16], Inverted Generational Distance (IGD) [43], Generational Distance (GD) [44], and Pure Diversity (PD) [45]. The Median Attainment Surface could intuitively indicate the diversity and convergence of obtained solutions in the objective space (normally less than 3 objectives). HV and IGD are the most widely used metrics that could evaluate the diversity and convergence simultaneously. A greater value of HV and a smaller value of IGD both indicate that the solutions perform better. At the same time, a smaller value of GD and a larger value of PD indicates better performance in convergence and diversity, respectively. Note that the PD metric could assess the diversity of solutions not only in the dimensions of spread and uniformity but also dissimilarity.

4.3. Competitor MOEAs. To evaluate the performance of the proposed MOEA/D-AW, 4 recent published outstanding MOEAs, which are Reference Vector Guided Evolutionary Algorithm (RVEA) [36], A-NSGA-III [34], Preference Inspired Coevolutionary Algorithm Using Goals (PICEA-g) [46], and Ensemble Fitness Ranking with Ranking Restriction Scheme (EFR-RR) [47], are introduced to be compared with.

In Reference Vector Guided Evolutionary Algorithm (RVEA), a set of evenly distributed reference vectors is prescribed to divide the objective space into the same number of subspaces as that of reference vectors. At each iteration, RVEA combines the parent and the offspring to obtain the next generation based on the elitism strategy of NSGA-II [17]. Angle-Penalized Distance (APD) is introduced to balance the performance of RVEA in diversity and convergence. APD is calculated with reference to the number of iterations so that RVEA could be more inclined to converge at the early iterations while spreading at the final iterations. In the presence of the performance degeneration of RVEA in dealing with irregular Pareto fronts, RVEA introduces a strategy that could adaptively regenerate the reference vector to alleviate the degeneration. Specifically, after subpopulations are generated by the reference vectors, the reference vectors with respect to empty subpopulations are replaced to be random unit vectors that are ranged from the maximum and minimum objective values of the current translated population.

A-NSGA-III is the adaptive version of NSGA-III which is manifested in adaptively adding and reducing reference points. Similar to NSGA-II, at iteration t , NSGA-III first select the nondominance levels $(F_{1,t}, F_{2,t}, \dots, F_{l-1,t})$ prior to the critical (last) nondominance level $(F_{l,t})$ to construct part

Initialization N : number of population, T : the size of the neighbors, iter_{\max} : maximum number of iteration

- (1) Generate an evenly distributed weight vector, $W \leftarrow \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N\}$
- (2) Generate an initialized population, $P^0 = \{p_1^0, p_2^0, \dots, p_N^0\}$ based on W and calculate the corresponding objective functions, $F_{\cdot}P^0 \leftarrow \text{obj}(P^0)$
- (3) Set the initialized ideal point based on P^0
- (4) Calculate the T closest weight vectors to \mathbf{w}_i to form the neighbor, $\lambda_i \leftarrow \{\mathbf{w}_i, \mathbf{w}_j, \dots, \mathbf{w}_T\}$
- (5) $\text{iter} \leftarrow 1$
- (6) **while** $\text{iter} < \text{iter}_{\max}$ **do**
- (7) $(F_{\cdot}P^{\text{iter}}, P^{\text{iter}}, W) \leftarrow \text{coEvolve}(F_{\cdot}P^{\text{iter}}, P^{\text{iter}}, W)$
- (8) $P^{\text{iter}} \leftarrow \text{GenerateOffspring}(P^{\text{iter}}, DE)$
- (9) $F_{\cdot}P^{\text{iter}} \leftarrow \text{obj}(P^{\text{iter}})$
- (10) **end while**
- (11) return non-dominated solutions in P

ALGORITHM 2: MOEA/D-AW.

Inputs DE : the operator, P^{iter} : population at iter
Outputs P^{iter}

- (1) **for** $i \leftarrow 1: N$ **do**
- (2) Randomly pick 3 individuals from the neighbors of p_i^{iter} , use DE operator to generate its offspring u_i^{iter}
- (3) $P^{\text{iter}} \leftarrow \text{Update}(u_i^{\text{iter}}, p_i^{\text{iter}})$
- (4) **end for**

ALGORITHM 3: GenerateOffspring.

Inputs u_i^{iter} : offspring, subproblem: p_i^{iter}
Outputs p_i^{iter} : updated subproblem

- (1) Based on the weight vector with respect to p_i^{iter} , $\mathbf{w}_i = \{w_1, w_2, \dots, w_M\}$, calculate the Chebyshev function of u_i^{iter} and the other solutions in p_i^{iter} , $f_{Tch}(u_i^{\text{iter}})$, $f_{Tch}(nox_{i,k}^{\text{iter}})$, $nox_{i,k}^{\text{iter}} \in p_i^{\text{iter}}$, $k = \{1, 2, \dots, T\}$
- (2) **for** $k \leftarrow 1: T$ **do**
- (3) **if** $f_{Tch}(u_i^{\text{iter}}) < f_{Tch}(nox_{i,k}^{\text{iter}})$ **then**
- (4) $nox_{i,k}^{\text{iter}} \leftarrow u_i^{\text{iter}}$
- (5) **end if**
- (6) **end for**

