Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 2491796, 17 pages
https://doi.org/10.1155/2021/2491796

Research Article

Adaptive Strategies Based on Differential Evolutionary Algorithm for Many-Objective Optimization

Yifei Sun,[1] **Kun Bian**,[1] **Zhuo Liu**,[1] **Xin Sun**,[1] **and Ruoxia Yao**[2]

[1] *School of Physics and Information Technology, Shaanxi Normal University, Xi'an 710119, China*
[2] *School of Computer Science, Shaanxi Normal University, Xi'an 710119, China*

Correspondence should be addressed to Yifei Sun; yifeis@snnu.edu.cn

Received 18 August 2021; Accepted 29 October 2021; Published 18 November 2021

Academic Editor: Rui Wang

The decomposition-based algorithm, for example, multiobjective evolutionary algorithm based on decomposition (MOEA/*D*), has been proved effective and useful in a variety of multiobjective optimization problems (MOPs). On the basis of MOEA/*D*, the MOEA/D-DE replaces the simulated binary crossover (SBX) operator with differential evolution (DE) operator, which is used to enhance the diversity of the solutions more effectively. However, the amplification factor and the crossover probability are fixed in MOEA/D-DE, which would lead to a low convergence rate and be more likely to fall into local optimum. To overcome such a prematurity problem, this paper proposes three different adaptive operators in DE with crossover probability and amplification factors to adjust the parameter settings adaptively. We incorporate these three adaptive operators in MOEA/D-DE and MOEA/D-PaS to solve MOPs and many-objective optimization problems (MaOPs), respectively. This paper also designs a sensitive experiment for the changeable parameter η in the proposed adaptive operators to explore how η would affect the convergence of the proposed algorithms. These adaptive algorithms are tested on many benchmark problems, including ZDT, DTLZ, WFG, and MaF test suites. The experimental results illustrate that the three proposed adaptive algorithms have better performance on most benchmark problems.

1. Introduction

In fields like industrial production and scientific research, the solutions for many practical problems are considered as a type of multiobjective optimization according to many researches. And there are many challenges in MOPs, which means that there is still room for improvement. An MOP, which is the main objective in this paper, is illustrated as follows:

$$\begin{cases} \min F(x) = \left(f_1(x), \ldots, f_m(x)\right), \\ \text{subject to } x \in \Omega, \end{cases} \tag{1}$$

where $(x_1, \ldots x_n)$ is a decision vector from the search space Ω (n is the number of decision variables) and $f_1(x), \ldots, f_m(x)$ are m objective functions. As these objectives conflict with one another, the algorithm will generate no single optimal solution, which is the output of single-objective optimization, but a group of solutions under the restriction of the balance of the m objective functions $f_1(x), \ldots, f_m(x)$, which means that any amelioration in one objective will impair at least one other objective. Such one group of solutions is Pareto optimal solutions (PS). The image of PS in the objective space is defined as the Pareto optimal front (PF). Decision-makers can select the probable solutions from a set of PF [1].

The effectiveness of solving MOPs by multiobjective evolutionary algorithms (MOEAs) has been demonstrated. These MOEAs can be classified into three categories by various selection ways [2]: (1) decomposition-based algorithms, for example, multiobjective evolutionary algorithms based decomposition (MOEA/*D*) [3] and multiple single-objective Pareto sampling (MSOPS) [4]; (2) Pareto-dominance-based algorithms, for example, the nondominated sorting genetic algorithm (NSGA-II) [5], multiobjective genetic algorithm (MOGA) [6], strength Pareto evolutionary algorithm (SPEA) [7], SPEA2 [8], niched Pareto genetic algorithm (NPGA) [9], the Pareto envelope-based selection algorithm for multiobjective optimization (PESA) [10], and PESA-II [11]; (3) performance indicator-based algorithms, for example, approximated hypervolume-based evolutionary algorithm (HypE) [12] and indicator-based evolutionary algorithm (IBEA) [13]. Decomposition-based MOEAs are becoming one of the most popular algorithms during these years. It is an essential way to combine decomposition with scalarization in conventional multiobjective optimization. MOEA/*D*, a representative decomposition-based algorithm, uses a scalarizing approach to divide an MOP into many subproblems with different weights. It also uses the coefficient method based on population search to solve these subproblems [14]. Li and Zhang [15] designed a DE operator in MOEA/D-DE to solve variant MOPs. Baatar et al. [16] designed an adaptive parameter in A-NRDE to solve MOPs. Recently, Zhan et al. [17] proposed an adaptive distributed differential evolution (ADDE) to tackle the difficulties of strategies' selections and parameters' settings. And Wang [18] used a niching method in AED in solving some optimization problems.

Many real-world problems, which contain more than three objectives [19–22], are usually named many-objective optimization problems (MaOPs). Compared with MOPs, MaOPs usually have more complex PFs and need higher performance requirements for algorithms. Because of the loss of selectivity [23], the traditional MOEAs have degradation in solving MaOPs [24]. For decomposition-based MOEAs, specifying a set of weight vectors in a high-dimensional target space is difficult, and its performance depends heavily on the consistency of the weight vectors and the shape of the PF, while for Pareto-dominated algorithms, it is difficult to provide efficient selectivity to the PFs when dealing with a variety of obtained solutions [18]. As for indicator-based algorithms, they usually need a lot of computational resources. To handle these problems, a lot of many-objective evolutionary algorithms (MaOEAs) were proposed for solving MaOPs in the few decades. Depending on the strategy for handling convergence enhancement and diversity maintenance, they can be generally divided into three classes [25].

The first class involves decomposition-based algorithms. MOEA/D-DD [26] combines dominance and decomposition-based strategies to solve MaOPs. MOEA/D-CRU used the chain-reaction solution update strategy to improve the diversity of the solutions. In MOEA/D-PaS [27], a Pareto adaptive scalarizing method was proposed to approximate the optimal value. In MOEA/D-LWS [14], a weighted sum method was applied in a local manner. The second class is the Pareto-based algorithms. Reference [28] proposed an ensemble fitness ranking method to balance the convergence and diversity in solving MaOPs. In [29], a shift-based density estimation method in SPEA2-SDE was proposed to reduce the loss of selection pressure. SPEA/R used a reference direction-based density estimator to solve MOPs and MaOPs. The third category involves indicator-based algorithms. In [30–32], several methods have been proposed to calculate HV in a more efficient way. Regarding the MaOPs with a variety of objectives, other performance targets, R2 [33, 34], Two_Arch2 [35], and SRA [36], were proposed. In recent years, there have been many new algorithms proposed to solve MaOPs. Liang [37] proposed a two-round selection strategy to generate good solutions between population diversity and convergence. Ma et al. [38] designed an adaptive localized decision variable analysis approach to solve MaOPs. A bottleneck objective learning strategy was proposed by Liu et al. to balance the diversity and convergence [39]. Zhang et al. proposed a DECAL algorithm to increase the diversity of the population for solving unconstrained MaOPs [40]. Ma et al. [41] proposed an adaptive reference vector reinforcement learning approach to decomposition-based algorithms for industrial copper burdening optimization. An orthogonal learning framework was proposed by Ma et al. [42] to improve the learning mechanism of brain storm optimization for solving complex problems.

In this study, we design three strategies, the linear variation, power function transformation, and the exponential transformation methods, to adjust the crossover probability and amplification factor in DE adaptively, and we incorporate these three adaptive operators in MOEA/*D* to solve MOPs and MOEA/D-PaS to solve MaOPs. We run these algorithms on ZDT, DTLZ, WFG, and MaF test functions. The experimental results illustrate that these proposed methods have advantages on most of the test functions.

There is the organization of this paper: we expound some basic knowledge in Section 2 and elaborate on three adaptive algorithms in Section 3. Experimental research and results analysis are detailed in Section 4. The conclusions of this paper and some future works are presented in Section 5.

2. Background

2.1. Basic Definitions. There are some basic definitions in multiobjective problems described as follows.

Definition 1. If y is Pareto dominated by x, then denoted as $x \prec y$, if and only if $\forall i \in \{1, 2, \ldots, m\}$, $f_i(x) \leq f_i(y)$ and $f_j(x) < f_j(y)$, there is at least one index $j \in \{1, 2, \ldots, m\}$.

Definition 2. A solution $x^* \in \Omega$ is Pareto optimal if and only if $\nexists x \in \Omega$ such that $x \prec x^*$.

Definition 3. The Pareto set (PS) is described as the set of all Pareto optimal solutions. The set of all Pareto optimal vectors, $\mathrm{PF} = \{F(x) \in R^m | x \in PS\}$, is called PF.

2.2. MOEA/D. An MOP can be divided into many subproblems in the form of single-objective optimization by MOEA/*D*. Every subproblem is optimized by different weighted aggregation [43]. The neighborhood of each subproblem is dependent on several weighted vectors which are close to the subproblem in distance. MOEA/*D* adopts a group of N uniformly distributed weight vectors, where N represents the number of subproblems. With the information of the neighborhood, every subproblem is optimized at the same time. There are many different editions of MOEA/

D. Our algorithms are based upon MOEA/D-DE, which is an enhanced edition of MOEA/*D*. We adopt the Tchebycheff method to aggregate the function:

$$\min g^{te}(x|\lambda, z^*) = \max_{1\le i\le m}\{\lambda_i|f_i(x) - z_i^*\}, \qquad (2)$$

where z^* represents the ideal point which is the point with minimum value in the i th objective. More details can be found in [3, 15].

2.3. Differential Evolutionary Algorithms. Kenneth Price and Rainer Storn proposed a variety of variation forms of differential evolution (DE) algorithms [44, 45]. We describe the different DE algorithms as a form of DE/X/Y/*Z*.

X represents the selection of the basis vector (the individual vector to be mutated) in the mutation operation, "rand" represents an individual that is chosen at random from the race, and "best" represents the individual that has the best performance. Y represents the number of different vectors. *Z* stands for crossover, and the binomial experiment described as "bino" is usually used for crossover operation. There are some frequently used algorithms described as follows:

$$\begin{aligned}
&\frac{\text{DE/rand}}{1/\text{bin}}\text{:}\ v_i = x_{r_1} + F\times\left(x_{r_2} - x_{r_3}\right),\\
&\frac{\text{DE/best}}{1/\text{bin}}\text{:}\ v_i = x_{\text{best}} + F\times\left(x_{r_2} - x_{r_3}\right),\\
&\frac{\text{DE/rand}}{2/\text{bin}}\text{:}\ v_i = x_{r_1} + F\times\left(x_{r_2} - x_{r_3}\right) + F\times\left(x_{r_4} - x_{r_5}\right),\\
&\frac{\text{DE/best}}{2/\text{bin}}v_i = x_{\text{best}} + F\times\left(x_{r_2} - x_{r_3}\right) + F\times\left(x_{r_4} - x_{r_5}\right),
\end{aligned} \qquad (3)$$

where r_1, r_2, r_3, r_4, and r_5 are randomly selected distinct integers from the set $\{1, 2, \ldots, N\}$ and r_{best} is the individual that has the best performance. F is an amplification factor, which expands the different vector.

2.4. DE in MOEA/D. Li and Zhang [15] replaced SBX [46] operator in MOEA/*D* with the DE/rand/1/bin operator and proposed MOEA/D-DE. The algorithm adopts three randomly selected individuals, r_1, r_2, and r_3, to generate the new solution from the neighborhood P:

$$\overset{,}{y}_k = \begin{cases} x_k^{r_1} + F\times\left(x_k^{r_2} - x_k^{r_3}\right), & \text{rand} < \text{CR},\\ x_k^{r_1}, & \text{rand} > \text{CR}, \end{cases} \qquad (4)$$

where CR is the parameter that controls the rate of crossover, F represents the amplification factor, and rand represents a random number whose numerical value is between 0 and 1.

