



# A multi-objective particle swarm optimization with a competitive hybrid learning strategy

Fei Chen<sup>1</sup> · Yanmin Liu<sup>2</sup> · Jie Yang<sup>2</sup> · Jun Liu<sup>3</sup> · Xianzi Zhang<sup>3</sup>

Received: 9 August 2023 / Accepted: 2 April 2024  
© The Author(s) 2024

## Abstract

To counterbalance the abilities of global exploration and local exploitation of algorithm and enhance its comprehensive performance, a multi-objective particle swarm optimization with a competitive hybrid learning strategy (CHLMOPSO) is put forward. With regards to this, the paper first puts forward a derivative treatment strategy of personal best to promote the optimization ability of particles. Next, an adaptive flight parameter adjustment strategy is designed in accordance with the evolutionary state of particles to equilibrate the exploitation and exploration abilities of the algorithm. Additionally, a competitive hybrid learning strategy is presented. According to the outcomes of the competition, various particles decide on various updating strategies. Finally, an optimal angle distance strategy is proposed to maintain archive effectively. CHLMOPSO is compared with other algorithms through simulation experiments on 22 benchmark problems. The results demonstrate that CHLMOPSO has satisfactory performance.

**Keywords** Multi-objective particle swarm optimization · Derivative treatment strategy · Hybrid learning · Optimal angle distance

## Introduction

In numerous instances of engineering practice and scientific research, multi-objective optimization problems (MOPs) [1] are prevalent. Their objectives often conflict with each other, and optimization of these objectives must be done simultaneously. As a consequence, what is usually obtained in MOPs is a group of solutions that are relatively superior among the

objectives, known as Pareto optimal solutions [2]. Obtaining a group of solutions that converge to the true Pareto front (PF) [3] as much as feasible and are uniformly distributed on it is the most critical purpose of solving MOPs.

Due to the universality and group-based search characteristics of multi-objective evolutionary algorithms (MOEAs), which can generate multiple solutions in parallel. As a result, MOEAs have been greatly developed, and a significant number of MOEAs have been successfully applied to solve MOPs, for instance, firefly algorithm (FA) [4], genetic algorithm (GA) [5], ant colony optimization (ACO) [6], differential evolution (DE) [7], and particle swarm optimization (PSO) [8]. Among them, PSO was inspired by the foraging behavior of birds. It has the advantages of fast convergence, a simple mechanism, and so on. After some studies, it was found that PSO has good potential to be extended to solve MOPs.

Since the presentation of multi-objective particle swarm optimization (MOPSO) [9] in 2002, many researchers have improved it. MOPSOs can be approximately subdivided into three categories according to different strategies. The first category is MOPSOs based on Pareto dominance relationship. The representative MOPSOs include the speed-constrained MOPSO (SMPSO) [10], the MOPSO based on

---

✉ Yanmin Liu  
yanmin7813@163.com

Fei Chen  
2475397530@qq.com

Jie Yang  
530966074@qq.com

Jun Liu  
1045131817@qq.com

Xianzi Zhang  
764811651@qq.com

<sup>1</sup> School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China

<sup>2</sup> Zunyi Normal College, Zunyi 563002, China

<sup>3</sup> School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550025, China

global margin ranking (MOPSO/GMR) [11], and the vortex MOPSO (MOVPSO) [12]. The second category is indicator-based MOPSOs. The representative MOPSOs include the MOPSO based on virtual Pareto front (MOPSO/vPF) [13], the MOPSO based on R2 indicator (ANMPSO) [14], and the MOPSO based on hypervolume (MOPSO<sub>h</sub>v) [15]. The third category is decomposition-based MOPSOs. The representative MOPSOs include the MOPSO based on decomposition (dMOPSO) [16], the decomposition-based MOPSO (MPSO/D) [17], the MOPSO based on multiple search strategies (MMOPSO) [18], and the external archive-guided MOPSO (AgMOPSO) [19].

Although the above algorithms improved MOPSO from different perspectives to solve MOPs, there are still two problems that need to be further solved in an expansion of PSO to settle MOPs. How to select appropriate personal best (pbest) and global best (gbest) is the first issue that needs to be solved. Because different pbest and gbest will pilot particles to approximate the true PF in different directions, which influences the search ability of algorithm in the evolutionary process. Selecting appropriate pbest and gbest can efficiently find optimal solutions with good properties. How to effectively maintain the external archive is the second issue that needs to be solved. Since the role of archive is to reserve non-dominated solutions, it provides good candidate solutions for the selection of gbest. Therefore, it is inevitably crucial to maintain the external archive effectively in order to reduce the computational complexity of the algorithm as well as to ensure that the non-dominated solutions in the external archive remain well-distributed.

For the past few years, numerous researchers have intensively investigated the above two problems and put forward many improved MOPSOs. For example, the new selection strategies of pbest and gbest were proposed in the literature [13, 20–22]. Li et al. [13] proposed a newly defined virtual generational distance indicator to select appropriate gbest, and designed an adaptive selection strategy of pbest based on different evolutionary states to promote the search efficiency of the algorithm and adaptively strengthen the exploitation and exploration abilities of particles. Wu et al. [20] designed a new selection strategy of gbest according to different evolutionary states monitored by ESE, which adaptively selected global best solutions to enhance the equilibrium between exploration and exploitation. Han et al. [21] chose an appropriate gbest adaptively by introducing the solution distribution entropy, thus improving the search performance in terms of convergence speed and convergence accuracy. Sharma et al. [22] chose the solution with the minimum penalty boundary intersection as gbest of the particle, which improves the performance of algorithm in terms of diversity and convergence. In addition, many researchers [23–25] effectively maintained external archive. Han et al. [23] used the gradient method to maintain the external archive

and used the gradient information to solve the MOPs, which can obtain good Pareto sets, achieve smaller test errors at a faster rate, and improve the speed of convergence and local exploitation during the evolutionary process. Cui et al. [24] designed the diversity archive and the convergence archive by introducing the two-archive strategy into MOPSO and updated them in different ways to achieve a balance between convergence and diversity. Li et al. [25] updated the external archive by using a method based on dominant difference to improve the discriminability of particles that are difficult to obtain in high-dimensional space, and the experimental results also show that the method has better performance in terms of convergence and diversity and low complexity. From the above-mentioned literature, it is evident that the performance of algorithms depends greatly on the choice of the two best solutions and archive maintenance.

On the basis of the above review and analysis, a multi-objective particle swarm optimization with a competitive hybrid learning strategy (CHLMOPSO) is put forward for the sake of advancing algorithm robustness and achieving harmonious equilibrium between the convergence and diversity in dealing with MOPs. The distinctive contributions of CHLMOPSO that have been put forward are as follows:

(1) CHLMOPSO proposes a new update strategy of pbest. For further enhancement of the optimization ability of the particles, the derivative treatment of the stagnant particles without better pbest is carried out, which is more helpful to find better pbest.

(2) An adaptive strategy is devised to adjust the flight parameters in accordance with the evolutionary state of the particles, which is adjusted by using their ability to find their personal best solutions. This strategy can equilibrate the exploitation and exploration abilities of algorithm.

(3) A competitive hybrid learning strategy is proposed. This strategy mainly combines the advantages of MOPSO and competitive swarm optimizer (CSO) [26], determines the winners and losers through multiple methods of competition, and then updates them by using the updating strategies of MOPSO and CSO, respectively.

(4) An optimal angle distance strategy is proposed to maintain the archive effectively. The optimal angle distance considers the convergence distance of solutions and their minimum angle with other solutions at the same time, so as to increase the convergence speed of algorithm.

The arrangement of the remainder of this paper is as follows. Sections "Preliminaries and Background" and "The Proposed CHLMOPSO" describe the related work and the details of CHLMOPSO, respectively. Sections "Experimental Settings" and "Comparison of Experimental Results" present the experimental portion, in which CHLMOPSO is compared with the comparison algorithms on three suits

of benchmark problems to assess the effectiveness of algorithms. In the end, the conclusions of CHLMOPSO are provided in section "Conclusion".

## Preliminaries and background

### MOPs

In a general case, a MOP [27, 28] can be expressed as:

$$\begin{aligned} \min F(x) &= (f_1(x), f_2(x), \dots, f_M(x))^T \\ \text{s.t. } x &= (x_1, x_2, \dots, x_D) \in X \end{aligned} \tag{1}$$

where  $F(x)$  represents the objective vector of  $M$ -dimension,  $f_i(x)$  represents the objective function of the  $i$ -th dimension,  $M$  is the number of objectives, and  $x$  is a  $D$ -dimensional decision vector in the decision space  $X \subset R^D$ .

The Pareto dominance relationship could be utilized to evaluate the quality of solutions in MOPs for the reason that the objectives of MOPs are in conflict with each other. That is, a solution  $x_\alpha$  dominates the other solution  $x_\beta$ , denoted as  $x_\alpha < x_\beta$ , if and only if  $\forall u \in \{1, 2, \dots, m\} : f_u(x_\alpha) \leq f_u(x_\beta) \wedge \exists v \in \{1, 2, \dots, m\} : f_v(x_\alpha) < f_v(x_\beta)$ . A solution in MOPs is generally referred to as a Pareto optimal solution if it is not dominated by any other solution. The set of all Pareto optimal solutions in the decision space is known as Pareto optimal set (PS), and its image in the objective space is termed as Pareto front (PF).

### MOPSO

Each particle  $i$  in MOPSO [9] encompasses a velocity vector  $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$  and a position vector  $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ , where  $i = 1, 2, \dots, N$ ,  $N$  denotes population size, and  $D$  is the dimension of decision space. Subsequently, throughout the  $t + 1$ -th iteration, the velocity of the  $i$ -th particle is updated as follows:

$$v_{i,d}^{t+1} = \omega v_{i,d}^t + c_1 r_1 (pb_{i,d}^t - x_{i,d}^t) + c_2 r_2 (gb_{i,d}^t - x_{i,d}^t) \tag{2}$$

where  $t$  is the number of iterations,  $d = 1, 2, \dots, D$ ,  $pb_i^t = (pb_{i,1}^t, pb_{i,2}^t, \dots, pb_{i,D}^t)$  and  $gb_i^t = (gb_{i,1}^t, gb_{i,2}^t, \dots, gb_{i,D}^t)$  represent pbest and gbest of the  $i$ -th particle, respectively,  $\omega$  stands for inertia weight,  $c_1$  and  $c_2$  denote learning factors, and  $r_1$  and  $r_2$  represent two random numbers uniformly generated in  $[0,1]$ . Then, its position is updated as follows:

$$x_{i,d}^{t+1} = x_{i,d}^t + v_{i,d}^{t+1} \tag{3}$$

## The proposed CHLMOPSO

The present paper puts forward a competitive hybrid learning MOPSO to increase the holistic performance of algorithm. The algorithm is primarily made up of four components, and its main framework is shown in Fig. 1. First and foremost, the derivative treatment strategy is carried out on the stagnant particles without better pbest, which is more helpful to find better pbest. In the next place, for the sake of balancing the abilities of exploration and exploitation of algorithm, the flight parameters are adjusted adaptively in accordance with the evolutionary state of particles. Additionally, the winners and losers are determined through multiple competitions, and the hybrid learning strategies of MOPSO and CSO are used to guide them to fly. Finally, the optimal angle distance strategy is set forward to maintain the archive efficiently. The following sections go into great depth on the essential components of CHLMOPSO.

### The derivative treatment strategy of stagnant particles

Most present MOPSOs choose their pbest by the Pareto dominance relationship between the existing pbest and the newly discovered solution, while ignoring the evolutionary state of population. Additionally, when the existing pbest and the newly discovered solution do not dominate each other, randomly selecting one of them may cause the particles to fall into the oscillation search process, thus affecting the search efficiency of population.