ALGORITHM 4: Update.

of the next generation (Q_{t+1}), S_t and much efforts are put into choosing remaining populations from $F_{l,t}$. Associate the solutions in S_t with reference points by calculating the closest perpendicular distance between a solution in S_t and an reference line j . Based on the number of associated solutions concerning reference point j , p_j , reference points with minimum p_j are first classified. In the case of $p_j = 0$, choose the population member from $F_{l,t}$ which is closest to reference line j to construct Q_{t+1} ; otherwise exclude the reference point j . In the case of $p_j \geq 1$, a randomly chosen member from $F_{l,t}$, if exists, would be included if it is associated with reference point j . Based on NSGA-III, referring to the case of $p_j > 1$, A-NSGA-III adds a corresponding number of reference points according to the number of objectives near the crowded point j and deletes all reference points that $p_j = 0$. By doing this, the ideal state (each $p_j = 1$)

could be achieved to obtain an evenly distributed Pareto front.

Based on the coevolution strategy, PICEA-g uses the range of current objective values to construct a set of goal vectors to divide the objective space. It combines the conventional Pareto dominant relationship and fitness calculation based on goal vector. Like that of NSGA-II, it sorts the fitness of the members in the combined population which is constructed by the parent and the offspring to choose the optimal generation.

Based on the Ensemble Fitness Ranking (EFR), EFR-RR introduces a ranking restriction scheme to formulate the outstanding EFR-RR. The fitness evaluation in EFR is barely executed based on the aggregating function, which would result in distant solutions to the weight vectors and poor performance in diversity. In view of this, EFR-RR devised a

ranking restriction scheme under which the solutions could only be sorted by calculating the fitness function formulated with its neighboring weight vectors. A set of diversely distributed solutions can then be obtained in accordance with an evenly distributed weight vector.

The reference vector in RVEAa, A-NSGA-III, and PICEA-g (goal vector in PICEA-g) could be adjusted adaptively based on the current population to obtain a set of more diverse solutions. At the same time, EFR-RR restricts the fitness sorting to be executed between the solution and its neighboring fitness function to obtain a set of more diverse solutions.

4.4. Parameters. The general parameters involved in the experiment are set as follows:

(1) Population size, N

The N in A-NSGA-III, RVEAa, PICEA-g, and EFR-RR are all set to be 100 when dealing with the 2, 4, and 7 objectives WFGs. The N in our proposed MOEA/D-AW is set to be 100, 126, and 210 with respect to the 2, 4, and 7 objectives WFGs, respectively.

(2) Rate of change of penalty function, α , and the frequency of reference vector adaption, fr , in RVEAa α is set to be 2 and fr is set to be 0.1 in accordance with that of the original research [35].

(3) Number of goals in PICEA-g, N_{goal}

N_{goal} is set to be $N \cdot m/2$, where m the number of objectives in accordance with that of the original research [46].

(4) Number of neighboring weight vectors in EFR-RR, K

K is set to be 2 in accordance with that of the original research [47].

(5) Position parameter, k , and distance parameter, l , in WFGs

k is set to be 18 and l is set to be 14

(6) Mutation and crossover operators

All algorithms adopt the simulated binary crossover (SBX) and polynomial mutation to generate offspring. The distribution index, η_c , and crossover probability, p_c , in SBX are set to be 15 and 1.0, respectively. The distribution index, η_m , and the mutation probability, p_m , in polynomial mutation are set to be 20 and $1/32$, respectively.

Note that the parameter p in PD is set to be 0.1, suggested by Wang et al.

(7) Termination condition

The maximum number of iterations is set to be 250.

In addition, when calculating the performance metrics, all nondominated solutions are set to be 100.

5. Experimental Results

All algorithms are executed independently for 31 times. The results are developed from 3 aspects: Median

Attainment Surface, numerical statistics, and boxplot. Draw the Median Attainment Surface that approximates the Pareto optimal frontal surface obtained after running all the comparison algorithms 31 times, as shown in Figures 2 and 3. The statistical results of the values with respect to HV, IGD, GD, and PD metrics are listed in Tables 2 to 5.

Note that, in each illustration, ANSGAIII represents A-NSGA-III, PICEAg represents PICEA-g, EFRRR represents EFR-RR, and AW represents MOEA/D-AW.

5.1. Median Attainment Surface. We can see the following from Figure 2:

- (1) Referring to the discrete and convex WFG2, all algorithms cannot converge to the true Pareto optimal front. RVEAa is inferior to other algorithms in terms of convergence, while all algorithms perform the same in terms of diversity.
- (2) Referring to the linear and convex WFG3, all algorithms could converge to the true Pareto optimal front to some extent, while PICEA-g has a certain degree of volatility, and the convergence is slightly worse. In terms of diversity performance, PICEA-g and MOEA/D-AW are superior to other algorithms.
- (3) Referring to the multimodal, separable, and convex WFG4, MOEA/D-AW is superior to other algorithms in terms of both convergence and diversity.
- (4) Referring to the deceptive, separable, and concave WFG5, all algorithms could converge to the true Pareto optimal front to some extent while PICEA-g has a certain degree of volatility, and the convergence is slightly worse. In terms of diversity performance, PICEA-g and MOEA/D-AW are superior to other algorithms.