The polynomial mutation in DE is described as follows:

$$\begin{aligned}
y_k &= \begin{cases} \overset{,}{y}_k + \delta_k\times(b_k - a_k), & \text{with probability } p_m,\\ \overset{,}{y}_k & \text{with probability } 1 - p_m, \end{cases}\\
\delta_k &= \begin{cases} (2\times\text{rand})^{1/\omega+1} - 1, & \text{rand} < 0.5,\\ 1 - (2 - 2\times\text{rand})^{1/\omega+1}, & \text{otherwise}, \end{cases}
\end{aligned} \qquad (5)$$

where the distribution index ω and the mutation probability p_m are two parameters in the algorithm. a_k represents the lower boundary and b_k is the upper boundary.

2.5. MOEA/D-PaS. In the last few decades, many MOEAs have demonstrated their effectiveness in solving MOPs and MaOPs. Among these MOEAs, decomposition-based algorithms use evenly distributed vectors to keep the population diversity [37]. MOEA/D-PaS was proposed by Wang [27], using the Pareto adaptive scalarizing method to maximize the searchability of the algorithms and enhance the robustness to the PF.

In MOEA/*D*, a variety of decomposition scalarizing methods can be used to solve MOPs. A weighted scalarizing method can be described as follows:

$$g^{w\,d}(x|w, p) = \left(\sum_{i=1}^{m}\lambda_i\left(f_i(x) - z_i^*\right)^p\right)^{1/p}, \quad p\ge 1. \qquad (6)$$

When $p = 1$, the above formula represents the weighted sum method, and when $p = \infty$, the formula is the Tchebycheff method [27]. The different values can affect the search speed in the objective space. The MOEA/D-PaS uses a set of p values to select a suitable scalarizing method to find the optimal solutions. More details can be found in [27]. The framework of the adaptive scalarizing method is described as follows.

3. Proposed Algorithms

3.1. Adaptive Strategies. The contrast of other algorithms and MOEA/D-DE shows MOEA/D-DE lower time complexity and fast convergence speed [47]. However, there are weak points such as rough race dispersion and inefficient local search capability of the race in MOEA/D-DE. Besides, the amplification factor and the crossover probability are fixed in MOEA/D-DE, which would lead to a low convergence rate and be more likely to fall into local optimum. To overcome these shortcomings, we design three adaptive operators in MOEA/D-DE.

In DE, we encode the crossover probability (*CR*), and the amplification factor (*F*) evolved with the increase of iterations. The traditional differential evolutionary (DE) algorithm is to keep *F* and *CR* fixed in value, which will converge slowly and be difficult to search for the global optimal solution as a result of premature convergence [47]. So, we design three adaptive operators to adjust the values of *F* and *CR* dynamically.

```
    **Input**: weight vector: $w$, $p$ suite: $P = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \infty\}$, The current generation: $gen$, The maximum generation: maxGen.
**Output**: The best $p$ value for subproblem.
(1)  if rand > gen/maxGen
(2)      for each $p^k$ in $P$ do
(3)          $z^k \leftarrow \arg\min (g^{w\ d}(x|w, p^k))$
(4)      End
(5)      $z \leftarrow \arg\min \mathrm{dist}(z^k, w)$
(6)      Set $p$ to be the smallest $p^k$ with $z = z^k$
(7)  end
```

Algorithm 1: Adaptive scalarizing method.

The first method uses a linear variation named MOEA/D-DE-LAD:

$$\mathrm{CR} = \mathrm{CR}_0 + \eta * \left(\frac{\mathrm{gen}}{\max \mathrm{Gen}}\right),$$
$$F = F_0 - \eta * \left(\frac{\mathrm{gen}}{\max \mathrm{Gen}}\right). \tag{7}$$

The second method uses power function transformation called MOEA/D-DE-PAD:

$$\mathrm{CR} = \mathrm{CR}_0 + \eta * \left(\frac{\mathrm{gen}}{\max \mathrm{Gen}}\right)^2,$$
$$F = F_0 - \eta * \left(\frac{\mathrm{gen}}{\max \mathrm{Gen}}\right)^2. \tag{8}$$

The third method uses exponential transformation named MOEA/D-DE-EAD:

$$\mathrm{CR} = \eta^{(1-\ \mathrm{gen}/\max \mathrm{Gen})},$$
$$F = \eta^{(\mathrm{gen}/\max \mathrm{Gen})}. \tag{9}$$

Compared with the traditional DE, these three methods use different strategies to adjust the values of *CR* and *F* dynamically. The values of *CR* and *F* will change by the generations. η is an artificial parameter changed from 0.1 and 0.2 to 0.9 with step 0.1. We will discuss how the values of η would influence the three proposed algorithms in the following text. Algorithm 1 describes the situation that uses the LAD adaptive strategy to generate the new solution. While using the PAD strategy, we can replace lines 2 and 3 with $\mathrm{CR} = \mathrm{CR}_0 + \eta * (\mathrm{gen}/\max \mathrm{Gen})^2$ and $F = F_0 - \eta * (\mathrm{gen}/\max \mathrm{Gen})^2$. In the same case, when using the EAD strategy, we replace lines 2 and 3 with $\mathrm{CR} = \eta^{(1-\mathrm{gen}/\max \mathrm{Gen})}$ and $F = \eta^{(\mathrm{gen}/\max \mathrm{Gen})}$.

The impressions of η on the proposed algorithms will be discussed in the following text in detail. When $\eta = 0.5$, the values of *CR* and *F* among the three strategies are described as follows.

As shown in Figure 1, the three methods use different strategies to adjust the values of *CR* and *F*. The values of *CR* are increased by the generations. On the contrary, the values of *F* are decreased. When $\eta = 0.5$, the values of *CR* in MOEA/D-DE-LAD are always bigger than the other two methods, and the values of *F* in MOEA/D-DE-EAD are the minimum among three methods during the evolutionary.

3.2. Adaptive Strategies in MOEA/D. We use the different adaptive DE operators to generate the new solutions in MOEA/*D*. *CR* and *F* will change by the increase of generations so as to accelerate the pace of converging to the optimal solution. More details can be found from the following text.

3.3. Adaptive Strategies in MOEA/D-PaS. To enhance the performance of MOEA/D-PaS for MaOPs, we incorporated the adaptive DE methods proposed above into the MOEA/D-PaS and proposed MOEA/D-PaS-LAD denoted as PaS-LAD, MOEA/D-PaS-PAD denoted as PaS-PAD, and MOEA/D-PaS-EAD denoted as PaS-EAD. We use the adaptive DE operators in MOEA/D-PaS to generate the new solutions.

The framework of MOEA/D-PaS is based on MOEA/*D*, and in the initialization stage (line 1 in Algorithm 3), set the p value suite $p = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \infty\}$. And in every generation, update the p value using Algorithm 1 to select the suitable p value. The adaptive scalarizing method in MOEA/D-PaS can select the suitable p value to enhance searchability. The adaptive DE operators can generate better solutions according to the generation. In the following text, we will discuss the advantages of adaptive operators through the experiments.

4. Experimental Study and Results Analysis

4.1. For MOPs. To identify the effectiveness of these operators in solving MOPs and MaOPs, we run these algorithms on 2- and 3-objective test problems. The details can be found as follows.

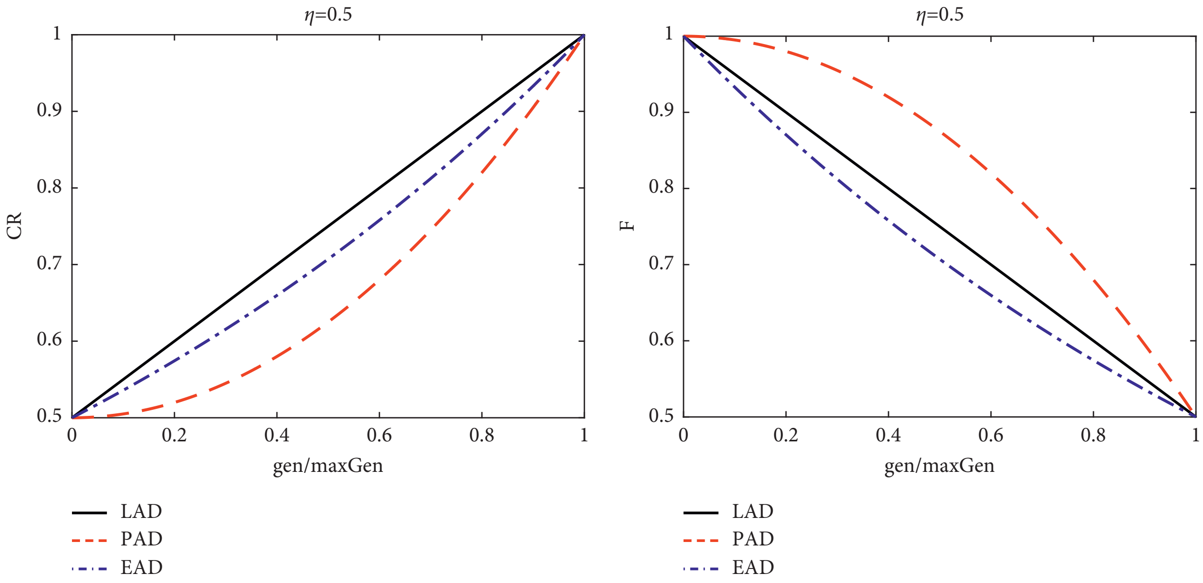
4.1.1. Benchmark Problems. In this section, on the purpose of testing the performance of the proposed algorithms in solving multiobjective optimization problems, the ZDT [48] test function set including ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 and the DTLZ [49] test function set including DTLZ1, DTLZ2, DTLZ3, and DTLZ4 are adopted in experiments.

4.1.2. Parameter Settings. The settings of the parameters in four algorithms are described in Table 1.

Each algorithm is run for 30 times on each test problem; each run is for 30000 function evaluations for the 2-objective problem and 60000 evaluations for the 3-objective problem.

```
Input: Current generation: $gen$, Maximum generation: $maxGen$, Three individuals: $r_1$, $r_2$, $r_3$
Output: the new solution $y'$
Using LAD strategy:
    $\mathrm{CR} = \mathrm{CR}_0 + \eta * (\mathrm{gen}/\max\mathrm{Gen})$
    $F = F_0 - \eta * (\mathrm{gen}/\max\mathrm{Gen})$
Calculate the values of $CR$ and $F$ according to the chosen adaptive strategy
Generate a new solution $y'$ :$y' = \begin{cases} x_{r_1} + F \times (x_{r_2} - x_{r_3}), & \mathrm{rand} < \mathrm{CR}, \\ x_{r_1}, & \mathrm{rand} > \mathrm{CR}, \end{cases}$
```

Algorithm 2: Adaptive strategies.



(a)

(b)

Figure 1: The values of (a) CR and (b) F.

F and *CR* are fixed parameters set in MOEA/D-DE. F_0 is the initial value of *F*, and CR_0 is the initial value of *CR*. The probability of mutation and its corresponding distribution index are set as $p_m = 1/n$ and $\eta_m = 20$.

All the experiments are tested on a computer (AMD Ryzen 5–4600H CPU (3.0 GHz) 16G RAM Windows 10 systems).