Here, a derivative treatment strategy is put forward for the sake of ameliorating the issues mentioned above. The algorithm initially determines if there is a dominance relationship between the newly discovered candidate pbest (marked as  $pbest_{new}$ ) and the existing pbest (denoted as  $pbest_{old}$ ) when the particle discovers a new candidate pbest. If  $pbest_{new}$  dominates  $pbest_{old}$ , it means that the pbest of the current generation of the particle is better than the pbest of the previous generation, thus indicating that the particle is learning and evolving without stagnation, then  $pbest_{new}$  is determined to be the pbest of the current generation of the particle. Otherwise, if  $pbest_{old}$  dominates  $pbest_{new}$  or the two do not dominate each other, which means that the pbest of the particle has not been updated to get new information, that is, there is a stagnation phenomenon, which we call the stagnant particle, and  $pbest_{old}$  of the particle will be derived. This strategy helps the stagnant particles to find new possible candidate pbest. The particular description of the derivative process of pbest of the stagnant particles is presented hereunder.

*Procedure 1:* Adaptively determine the derivative number of new candidate pbest for the stagnant particles during each iteration, where the number of derivatives is constrained by the upper and lower limits of derivatives. Additionally, the

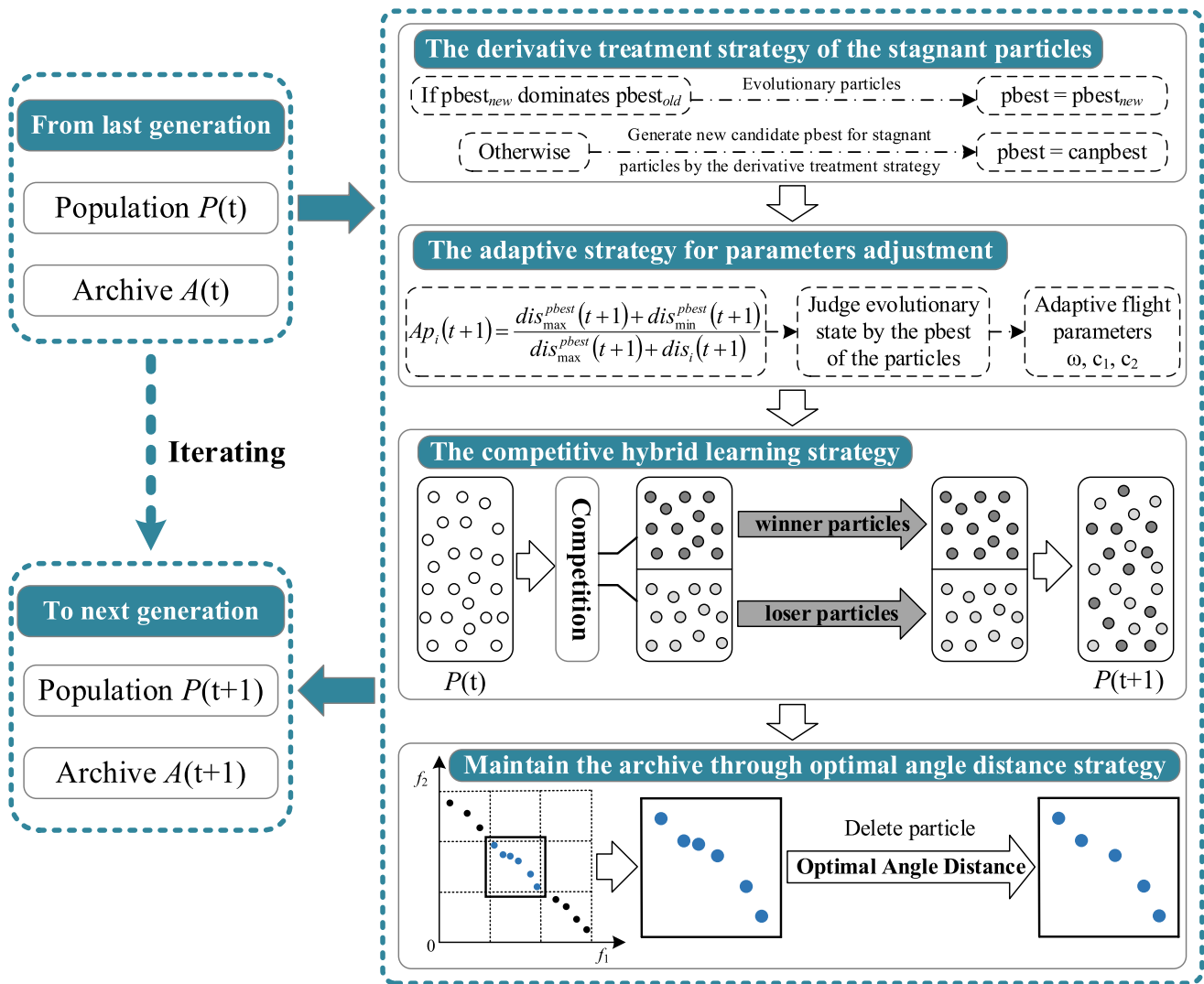


Fig. 1 The framework of CHLMOPSO

number of candidate pbest here is adaptively reduced with the number of iterations. This is to derive a more appropriate number of candidate pbest and improve the optimization ability of particles. Inspired by [29], at the  $t$ -th iteration, the derivative number of  $pbest_{old}$  of each stagnant particle is determined by Eq. (4):

$$p(t) = \begin{cases} (up - lp) \times (1 - 2 \times (t/T)^2 + lp / (up - lp)), & 0 \leq t \leq (T/2) \\ (up - lp) \times (2 \times ((T - t)/T)^2 + lp / (up - lp)), & (T/2) \leq t \leq T \end{cases} \quad (4)$$

where  $p(t)$  is the quantity of candidate pbest that each stagnant particle created for the  $t$ -th iteration,  $T$  stands for the number of maximum iteration,  $1 \leq lp \leq up \leq 5$ ,  $up$  is the maximum amount of derivatives, and  $lp$  is the minimum amount of derivatives.

*Procedure 2:* After determining the number of candidate pbest to be derived, new candidate personal best solutions  $canpbest = \{canpbest_1, canpbest_2, \dots, canpbest_{p(t)}\}$  are generated by deriving the  $pbest_{old}$  of the stagnant particles. These are the concrete steps:

$$canpbest_{i,j}(t) = pbest_{old_{i,j}}(t) - \Delta x(t)(ub_j - lb_j) \quad (5)$$

$$\Delta x(t) = \begin{cases} (x_u - x_l) \times (2r^2 + x_l / (x_u - x_l)), & 0 \leq r \leq 0.5 \\ (x_u - x_l) \times (1 - 2(r - 1)^2 + x_l / (x_u - x_l)), & 0.5 < r \leq 1 \end{cases} \quad (6)$$

the pseudo-code of the derivative treatment strategy of stagnant particles.

---

**Algorithm 1:** The derivative treatment strategy of the stagnant particles

---

**Input:**  $pbest_{old}$ ,  $pbest_{new}$ ,  $up$ ,  $lp$ ,  $r_u$ , and  $r_l$ .

**Output:** The updated  $pbest$ .

```

1: if  $pbest_{new} < pbest_{old}$  then
2:    $pbest = pbest_{new}$ ;
3: else
4:   Calculate the number  $p(t)$  of derivative candidate  $pbest$  of stagnant particle by Eq. (4);
5:   for  $i=1$  to  $p(t)$  do
6:     Compute the derivative additional distance  $\Delta x(t)$  by Eq. (6) and Eq. (7);
7:     Generate new candidate  $canpbest_i$  for personal best by Eq. (5);
8:     if  $canpbest_i < pbest_{old}$  then
9:       Reserve  $canpbest_i$ ;
10:    else
11:      Remove  $canpbest_i$ ;
12:    end if
13:  end for
14:  if  $isempty(canpbest)$  do
15:     $pbest = pbest_{old}$ ;
16:  else
17:    The reserved individuals are sorted by Pareto dominance relationship and the better one is selected as the  $pbest$  of the stagnant particle;
18:  end if
19: end if

```

---

$$x_u = r_u \times ub_j, x_l = r_l \times lb_j, \quad (7)$$

where  $canpbest_{i,j}(t)$  is the  $j$ -th dimension of the  $i$ -th derivative candidate  $pbest$ ,  $ub_j$  and  $lb_j$  represent the upper limit and lower limit, respectively.  $\Delta x(t)$  represents the additional distance derived,  $x_u$  represents the maximum additional distance derived, which is determined by the product of the user-defined upper limit ratio  $r_u$  and  $ub_j$ ,  $x_l$  represents the minimum additional distance derived, which is determined by the product of the user-defined lower limit ratio  $r_l$  and  $lb_j$ , and  $r$  is the absolute value of a Gaussian random number with a mean of 0 and a variance of 1/9. Similar to the literature [29], the parameters  $r_u$  and  $r_l$  in this paper are selected in the ranges (0.02, 0.7) and [0, 0.02], respectively.

*Procedure 3:* Firstly, the derivative particles are compared with the  $pbest_{old}$  by Pareto dominance. If there is a dominance relationship, the algorithm will determine who becomes  $pbest$  of the stagnant particle. If there is no dominance relationship, the algorithm randomly selects one of them as the  $pbest$  of the stagnant particles. Algorithm 1 gives

### Adaptive strategy for parameters adjustment

The three flight parameters  $\omega$ ,  $c_1$ , and  $c_2$  are critical in MOPSO. The majority of experiments [21, 23, 24] have demonstrated that the larger  $\omega$  and  $c_1$ , and the smaller  $c_2$  can stimulate better global exploration, while the smaller  $\omega$  and  $c_1$ , and the larger  $c_2$  are conducive to better local exploitation. As a consequence, for the purpose of equilibrating the exploration and exploitation abilities, an adaptive flight parameter adjustment strategy is designed through the evolutionary state of each particle.

*Procedure 1:* Through the above analysis, the evolutionary state of each particle is first evaluated with respect to the dominance relationship between the current  $pbest$  and the previous  $pbest$ , thereby designing an adaptive flight parameter strategy.

$$Ap_i(t) = \frac{dis_{\max}^{pbest}(t) + dis_{\min}^{pbest}(t)}{dis_{\max}^{pbest}(t) + dis_i(t)} \quad (8)$$

where  $Ap_i(t)$  is the adaptive parameter of the  $i$ -th particle,  $dis_{\max}^{pbest}(t)$  and  $dis_{\min}^{pbest}(t)$  stand for the maximum and minimum distance between all particles and pbest, respectively, and  $dis_i(t)$  signifies the distance between the  $i$ -th particle and its corresponding pbest.

*Procedure 2:* Based on the above considerations, according to the adaptive parameters of each particle, the adaptive flight parameters proposed in this paper are expressed as:

$$\omega_i(t) = \begin{cases} \omega_o \times (1 - Ap_i(t)), & \text{if } pbest_i(t-1) < pbest_i(t) \\ \omega_o \times (1 + Ap_i(t)), & \text{if } pbest_i(t-1) > pbest_i(t) \\ \omega_o, & \text{if } pbest_i(t-1) <> pbest_i(t) \end{cases} \quad (9)$$

$$c_{1,i}(t) = \begin{cases} c_{1_o} \times (1 - Ap_i(t)), & \text{if } pbest_i(t-1) < pbest_i(t) \\ c_{1_o} \times (1 + Ap_i(t)), & \text{if } pbest_i(t-1) > pbest_i(t) \\ c_{1_o}, & \text{if } pbest_i(t-1) <> pbest_i(t) \end{cases} \quad (10)$$

$$c_{2,i}(t) = \begin{cases} c_{2_o} \times (1 - Ap_i(t)), & \text{if } pbest_i(t-1) > pbest_i(t) \\ c_{2_o} \times (1 + Ap_i(t)), & \text{if } pbest_i(t-1) < pbest_i(t) \\ c_{2_o}, & \text{if } pbest_i(t-1) <> pbest_i(t) \end{cases} \quad (11)$$

## The competitive hybrid learning strategy

In most meta-heuristic algorithms, it is fundamental to effectively promote convergence and diversity. In the standard MOPSO [9], every particle in the population learns in the same way, so the learning mode of particles is too single, which affects the exploration ability and the diversity. Therefore, inspired by the way of improving population diversity in CSO [26], this part puts forward a competitive hybrid learning strategy to update the particles. Algorithm 2 displays the corresponding pseudo-code.