We can see the following from Figure 3:

- (1) Referring to the nonseparable, unimodal, and concave WFG6, all algorithms could converge to the true Pareto optimal front to some extent while PICEA-g has a certain degree of volatility, and the convergence is slightly worse. In terms of diversity performance, PICEA-g and MOEA/D-AW are superior to other algorithms.
- (2) Referring to the separable, unimodal, and concave WFG7, all algorithms cannot converge to the true Pareto optimal front. In terms of diversity performance, EFR-RR is superior to other algorithms.
- (3) Referring to the nonseparable, unimodal, and concave WFG8, all algorithms cannot converge to the true Pareto optimal front. EFR-RR and RVEAa are superior to other algorithms in terms of convergence.
- (4) Referring to the deceptive, multimodal, nonseparable, and concave WFG9, all algorithms could converge to the true Pareto optimal front to some

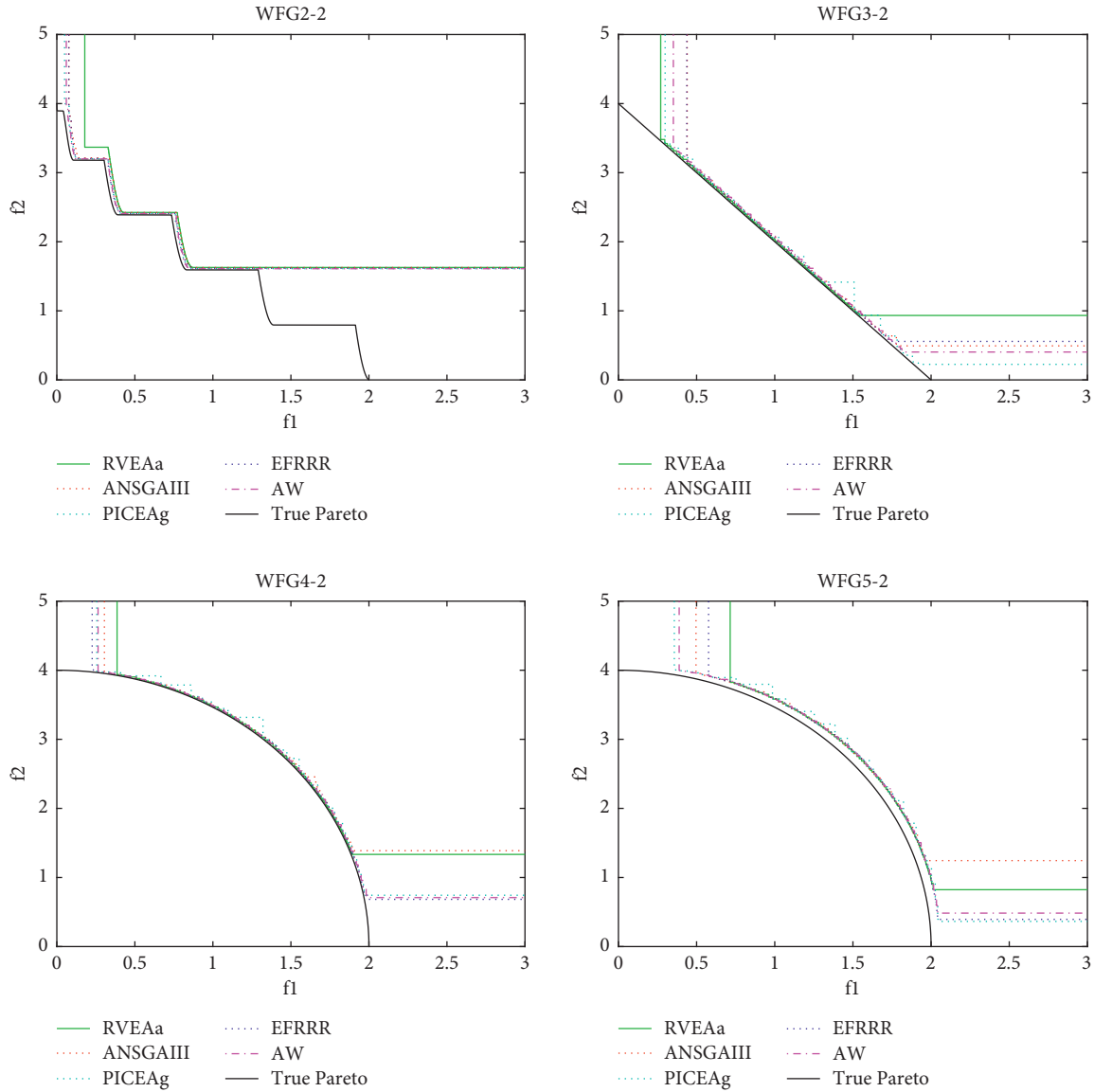


FIGURE 2: Median attainment surfaces of the 2-objective WFG2 to WFG5.

extent while PICEA-g has a certain degree of volatility, and the convergence is slightly worse. In terms of diversity performance, PICEA-g and MOEA/D-AW are superior to other algorithms.