4.1.3. Performance Metrics.

(1) *IGD Metric*. The reverse generation distance (IGD) [50] is adopted to evaluate the quality of one solution set *P* in the experiments set in this paper. Assuming that P^* is the true Pareto front, *P* is the practical Pareto front found by an algorithm. The distance between P^* and *P* is defined as follows:

$$\mathrm{IGD}(P^*, P) = \frac{\sum_{v \in p^*} d(v, P)}{|P^*|}, \tag{10}$$

where $d(v, P)$ is the minimum distance between points *v* and *P*. The algorithms with solutions of smaller values of IGD will be considered to have better performance.

(2) *HV Metric*. The HV metric evaluates the MOEA performance by calculating the supervolume value of the space between the nondominant solution set and the reference point. The HV metric can be defined as

$$\mathrm{HV} = \lambda\left(\cup_{i=1}^{|S|} v_i\right), \tag{11}$$

where λ stands for the Lebesgue measure, which is used to measure volume. v_i represents the supervolume of reference points and nondominant individuals. $|S|$ is the number of nondominant solution sets. The value of HV can comprehensively reflect the convergence and distribution breadth of the solution set. The solutions with higher values of HV are closer to the global Pareto optimal solution. And the algorithms with such solutions are considered to have better performance.

4.1.4. Computational Complexity. Calculation of the adaptive operators in MOEA/D-DE runs at $O(N \times N_w)$, where *N* is the number of solutions. N_w is the number of weighted vectors, and the number of weighted vectors is the same as solutions. So the time complexity is $O(N^2)$.

```
Input: The number of populations: $N$, the number of the weight vectors from neighborhood: $T$, and the maximum generation: maxGen.
Output: Optimal solutions.
Initialization:
Generate $N$ weight vectors $\lambda^1, \lambda^2, \ldots, \lambda^N$ from objective space evenly
Calculate the distances between every pair of the weighted vectors. On the basis of the distances, find $T$ closest weight vectors $\lambda^{i_1}, \ldots, \lambda^{i_T}$ from $\lambda^i$ as the neighborhood of $\lambda^i$. For every $i = 1, \ldots, N$, there are $B(i) = (i_1, \ldots, i_T)$
Initialize $z = (z_1, \ldots, z_m)'$, where $z_j = \min f_j(x^i)$ is the value which is minimum from objective space
while gen < maxGen do
  for $i \leftarrow 1$ to $N$ do
    if $rand < \delta$
      $Q \leftarrow B(i)$
    else
      $Q \leftarrow S$
    end
    Randomly select three solutions $x^{r_1}, x^{r_2}, x^{r_3}$ from mating pool $Q$
    Generate a new solution $y'$ by the adaptive DE operators in Algorithm 2
    If a dimension in $y$ is beyond the search boundary, instead it of the value which is chosen from the value which is within the boundary at random
    Update the ideal point $z$: for each $j = 1, \ldots, m$, if $z_j > f_j(y)$, then set $z_j = f_j(y)$
    Update the solutions: if $g(y|\lambda^j, z) \leq g(x^j|\lambda^j, z)$, then set $x^j = y$
  end
  Find the nondominated solutions during every generation
  gen $\leftarrow$ gen + 1
end
Output these nondominated solutions as optimal solutions
```

Algorithm 3: Adaptive strategies in MOEA/*D*.

Table 1: The settings of parameters.

Parameter	Value	Description
N-2	100	2-objective
N-3	200	3-objective
F	0.5	Amplification factor
CR	0.9	Crossover rate
F_0	1.0	Initial value of F
CR_0	0.5	Initial value of CR

4.1.5. Experiments Analysis. In this paper, the three proposed algorithms, MOEA/D-DE-LAD denoted as LAD, MOEA/D-DE-PAD denoted as PAD, and MOEA/D-DE-EAD denoted as EAD, are compared with MOEA/D-DE. In every test function, the four algorithms are set to run thirty times independently on the same computer.

We firstly set different values of η to see how IGD and HV would change, and then we run these proposed algorithms on more test functions with probable η compared with MOEA/D-DE to see which one performs better on these test functions. The probable value of η was decided by the IGD and HV values that most algorithms get best on these test functions. More details can be found as follows.

To explore the influence of the change of η parameter on the values of IGD and HV, we set $\eta = [0.1, 0.2, \ldots, 0.9]$ and see how IGD and HV would change on the 2-objective test functions and 3-objective test functions. The values of η will influence the values of *CR* and *F* changed by generations so as to get different values of IGD and HV. The results of related experiments are shown in the following figures.

From Figure 2, we can see that

(1) On ZDT1 and ZDT2, the IGD values of LAD and PAD increase as η increases. However, LAD has a sharper increase than PAD. On these two test problems, PAD performs best among the three algorithms. When $\eta = 0.1$, the IGD values of LAD and PAD are minimum. The values of IGD in EAD are minimum when $\eta = 0.9$. For these 2-objective problems, PAD keeps a lower value than the other two algorithms.

(2) On test function DTLZ1 and DTLZ2 test problem, there are no obvious relationships between IGD and η. When $\eta = 0.7$, EAD has the best performance in comparison compared with LAD and PAD. For 3-objective problems, there is a much more complex possibility than the 2-objective problem. The IGD

values are more susceptible to the values of η. We need other methods to test the algorithms.

(3) The IGD values of EAD are complicated and changeable, and there are no obvious patterns compared with LAD and PAD. Moreover, in 2-objective test problems, LAD and PAD changed more regularly than in 3-objective problems.

From Figure 3, we could see that

(1) The values of HV in LAD and PAD decrease as η increases on ZDT1, ZDT2, and DTLZ1 test functions. On DTLZ2, the overall trends of LAD and PAD are downward. When $\eta = 0.1$, the HV values of LAD and PAD are maximum. LAD and PAD changed more regularly than EAD on these test functions.

(2) On ZDT1 and ZDT2, the values of HV in EAD are flexible; there is no obvious pattern on the 2-objective test problems. When $\eta = 0.9$, the values of HV are the maximum.

(3) On test function DTLZ1 and DTLZ2 test problem, the HV values of EAD are downward as η changes. The HV values are maximum when η is 0.1. Moreover, in 3-objective test problems, EAD changed more regularly than in 2-objective problems.

The values of *CR* and *F* are controlled by the following parameters: CR_0, F_0, *gen*, *maxGen*, and η. CR_0, F_0, and *maxGen* are the initial parameters. The different values of η can reflect the speed of change of *CR* and *F* values, which would affect the values of IGD and HV.

From the figures shown in this paper, the values of IGD and HV would dynamically change by η. These algorithms perform better on those test functions with $\eta = 0.1$, so we set $\eta = 0.1$ in these proposed algorithms. We run these algorithms and MOEA/D-DE on ZDT series test problems. The performance of the four algorithms on 2-objective test functions is shown in the figure. In the following text, the true PF is the real Pareto front of the test functions. According to the comparison with the true PF, we can inform that which algorithm performs better among these proposed algorithms.

As shown in Figure 4, we enlarge some areas to explore which algorithm has the better performance through comparing the distance between the true PF and the PF of the algorithm. The smaller distance between the PF and the algorithm is, the better performance the algorithm has. In the enlarged area, the horizontal axis represents the value of $f1$, and the vertical axis is the value of $f2$. Compared with MOEA/D-DE, the proposed algorithms have better performance on these test functions. On ZDT1, PAD almost converges to the PF completely. LAD has the best convergence performance on ZDT2. On ZDT3, the proposed algorithms are approximate. Besides, EAD has the best performance among the four algorithms on ZDT4. The PFs of the three proposed algorithms are much closer than MOEA/D-DE to the real PF. On these 2-objective test problems, using adaptive strategies can make the distance between the PF of the algorithms and the real PF much closer.

For the 2-objective problems, we made the figures which obtain the comparison of the true PF and the PF of the four algorithms, and we can distinguish which algorithm is better from the figures easily. However, for 3-objective problems, it is uneasy to recognize which performs better from the comparisons of the four algorithms in one figure. Besides, the performance metrics can analyze the performance of the algorithm from various aspects. So we select IGD and HV as criteria to judge which performs better on these test functions.

To further explore the performance of these four algorithms on the test problems, we made the statistics of IGD and HV in the tables.

It can be seen from Table 2 that

(1) Compared with other algorithms, PAD performs better on the ZDT1, ZDT2, ZDT4, DTLZ2, DTLZ3, and DTLZ4 test problems. PAD has huge advantages over MOEA/D-DE on these test functions except ZDT6 and DTLZ1.

(2) On test functions ZDT3 and DTLZ1, EAD performs best. LAD and PAD perform better than MOEA/D-DE. EAD has tiny advantages over LAD and PAD.

(3) MOEA/D-DE has the best performance on the ZDT6 test problem. The IGD values of LAD and PAD are close to MOEA/D-DE on ZDT6. The IGD value of EAD is the biggest among these four algorithms.

(4) Compared with MOEA/D-DE, the proposed algorithms have better performance on most test functions. Besides, PAD has better advantages over LAD and EAD on six test functions and only performs worse on three of nine functions. Still, on these test problems which PAD has not the best performance, PAD has more advantages over MOEA/D-DE except on ZDT6.

From Table 3, the following results are observed:

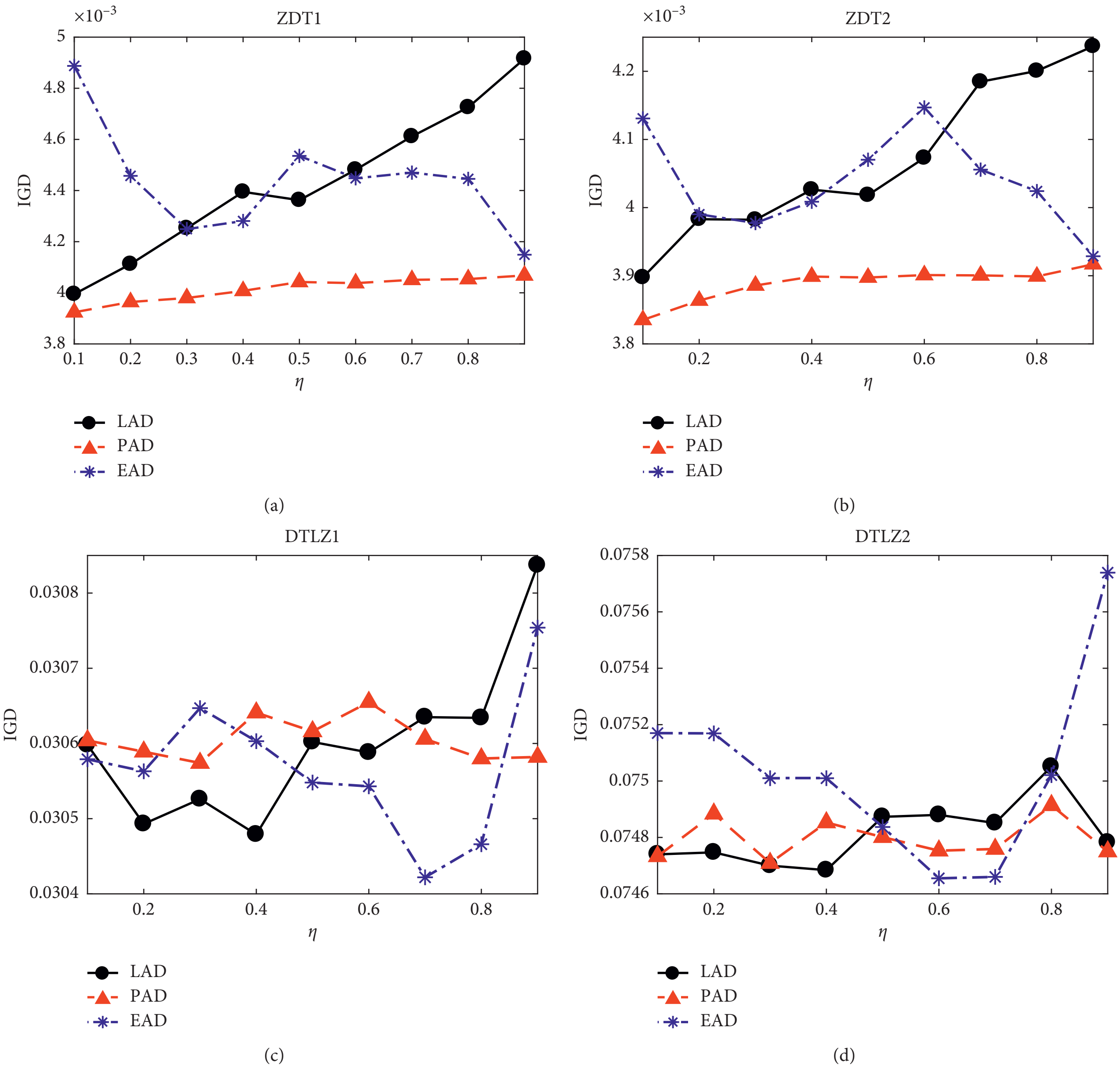
(1) PAD performs best on ZDT1, ZDT2, ZDT4, DTLZ1, and DTLZ3 test problems and has huge advantages over MOEA/D-DE on these test problems.