---

### Algorithm 2: The competitive hybrid learning strategy

---

**Input:** The population  $P$ , the population size  $N$ ,  $\omega$ ,  $c_1$ , and  $c_2$ .

**Output:** The updated particles.

```

1: for  $i=1$  to  $\text{floor}(N/2)$  do    % The floor function is integer in the direction of negative infinity.
2:   Randomly select two particles  $x_1$  and  $x_2$  from population  $P$ ;
3:   if  $x_1 < x_2$  then          % Compare the dominance relationship between the two particles.
4:      $x_w = x_1$ ,  $x_l = x_2$ ;    %  $x_w$  is winner and  $x_l$  is loser.
5:   else if  $x_2 < x_1$  then
6:      $x_w = x_2$ ,  $x_l = x_1$ ;
7:   else
8:     if  $E_1 > E_2$  then      %Compare the evolutionary ability of the two particles.
9:        $x_w = x_1$ ,  $x_l = x_2$ ;
10:    else
11:       $x_w = x_2$ ,  $x_l = x_1$ ;
12:    end if
13:  end if
14:  Remove  $x_1$  and  $x_2$  from the old population, add  $x_w$  and  $x_l$  to the new population;
15: end for
16: Update the winners by Eq. (2) and Eq. (3);
17: Update the losers by Eq. (13) and Eq. (14);
18: return particles.

```

---

where  $\omega_o = 0.4$ ,  $c_{1_o} = 2$ ,  $c_{2_o} = 2$ . During the search process, the proposed adaptive flight parameter adjustment strategy can find the appropriate flight parameters for each particle. This is beneficial to accelerate convergence speed of the algorithm approaching the true PF and balance global exploration and local exploitation abilities of the algorithm.

*Procedure 1:* The pairwise competition mechanism in typical CSO only compares the fitness values of two particles, so the method of comparison is relatively simple. Therefore, in this section, a multi-competition method is proposed to determine the winners and losers. Firstly, the basic principle of Pareto dominance is introduced into the competition mechanism to judge whether there is dominance relationship between two randomly selected particles. The non-dominated particle wins and the dominated particle loses

if there is a dominance relationship. Otherwise, an evolutionary ability comparison is used to distinguish the winner and loser. The particle with strong evolutionary ability is the winner, and the particle with weak evolutionary ability is the loser. The evolutionary ability here stands for the ability of particles to discover their own personal best. The evolutionary ability of the  $i$ -th particle is described as:

$$E_i(t) = \sqrt{\sum_{m=1}^M \left( f_{i,m}^{pbest}(t) - f_{i,m}^{pbest}(t-1) \right)^2} \quad (12)$$

where  $f_{i,m}^{pbest}(t)$  and  $f_{i,m}^{pbest}(t-1)$  denote the  $m$ -th fitness value of the personal best solutions of the  $i$ -th particle, and  $M$  is the number of objectives.

*Procedure 2:* After the above competition, the hybrid learning strategy is proposed to update the winners and losers. The winners and losers are updated in different ways, and the MOPSO method and CSO method update the winners and losers, respectively, thus increasing convergence and diversity. By combining the updating strategies in MOPSO and CSO, the algorithm has attained favorable equilibrium in convergence and diversity. Therefore, the winners are updated by (2) and (3). Accordingly, the losers are updated to:

$$v_i^{t+1} = r_3 v_i^t + r_4 (NS_i^t - x_i^t) \quad (13)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (14)$$

where  $r_3$  and  $r_4$  represent two random numbers uniformly generated in  $[0,1]$ , and  $NS_i^t$  is a sample of learning randomly selected by the losers from the archive.

### The archive maintenance of optimal angle distance

In most existing MOPSOs, as the number of iterations of the algorithm increases, so does the number of non-dominated solutions. Due to the limited size of the archive, an adequate strategy is needed for the effective maintenance of archive. Therefore, in CHLMOPSO, a strategy combining the adaptive grid and the optimal angle distance (OAD) is proposed to maintain the external archive. As shown in Algorithm 3,

when the number of non-dominated solutions in the archive is larger than predefined archive size, the solutions with high density in the adaptive grid are deleted by using the OAD strategy, so that the number of stored solutions does not exceed the predefined archive size, so as to improve convergence speed of the algorithm in the evolution process.

*Procedure 1:* The optimal angle distance considers both the convergence distance (ConD) of solutions and their minimum angle to other solutions with high density. First of all, the convergence distance of the non-dominated solutions is calculated, and the fitness values of solutions need to be transformed before calculation. Denote the fitness values of all solutions in the grid  $G$  which have the largest density as  $F_G = \{F(NS_1), F(NS_2), \dots, F(NS_{|G|})\}$ , where  $|G|$  is the number of solutions in  $G$ . Transform  $F_G$  to  $F'_G = \{F'(NS_1), F'(NS_2), \dots, F'(NS_{|G|})\}$  by the following method:

$$F'(NS_i) = F(NS_i) - z_G^{\min} \quad (15)$$

where  $NS_i$  is the  $i$ -th non-dominated solution in the grid  $G$ ,  $i = 1, 2, \dots, |G|$ ,  $F(NS_i)$  and  $F'(NS_i)$  are the fitness values before and after the transformation of the  $i$ -th non-dominated solution in the grid  $G$ , respectively, and  $z_G^{\min} = (z_{G,1}^{\min}, z_{G,2}^{\min}, \dots, z_{G,m}^{\min})$  represents the smallest fitness value in each dimension of  $F_G$ . Then, the convergence distance of the solutions is calculated by Eq. (16).

$$ConD(NS_i) = \|F'(NS_i)\|_2 \quad (16)$$

*Procedure 2:* Next, the optimal angle distance strategy for external archive maintenance is described. As shown in Eq. (17), the OAD consists of the ratio of the convergence distance of the non-dominated solution to the minimum angle, hence, the solutions can be evaluated more comprehensively. The smaller OAD value, the better quality of the solutions. The OAD is defined as:

$$OAD(NS_i) = \frac{ConD(NS_i)}{\min_{NS_j \in G, j \neq i} \{F(NS_i), F(NS_j)\}} \quad (17)$$

where  $F(NS_i)$  and  $F(NS_j)$  stand for the fitness values of the solutions  $NS_i$  and  $NS_j$ , respectively, and  $\angle(F(NS_i), F(NS_j))$  is the angle between the vectors  $F(NS_i)$  and  $F(NS_j)$ .

At first, the initial population  $P$  and archive  $A$  are generated in the initialization stage of lines 1–2. After that, CHLMOPSO begins the cycle of the population search pro-

---

**Algorithm 3:** The archive maintenance of optimal angle distance

---

**Input:** The updated population  $P$ , the archive  $A$ , the predefined archive size  $N_A$ .

**Output:** The updated archive  $A$ .

- 1: Add the updated population to the archive  $A$ ;
- 2: Update the archive by Pareto dominance relationship firstly;
- 3: Calculate the density of particles by adaptive grid strategy;
- 4: **while**  $\text{size}(A) > N_A$  **do**
- 5: Find the grid  $G$  with the most non-dominated solutions;
- 6: Translate the fitness values by Eq. (15);
- 7: Calculate the ConD values by Eq. (16);
- 8: Calculate the optimal angle distance of the solutions by Eq. (17);
- 9: Sort the OAD of the solutions in ascending order;
- 10: Remove the solutions with larger OAD values from grid  $G$ ;
- 11: **end while**
- 12: **return**  $A$ .

---

## General framework of CHLMOPSO

The main components of CHLMOPSO are introduced in detail above, namely the derivative treatment of pbest of the stagnant particles, the adaptive adjustment of flight parameters, the competitive hybrid learning strategy, and the maintenance of external archive through the optimal angle distance. The pseudo-code of CHLMOPSO is shown in Algorithm 4.

cess in lines 3–8. In line 4, firstly, the pbest of stagnant particles is derived by Algorithm 1. Then, in line 5, as described in section "Adaptive strategy for parameters adjustment", the flight parameters are adjusted adaptively with respect to the evolutionary state of particles to balance the exploration and exploitation abilities. Then, in line 6, the competitive hybrid learning strategy in Algorithm 2 is used to update the particles. Finally, in line 7, the optimal angle distance of Algorithm 3 is used to effectively maintain the archive. The above-mentioned evolutionary process is duplicated until the maximum number of iterations is achieved and the final archive  $A$  is output.

---

**Algorithm 4:** The general framework of CHLMOPSO

---

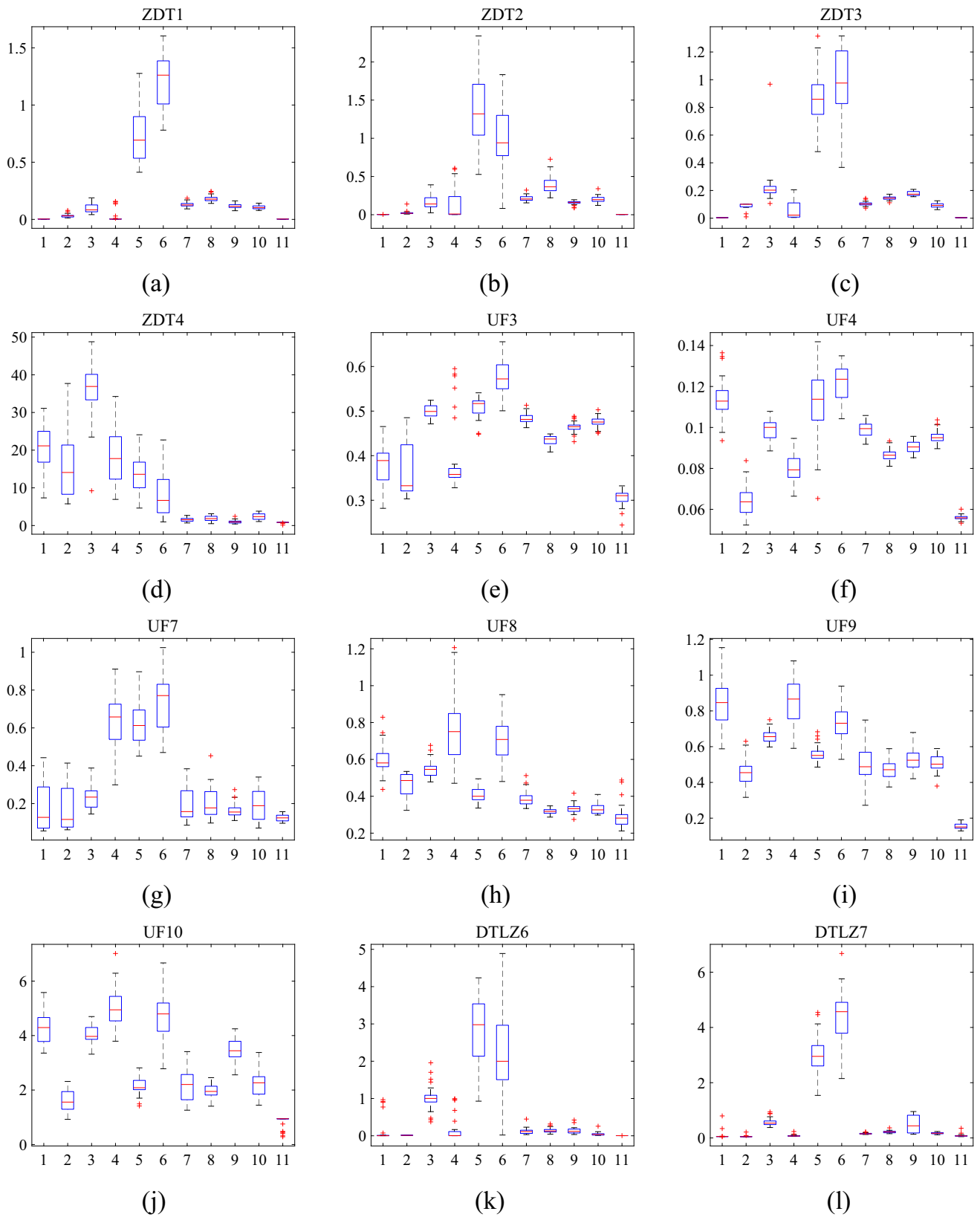
**Input:**  $N$ ,  $N_A$ , and  $T$ .