In all, our proposed MOEA/D-AW and PICEA-g are superior to other algorithms in terms of diversity in most cases, while MOEA/D-AW is superior to PICEA-g in terms of convergence in most cases.

5.2. Statistical Results. Since all algorithms are essentially random search methods, it is difficult to determine whether an algorithm is good or bad based on the results of a single operation. Therefore, in this section, the average statistics of the results of 31 independent operations are compared with respect to different performance metrics. The best performing algorithms in each of these problems are given in bold in Tables 2 to 5.

As shown in Table 2, PICEA-g is slightly superior to other algorithms in terms of HV metric on most WFGs. MOEA/D-AW performs well compared to other algorithms in the 4-objective and 7-objective WFGs, except for being slightly inferior to EFR-RR and PICEA-g.

As shown in Table 3, PICEA-g is superior to other algorithms in terms of IGD metric on most 4- and 7-objective WFGs. The average IGD metric index obtained by MOEA/D-AW on WFG2-7 is superior to other algorithms, and it performs poorly on other test problems. Based on the comparison results in Table 2, the average IGD metric indexes obtained by MOEA/D-AW on most WFGs are mainly limited by its convergence, and it reveals that MOEA/D-AW is superior to other algorithms in terms of diversity.

As shown in Table 4, MOEA/D-AW is superior to other algorithms on WFG3-4, WFG3-7, WFG4-4, WFG4-7, WFG5-7, and WFG6-4, while RVEAa is superior to