(2) Compared with the proposed algorithms, MOEA/D-DE has the best performance on ZDT6. However, the HV values of LAD and PAD are so closed to MOEA/D-DE.

(3) On DTLZ2 and DTLZ4 test problems, LAD has the maximum HV values. And on ZDT3, EAD performs best.

(4) These three proposed algorithms have better performance on most test functions except on the ZDT6 test problem. Moreover, PAD has better advantages over LAD and EAD on five of nine test problems.

From the two tables shown above, the proposed three algorithms have advantages on most of the test functions. MOEA/D-DE only has the best values of IGD and HV on ZDT6; however, LAD and PAD are close to it on ZDT6.



(a) (b) (c) (d)

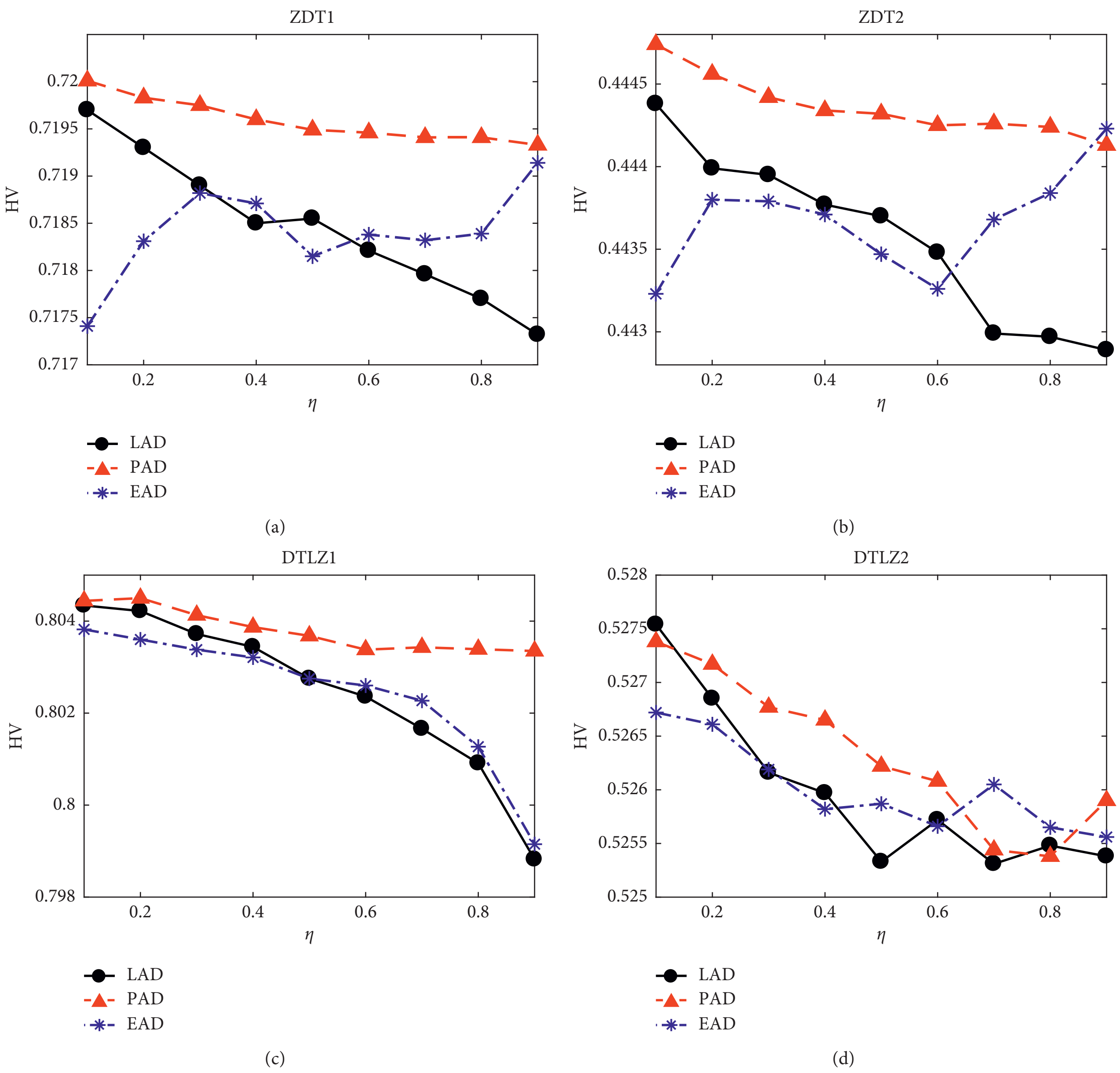
Figure 2: The values of IGD changed by η.

From the results shown in the tables, PAD has advantages over LAD and EAD, and these proposed adaptive strategies are effective on these test problems.

4.2. For MaOPs. For MaOPs, we run these algorithms on some 4-, 7-, and 10-objective test functions to explore which method can help MOEA/D-PaS get more nondominated solutions.

4.2.1. Benchmark Functions and Performance Measures. We use two test suites, WFG test functions WFG1-WFG8 [51] and MaF test problems MaF1-MaF8 [52], to test the performance of the algorithms. WFG test suite is a classic test problem with different scaled objectives for MaOPs and is wildly used in [18, 25, 38, 40, 53–55]. WFG test set has different PF shapes, which obtain nonseparable disconnected and biased PFs. And for MaF, it is a new test suite with complicated PFs and is more challenging to the MOEAs. For these test problems, each test function is tested for 4-, 7-, and 10-objective instances. The settings of other parameters can be found in the table.

The hypervolume (HV) has brilliant theoretical qualities [56–58], so we use the values of HV to access the performance of the proposed algorithms. The HV value can reflect the quality of the solutions by calculating the volume of a region in the objective space bounded between a nondominant solution set and a reference point. A larger HV value means that the obtained solutions set is closer to the true PF.



(a)

(b)

(c)

(d)

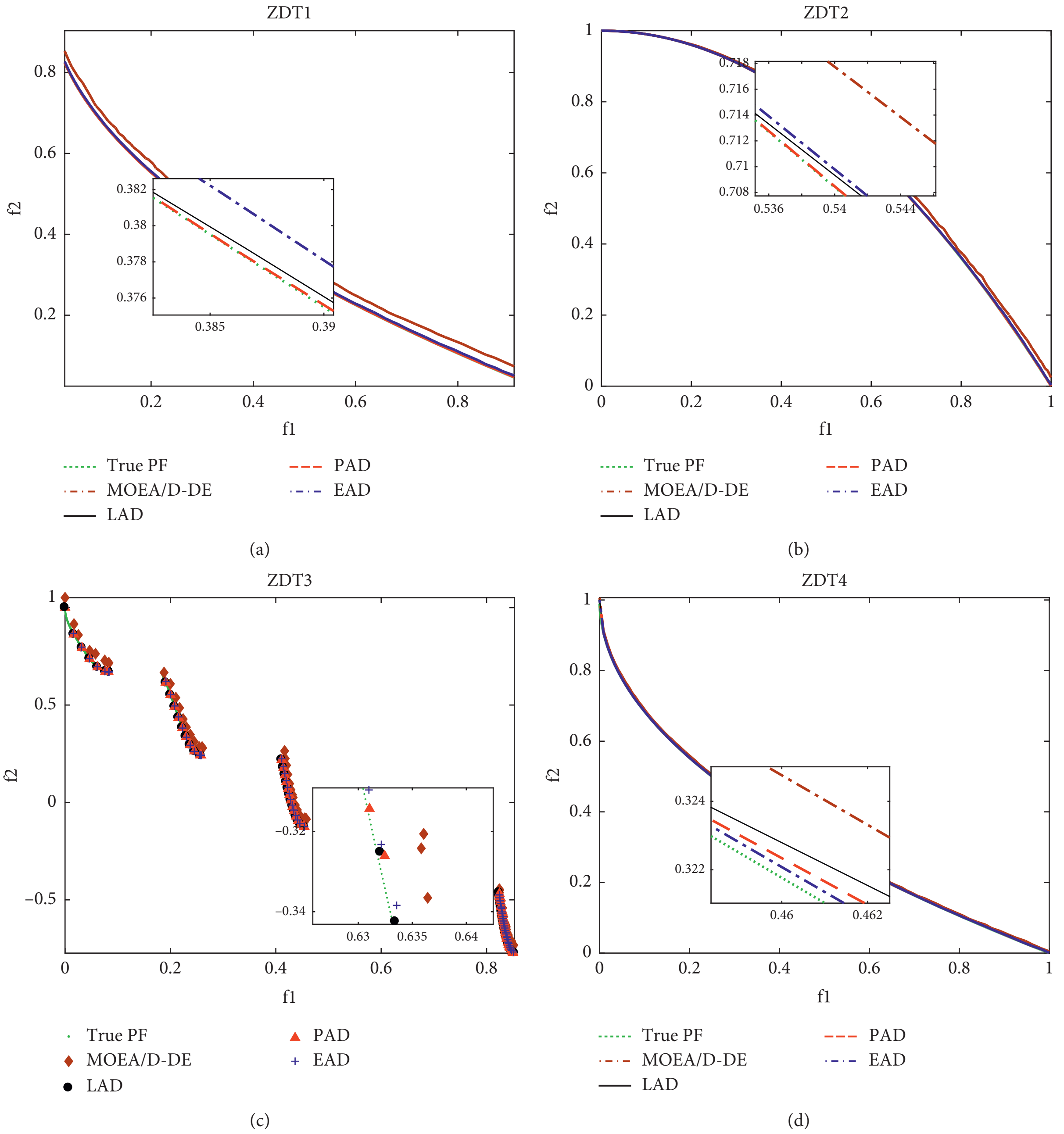
Figure 3: The values of HV changed by η.

4.2.2. Parameter Settings. We run these algorithms on WFG and MaF test suites to explore which adaptive DE operators perform better for the MaOPs. The value of η is 0.1. For a fair comparison, the population size is set to 200 for four objective problems, 240 for seven objective problems, and 280 for ten objective problems. The evaluations are set at 100000, 168000, and 196000 for 4-, 7-, and 10-objective test functions. All the algorithms are implemented in PlatEMO [59]. To make the experiment results more convincing, every algorithm is run thirty times on each test function independently. All the experimental results are shown in the following tables and figures.