**Output:** The external archive  $A$ .

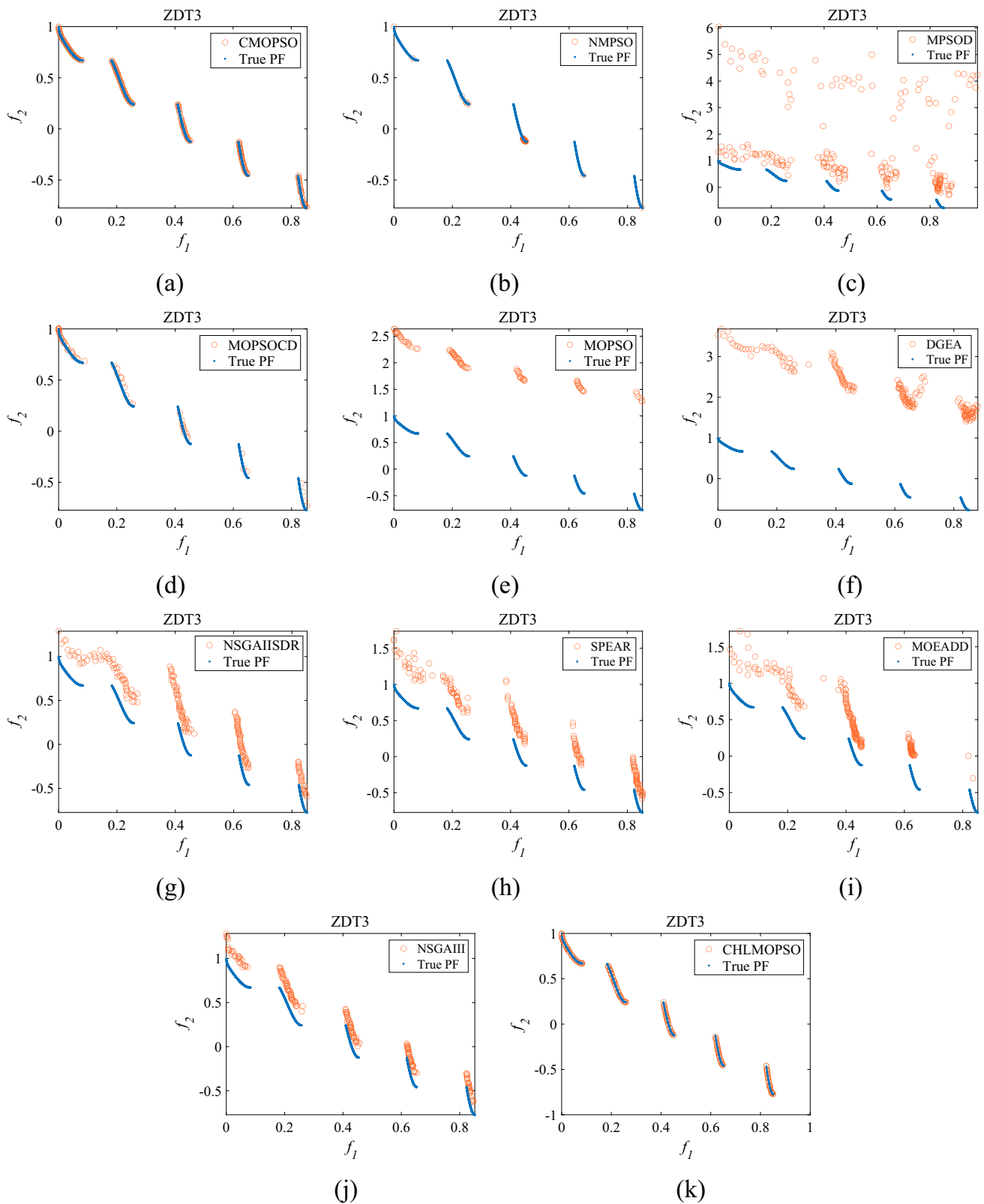
- 1: Initialize population  $P$ ;
- 2: Initialize an empty archive  $A$ ;
- 3: **while**  $t < T$  **do**
- 4: Update pbest using the **Algorithm 1**;
- 5: Adjust the flight parameters adaptively according to Section 3.2;
- 6: Update the winners and losers using the **Algorithm 2**;
- 7: Update the archive  $A$  using the **Algorithm 3**;
- 8: **end while**
- 9: **return**  $A$ .

---

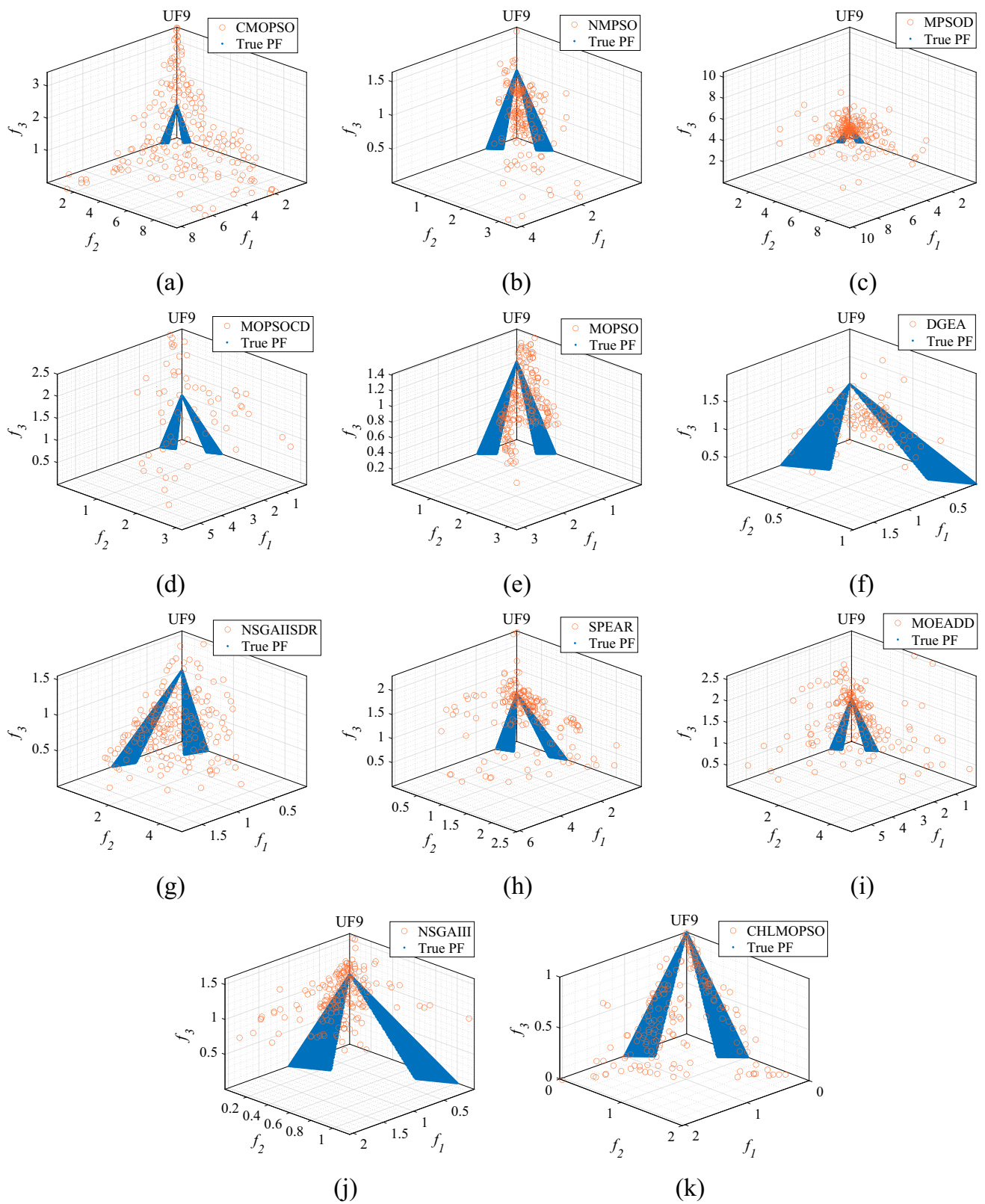




**Fig. 2** Box statistics of the IGD values of eleven algorithms on different test problems



**Fig. 3** Approximation PF of eleven algorithms on ZDT3 test problem



**Fig. 4** Approximation PF of eleven algorithms on UF9 test problem

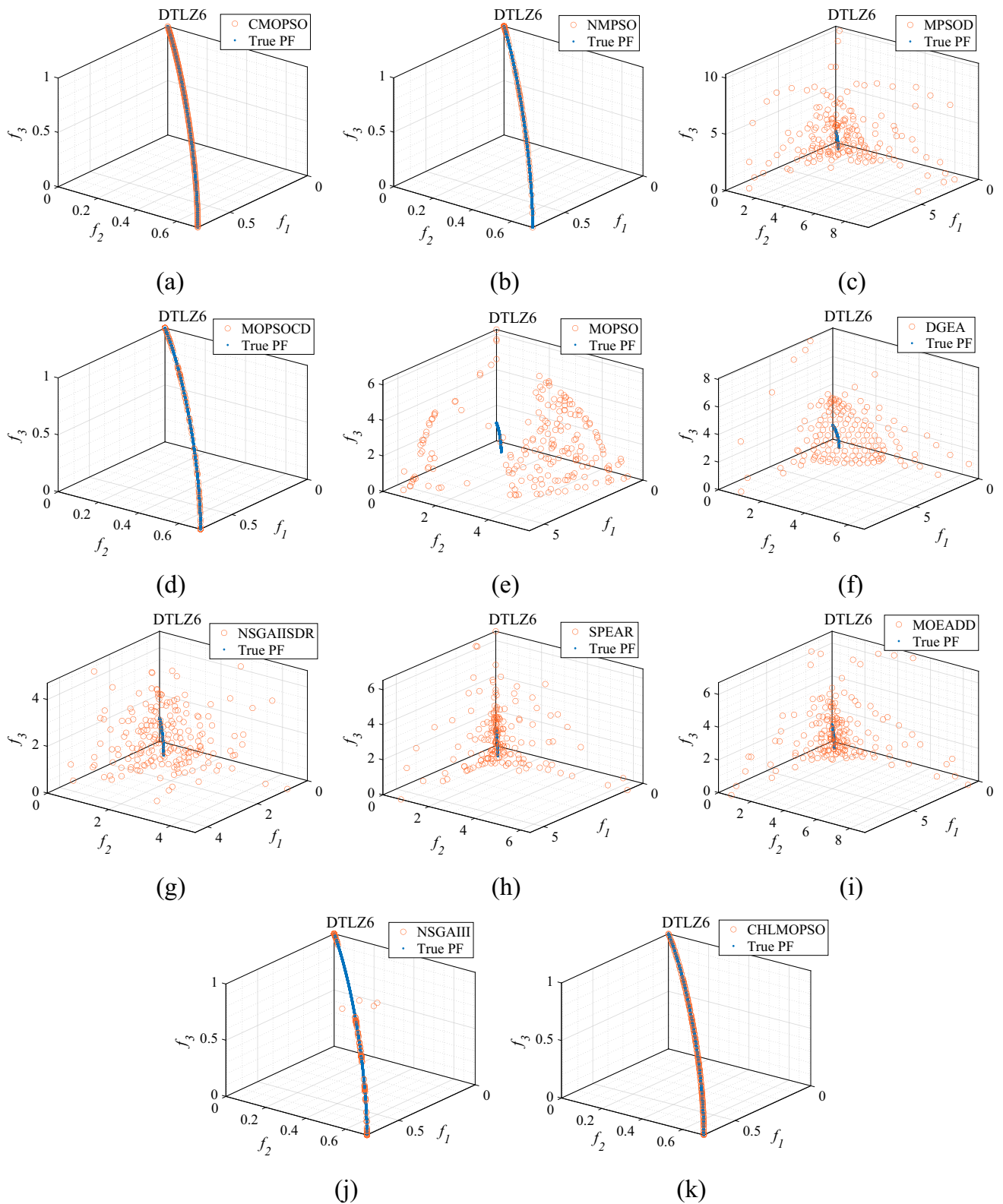
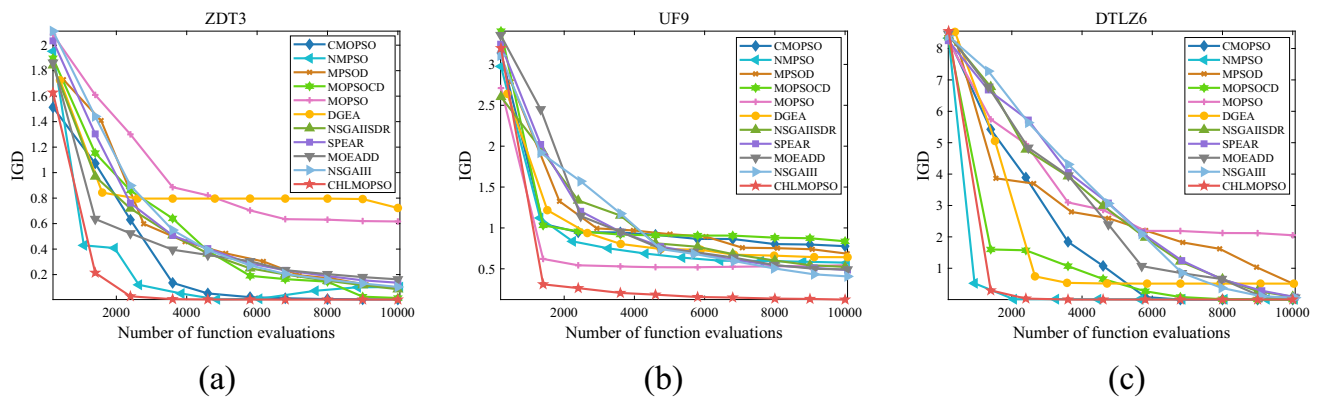