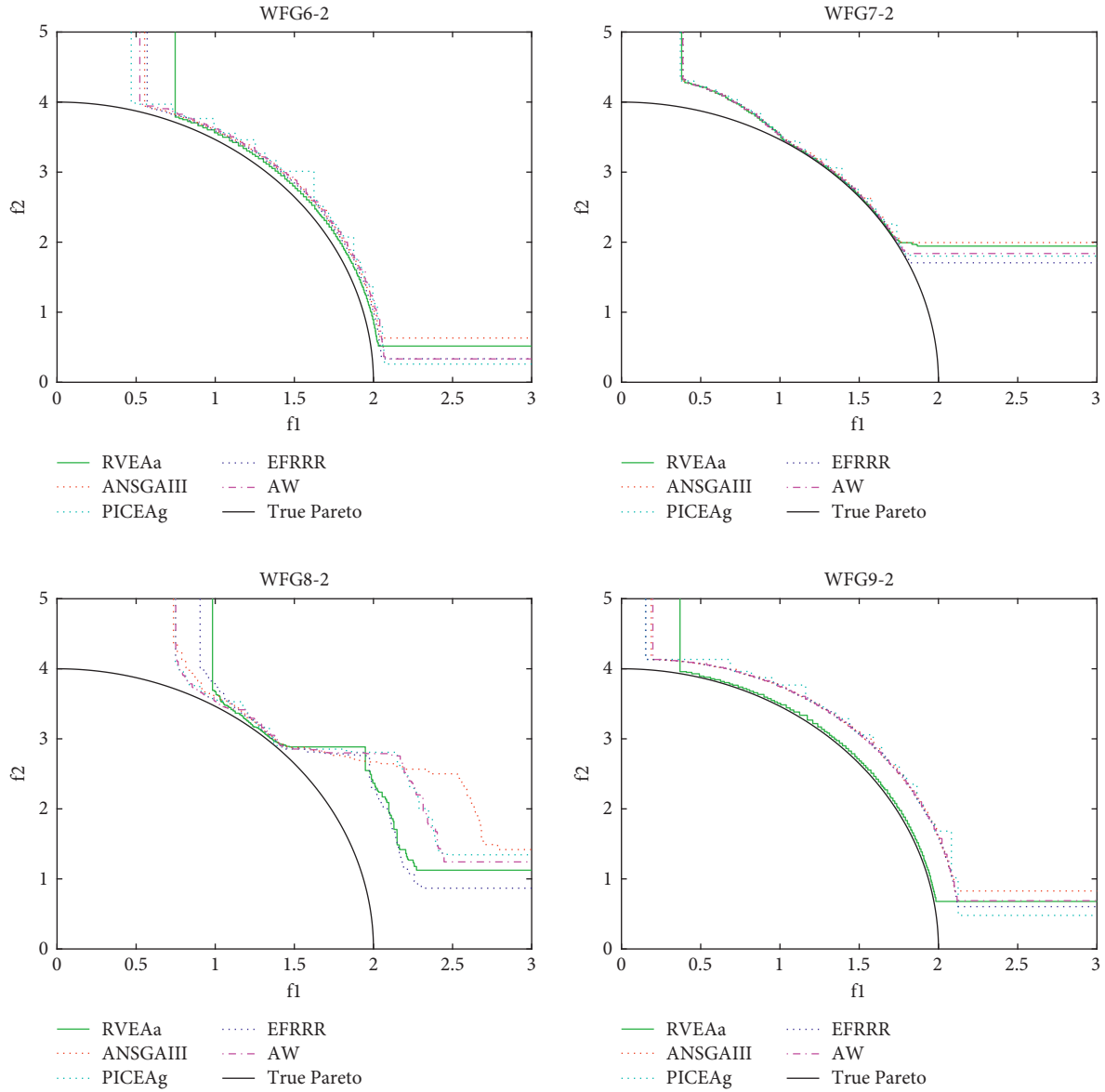


FIGURE 3: Median attainment surfaces of the 2-objective WFG6 to WFG9.

TABLE 2: The statistical results (mean) of the HV values obtained by 5 algorithms on WFG2 to WFG9.

	Obj	ANSGAIII	RVEAa	PICEAg	EFRRR	MOEA/D-AW
WFG2	2	0.568387	0.550097	0.573903	0.564387	0.570258
	4	0.793419	0.764000	0.805484	0.819484	0.808258
	7	0.874677	0.835290	0.884032	0.840258	0.899903
WFG3	2	0.571097	0.545677	0.599936	0.591290	0.586484
	4	0.328194	0.203710	0.364419	0.329484	0.183065
	7	0.089355	0.000000	0.249419	0.054839	0.000000
WFG4	2	0.374065	0.384000	0.403097	0.410258	0.401710
	4	0.563903	0.619419	0.701194	0.718032	0.703032
	7	0.819936	0.713194	0.667677	0.808936	0.823839
WFG5	2	0.358323	0.358161	0.380000	0.376032	0.368903
	4	0.599968	0.577742	0.695419	0.697936	0.670677
	7	0.790548	0.652871	0.736645	0.762677	0.782097

TABLE 2: Continued.

	Obj	ANSGAIII	RVEAa	PICEAg	EFRRR	MOEA/D-AW
WFG6	2	0.352774	0.351419	0.377226	0.372161	0.370839
	4	0.613484	0.601355	0.692323	0.693065	0.664129
	7	0.794323	0.694419	0.866516	0.799742	0.786710
WFG7	2	0.327968	0.334581	0.345548	0.349742	0.343419
	4	0.606581	0.615129	0.744097	0.706194	0.716194
	7	0.845871	0.718581	0.820613	0.766000	0.872710
WFG8	2	0.263581	0.245129	0.254903	0.265387	0.274645
	4	0.510323	0.459290	0.591032	0.550452	0.570258
	7	0.733419	0.607516	0.773774	0.653710	0.706839
WFG9	2	0.356129	0.368161	0.361839	0.371710	0.365677
	4	0.571419	0.582323	0.599936	0.594645	0.608484
	7	0.680774	0.657742	0.699677	0.653581	0.703258

The best results are given in bold.

TABLE 3: The statistical results (mean) of the IGD values obtained by 5 algorithms on WFG2 to WFG9.

	Obj	ANSGAIII	RVEAa	PICEAg	EFRRR	MOEA/D-AW
WFG2	2	0.027742	0.056307	0.024923	0.075312	0.027174
	4	0.318926	0.319134	0.244438	0.397874	0.265584
	7	2.478501	2.573738	3.055701	2.932787	2.236648
WFG3	2	0.022826	0.029178	0.020397	0.016903	0.028081
	4	0.706847	0.618760	0.018508	0.226327	1.491903
	7	0.793979	1.865569	0.008584	0.482924	3.875040
WFG4	2	0.011243	0.006770	0.010859	0.007660	0.011409
	4	0.144125	0.207221	0.119248	0.126419	0.158724
	7	0.968558	1.121720	0.552663	0.801419	1.055519
WFG5	2	0.068020	0.064636	0.064953	0.064900	0.068315
	4	0.246305	0.211099	0.153042	0.156273	0.178305
	7	0.991691	1.170598	0.727007	0.876354	1.117266
WFG6	2	0.078930	0.071754	0.075731	0.076946	0.077206
	4	0.294787	0.269174	0.170637	0.164501	0.214040
	7	0.911231	1.239905	0.596021	0.712028	1.083449
WFG7	2	0.096721	0.066667	0.096891	0.086643	0.088512
	4	0.183409	0.185135	0.113988	0.145567	0.142149
	7	0.900902	1.247910	0.576098	1.049238	0.963121
WFG8	2	0.228603	0.267558	0.239812	0.292829	0.244698
	4	0.441809	0.634984	0.224257	0.593517	0.342018
	7	1.103588	1.792134	0.572861	1.508154	1.435946
WFG9	2	0.155992	0.116460	0.124575	0.123171	0.126931
	4	0.300874	0.270670	0.234286	0.246548	0.248221
	7	1.370617	1.507547	1.093408	1.396443	1.326873

The best results are given in bold.

other algorithms on the other WFGs. The average PD metric indexes obtained by MOEA/D-AW on the WFGs that RVEAa performs best are not much different from that of RVEAa and are in the same order of magnitude. The obtained results reveal that MOEA/D-AW is superior to other outstanding algorithms in terms of diversity on most problems that are linear, convex, multimodal, separable, deceptive, and concave, which supports the regularities revealed by the obtained results from Tables 2 and 3 and complies to the motivation of the proposed algorithm.