4.2.3. Computational Complexity. Calculation of the adaptive operators in MOEA/D-PaS runs at $O(N \times N_W \times N_p)$, where N is the number of solutions. N_w is the number of weighted vectors, and the number of weighted vectors is the same as solutions. N_p is the number of p values. So the time complexity is $O(N^2 \times N_p)$.

4.2.4. Experiments Analysis. Table 4 shows the comparison of HV values in four algorithms on WFG1-8 test problems. PaS-PAD has the best results in 12 out of 24 cases, while PaS-LAD, PaS-EAD, and MOEA/D- PaS perform best in 6, 4, and 2 cases, which indicate the superiority of PaS-PAD on these WFG problems. The algorithms with adaptive DE operators have obvious advantages over MOEA/D-PaS on the WFG test set.

For the four and seven objective problems, MOEA/D-PaS performs worst among the four algorithms. But with the dimension increases, for ten objective test problems,



(a)

(b)

(c)

(d)

Figure 4: Attainment surfaces for (a) ZDT1, (b) ZDT2, (c) ZDT3, and (d) ZDT4.

Table 2: IGD metric (mean/std) results.

Problem	MOEA/D-DE	MOEA/D-de-LAD	MOEA/D-de-PAD	MOEA/D-de-EAD
ZDT1	1.3148e − 2 (3.39e − 3)	3.9948e − 3 (4.49e − 5)	**3.9238 e − 3 (2.09 e − 5)**	4.8875e − 3 (3.08e − 4)
ZDT2	8.5105e − 3 (2.11e − 3)	3.8976e − 3 (3.99e − 5)	**3.8352 e − 3 (1.22 e − 5)**	4.1308e − 3 (1.06e − 4)
ZDT3	1.7491e − 2 (4.73e − 3)	1.0911e − 2 (3.45e − 5)	1.0912e − 2 (2.85e − 5)	**1.0799 e − 2 (7.83 e − 5)**
ZDT4	1.8778e − 1 (1.58e − 1)	8.1790e − 3 (4.15e − 3)	**5.6242 e − 3 (5.83 e − 4)**	1.4027e − 2 (1.93e − 2)
ZDT6	**3.1147e − 3 (1.62 e − 5)**	3.1189e − 3 (1.66e − 5)	3.1220e − 3 (1.52e − 5)	8.5842e − 3 (1.80e − 2)
DTLZ1	3.0854e − 2 (5.15e − 4)	3.0598e − 2 (1.41e − 4)	3.0604e − 2 (1.58e − 4)	**3.0579 e − 2 (1.26 e − 4)**
DTLZ2	7.5061e − 2 (6.13e − 4)	7.4740e − 2 (3.27e − 4)	**7.4733 e − 2 (3.14 e − 4)**	7.5170e − 2 (4.48e − 4)
DTLZ3	2.6452e − 1 (4.20e − 1)	7.5338e − 2 (1.41e − 3)	**7.4498 e − 2 (7.52 e − 4)**	4.1221e − 1 (5.96e − 1)
DTLZ4	1.1675e − 1 (6.70e − 2)	7.5895e − 2 (8.73e − 4)	**7.5531 e − 2 (8.33 e − 4)**	7.7210e − 2 (6.94e − 3)

Table 3: HV metric (mean/std) results.

Problem	MOEA/D-DE	MOEA/D-de-LAD	MOEA/D-de-PAD	MOEA/D-de-EAD
ZDT1	7.0542e − 1 (4.39e − 3)	7.1970e − 1 (1.76e − 4)	**7.2001 e − 1 (9.40 e − 5)**	7.1741e − 1 (6.21e − 4)
ZDT2	4.3311e − 1 (4.38e − 3)	4.4438e − 1 (1.98e − 4)	**4.4474 e − 1 (8.20 e − 5)**	4.4323e − 1 (4.10e − 4)
ZDT3	5.9565e − 1 (6.14e − 3)	5.9803e − 1 (1.25e − 4)	5.9802e − 1 (1.09e − 4)	**5.9857 e − 1 (4.03 e − 4)**
ZDT4	5.0235e − 1 (1.57e − 1)	7.1262e − 1 (6.02e − 3)	**7.1646 e − 1 (9.37 e − 4)**	7.0274e − 1 (3.09e − 2)
ZDT6	**3.8868 e − 1 (3.53 e − 4)**	3.8865e − 1 (3.54e − 4)	3.8863e − 1 (3.26e − 4)	3.8139e − 1 (2.38e − 2)
DTLZ1	7.9954e − 1 (4.20e − 3)	8.0434e − 1 (4.25e − 4)	**8.0444 e − 1 (5.08 e − 4)**	8.0382e − 1 (5.68e − 4)
DTLZ2	5.2575e − 1 (1.21e − 3)	**5.2754 e − 1 (1.00 e − 3)**	5.2738e − 1 (1.09e − 3)	5.2672e − 1 (7.98e − 4)
DTLZ3	4.1994e − 1 (1.93e − 1)	5.2825e − 1 (2.98e − 3)	**5.2847 e − 1 (1.81 e − 3)**	3.6518e − 1 (2.27e − 1)
DTLZ4	5.2134e − 1 (2.15e − 2)	**5.3208 e − 1 (2.04 e − 3)**	5.3051e − 1 (1.92e − 3)	5.3067e − 1 (3.12e − 3)

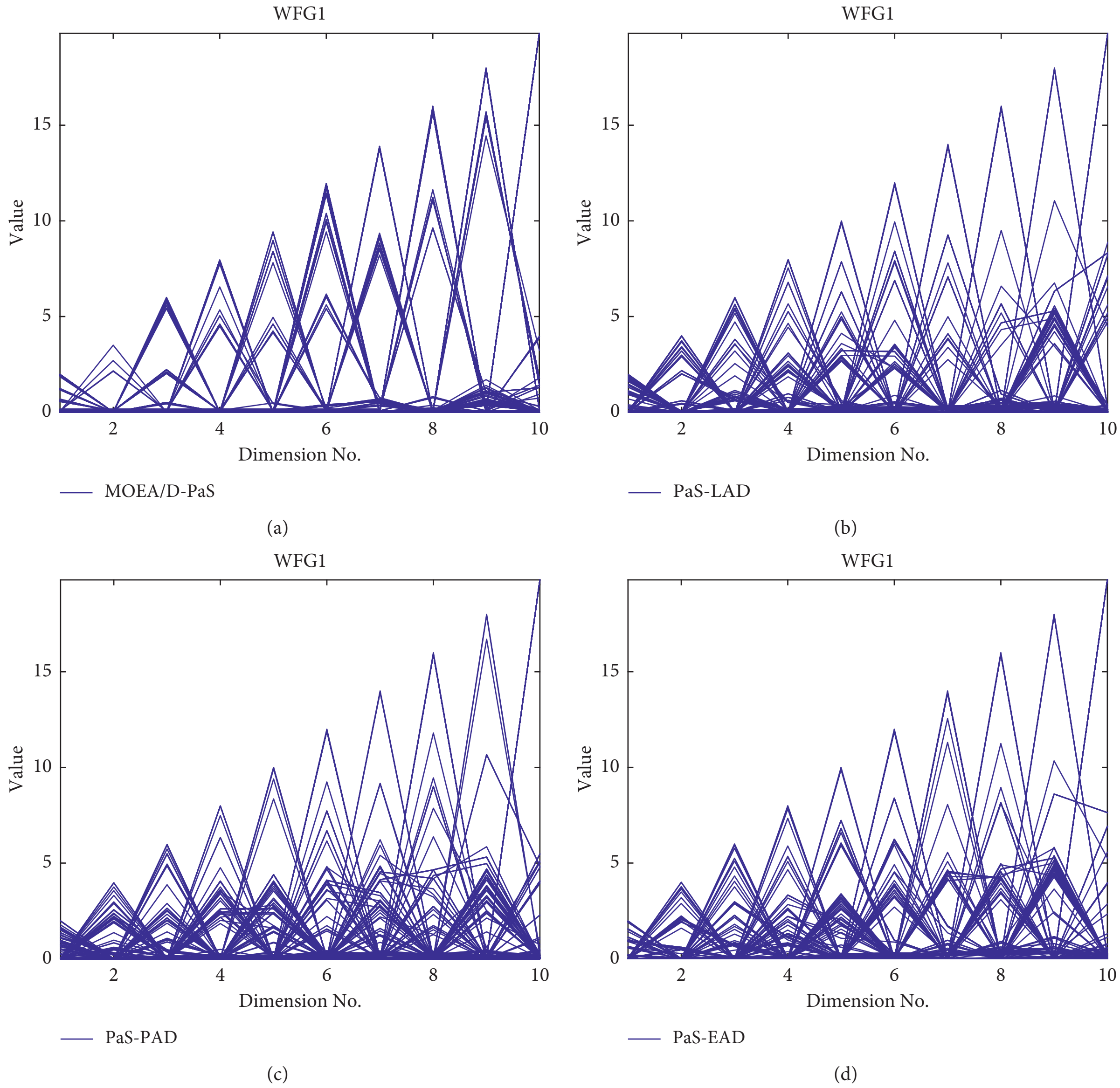
Table 4: HV metric (mean/std) comparison results on WFG.