Fig. 5 Approximation PF of eleven algorithms on DTLZ6 test problem



**Fig. 6** IGD convergence trajectories of CHLMOPSO and ten comparison algorithms on ZDT3, UF9, and DTLZ6 test problems

**Table 1** Parameter settings of test problems

Problems	N	M	D	FEs
ZDT1 ~ ZDT3	200	2	30	10,000
ZDT4 and ZDT6	200	2	10	10,000
UF1 ~ UF7	200	2	30	10,000
UF8 ~ UF10	200	3	30	10,000
DTLZ1	200	3	7	10,000
DTLZ2 ~ DTLZ6	200	3	12	10,000
DTLZ7	200	3	22	10,000

**Table 2** IGD values of CMOPSO, NMPSO, MPSOD, MOPSOCD, MOPSO, and CHLMOPSO on 22 test problems

Problem	IGD	CMOPSO	NMPSO	MPSOD	MOPSOCD	MOPSO	CHLMOPSO
ZDT1	A	3.1961e-3	3.2785e-2	9.7802e-2	2.3465e-2	7.3347e-1	<b>3.0056e-3</b>
	S	4.79e-4	1.52e-2	4.15e-2	5.03e-2	2.22e-1	<b>1.18e-4</b>
	W	≈	—	—	≈	—	
ZDT2	A	<b>2.9273e-3</b>	2.4506e-2	1.6670e-1	1.4050e-1	1.3435e+0	3.2928e-3
	S	<b>4.12e-4</b>	2.33e-2	9.12e-2	1.89e-1	4.12e-1	1.71e-4
	W	+	—	—	—	—	
ZDT3	A	3.8764e-3	8.9527e-2	2.2761e-1	5.5138e-2	8.6668e-1	<b>3.8435e-3</b>
	S	8.76e-4	2.10e-2	1.44e-1	5.47e-2	1.96e-1	<b>2.43e-4</b>
	W	≈	—	—	—	—	
ZDT4	A	2.0282e+1	1.6472e+1	3.5312e+1	1.8825e+1	1.3601e+1	<b>7.9258e-1</b>
	S	6.02e+0	9.59e+0	7.84e+0	7.24e+0	4.90e+0	<b>1.37e-1</b>
	W	—	—	—	—	—	
ZDT6	A	<b>1.5779e-3</b>	2.2303e-3	1.7975e-2	3.5327e-3	3.1725e-1	2.4231e-3
	S	<b>2.90e-5</b>	1.91e-4	1.20e-2	2.59e-3	1.02e+0	9.74e-5
	W	+	+	—	—	—	
UF1	A	<b>9.7306e-2</b>	1.2996e-1	2.7838e-1	6.3427e-1	5.3461e-1	1.4149e-1

**Table 2** (continued)

Problem	IGD	CMOPSO	NMPSO	MPSOD	MOPSOCD	MOPSO	CHLMOPSO
UF2	S	<b>2.37e-2</b>	3.09e-2	5.59e-2	1.20e-1	1.00e-1	1.42e-2
	W	+	+	-	-	-	
	A	<b>6.4510e-2</b>	8.2393e-2	1.1551e-1	1.4207e-1	1.0389e-1	8.0615e-2
UF3	S	<b>6.97e-3</b>	6.75e-3	8.13e-3	1.51e-2	9.48e-3	5.63e-3
	W	+	≈	-	-	-	
	A	3.7866e-1	3.6336e-1	4.9795e-1	3.9439e-1	5.0913e-1	<b>3.0547e-1</b>
UF4	S	4.46e-2	5.66e-2	1.44e-2	8.20e-2	2.30e-2	<b>1.77e-2</b>
	W	-	-	-	-	-	
	A	1.1398e-1	6.3932e-2	9.8983e-2	7.9956e-2	1.1184e-1	<b>5.5948e-2</b>
UF5	S	9.87e-3	7.39e-3	4.98e-3	7.40e-3	1.69e-2	<b>1.28E-03</b>
	W	-	-	-	-	-	
	A	<b>9.3327e-1</b>	1.6477e+0	2.7466e+0	3.7870e+0	3.3339e+0	1.6084e+0
UF6	S	<b>2.24e-1</b>	3.92e-1	2.82e-1	5.12e-1	3.62e-1	2.00e-1
	W	+	≈	-	-	-	
	A	<b>3.9479e-1</b>	6.6775e-1	1.3760e+0	2.8768e+0	2.2673e+0	7.1366e-1
UF7	S	<b>6.16e-2</b>	1.50e-1	2.16e-1	7.53e-1	4.23e-1	9.24e-2
	W	+	≈	-	-	-	
	A	1.8428e-1	1.7558e-1	2.4152e-1	6.3686e-1	6.2616e-1	<b>1.2309e-1</b>
UF8	S	1.24e-1	1.16e-1	6.69e-2	1.51e-1	1.08e-1	<b>1.67e-2</b>
	W	-	≈	-	-	-	
	A	5.9969e-1	4.6498e-1	5.4983e-1	7.5343e-1	4.0805e-1	<b>2.9240e-1</b>
UF9	S	8.23e-2	6.43e-2	4.63e-2	1.73e-1	3.97e-2	<b>6.51e-2</b>
	W	-	-	-	-	-	
	A	8.5292e-1	4.6245e-1	6.5764e-1	8.6314e-1	5.5987e-1	<b>1.5506e-1</b>
UF10	S	1.25e-1	7.05e-2	3.73e-2	1.27e-1	4.57e-2	<b>1.74e-2</b>
	W	-	-	-	-	-	
	A	4.3049e+0	1.5942e+0	4.0568e+0	4.9685e+0	2.1334e+0	<b>8.3000e-1</b>
DTLZ1	S	5.86e-1	4.13e-1	3.16e-1	7.28e-1	2.91e-1	<b>2.24e-1</b>
	W	-	-	-	-	-	
	A	1.4001e+	<b>5.4379e+</b>	1.1087e+1	1.8504e+1	9.9132e+0	1.9836e+1
DTLZ2	S	4.10e+0	3.13e+0	2.42e+0	4.02e+0	4.23e+0	4.59e+0
	W	+	+	+	≈	+	
	A	<b>4.1147e-2</b>	5.5989e-2	4.4783e-2	9.1093e-2	5.4359e-2	2.5435e-1
DTLZ3	S	<b>6.57e-4</b>	1.92e-3	1.28e-3	7.83e-3	3.80e-3	3.95e-2
	W	+	+	+	+	+	
	A	1.8571e+2	<b>1.1626e+2</b>	1.4668e+2	1.3602e+2	1.8989e+2	1.4550e+2

## Experimental settings

### Experimental comparison algorithms

For the sake of estimating the performance of CHLMOPSO more objectively, ten algorithms are selected for comparison, including five advanced MOPSOs (CMOPSO [30], NMPSO [31], MPSOD [17], MOPSOCD [32], and MOPSO [9]) and five competitive MOEAs (DGEA [33], NSGAII [34],

SPEAR [35], MOEADD [36], and NSGAIII [37]). All relevant parameters set by the comparison algorithms remain consistent with the original references to guarantee the impartiality of algorithm performance comparison. Each algorithm runs independently for 30 times on each test problem, and the average and standard deviation of the IGD and HV values are recorded. The experimental results of all algorithms are implemented on an Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz Windows 7 system by MATLAB

**Table 2** (continued)

Problem	IGD	CMOPSO	NMPSO	MPSOD	MOPSOCD	MOPSO	CHLMOPSO
DTLZ4	S	2.73e+1	1.76e+1	1.44e+1	4.02e+1	4.76e+1	2.93e+1
	W	—	+	≈	≈	—	
	A	<b>4.3279e−2</b>	7.4004e−2	1.4083e−1	3.3113e−1	3.1068e−1	2.9143e−1
	S	<b>1.41e−3</b>	8.83e−2	3.50e−2	4.61e−2	5.33e−2	3.58e−2
DTLZ5	W	+	+	+	—	≈	
	A	<b>4.6382e−3</b>	7.1710e−3	5.6792e−2	3.4002e−2	8.5109e−3	1.2344e−2
	S	<b>3.64e−4</b>	8.70e−4	5.94e−3	8.30e−3	2.64e−3	1.98e−3
DTLZ6	W	+	+	—	—	+	
	A	1.2468e−1	1.3457e−2	1.0284e+0	1.5378e−1	2.8161e+0	<b>3.7187e−3</b>
	S	3.07e−1	3.54e−3	3.33e−1	3.13e−1	8.75e−1	<b>3.39e−4</b>
DTLZ7	W	≈	—	—	—	—	
	A	8.5583e−2	<b>5.3346e−2</b>	5.6886e−1	8.1282e−2	3.0066e+0	8.5440e−2
	S	1.44e−1	<b>3.10e−2</b>	1.48e−1	3.55e−2	7.01e−1	5.57e−2
	W	—	+	+	≈	—	
+/-/≈		10/9/3	8/10/4	4/17/1	1/17/4	3/18/1	—
Best/all		9/22	3/22	0/22	0/22	0/22	10/22

**Table 3** IGD values of DGEA, NSGAIIISDR, SPEAR, MOEADD, NSGAIII, and CHLMOPSO on 22 test problems

Problem	IGD	DGEA	NSGAIIISDR	SPEAR	MOEADD	NSGAIII	CHLMOPSO
ZDT1	A	1.2077e+0	1.2905e−1	1.8116e−1	1.1602e−1	1.0424e−1	<b>3.0056e−3</b>
	S	2.25e−1	2.16e−2	2.57e−2	1.98e−2	1.60e−2	<b>1.18e−4</b>
	W	—	—	—	—	—	
ZDT2	A	9.9849e−1	2.1416e−1	3.9212e−1	1.5974e−1	2.0141e−1	<b>3.2928e−3</b>
	S	3.85e−1	3.56e−2	1.18e−1	2.54e−2	4.43e−2	<b>1.71e−4</b>
	W	—	—	—	—	—	
ZDT3	A	9.6505e−1	1.0455e−1	1.4539e−1	1.7759e−1	9.3487e−2	<b>3.8435e−3</b>
	S	2.43e−1	1.47e−2	1.35e−2	1.60e−2	1.62e−2	<b>2.43e−4</b>
	W	—	—	—	—	—	
ZDT4	A	8.3958e+0	1.5269e+0	1.8607e+0	9.8417e−1	2.4152e+0	<b>7.9258e−1</b>
	S	6.32e+0	5.01e−1	6.67e−1	4.29e−1	8.24e−1	<b>1.37e−1</b>
	W	—	—	—	≈	—	
ZDT6	A	1.7504e−2	1.0253e+0	1.1952e+	6.6621e−1	1.4526e+0	<b>2.4231e−3</b>
	S	8.46e−2	2.01e−1	2.14e−1	1.66e−1	3.15e−1	<b>9.74e−5</b>
	W	—	—	—	—	—	
UF1	A	6.9052e−1	1.4953e−1	1.5014e−1	1.5269e−1	1.5160e−1	<b>1.4149e−1</b>

R2021b. The PlatEMO [38] provides source codes of comparison algorithms.