As shown in Table 5, PICEA-g and RVEAa are superior to other algorithms in terms of convergence on most WFGs,

while MOEA/D-AW is barely superior to other algorithms on WFG2-4, WFG2-7, and WFG5-4. The gap between the average GD metric indexes obtained by MOEA/D-AW and the best algorithm on other WFGs is not big. The obtained results reveal that the proposed MOEA/D is inferior to other outstanding algorithms in terms of convergence in most cases.

5.3. Further Discussion. In this section, we will discuss why MOEA/D-AW performs badly compared with those outstanding MOEAs in terms of convergence on many occasions.

TABLE 4: The statistical results (mean) of the PD values obtained by 5 algorithms on WFG2 TO WFG9.

	Obj	ANSGAIII	RVEAa	PICEAg	EFRRR	MOEA/D-AW
WFG2	2	2.59E-01	9.97E-01	9.93E-03	1.14E+00	2.41E-01
	4	1.53E+04	1.38E+04	5.36E+02	2.86E+03	5.70E+03
	7	1.03E+06	2.84E+06	2.84E+04	7.31E+05	1.54E+06
WFG3	2	3.11E+00	5.96E+00	9.31E-02	8.16E+00	3.72E+00
	4	1.25E+03	1.18E+04	2.14E+01	2.54E+01	2.12E+04
	7	2.30E+05	6.21E+06	2.18E+02	6.22E+04	9.84E+06
WFG4	2	3.04E+00	6.17E+00	2.98E-02	3.22E+00	3.09E+00
	4	1.14E+04	4.37E+04	1.27E+03	6.25E+03	4.55E+04
	7	4.73E+06	1.35E+07	1.59E+03	4.45E+06	1.41E+07
WFG5	2	3.61E+00	5.60E+00	3.25E-02	3.79E+00	3.30E+00
	4	3.47E+04	4.45E+04	1.23E+03	1.16E+04	3.94E+04
	7	9.21E+06	1.52E+07	1.92E+04	6.22E+06	1.53E+07
WFG6	2	3.21E+00	4.92E+00	4.68E-02	3.75E+00	2.49E+00
	4	3.23E+04	4.37E+04	7.15E+02	2.28E+03	4.43E+04
	7	8.07E+06	1.32E+07	2.87E+04	1.78E+06	1.23E+07
WFG7	2	2.75E+00	8.02E+00	2.16E-02	4.16E+00	2.20E+00
	4	6.31E+03	5.33E+04	7.31E+02	3.39E+03	4.46E+04
	7	1.14E+06	2.09E+07	3.30E+04	5.70E+06	9.43E+06
WFG8	2	3.82E-01	2.56E+00	3.05E-02	1.70E+00	7.31E-01
	4	4.91E+03	5.51E+04	4.64E+02	5.49E+03	3.99E+04
	7	1.25E+06	2.06E+07	3.49E+04	4.41E+06	1.72E+07
WFG9	2	2.50E+00	5.78E+00	2.17E-02	3.48E+00	3.02E+00
	4	4.97E+04	5.98E+04	1.01E+03	5.65E+03	4.45E+04
	7	7.87E+06	2.28E+07	9.12E+04	5.56E+06	1.35E+07

The best results are given in bold.

TABLE 5: The statistical results (mean) of the GD values obtained by 5 algorithms on WFG2 to WFG9.

	Obj	ANSGAIII	RVEAa	PICEAg	EFRRR	MOEA/D-AW
WFG2	2	0.002806	0.006392	0.002599	0.010613	0.002742
	4	0.033624	0.032956	0.025543	0.056684	0.021916
	7	0.183074	0.189915	0.230406	0.331688	0.161590
WFG3	2	0.002307	0.002955	0.002062	0.001708	0.002825
	4	0.089108	0.062784	0.002019	0.033096	0.126013
	7	0.066543	0.138259	0.000780	0.056480	0.261087
WFG4	2	0.001170	0.000721	0.001142	0.000805	0.001196
	4	0.013677	0.019292	0.011413	0.014473	0.012318
	7	0.070404	0.082590	0.043364	0.089228	0.070061
WFG5	2	0.006819	0.006484	0.006526	0.006517	0.006854
	4	0.022822	0.019645	0.014288	0.017428	0.013727
	7	0.070445	0.084546	0.053274	0.094565	0.073117
WFG6	2	0.007920	0.007206	0.007612	0.007738	0.007751
	4	0.027430	0.024996	0.015878	0.018369	0.016379
	7	0.065424	0.089358	0.044954	0.078487	0.071123
WFG7	2	0.014487	0.012465	0.015652	0.014600	0.014396
	4	0.017162	0.017502	0.011000	0.017954	0.011135
	7	0.065801	0.090326	0.045114	0.116701	0.064698
WFG8	2	0.031439	0.037681	0.034324	0.035284	0.031517
	4	0.054451	0.064436	0.027970	0.077218	0.031472
	7	0.080463	0.128387	0.044472	0.170342	0.095627
WFG9	2	0.016028	0.012483	0.013274	0.013520	0.013388
	4	0.027862	0.025002	0.021662	0.027273	0.018952
	7	0.096764	0.106072	0.078735	0.150570	0.085949

The best results are given in bold.