Problem	MOEA/D-PaS	PaS-LAD	PaS-PAD	PaS-EAD
4-Objective				
WFG1	7.7750e − 1 (8.45e − 2)	9.8382e − 1 (1.04e − 2)	**9.8765 e − 1 (3.36 e − 3)**	9.3344e − 1 (6.70e − 2)
WFG2	9.6159e − 1 (6.95e − 3)	9.8842e − 1 (1.08e − 3)	9.8855e − 1 (9.68e − 4)	**9.8926 e − 1 (2.25 e − 3)**
WFG3	2.8583e − 1 (8.25e − 3)	**2.9166 e − 1 (7.53 e − 3)**	2.8348e − 1 (8.51e − 3)	2.8803e − 1 (7.76e − 3)
WFG4	6.1657e − 1 (1.30e − 2)	6.5520e − 1 (1.13e − 2)	**6.7087 e − 1 (6.20 e − 3)**	6.6817e − 1 (8.13e − 3)
WFG5	6.0973e − 1 (9.17e − 3)	6.1241e − 1 (1.18e − 2)	**6.1912 e − 1 (8.25 e − 3)**	6.1157e − 1 (1.09e − 2)
WFG6	5.8802e − 1 (5.50e − 2)	5.1429e − 1 (9.13e − 4)	5.1408e − 1 (1.14e − 3)	**6.1702 e − 1 (3.72 e − 2)**
WFG7	6.5290e − 1 (1.00e − 2)	**6.8584 e − 1 (2.13 e − 3)**	6.8536e − 1 (2.07e − 3)	6.8268e − 1 (2.28e − 3)
WFG8	4.9088e − 1 (1.53e − 2)	5.2406e − 1 (9.59e − 3)	**5.2891 e − 1 (6.16 e − 3)**	5.2782e − 1 (6.52e − 3)
7-Objective				
WFG1	9.8673e − 1 (2.32e − 2)	9.9962e − 1 (1.60e − 4)	**9.9973 e − 1 (1.43 e − 4)**	9.1178e − 1 (9.24e − 2)
WFG2	9.9647e − 1 (2.13e − 3)	**9.9995 e − 1 (7.00 e − 6)**	9.9995e − 1 (8.44e − 6)	9.9994e − 1 (6.03e − 5)
WFG3	1.3416e − 1 (3.23e − 2)	**1.6627 e − 1 (9.11 e − 3)**	1.6191e − 1 (1.21e − 2)	1.0693e − 1 (3.39e − 2)
WFG4	7.8348e − 1 (1.41e − 2)	8.6776e − 1 (8.94e − 3)	**8.8070 e − 1 (4.25 e − 3)**	8.7302e − 1 (7.85e − 3)
WFG5	7.2890e − 1 (1.26e − 2)	7.6889e − 1 (9.14e − 3)	**7.7767 e − 1 (8.78 e − 3)**	7.7617e − 1 (8.13e − 3)
WFG6	7.5305e − 1 (6.55e − 2)	6.8311e − 1 (1.28e − 3)	6.8345e − 1 (1.48e − 3)	**8.0685 e − 1 (4.69 e − 2)**
WFG7	8.0731e − 1 (2.62e − 2)	8.9357e − 1 (1.31e − 3)	**8.9410 e − 1 (1.57 e − 3)**	8.6221e − 1 (1.46e − 1)
WFG8	5.9494e − 1 (3.36e − 2)	**6.7355 e − 1 (1.85 e − 2)**	6.7035e − 1 (1.34e − 2)	6.4979e − 1 (7.45e − 2)
10-Objective				
WFG1	9.9493e − 1 (5.56e − 3)	9.9997e − 1 (2.22e − 5)	**9.9998 e − 1 (1.25 e − 5)**	9.7977e − 1 (5.21e − 2)
WFG2	2.3412e − 1 (2.19e − 1)	**1.0000e + 0 (1.28 e − 6)**	1.0000e + 0 (1.45e − 6)	1.7390e − 1 (1.48e − 1)
WFG3	**9.0889 e − 2 (4.25 e − 5)**	8.4467e − 2 (1.80e − 2)	7.7307e − 2 (2.63e − 2)	7.8137e − 2 (1.02e − 2)
WFG4	3.8941e − 1 (3.84e − 1)	9.3650e − 1 (4.64e − 3)	**9.4885 e − 1 (5.52 e − 3)**	2.1424e − 1 (2.32e − 1)
WFG5	9.0472e − 2 (3.13e − 2)	1.0183e − 1 (4.38e − 2)	9.5718e − 2 (3.96e − 2)	**1.5242 e − 1 (1.99 e − 1)**
WFG6	3.0037e − 1 (1.68e − 1)	7.4836e − 1 (2.98e − 4)	**7.4855 e − 1 (3.72 e − 4)**	9.4173e − 2 (2.73e − 2)
WFG7	1.1521e − 1 (6.55e − 2)	9.6635e − 1 (1.29e − 3)	**9.6725 e − 1 (1.41 e − 3)**	1.0418e − 1 (5.19e − 2)
WFG8	**9.8960 e − 2 (1.96 e − 2)**	9.2377e − 2 (6.28e − 3)	9.2756e − 2 (8.97e − 3)	9.3403e − 2 (8.65e − 3)
Best/all	2/24	6/24	12/24	4/24

MOEA/D-PaS has the best performance on WFG3 and WFG8. For WFG1 with a convex, biased, and mixed PF, PaS-PAD has the best performance in all cases. Regarding WFG2 with a disconnected PF, PaS-EAD is the best in four objectives, while the three proposed algorithms have close performance in seven objectives, and in ten objectives, PaS-LAD and PaS-PAD perform better than the other two algorithms. For the WFG3 test problem with linear PF, PaS-LAD has the best performance in four and seven objectives, while MOEA/D-PaS obtains the best result in ten objectives. Considering WFG4 with a convex PF, it is observed that PaS-PAD shows an advantageous performance over the other three algorithms. Meanwhile, PaS-PAD also shows its advantages on WFG5 in four and seven objectives, while in ten objectives, PaS-EAD has superior performance than other algorithms. PaS-EAD also has the best performance on WFG6 in four and seven objectives, while PaS-LAD and PaS-PAD perform better in ten objectives. Concerning WFG7 with a convex PF, PaS-PAD performs best in seven and ten objectives, and in four objectives, PaS-PAD has close performance with PaS-LAD. On the WFG8 test function, MOEA/D-PaS performs best in ten objectives, while PaS-PAD has superior performance in four and seven objectives.

To better visualize the distribution of the final population, we plot the final population of the four algorithms on the WFG1 test problem.

It can be seen from Figure 5 that, on the WFG1 test problem with flat bias and a mixed structure of the PF, the proposed algorithms with adaptive operators have better convergence and diversity compared to MOEA/D-PaS. Though MOEA/D-PaS can still achieve the value, it loses the diversity of the solutions. Among the proposed algorithms, the PaS-PAD has better distributed solutions than PaS-LAD and PaS-EAD.



(a)

(b)

(c)

(d)

FIGURE 5: Visualization of PF approximation by four algorithms on the 10-objective WFG1: (a) MOEA/D-PaS, (b) PaS-LAD, (c) PaS-PAD, and (d) PaS-EAD.

On the WFG test set, PaS-PAD has better performance overall. MOEA/D-PaS only has the best results in 2 out of 8 cases in ten objectives. PaS-LAD and PaS-EAD also have advantages on some test functions compared with MOEA/D-PaS.

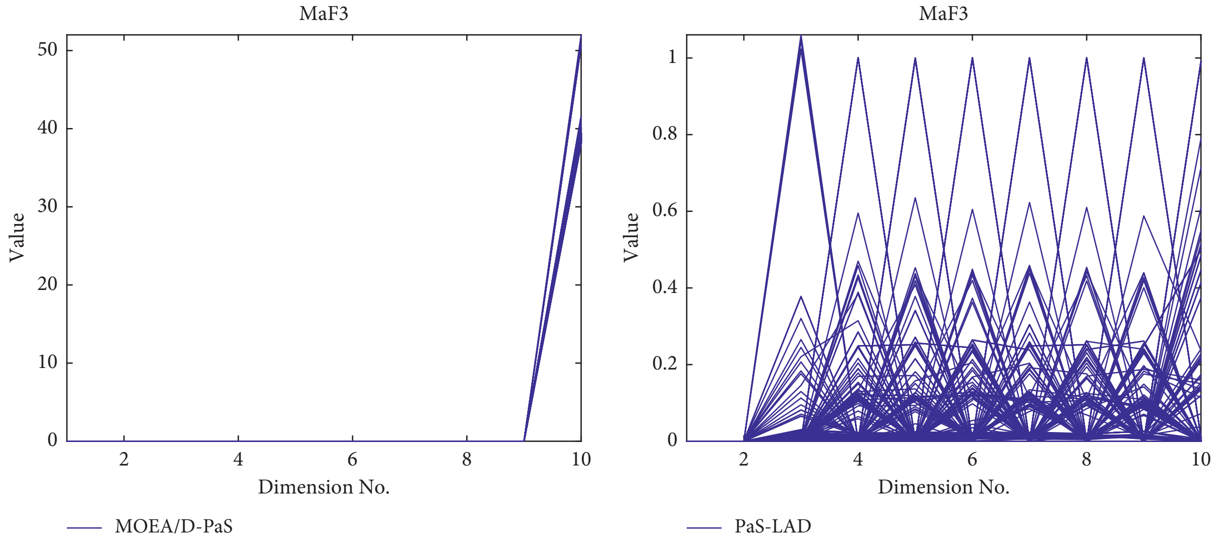
To further explore which adaptive operator has superior performance than its competitors, we also run these algorithms on a newly benchmark MaF test suite. The HV values of MaF1–MaF8 are shown in the table.

Table 5 collects the HV comparison results in the four algorithms on MaF1–MaF8 with 4-, 7-, and 10-objectives. PaS-PAD has the best performance in 9 out of 24 cases. MOEA/D-PaS, PaS-EAD, and PaS-LAD perform best in 6, 5, and 4 out of 24 cases. Compared with the other three algorithms, PaS-PAD has superior performance.

The PF of MaF1 is gotten by inverting the DTLZ1 PF [51]. For the four and ten objectives, MOEA/D-PaS performs better than proposed algorithms, while in seven objectives, PaS-EAD has a better performance. MaF2 is gotten from DTLZ2 by enhancing the difficulty of convergence. For getting the real PF, all the objectives should be optimized on MaF2 at the same time. On MaF2, PaS-EAD has superior performance in four and seven objectives, and PaS-PAD has the best performance in ten objectives. Regarding MaF3 with a convex PF and a lot of local PFs, PaS-PAD shows an advantageous performance over its competitors in four, seven, and ten objectives. The PF of MaF4 is obtained by inverting the DTLZ3 PF shape. MOEA/D-PaS performs best on MaF4 in seven and ten objectives, while in four objectives, PaS-PAD and PaS-LAD perform better than the other two algorithms. On MaF5, PaS-PAD has superior

Table 5: HV metric (mean/std) comparison results on MaF.