### Test problems

Moreover, we selected 22 test problems from three different benchmark test suits, ZDT [39], UF [40], and DTLZ [41], to test the performance of these algorithms. They have different characteristics and complex Pareto front features, such

as concavity and convexity, multi-mode, disconnected, and irregular PF shapes, etc., which can well verify the reliability of the algorithm. There are relevant parameter settings in Table 1, where FEs represents the maximum number of evaluations.

Table 3 (continued)

Problem	IGD	DGEA	NSGAIISDR	SPEAR	MOEADD	NSGAIII	CHLMOPSO
UF2	S	1.37e-1	4.24e-2	2.95e-2	3.64e-2	3.86e-2	<b>1.42e-2</b>
	W	—	≈	≈	≈	≈	
	A	1.7395e-1	8.5657e-2	7.4438e-2	<b>7.3212e-2</b>	8.1883e-2	8.06E-02
UF3	S	2.05e-2	5.44e-3	8.75e-3	<b>5.32e-3</b>	5.39e-3	5.63e-3
	W	—	—	+	+	≈	
	A	5.7249e-1	4.8299e-1	4.3379e-1	4.6408e-1	4.7550e-1	<b>3.0547e-1</b>
UF4	S	3.94e-2	1.19e-2	1.15e-2	1.21e-2	1.19e-2	<b>1.77e-2</b>
	W	—	—	—	—	—	
	A	1.2199e-1	9.9070e-2	8.6396e-2	9.0427e-2	9.5550e-2	<b>5.5948e-2</b>
UF5	S	8.84e-3	3.54e-3	2.98e-3	2.82e-3	3.15e-3	<b>1.28e-3</b>
	W	—	—	—	—	—	
	A	3.0556e+0	1.3926e+0	<b>1.1443e+0</b>	1.3742e+0	1.3892e+0	1.6084e+0
UF6	S	4.62e-1	3.25e-1	<b>1.57e-1</b>	2.70e-1	3.14e-1	2.00E-01
	W	—	+	+	+	+	
	A	2.2859e+0	7.3621e-1	7.1858e-1	8.1831e-1	7.5425e-1	<b>7.1366e-1</b>
UF7	S	7.18e-1	1.18e-1	2.85e-1	1.14e-1	1.72e-1	<b>9.24e-2</b>
	W	—	≈	≈	—	≈	
	A	7.4477e-1	2.0669e-1	1.9993e-1	1.6423e-1	1.9537e-1	<b>1.2309e-1</b>
UF8	S	1.40e-1	9.36e-2	8.14e-2	4.12e-2	8.00e-2	<b>1.67e-2</b>
	W	—	—	—	—	—	
	A	7.0365e-1	3.8627e-1	3.1851e-1	3.3305e-1	3.3350e-1	<b>2.9240e-1</b>
UF9	S	1.23e-1	4.01e-2	1.50e-2	2.53e-2	3.08e-2	<b>6.51e-2</b>
	W	—	—	—	—	—	
	A	7.2837e-1	5.0192e-1	4.7259e-1	5.2958e-1	5.0618e-1	<b>1.5506e-1</b>
UF10	S	9.36e-2	1.00e-1	4.71e-2	6.40e-2	4.58e-2	<b>1.74e-2</b>
	W	—	—	—	—	—	
	A	4.7449e+0	2.1774e+	1.9678e+0	3.4898e+0	2.2698e+0	<b>8.3000e-1</b>
DTLZ1	S	9.32e-1	5.75e-1	2.35e-1	4.50e-1	5.19e-1	<b>2.24e-1</b>
	W	—	—	—	—	—	
	A	1.4659e+1	<b>5.2426e-1</b>	9.8136e-1	1.3217e+0	9.2275e-1	1.9836e+1
DTLZ2	S	7.84e+0	<b>3.63e-1</b>	3.81e-1	5.91e-1	3.44e-1	4.59e+0
	W	+	+	+	+	+	
	A	1.1855e-1	4.2239e-1	4.3210e-2	<b>3.9378e-2</b>	3.9824e-2	2.54E-01
DTLZ3	S	1.63e-2	4.28e-2	1.95e-3	<b>4.62e-4</b>	6.87e-4	3.95e-2
	W	+	—	+	+	+	
	A	1.1440e+2	<b>2.1763e+1</b>	2.9695e+1	4.3925e+1	3.3734e+1	1.4550e+2

### Comprehensive performance evaluation indicators

The performance of algorithms is assessed by adopting the extensively acknowledged inverted generational distance (IGD) [42] and hypervolume (HV) [43]. The quality of the Pareto optimal solution set that the algorithm ultimately generates is superior with the smaller IGD value (or the larger HV value) of an algorithm.

### Comparison of experimental results

In this paper, a competitive hybrid learning MOPSO is put forward. We compared CHLMOPSO with other MOPSOs and MOEAs to further assess its effectiveness and analyzed the overall performance of CHLMOPSO from the following aspects.



**Table 3** (continued)

Problem	IGD	DGEA	NSGAIISDR	SPEAR	MOEADD	NSGAIII	CHLMOPSO
DTLZ4	S	6.25e+1	<b>6.85e+0</b>	8.29e+0	8.10e+0	8.16e+0	2.93e+1
	W	+	+	+	+	+	
	A	4.0165e-1	4.5289e-1	4.4194e-2	<b>4.0048e-2</b>	4.1080e-2	2.91E-01
DTLZ5	S	1.00e-1	1.38e-1	1.51e-3	<b>7.72e-4</b>	7.15e-4	3.58e-2
	W	-	-	+	+	+	
	A	5.9983e-2	3.0359e-2	2.2011e-2	2.3132e-2	<b>7.0994e-3</b>	1.2344e-2
DTLZ6	S	1.12e-2	3.88e-3	1.85e-3	1.47e-3	<b>6.75e-4</b>	1.98e-3
	W	-	-	-	-	+	
	A	2.1087e+0	1.2027e-1	1.3967e-1	1.3655e-1	4.7538e-2	<b>3.72E-03</b>
DTLZ7	S	1.21e+0	7.99e-2	6.61e-2	8.73e-2	4.63e-2	<b>3.39e-4</b>
	W	-	-	-	-	-	
	A	4.3938e+0	1.5527e-1	2.2097e-1	4.9782e-1	1.7125e-1	<b>8.5440e-2</b>
+/-/≈	S	9.64e-1	2.34e-2	4.30e-2	3.15e-1	3.30e-2	<b>5.57e-2</b>
	W	-	-	-	-	-	
	Best/all	3/19/0	3/17/2	6/14/2	6/14/2	6/13/3	—
Best/all	0/22	2/22	1/22	3/22	1/22	15/22	

**Table 4** HV values of CMOPSO, NMPSO, MPSOD, MOPSOCD, MOPSO, and CHLMOPSO on 22 test problems

Problem	HV	CMOPSO	NMPSO	MPSOD	MOPSOCD	MOPSO	CHLMOPSO
ZDT1	A	7.1985e-1	6.8722e-1	5.7991e-1	6.9484e-1	9.0486e-2	<b>7.2124e-1</b>
	S	7.51e-4	1.53e-2	5.69e-2	6.32e-2	8.72e-2	<b>1.44e-4</b>
	W	-	-	-	-	-	
ZDT2	A	4.4472e-1	4.2762e-1	2.4874e-1	3.2089e-1	1.0419e-3	<b>4.4582e-1</b>
	S	7.09e-4	2.71e-2	8.92e-2	1.50e-1	5.71e-3	<b>1.26e-4</b>
	W	-	-	-	-	-	
ZDT3	A	5.9946e-1	5.7217e-1	4.4330e-1	5.6876e-1	6.7854e-2	<b>6.0007e-1</b>
	S	9.78e-4	8.19e-3	8.88e-2	3.84e-2	7.69e-2	<b>6.33e-4</b>
	W	-	-	-	-	-	
ZDT4	A	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	<b>1.3220e-1</b>
	S	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	<b>1.05e-1</b>
	W	-	-	-	-	-	
ZDT6	A	<b>3.9034e-1</b>	3.8983e-1	3.7443e-1	3.8848e-1	3.0174e-1	3.8935e-1
	S	<b>3.41e-5</b>	1.65e-4	1.15e-2	2.59e-3	1.39e-1	1.64e-4
	W	+	+	-	≈	-	
UF1	A	<b>5.7906e-1</b>	5.2221e-1	3.4986e-1	7.4320e-2	1.4899e-1	5.0017e-1

**Comparison of IGD data**

The average and standard deviation of the IGD values for CHLMOPSO and other MOPSOs and MOEAs on 22 test problems are displayed in Tables 2 and 3, respectively, and are denoted by ‘A’ and ‘S’. The best IGD values are displayed in bold. Furthermore, for the purpose of ensuring statistically reliable conclusions, and the data of all algorithms are tested for normality, it is found that some data do not conform to the normal distribution, so we introduced a non-parametric

statistical hypothesis test to process the experimental results. At the significance level of  $\alpha = 0.05$ , the Wilcoxon rank sum test [44] is utilized to evaluate if there are meaningful differences between the algorithms. The corresponding symbols ‘+’, ‘-’, and ‘≈’ represent that the results of other algorithms are apparently superior to CHLMOPSO, apparently inferior to CHLMOPSO, and statistically similar to CHLMOPSO, respectively.