One major reason is about setting the crossover and mutation operators in the algorithm. It can be seen from the obtained results that the SBX operator could generate converged offspring in terms of 2-objective WFGs in the decomposition-based framework, MOEA/D while degenerating in terms of high dimension occasions (WFG4s and WFG7s). The competitors involved in the experiment essentially generate the offspring under the dominance-based framework, which could perform well in terms of convergence in most cases while degenerating in terms of diversity. The DE operator, which is superior in terms of convergence, has been adopted to solve complex problems and is validated to be efficient [30]. In this paper, for the sake of fairness, all algorithms adopt the simulated binary crossover (SBX) and polynomial mutation to generate offspring.

6. Conclusion

In this paper, MOEA/D-AW is proposed to enhance the diversity of decomposition-based MOEAs. The basic idea is to devise a coevolved adaptive weight vector generation mechanism to adjust the evolution during the iteration. The weight vector is updated based on the sparsity of the mapping of the current population in the objective space, deletion of crowded weight vectors, and addition of sparse weight vectors are conducted. Moreover, the mechanism makes full use of the convergence information of the current population, the current utopian point, to generate optimal weight vectors to update the evolution direction.

In the experimental studies, MOEA/D-AW is compared to 4 state-of-the-art MOEAs which are proven to effectively enhance diversity. 8 test instances with up to 7 objectives from WFG2 to WFG9 test suits are introduced for comparison. The results indicate that MOEA/D-AW is superior to other competitor MOEAs in terms of diversity on most test problems while shows a slight disadvantage over other competitors MOEAs in terms of convergence on many objective test problems.

In further studies, many efforts would be carried to devise a more efficient mechanism that would adaptively select different operators based on the spread of the mapping of current solutions in the objective space during the iteration. Also, it has been proven that different operators show different performances in exploration and exploitation in EAs [48].

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by funds from Sichuan Science and Technology Program under contract no. 2019ZDZX0001.

References

- [1] K. Deb, K. Sindhya, and J. Hakanen, "Multiobjective optimization," in *Decision Sciences: Theory and Practice* CRC Press, Boca Raton, FL, USA, 2016.
- [2] T. Xiahou, Y. Liu, and Q. Zhang, "Multiobjective redundancy allocation for multi-state system design under epistemic uncertainty of component states," *Journal of Mechanical Design*, vol. 142, no. 11, 2020.
- [3] G.-Z. Fu, H.-Z. Huang, Y.-F. Li, and J. Zhou, "An adaptive hybrid evolutionary algorithm and its application in aero-engine maintenance scheduling problem," *Soft Computing*, vol. 25, no. 8, pp. 6527–6538, 2021.
- [4] G.-Z. Fu, H.-Z. Huang, Y.-F. Li, B. Zheng, and T. Jin, "Multiobjective design optimization for a two-stage transmission system under heavy load condition," *Mechanism and Machine Theory*, vol. 122, pp. 308–325, 2018.
- [5] G.-Z. Fu, Y.-F. Li, Y. Tao, and H.-Z. Huang, "An interactive preference-based evolutionary algorithm for multi-criteria satisficing optimization," *Journal of Intelligent and Fuzzy Systems*, vol. 34, no. 4, pp. 2503–2511, 2018.
- [6] H. Li, Z.-M. Deng, N. A. Golilarz, and C. Guedes Soares, "Reliability analysis of the main drive system of a CNC machine tool including early failures," *Reliability Engineering & System Safety*, vol. 215, Article ID 107846, 2021.
- [7] H. Li, A. P. Teixeira, and C. Guedes Soares, "A two-stage Failure Mode and Effect Analysis of offshore wind turbines," *Renewable Energy*, vol. 162, pp. 1438–1461, 2020.
- [8] H. Li, C. Guedes Soares, and H.-Z. Huang, "Reliability analysis of a floating offshore wind turbine using Bayesian networks," *Ocean Engineering*, vol. 217, Article ID 107827, 2020.
- [9] H. Li, H. Diaz, and C. Guedes Soares, "A failure analysis of floating offshore wind turbines using AHP-FMEA methodology," *Ocean Engineering*, vol. 234, Article ID 109261, 2021.
- [10] H. Li, H. Diaz, and C. Guedes Soares, "A developed failure mode and effect analysis for floating offshore wind turbine support structures," *Renewable Energy*, vol. 164, pp. 133–145, 2021.
- [11] H. Li, H.-Z. Huang, Y.-F. Li, J. Zhou, and J. Mi, "Physics of failure-based reliability prediction of turbine blades using multi-source information fusion," *Applied Soft Computing*, vol. 72, pp. 624–635, 2018.
- [12] K. Miettinen and M. M. Mäkelä, "On scalarizing functions in multiobjective optimization," *OR Spectrum*, vol. 24, no. 2, pp. 193–213, 2002.
- [13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Longman Publishing, Boston, MA, USA, 1989.
- [14] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation discussion and generalization," in *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423, San Francisco, CA, USA, June 1993.
- [15] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," in *Proceedings of the EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., pp. 95–100, Athens, Greece, September 2001.
- [16] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *Proceedings of the International Conference on Parallel*