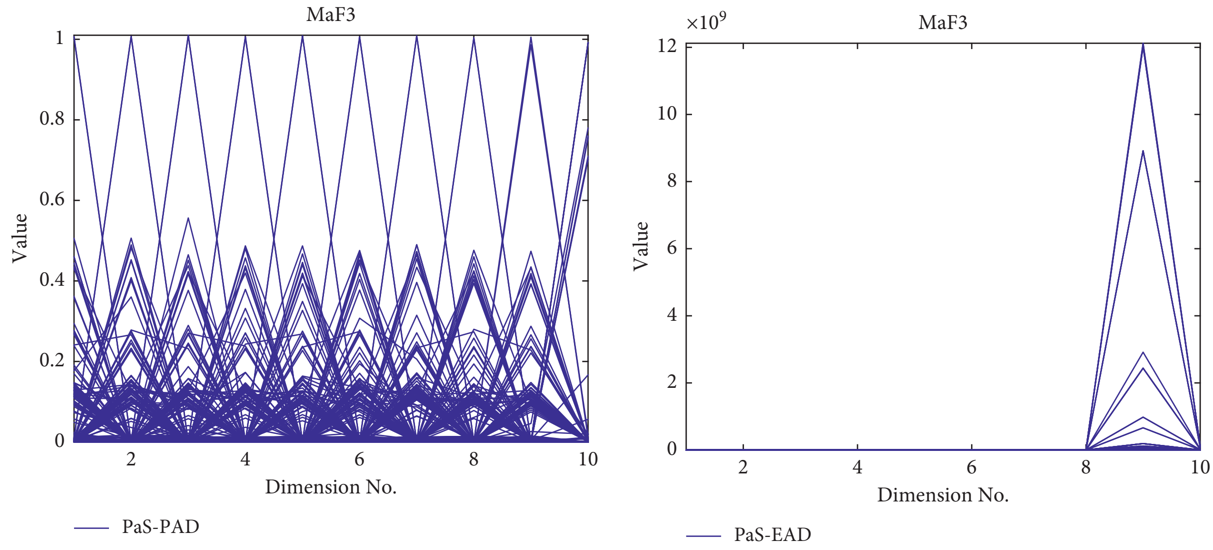
Problem	MOEA/D-PaS	PaS-LAD	PaS-PAD	PaS-EAD
4-Objective				
MaF1	**3.3027e − 2 (2.53e − 4)**	3.2898e − 2 (2.14e − 4)	3.2923e − 2 (1.41e − 4)	3.2949e − 2 (2.28e − 4)
MaF2	2.1235e − 1 (2.41e − 3)	2.1637e − 1 (1.73e − 3)	2.1641e − 1 (1.58e − 3)	**2.1830e − 1 (1.52e − 3)**
MaF3	6.8649e − 1 (4.16e − 1)	9.9230e − 1 (1.22e − 2)	**9.9563e − 1 (1.06e − 4)**	9.0867e − 1 (2.53e − 1)
MaF4	1.6656e − 1 (9.62e − 2)	2.2714e − 1 (3.50e − 3)	**2.2947e − 1 (1.36e − 3)**	2.0319e − 1 (6.94e − 2)
MaF5	6.6667e − 1 (4.36e − 2)	6.6824e − 1 (5.19e − 2)	**6.8048e − 1 (1.29e − 3)**	6.1379e − 1 (1.16e − 1)
MaF6	**1.3487e − 1 (2.12e − 3)**	1.3385e − 1 (1.85e − 3)	1.3390e − 1 (2.26e − 3)	1.3378e − 1 (1.43e − 3)
MaF7	2.6213e − 1 (1.20e − 2)	**2.7051e − 1 (9.46e − 4)**	2.6184e − 1 (4.77e − 2)	2.5460e − 1 (4.88e − 2)
MaF8	0.0000e + 0 (0.00e + 0)	3.0523e − 3 (9.61e − 3)	**7.7592e − 3 (3.88e − 2)**	2.9531e − 3 (1.08e − 2)
7-Objective				
MaF1	1.9797e − 4 (8.93e − 6)	1.9797e − 4 (5.02e − 6)	1.9549e − 4 (5.87e − 6)	**2.0684e − 4 (8.28e − 6)**
MaF2	1.6956e − 1 (2.98e − 3)	1.7424e − 1 (2.83e − 3)	1.7338e − 1 (2.64e − 3)	**1.7579e − 1 (3.53e − 3)**
MaF3	3.6908e − 1 (4.48e − 1)	8.9982e − 1 (3.05e − 1)	**8.9991e − 1 (3.05e − 1)**	8.2802e − 1 (3.77e − 1)
MaF4	**7.2824e − 3 (3.34e − 4)**	5.6786e − 3 (1.77e − 4)	5.6554e − 3 (1.46e − 4)	6.1162e − 3 (2.34e − 4)
MaF5	8.7820e − 1 (1.23e − 2)	**8.8009e − 1 (1.78e − 3)**	8.7854e − 1 (2.16e − 3)	8.5780e − 1 (3.64e − 2)
MaF6	8.4362e − 2 (5.43e − 3)	9.0020e − 2 (1.21e − 3)	8.9030e − 2 (5.98e − 3)	**9.0906e − 2 (9.54e − 6)**
MaF7	2.2109e − 1 (2.18e − 3)	**2.2492e − 1 (1.73e − 3)**	2.2470e − 1 (1.38e − 3)	2.0695e − 1 (5.23e − 2)
MaF8	0.0000e + 0 (0.00e + 0)	**2.2588e − 3 (8.31e − 3)**	9.5019e − 4 (2.86e − 3)	2.2216e − 3 (8.15e − 3)
10-Objective				
MaF1	**5.2740e − 7 (3.83e − 8)**	4.6605e − 7 (2.00e − 8)	4.7678e − 7 (2.81e − 8)	5.2065e − 7 (2.31e − 8)
MaF2	5.0267e − 2 (5.66e − 3)	4.7869e − 2 (5.28e − 3)	**5.0656e − 2 (6.15e − 3)**	4.9576e − 2 (5.11e − 3)
MaF3	3.4330e − 2 (1.24e − 1)	5.6636e − 1 (5.04e − 1)	**7.0284e − 1 (4.61e − 1)**	0.0000e + 0 (0.00e + 0)
MaF4	**2.8935e − 4 (5.49e − 5)**	2.0454e − 4 (9.61e − 6)	1.9942e − 4 (6.41e − 6)	2.3963e − 4 (1.25e − 5)
MaF5	9.5392e − 1 (5.79e − 3)	9.6577e − 1 (8.78e − 3)	**9.6664e − 1 (5.42e − 3)**	9.5800e − 1 (1.91e − 2)
MaF6	8.0122e − 2 (2.15e − 2)	9.0859e − 2 (9.33e − 5)	9.0868e − 2 (8.73e − 5)	**9.0903e − 2 (1.50e − 5)**
MaF7	**1.8595e − 1 (7.27e − 3)**	1.8555e − 1 (1.85e − 3)	1.8488e − 1 (1.56e − 3)	1.4057e − 1 (5.80e − 2)
MaF8	0.0000e + 0 (0.00e + 0)	3.1040e − 5 (1.62e − 4)	**9.9174e − 5 (2.80e − 4)**	6.9384e − 5 (1.85e − 4)
Best/all	6/24	4/24	9/24	5/24



(a)

(b)

Figure 6: Continued.



(c)

(d)

Figure 6: Visualization of PF approximation by four algorithms on the 10-objective MaF3: (a) MOEA/D-PaS, (b) PaS-LAD, (c) PaS-PAD, and (d) PaS-EAD.

Table 6: Runtimes (mean/std) comparison results on WFG.

Problem	PaS-LAD	PaS-PAD	PaS-EAD	MOEA/D-PaS
2-Objective				
WFG1	5.4306e + 1 (8.58e − 1)	5.4372e + 1 (8.35e − 1)	6.1241e + 1 (2.75e + 0)	**5.3998 e + 1 (6.49 e − 1)**
WFG2	5.3618e + 1 (7.04e − 1)	5.3681e + 1 (7.34e − 1)	6.0409e + 1 (2.68e + 0)	**5.3422 e + 1 (7.62 e − 1)**
WFG3	5.3953e + 1 (7.03e − 1)	5.3853e + 1 (7.07e − 1)	6.0820e + 1 (2.67e + 0)	**5.3727 e + 1 (6.97 e − 1)**
WFG4	5.3485e + 1 (8.39e − 1)	5.3718e + 1 (8.19e − 1)	6.0738e + 1 (2.80e + 0)	**5.2700 e + 1 (8.47 e − 1)**
WFG5	5.4351e + 1 (8.03e − 1)	5.4386e + 1 (7.10e − 1)	6.0595e + 1 (2.49e + 0)	**5.3689 e + 1 (8.18 e − 1)**
WFG6	5.3315e + 1 (6.63e − 1)	5.3524e + 1 (6.94e − 1)	6.0993e + 1 (2.74e + 0)	**5.3179 e + 1 (8.40 e − 1)**
WFG7	5.5593e + 1 (8.18e − 1)	5.5607e + 1 (8.21e − 1)	6.2865e + 1 (2.87e + 0)	**5.4881 e + 1 (9.31 e − 1)**
WFG8	5.4341e + 1 (7.90e − 1)	**5.4276 e + 1 (6.72 e − 1)**	6.1010e + 1 (2.48e + 0)	5.4376e + 1 (7.65e − 1)
4-Objctive				
WFG1	1.1602e + 2 (2.43e + 0)	**1.1598 e + 2 (2.48 e + 0)**	1.2543e + 2 (3.31e + 0)	1.1832e + 2 (3.01e + 0)
WFG2	**1.1882 e + 2 (2.76 e + 0)**	1.1890e + 2 (2.82e + 0)	1.2971e + 2 (2.93e + 0)	1.1953e + 2 (2.72e + 0)
WFG3	1.2042e + 2 (3.28e + 0)	1.2010e + 2 (2.96e + 0)	**1.0514 e + 2 (1.44 e + 1)**	1.1744e + 2 (7.23e + 0)
WFG4	1.1969e + 2 (3.42e + 0)	1.2107e + 2 (3.24e + 0)	1.3144e + 2 (3.19e + 0)	**1.1801 e + 2 (2.70 e + 0)**
WFG5	1.2069e + 2 (2.77e + 0)	1.2098e + 2 (2.97e + 0)	1.3135e + 2 (2.71e + 0)	**1.1969 e + 2 (2.90 e + 0)**
WFG6	1.1705e + 2 (2.18e + 0)	1.1841e + 2 (2.58e + 0)	1.3186e + 2 (3.32e + 0)	**1.1543 e + 2 (3.64 e + 0)**
WFG7	1.2524e + 2 (2.71e + 0)	1.2508e + 2 (2.76e + 0)	1.3525e + 2 (6.57e + 0)	**1.2080 e + 2 (2.86 e + 0)**
WFG8	**1.1980 e + 2 (2.72 e + 0)**	1.2002e + 2 (2.60e + 0)	1.2913e + 2 (3.65e + 0)	1.2045e + 2 (2.53e + 0)
10-Objective				
WFG1	**1.5182 e + 2 (2.03 e + 0)**	1.5255e + 2 (3.52e + 0)	1.7691e + 2 (1.59e + 1)	1.5425e + 2 (1.66e + 0)
WFG2	1.5776e + 2 (1.41e + 1)	1.5788e + 2 (1.44e + 1)	**1.4846 e + 2 (1.24 e + 1)**	1.5138e + 2 (1.24e + 1)
WFG3	1.2879e + 2 (1.35e + 1)	**1.1707 e + 2 (8.71 e + 0)**	1.3424e + 2 (4.75e + 0)	1.4064e + 2 (7.13e + 0)
WFG4	1.5341e + 2 (1.52e + 0)	1.5452e + 2 (1.90e + 0)	1.6346e + 2 (7.38e + 0)	**1.4539 e + 2 (7.75 e + 0)**
WFG5	1.3898e + 2 (8.69e + 0)	**1.3485 e + 2 (8.61 e + 0)**	1.6277e + 2 (1.16e + 1)	1.4766e + 2 (7.95e + 0)
WFG6	1.4951e + 2 (5.74e + 0)	1.5088e + 2 (3.88e + 0)	1.3634e + 2 (1.19e + 1)	**1.1097 e + 2 (1.18 e + 1)**
WFG7	1.6068e + 2 (1.90e + 0)	1.6066e + 2 (1.72e + 0)	1.4212e + 2 (1.41e + 1)	**1.3254 e + 2 (1.55 e + 1)**
WFG8	1.3959e + 2 (1.25e + 1)	**1.3083 e + 2 (1.39 e + 1)**	1.3834e + 2 (8.34e + 0)	1.3848e + 2 (1.84e + 1)
best/all	2/24	5/24	2/24	14/24

The bold values represent the minimum of each rows, which denote the fastest algorithms for each test functions.

performance in four and ten objectives, while in seven objectives, PaS-LAD performs best. Concerning that MaF6 has degenerate PF, PaS-EAD performs better than other algorithms in seven and ten objectives, and the four algorithms perform close in four objectives. For MaF7 with a disconnected PF, PaS-LAD has advantageous performance in four and seven objectives, while in ten objectives, PaS-LAD has a close performance with MOEA/D-PaS. On MaF8, the proposed algorithms have the smallest HV values larger than 0, while the HV values of MOEA/D-PaS are 0 in four, seven, and ten objectives. All algorithms could not find the obtained solutions overall, but adaptive operators can still help the algorithms to get some solutions from the results shown above.

Besides, we plot the final population of the four algorithms on the MaF3 test problem. From Figure 6, we can see that, on the MaF3 test problem with a convex PF, the MOEA/D-PaS and PaS-EAD cannot achieve the solutions. PaS-LAD and PaS-PAD have achieved solutions with better distributed solutions than the other two algorithms. PaS-PAD has the best convergence and diversity among these algorithms.

On the MaF test suite, PaS-PAD still has better performance than other algorithms, but the other two proposed algorithms perform not so well compared with MOEA/D-PaS overall. The adaptive operators can help algorithms get more obtained solutions on some test problems. The PAD method is better than the other two methods overall.