The comprehensive performance of CHLMOPSO is superior to MOPSOs and MOEAs. In accordance with the results

Table 4 (continued)

Problem	HV	CMOPSO	NMPSO	MPSOD	MOPSOCD	MOPSO	CHLMOPSO
UF2	S	<b>2.02e-2</b>	4.20e-2	6.04e-2	5.12e-2	6.90e-2	2.17e-2
	W	+	+	-	-	-	
	A	<b>6.4205e-1</b>	6.1909e-1	5.7659e-1	5.3607e-1	6.0304e-1	6.2844e-1
UF3	S	<b>7.60e-3</b>	6.67e-3	9.09e-3	2.07e-2	8.62e-3	5.74e-3
	W	+	-	-	-	-	
	A	2.7318e-1	2.7536e-1	1.7472e-1	2.3895e-1	1.6092e-1	<b>3.3451e-1</b>
UF4	S	3.26e-2	5.27e-2	1.34e-2	6.31e-2	1.64e-2	<b>2.23e-2</b>
	W	-	-	-	-	-	
	A	2.8584e-1	3.5965e-1	3.0804e-1	3.3164e-1	2.9249e-1	<b>3.6438e-1</b>
UF5	S	1.20e-2	1.07e-2	6.06e-3	9.26e-3	2.08e-2	<b>2.22e-3</b>
	W	-	≈	-	-	-	
	A	<b>8.3362e-3</b>	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
UF6	S	<b>2.38e-2</b>	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
	W	+	≈	≈	≈	≈	
	A	<b>1.4132e-1</b>	3.3171e-2	0.0000e+0	0.0000e+0	0.0000e+0	3.0084e-3
UF7	S	<b>6.86e-2</b>	3.08e-2	0.00e+0	0.00e+0	0.00e+0	5.91e-3
	W	+	+	-	-	-	
	A	3.9666e-1	3.8885e-1	2.6557e-1	3.1819e-2	3.9082e-2	<b>4.0005e-1</b>
UF8	S	8.80e-2	9.28e-2	6.74e-2	4.70e-2	3.27e-2	<b>2.28e-2</b>
	W	≈	≈	-	-	-	
	A	1.2540e-2	2.9571e-1	5.6163e-2	1.2088e-2	1.7199e-1	<b>3.3336e-1</b>
UF9	S	1.42e-2	3.07e-2	2.10e-2	1.63e-2	3.61e-2	<b>1.36e-2</b>
	W	-	-	-	-	-	
	A	3.2290e-2	3.1536e-1	1.1635e-1	2.9801e-2	1.9774e-1	<b>5.9686e-1</b>
UF10	S	3.48e-2	6.47e-2	2.74e-2	3.78e-2	3.83e-2	<b>2.08e-2</b>
	W	-	-	-	-	-	
	A	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	<b>1.1310e-1</b>
DTLZ1	S	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	<b>5.08e-2</b>
	W	-	-	-	-	-	
	A	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
DTLZ2	S		0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
	W	≈	≈	≈	≈	≈	
	A	5.5929e-1	<b>5.6939e-1</b>	5.5152e-1	4.8237e-1	5.3133e-1	3.3691e-1
	S	1.43e-3	<b>8.40e-4</b>	2.56e-3	1.18e-2	8.12e-3	2.23e-2
	W	+	+	+	+	+	

of the Wilcoxon rank sum test from Table 2, among all the 22 test problems considered, the comparison algorithms CMOPSO, NMPSO, MPSOD, MOPSOCD, and MOPSO perform significantly worse than CHLMOPSO on 9, 10, 17, 17, and 18 test problems, respectively, and obtain similar results with CHLMOPSO on 3, 4, 1, 4, and 1 test problems, respectively. Additionally, as can be observed from Table 3, CHLMOPSO is noticeably superior to the comparison algorithms DGEA, NSGAIISDR, SPEAR, MOEADD, and NSGAI on 19, 17, 14, 14, and 13 of the 22 test problems, respectively, and similar results are obtained on 0, 2, 2,

2, and 3 test problems, respectively. Moreover, the standard deviations demonstrate that when compared with the other comparison algorithms, CHLMOPSO has a certain advantage in terms of stability.

The outcomes listed above demonstrate that CHLMOPSO performs well on most of the test problems. This manifests that the competitive hybrid learning strategy improves the exploration ability and effectively balances the exploitation and exploration abilities.

**Table 4** (continued)

Problem	HV	CMOPSO	NMPSO	MPSOD	MOPSOCD	MOPSO	CHLMOPSO
DTLZ3	A	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
	S	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
	W	≈	≈	≈	≈	≈	
DTLZ4	A	5.5534e-1	<b>5.6140e-1</b>	4.2138e-1	2.6829e-1	3.3707e-1	3.3379e-1
	S	3.22e-3	<b>4.05e-2</b>	4.39e-2	6.36e-2	5.78e-2	5.06e-2
	W	+	+	+	-	≈	
DTLZ5	A	<b>1.9886e-1</b>	1.9794e-1	1.3831e-1	1.7376e-1	1.9385e-1	1.9102e-1
	S	<b>2.94e-4</b>	5.13e-4	8.67e-3	8.25e-3	3.24e-3	1.92e-3
	W	+	+	-	-	+	
DTLZ6	A	1.7301e-1	1.9790e-1	9.2771e-3	1.6595e-1	0.0000e+0	<b>2.0034e-1</b>
	S	6.91e-2	8.53e-4	2.87e-2	6.43e-2	0.00e+0	<b>3.19e-4</b>
	W	≈	-	-	-	-	
DTLZ7	A	2.6796e-1	<b>2.8071e-1</b>	6.3204e-2	2.6597e-1	3.0175e-3	1.4812e-1
	S	1.38e-2	<b>4.54e-3</b>	3.76e-2	1.13e-2	1.65e-2	9.46e-3
	W	+	+	-	+	-	
+/-/≈		9/9/4	7/10/5	2/17/3	2/16/4	2/16/4	—
Best/all		6/22	3/22	0/22	0/22	0/22	11/22

**Table 5** HV values of DGEA, NSGAIISDR, SPEAR, MOEADD, NSGAI, and CHLMOPSO on 22 test problems

Problem	HV	DGEA	NSGAIISDR	SPEAR	MOEADD	NSGAI	CHLMOPSO
ZDT1	A	1.9351e-3	5.5501e-1	4.8905e-1	5.6540e-1	5.8600e-1	<b>7.2124e-1</b>
	S	6.21e-3	2.35e-2	2.35e-2	2.53e-2	1.90e-2	<b>1.44e-4</b>
	W	-	-	-	-	-	
ZDT2	A	1.5381e-2	1.9493e-1	7.9008e-2	2.3956e-1	2.1027e-1	<b>4.4582e-1</b>
	S	6.44e-2	2.80e-2	5.65e-2	2.65e-2	3.54e-2	<b>1.26e-4</b>
	W	-	-	-	-	-	
ZDT3	A	4.7343e-2	5.3209e-1	5.0615e-1	4.7779e-1	5.3934e-1	<b>6.0007e-1</b>
	S	7.81e-2	1.12e-2	1.76e-2	1.52e-2	1.10e-2	<b>6.33e-4</b>
	W	-	-	-	-	-	
ZDT4	A	0.0000e+0	3.0638e-3	8.3073e-3	4.0321e-2	0.0000e+0	<b>1.3220e-1</b>
	S	0.00e+0	1.31e-2	3.31e-2	6.46e-2	0.00e+0	<b>1.05e-1</b>
	W	-	-	-	-	-	
ZDT6	A	3.8245e-1	1.6484e-4	0.0000e+0	1.2111e-2	0.0000e+	<b>3.8935e-1</b>
	S	4.08e-2	9.03e-4	0.00e+0	3.10e-2	0.00e+0	<b>1.64e-4</b>
	W	-	-	-	-	-	

**Comparison of HV data**

The outcomes of the best HV values in Table 4 are similar to those of IGD values in Table 2. Compared with the five MOPSOs, CHLMOPSO obtains 11 best HV values on 22 test problems. Moreover, CHLMOPSO performs better than CMOPSO, NMPSO, MPSOD, MOPSOCD, and MOPSO on 9, 10, 17, 16, and 16 test problems, respectively. Similarly, Table 5 displays that CHLMOPSO acquires 13 best HV values on 22 test problems compared with five MOEAs.

In addition, CHLMOPSO performs better than the comparison algorithms DGEA, NSGAIISDR, SPEAR, MOEADD, and NSGAI on 18, 14, 14, 14, and 13 of 22 test problems, respectively. In addition, from the comparison of standard deviations of algorithms in the two tables, it can also be found that CHLMOPSO has favorable stability.

All the above statistical results prove that CHLMOPSO is still very competitive compared with the selected algorithms. CHLMOPSO has better convergence and diversity in contrast to the selected comparison algorithms on ZDT and

Table 5 (continued)

Problem	HV	DGEA	NSGAIISDR	SPEAR	MOEADD	NSGAIII	CHLMOPSO
UF1	A	6.7111e-2	4.9828e-1	<b>5.0042e-1</b>	4.9562e-1	4.9953e-1	5.0017e-1
	S	6.22e-2	5.42e-2	<b>3.50e-2</b>	4.63e-2	4.96e-2	2.17e-2
	W	—	≈	≈	≈	≈	
UF2	A	5.1016e-1	6.0810e-1	6.2421e-1	6.2448e-1	6.1408e-1	<b>6.2844e-1</b>
	S	2.09e-2	6.96e-3	8.61e-3	6.46e-3	6.23e-3	<b>5.74e-3</b>
	W	—	—	—	—	—	
UF3	A	1.2160e-1	1.9201e-1	2.2046e-1	2.0287e-1	1.8933e-1	<b>3.3451e-1</b>
	S	2.06e-2	1.10e-2	1.15e-2	1.21e-2	9.08e-3	<b>2.23e-2</b>
	W	—	—	—	—	—	
UF4	A	2.7779e-1	3.0741e-1	3.2396e-1	3.1892e-1	3.1258e-1	<b>3.6438e-1</b>
	S	8.89e-3	4.45e-3	3.80e-3	3.09e-3	3.77e-3	<b>2.22e-3</b>
	W	—	—	—	—	—	
UF5	A	0.0000e+0	<b>5.6647e-4</b>	0.0000e+0	0.0000e+0	8.3424e-6	0.0000e+0
	S	0.00e+0	<b>3.10e-3</b>	0.00e+0	0.00e+0	4.57e-5	0.00e+0
	W	≈	≈	≈	≈	≈	
UF6	A	0.0000e+0	8.5768e-3	<b>1.5075e-2</b>	2.3512e-3	7.1497e-3	3.0084e-3
	S	0.00e+0	1.38e-2	<b>1.73e-2</b>	5.47e-3	1.39e-2	5.91e-3
	W	—	≈	+	≈	≈	
UF7	A	2.1266e-2	3.3428e-1	3.4003e-1	3.5538e-1	3.3420e-1	<b>4.0005e-1</b>
	S	3.57e-2	7.01e-2	6.04e-2	4.41e-2	7.83e-2	<b>2.28e-2</b>
	W	—	—	—	—	—	
UF8	A	2.9674e-2	1.0868e-1	1.8583e-1	1.5408e-1	2.2705e-1	<b>3.3336e-1</b>
	S	2.74e-2	3.52e-2	2.43e-2	3.70e-2	4.01e-2	<b>1.36e-2</b>
	W	—	—	—	—	—	
UF9	A	8.6029e-2	2.3222e-1	2.7185e-1	2.1465e-1	2.4249e-1	<b>5.9686e-1</b>
	S	5.66e-2	8.56e-2	4.81e-2	5.16e-2	4.07e-2	<b>2.08e-2</b>
	W	—	—	—	—	—	
UF10	A	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	<b>1.1310e-1</b>
	S	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	<b>5.08e-2</b>
	W	—	—	—	—	—	
DTLZ1	A	0.0000e+0	<b>1.1409e-1</b>	3.7883e-3	2.5080e-3	3.7897e-3	0.0000e+0
	S	0.00e+0	<b>2.02e-1</b>	2.04e-2	1.37e-2	1.87e-2	0.00e+0
	W	≈	+	≈	≈	≈	
DTLZ2	A	4.2024e-1	2.9050e-1	5.5627e-1	<b>5.6216e-1</b>	5.6186e-1	3.3691e-1
	S	2.64e-2	3.55e-2	2.81e-3	<b>1.44e-3</b>	1.73e-3	2.23e-2
	W	+	+	+	+	+	
DTLZ3	A	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+	0.0000e+0

UF test problems on 22 test problems, which may be due to the competitive hybrid learning strategy of our proposed algorithm can obtain a better performance guarantee.

### Friedman rank test

In the above two sections, CHLMOPSO is compared with MOPSOs and MOEAs in IGD indicator and HV indicator.

Next, we also conduct the Friedman rank test [45] on the results in Tables 2, 3, 4, 5 to further compare the overall performance of CHLMOPSO and the selected comparison algorithms, and the smaller value of the test result, the better. Tables 6 and 7 show the Friedman rank test and ranking of IGD and HV values of all algorithms on ZDT, UF, and DTLZ, respectively. The minimum values of the Friedman test and the minimum rankings in the tables are displayed in bold.