- Problem Solving from Nature*, pp. 292–301, Amsterdam, Netherlands, September 1998.
- [17] 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.
 - [18] K. Ikeda, H. Kita, and S. Kobayashi, “Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?” in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, vol. 2, pp. 957–962, Seoul, Republic of Korea, May 2001.
 - [19] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, “Combining convergence and diversity in evolutionary multiobjective optimization,” *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.
 - [20] K. Deb, M. Mohan, and S. Mishra, “Evaluating the ϵ -domination based multiobjective evolutionary algorithm for a quick computation of Pareto-optimal solutions,” *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.
 - [21] S. Yang, M. Li, X. Liu, and J. Zheng, “A grid-based evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.
 - [22] H. Li and D. Landa Silva, “Evolutionary multiobjective Simulated Annealing with adaptive and competitive search direction,” in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3311–3318, Hong Kong, China, June 2008.
 - [23] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, “A preference-based evolutionary algorithm for multiobjective optimization,” *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, 2009.
 - [24] K. Deb and A. Kumar, “Interactive evolutionary multi-objective optimization and decision-making using reference direction method,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, London, UK, July 2007.
 - [25] N. Beume, B. Naujoks, and M. Emmerich, “SMS-EMOA: multiobjective selection based on dominated hypervolume,” *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.
 - [26] J. Bader and E. Zitzler, “HypE: an algorithm for fast hypervolume-based many-objective optimization,” *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
 - [27] L. While, L. Bradstreet, and L. Barone, “A fast way of calculating exact hypervolumes,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, 2012.
 - [28] 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.
 - [29] H. Ishibuchi and T. Murata, “A multiobjective genetic local search algorithm and its application to flowshop scheduling,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 3, pp. 392–403, 1998.
 - [30] 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.
 - [31] I. Das and J. E. Dennis, “Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems,” *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
 - [32] 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*, vol. 18, no. 4, pp. 577–601, 2014.
 - [33] A. Rodríguez Sánchez, A. Ponsich, A. L. Jaimes, and S. Z. Martínez, “A parallel tabu search heuristic to approximate uniform designs for reference set based MOEAs,” in *Evolutionary Multi-Criterion Optimization*, pp. 254–265, Springer International Publishing, Berlin, Germany, 2019.
 - [34] H. Jain and K. Deb, “An Evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: handling constraints and extending to an adaptive approach,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
 - [35] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, “MOEA/D with adaptive weight adjustment,” *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.
 - [36] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, “A reference vector guided evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
 - [37] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, “Generalized decomposition,” in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, pp. 428–442, Sheffield, UK, March 2013.
 - [38] R. Wang, R. C. Purshouse, and P. J. Fleming, “Preference-inspired co-evolutionary algorithms using weight vectors,” *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.
 - [39] M. Li and X. Yao, “What weights work for you? adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation,” *Evolutionary Computation*, vol. 28, no. 2, pp. 227–253, 2020.
 - [40] S. Das and P. N. Suganthan, “Differential evolution: a survey of the state-of-the-art,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
 - [41] S. Huband, L. Barone, L. While, and P. Hingston, “A scalable multiobjective test problem toolkit,” in *Proceedings of the Third international conference on Evolutionary Multi-Criterion Optimization*, pp. 280–295, Guanajuato, Mexico, March 2005.
 - [42] C. M. Fonseca and P. J. Fleming, “An overview of evolutionary algorithms in multiobjective optimization,” *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.
 - [43] P. Czyżżak and A. Jaszkiewicz, “Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization,” *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–47, 1998.
 - [44] D. A. Van Veldhuizen and G. B. Lamont, *Evolutionary Computation and Convergence to a Pareto Front*, Stanford University, Stanford, CA, USA, 1998.
 - [45] H. Wang, Y. Jin, and X. Yao, “Diversity assessment in many-objective optimization,” *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1510–1522, 2017.
 - [46] R. Wang, R. C. Purshouse, and P. J. Fleming, “Preference-inspired coevolutionary algorithms for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2013.
 - [47] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, “Balancing convergence and diversity in decomposition-based many-objective optimizers,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 180–198, 2016.
 - [48] M. Črepinšek, S.-H. Liu, and M. Mernik, “Exploration and exploitation in evolutionary algorithms: a survey,” *ACM Computing Surveys*, vol. 45, no. 3, pp. 1–33, 2013.