Table 6 represents the runtimes of every algorithm. It can be seen from the table that when using adaptive strategy, it would cost extra computing resources to get the optimal solutions. MOEA/D-PaS has the best performance in 14 out of 24 cases. PaS-LAD, PaS-PAD, and PaS-EAD perform best in 2, 5, and 2 out of 24 cases. So the runtimes in adaptive strategies are longer than MOEA/D-PaS. The values of *CR* and *F* in the proposed three strategies are changed by the generations. During every generation, it is necessary to calculate the values of *CR* and *F* using extra computing resources. But from the values of HV shown in Tables 3 and 4, we can know that it is worth spending some extra computing resources to get better results.

5. Conclusion

MOEA/*D* and MOEA/D-DE have been demonstrated to be effective and useful in solving MOPs. However, the parameters are fixed, which would affect the convergence of the algorithm. This paper proposes three algorithms using different self-adaptive DE operators to automatically adjust the setting of parameters *CR* and *F* in different problems based on MOEA/D-DE. The experiments have demonstrated that the adaptive strategies have effectiveness on the 2-objective and 3-objective test functions. Moreover, PAD has better performance than LAD and EAD from the tables of IGD and HV values. For MaOPs, we incorporate the adaptive DE methods into the MOEA/D-PaS and run these algorithms on WFG and MaF test suites for 4-, 7-, and 10-objective problems. According to the HV values, for WFG problems, the proposed algorithms have huge advantages compared with MOEA/D-PaS. Moreover, PaS-PAD has superior performance among these adaptive methods. PaS-LAD and PaS-EAD do not perform as well as PaS-PAD but still have better performance than MOEA/D-PaS. Besides, on the MaF test functions, PaS-PAD still has advantages compared with other algorithms. In conclusion, for solving MOPs and MaOPs, the adaptive methods can help the algorithms to get more obtained solutions to converge much closer to the real PFs. Among these adaptive methods, the PAD method has the best performance.

However, there are still some problems to be solved; for example, for many real-world problems, the effectiveness of the adaptive methods needs to be proved. So we need to run these adaptive algorithms on more complex problems to identify their effectiveness.

For further study, firstly, we need to use these three adaptive strategies to run more complex test functions to identify if the PAD method still has advantages over LAD and EAD. Secondly, we would like to use niching technologies in population to accelerate the pace of converging to the optimal solution in the three adaptive algorithms. Thirdly, we would like to apply these proposed algorithms to practical problems like community detection, recommendation system, and so on [60]

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

This work about multiobjective optimization problems has been accepted in the Second International Conference, NCAA 2021 [60]. In this paper, the authors developed new algorithms to solve the many-objective optimization problems (MaOPs), which are not included in the conference paper. The authors incorporate the three adaptive strategies into MOEA/D-PaS to solve the MaOPs. The authors run these algorithms on 4-, 7-, and 10-objective test problems to verify the effectiveness of the adaptive strategies. The HV values of WFG and MaF test suites demonstrate the advantages of the proposed strategies.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 61703256 and 61806119), Natural Science Basic Research Plan in Shaanxi Province of China (Program no. 2017JQ6070), and the Fundamental Research Funds for the Central Universities (Program no. GK201803020).

References

[1] S. Mardle and K. Miettinen, "Nonlinear multiobjective optimization," *Journal of the Operational Research Society*, vol. 51, no. 2, p. 246, 1999.

[2] R. Wang, P. J. Fleming, and R. C. Purshouse, "General framework for localised multi-objective evolutionary algorithms," *Information Sciences*, vol. 258, no. 3, pp. 29–53, 2014.

[3] Q. F. 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, 2008.

[4] E. J. Hughes, “Multiple single objective pareto sampling,” *The 2003 Congress on Evolutionary Computation*, vol. 4, pp. 2678–2684, 2003.

[5] 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.

[6] C. M. Fonseca and P. J. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, 1998.

[7] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[8] S. Bleuler, M. Brack, L. Thiele, and E. Zitzler, “Multiobjective genetic programming: reducing bloat using SPEA2,” *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 536–543, 2001.

[9] J. Horn, N. Nafpliotis, and D. E. Goldberg, “A niched pareto genetic algorithm for multiobjective optimization,”vol. 1, pp. 82–87, in *Proceedings of the First IEEE Conference on Evolutionary Computation*, vol. 1, pp. 82–87, IEEE World Congress on Computational Intelligence, Orlando, FL, USA, June 1994.

[10] D. W. Corne, J. D. Knowles, and M. J. Oates, “The pareto envelope-based selection algorithm for multiobjective optimization,” *Lecture Notes in Computer Science*, Springer, vol. 1917, pp. 839–848, , Berlin, Germany, 2000.

[11] D. W. Corne, N. R. Jerram, and J. D. Knowles, “PESA-II: Region-based selection in evolutionary multiobjective optimization,” in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pp. 283–290, Madrid, Spain, July 2001.

[12] 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.

[13] E. Zitzler and S. Künzli S, “Indicator-Based selection in multiobjective search,” in *Lecture Notes in Computer Science*Vol. 3242, Springer, Berlin, Germany, 2004.

[14] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, “Localized weighted sum method for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2018.

[15] 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.

[16] N. Baatar, K.-Y. Jeong, and C.-S. Koh, “Adaptive parameter controlling non-dominated ranking differential evolution for multi-objective optimization of electromagnetic problems,” *IEEE Transactions on Magnetics*, vol. 50, no. 2, pp. 709–712, 2014.

[17] Z.-H. Zhan, Z.-J. Wang, H. Jin, and J. Zhang, “Adaptive distributed differential evolution,” *IEEE Transactions on Cybernetics*, vol. 50, no. 11, pp. 4633–4647, 2020.

[18] Q. Lin, S. Liu, K.-C. Wong et al., “A clustering-based evolutionary algorithm for many-objective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 391–405, 2019.

[19] O. Chikumbo, E. Goodman, and K. Deb, “Approximating a multidimensional pareto front for a land use management problem: a modified moea with an epigenetic silencing metaphor,” in *Proceedings of the 2012 IEEE Congress on Evolutionary Computation*, pp. 1–9, Brisbane, Australia, June 2012.

[20] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, “Evolutionary many-objective optimization of hybrid electric vehicle control: from general optimization to preference articulation,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 2, pp. 97–111, 2017.

[21] M. Ming, R. Wang, Y. Zha, and T. Zhang, “Pareto adaptive penalty-based boundary intersection method for multi-objective optimization,” *Information Sciences*, vol. 414, pp. 158–174, 2017.

[22] R. J. Lygoe, M. Cary, and P. J. Fleming, “A real-world application of a many-objective optimisation complexity reduction process,” in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, pp. 641–655, Sheffield, UK, March 2013.

[23] 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.

[24] 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.

[25] L. Pan, L. Li, C. He, and K. C. Tan, “A subregion division-based evolutionary algorithm with effective mating selection for many-objective optimization,” *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3477–3490, 2020.

[26] 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*, vol. 19, no. 5, pp. 694–716, 2015.

[27] R. Wang, Q. Zhang, and T. Zhang, “Decomposition-based algorithms using pareto adaptive scalarizing methods,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[28] 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.

[29] M. Li, S. Yang, and X. Liu, “Shift-based density estimation for pareto-based algorithms in many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.

[30] K. Bringmann and T. Friedrich, “An efficient algorithm for computing hypervolume contributions,” *Evolutionary Computation*, vol. 18, no. 3, pp. 383–402, 2010.

[31] 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.

[32] L. M. S. Russo and A. P. Francisco, “Quick hypervolume,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 481–502, 2014.

[33] D. Brockhoff, T. Wagner, and H. Trautmann, “On the properties of the R2 indicator,” in *Proceedings of the 14th Annu Conference Genetic. Evolutionary Computation*, pp. 465–472, Philadelphia, PA, USA, July 2012.

[34] R. H. Gómez and C. A. C. Coello, “Mombi: a new metaheuristic for many-objective optimization based on the R2 indicator,” in *Proceedings of the IEEE Congress Evolutionary Computation*, pp. 2488–2495, Cancun, Mexico, Jnue 2013.

[35] H. Wang, L. Jiao, and X. Yao, “Two_Arch2: an improved two-archive algorithm for many-objective optimization,” *IEEE*

Transactions on Evolutionary Computation, vol. 19, no. 4, pp. 524–541, 2015.

[36] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 924–938, 2016.

[37] Z. Liang, K. Hu, X. Ma, and Z. Zhu, "A many-objective evolutionary algorithm based on a two-round selection strategy," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1417–1429, 2021.

[38] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 2021, Article ID 3041212, 13 pages, 2021.

[39] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 587–602, 2019.

[40] Y.-H. Zhang, Y.-J. Gong, T.-L. Gu et al., "DECAL: decomposition-based coevolutionary algorithm for many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 27–41, 2019.

[41] L. Ma, N. Li, Y. Wang, S. Yang, M. Huang, and H. Zhang, "Learning to optimize: reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Transactions on Cybernetics*, vol. 2021, Article ID 3086501, 14 pages, 2021.

[42] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6723–6742, 2021.

[43] A. G. Richard, C. P. Almeida, J. N. Kuk, and A. Pozo, "MOEA/d with adaptive operator selection for the environmental/economic dispatch problem," in *Proceedings of the 2015 Latin America Congress on Computational Intelligence*, pp. 1–6, Curitiba, PR, Brazil, October 2015.

[44] R. Storn and K. Price, "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[45] K. V. Price, "Differential evolution: a fast and simple numerical optimizer," in *Proceedings of the North American Fuzzy Information Processing*, pp. 524–527, Berkeley, CA, USA, January 1996.

[46] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK, 2001.

[47] J. Han, M. He, and X. Wang, "Improvement of differential evolution multiobjective optimization algorithm based on decomposition," *Journal of Physics: Conference Series*, vol. 1213, no. 3, p. 7, 2019.

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

[49] K. Deb, L. Thiele, and M. Laumanns, *Scalable Test Problems for Evolutionary Multi-Objective Optimization*, pp. 9-10, Springer, Berlin, Germany, 2006.

[50] 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.

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

[52] R. Cheng, M. Li, Y. Tian et al., "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[53] Z. Liang, K. Hu, X. Ma, and Z. Zhu, "A many-objective evolutionary algorithm based on a two-round selection strategy," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1417–1429, 2021.

[54] Z. He and G. G. Yen, "Many-objective evolutionary algorithm: objective space reduction and diversity improvement," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 145–160, 2016.

[55] L. Li, L. Chang, T. Gu, W. Sheng, and W. Wang, "On the norm of dominant difference for many-objective particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 2055–2067, 2021.

[56] I. Yevseyeva, A. P. Guerreiro, M. T. M. Emmerich, and C. M. Fonseca, "A portfolio optimization approach to selection in multiobjective evolutionary algorithms," *Parallel Problem Solving from Nature - PPSN XIII*, Springer, Berlin, Germany, pp. 672–681, 2014.

[57] A. P. Guerreiro and C. M. Fonseca, "Hypervolume sharpe-ratio indicator: formalization and first theoretical results," *Parallel Problem Solving from Nature - PPSN XIV*, Springer, Berlin, Germany, pp. 814–823, 2016.

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

[59] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: a matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[60] K. Bian, Y. Sun, S. Cheng, Z. Liu, and X. Sun, "Adaptive methods of differential evolution multi-objective optimization algorithm based on decomposition," in *Proceedings of the Second International Conference, Neural Computing for Advanced Applications 2021*, pp. 458–472, Guangzhou, China, August 2021.