**Table 5** (continued)

Problem	HV	DGEA	NSGAIISDR	SPEAR	MOEADD	NSGAI	CHLMOPSO
	S	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
	W	≈	≈	≈	≈	≈	
DTLZ4	A	2.1618e−1	3.0606e−1	5.5376e−1	<b>5.6129e−1</b>	5.6032e−1	3.3379e−1
	S	8.43e−2	8.00e−2	3.55e−3	<b>2.17e−3</b>	1.71e−3	5.06e−2
	W	−	≈	+	+	+	
DTLZ5	A	1.4315e−1	1.7497e−1	1.8593e−1	1.8448e−1	<b>1.9647e−1</b>	1.9102e−1
	S	1.88e−2	2.95e−3	1.50e−3	1.70e−3	<b>7.71e−4</b>	1.92e−3
	W	−	−	−	−	+	
DTLZ6	A	1.7314e−2	1.0746e−1	1.0357e−1	1.0098e−1	1.7093e−1	<b>2.0034e−1</b>
	S	5.33e−2	4.50e−2	3.70e−2	4.22e−2	2.70e−2	<b>3.19e−4</b>
	W	−	−	−	−	−	
DTLZ7	A	6.6795e−7	<b>2.1793e−1</b>	1.9382e−1	2.0328e−1	2.0891e−1	1.4812e−1
	S	3.66e−6	<b>9.63e−3</b>	1.25e−2	2.18e−2	1.29e−2	9.46e−3
	W	−	+	+	+	+	
+/-/≈		1/18/3	3/14/5	4/14/4	3/14/5	4/13/5	−
Best/all		0/22	3/22	2/22	2/22	1/22	13/22

**Table 6** Friedman rank test of IGD values of CHLMOPSO and all comparison algorithms

Algorithm	ZDT: ZDT1-4, 6		UF: UF1-10		DTLZ: DTLZ1-7		Overall	
	Friedman test	Rank	Friedman test	Rank	Friedman test	Rank	Friedman test	Rank
CMOPSO	3.20	2	4.90	5	4.86	5	4.50	3
NMPSO	4.20	3	3.50	2	4.14	2	3.86	2
MPSOD	7.40	8	8.10	8	7.86	9	7.86	9
MOPSOCD	4.60	4	9.20	10	7.57	8	7.64	8
MOPSO	9.00	11	8.10	8	8.00	10	8.27	10
DGEA	8.60	10	10.30	11	9.29	11	9.59	11
NSGAIISDR	6.80	6	5.80	7	5.86	6	6.05	7
SPEAR	7.80	9	3.70	3	4.71	4	4.95	5
MOEADD	6.00	5	4.80	4	4.43	3	4.95	5
NSGAI	6.80	6	5.40	6	<b>2.86</b>	<b>1</b>	4.91	4
CHLMOPSO	<b>1.60</b>	<b>1</b>	<b>2.20</b>	<b>1</b>	6.43	7	<b>3.41</b>	<b>1</b>

Table 6 clearly demonstrates that CHLMOPSO carries out significantly better than other algorithms on ZDT and UF with respect to the ranking of IGD values, and its ranking of the Friedman rank test is the first. Although the ranking on DTLZ is not ideal, from the overall result, CHLMOPSO still ranks first in Friedman rank test. Similarly, according to the ranking of HV values in Table 7, CHLMOPSO ranks first in Friedman rank test for both ZDT and UF test problems. Although it ranks second in the overall ranking, it is also second only to NMPSO. The aforementioned results illustrate that CHLMOPSO is considerably better overall than the other ten algorithms on the chosen test problems.

### Comparison of box plots

To intuitively compare the stability on three suits of benchmark test problems ZDT, UF, and DTLZ, Fig. 2 exhibits statistical box plots of IGD values distribution acquired by CHLMOPSO and other ten algorithms running independently for 30 times on some test problems. Among them, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11 on the horizontal axis of the box plots are CMOPSO, NMPSO, MPSOD, MOPSOCD, MOPSO, DGEA, NSGAIISDR, SPEAR, MOEADD, NSGAI, and CHLMOPSO in turn. It can be observed from the box plots that the gap between CHLMOPSO and CMOPSO is not obvious on

**Table 7** Friedman rank test of HV values of CHLMOPSO and all comparison algorithms

Algorithm	ZDT: ZDT1-4, 6		UF: UF1-10		DTLZ: DTLZ1-7		Overall	
	Friedman test	Rank	Friedman test	Rank	Friedman test	Rank	Friedman test	Rank
CMOPSO	3.00	2	4.55	4	4.00	3	4.02	3
NMPSO	4.00	3	3.40	2	<b>3.00</b>	<b>1</b>	<b>3.41</b>	<b>1</b>
MPSOD	6.80	6	8.05	8	8.00	10	7.75	9
MOPSOCD	4.60	4	8.25	10	7.00	8	7.02	8
MOPSO	9.20	11	8.05	8	7.57	9	8.16	10
DGEA	9.00	10	9.65	11	9.14	11	9.34	11
NSGAIISDR	7.00	7	5.85	7	6.43	6	6.30	7
SPEAR	7.70	9	4.50	3	5.57	5	5.57	5
MOEADD	6.20	5	5.80	6	5.00	4	5.64	6
NSGAI	7.10	8	5.35	5	3.71	2	5.23	4
CHLMOPSO	<b>1.40</b>	<b>1</b>	<b>2.55</b>	<b>1</b>	6.57	7	3.57	2

**Table 8** The comparison results of IGD and HV for CHLMOPSO and variant CHLMOPSO-I on 22 test problems

Problem	M	IGD		HV	
		CHLMOPSO-I	CHLMOPSO	CHLMOPSO-I	CHLMOPSO
ZDT1	2	3.0139e-3(1.44e-4)≈	<b>3.0056e-3(1.18e-4)</b>	7.2110e-1(1.85e-4) -	<b>7.2124e-1(1.44e-4)</b>
ZDT2	2	3.3054e-3(1.52e-4)≈	<b>3.2928e-3(1.71e-4)</b>	4.4578e-1(1.33e-4)≈	<b>4.4582e-1(1.26e-4)</b>
ZDT3	2	3.8524e-3(1.77e-4)≈	<b>3.8435e-3(2.43e-4)</b>	5.9993e-1(4.94e-4)≈	<b>6.0007e-1(6.33e-4)</b>
ZDT4	2	8.1314e-1(1.08e-1) -	<b>7.9258e-1(1.37e-1)</b>	1.0995e-1(7.40e-2) -	<b>1.3220e-1(1.05e-1)</b>
ZDT6	2	2.4320e-3(9.92e-5)≈	<b>2.4231e-3(9.74e-5)</b>	3.8930e-1(1.52e-4)≈	<b>3.8935e-1(1.64e-4)</b>
UF1	2	1.4226e-1(1.44e-2)≈	<b>1.4149e-1(1.42e-2)</b>	<b>5.0103e-1(2.04e-2)≈</b>	5.0017e-1(2.17e-2)
UF2	2	<b>7.3777e-2(6.41e-3) +</b>	8.0615e-2(5.63e-3)	<b>6.3726e-1(6.42e-3) +</b>	6.2844e-1(5.74e-3)
UF3	2	<b>3.0374e-1(2.40e-2)≈</b>	3.0547e-1(1.77e-2)	<b>3.4276e-1(2.40e-2)≈</b>	3.3451e-1(2.23e-2)
UF4	2	7.4773e-2(1.05e-2) -	<b>5.5948e-2(1.28e-3)</b>	3.4134e-1(1.29e-2) -	<b>3.6438e-1(2.22e-3)</b>
UF5	2	1.7306e + 0(1.86e-1) -	<b>1.6084e + 0(2.00e-1)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
UF6	2	7.7582e-1(1.43e-1)≈	<b>7.1366e-1(9.24e-2)</b>	2.5555e-3(5.42e-3)≈	<b>3.0084e-3(5.91e-3)</b>
UF7	2	<b>1.1930e-1(1.19e-2)≈</b>	1.2309e-1(1.67e-2)	<b>4.0785e-1(1.69e-2)≈</b>	4.0005e-1(2.28e-2)
UF8	3	3.1107e-1(9.08e-2)≈	<b>2.9240e-1(6.51e-2)</b>	3.3247e-1(2.14e-2)≈	<b>3.3336e-1(1.36e-2)</b>
UF9	3	<b>1.4107e-1(2.21e-2) +</b>	1.5506e-1(1.74e-2)	<b>6.0437e-1(3.30e-2)≈</b>	5.9686e-1(2.08e-2)
UF10	3	9.1253e-1(1.27e-1) -	<b>8.3000e-1(2.24e-1)</b>	1.0046e-1(3.86e-2)≈	<b>1.1310e-1(5.08e-2)</b>
DTLZ1	3	<b>1.9182e + 1(4.20e + 0)≈</b>	1.9836e + 1(4.59e + 0)	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
DTLZ2	3	4.0765e-1(8.78e-2) -	<b>2.5435e-1(3.95e-2)</b>	2.5244e-1(4.96e-2) -	<b>3.3691e-1(2.23e-2)</b>
DTLZ3	3	<b>1.3540e + 2(4.05e + 1)≈</b>	1.4550e + 2(2.93e + 1)	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
DTLZ4	3	<b>2.6406e-1(5.12e-2) +</b>	2.9143e-1(3.58e-2)	<b>4.0001e-1(4.67e-2) +</b>	3.3379e-1(5.06e-2)
DTLZ5	3	<b>5.9102e-3(8.80e-4) +</b>	1.2344e-2(1.98e-3)	<b>1.9755e-1(8.83e-4) +</b>	1.9102e-1(1.92e-3)
DTLZ6	3	3.9457e-3(4.02e-4) -	<b>3.7187e-3(3.39e-4)</b>	<b>2.0038e-1(3.64e-4)≈</b>	2.0034e-1(3.19e-4)
DTLZ7	3	2.3626e-1(1.57e-1) -	<b>8.5440e-2(5.57e-2)</b>	1.1244e-1(3.42e-2) -	<b>1.4812e-1(9.46e-3)</b>
+ / - / ≈		4/7/11	—	3/5/14	—
Best/all		8/22	14/22	8/22	11/22

**Table 9** The comparison results of IGD and HV for CHLMOPSO and variant CHLMOPSO-II on 22 test problems

Problem	M	IGD		HV	
		CHLMOPSO-II	CHLMOPSO	CHLMOPSO-II	CHLMOPSO
ZDT1	2	<b>2.9466e-3(1.03e-4)</b> ≈	3.0056e-3(1.18e-4)	7.2123e-1(1.51e-4)≈	<b>7.2124e-1(1.44e-4)</b>
ZDT2	2	3.3460e-3(1.83e-4)≈	<b>3.2928e-3(1.71e-4)</b>	4.4574e-1(1.82e-4) –	<b>4.4582e-1(1.26e-4)</b>
ZDT3	2	<b>3.8261e-3(2.16e-4)</b> ≈	3.8435e-3(2.43e-4)	5.9992e-1(4.83e-4)≈	<b>6.0007e-1(6.33e-4)</b>
ZDT4	2	8.0519e-1(1.25e-1)≈	<b>7.9258e-1(1.37e-1)</b>	1.1715e-1(8.91e-2)≈	<b>1.3220e-1(1.05e-1)</b>
ZDT6	2	2.4269e-3(1.01e-4)≈	<b>2.4231e-3(9.74e-5)</b>	3.8933e-1(1.61e-4)≈	<b>3.8935e-1(1.64e-4)</b>
UF1	2	1.4265e-1(1.19e-2)≈	<b>1.4149e-1(1.42e-2)</b>	4.9932e-1(1.94e-2)≈	<b>5.0017e-1(2.17e-2)</b>
UF2	2	<b>7.9659e-2(5.89e-3)</b> ≈	8.0615e-2(5.63e-3)	<b>6.2967e-1(5.81e-3)</b> ≈	6.2844e-1(5.74e-3)
UF3	2	<b>3.0369e-1(2.17e-2)</b> ≈	3.0547e-1(1.77e-2)	<b>3.4120e-1(2.37e-2)</b> ≈	3.3451e-1(2.23e-2)
UF4	2	5.6516e-2(1.26e-3) –	<b>5.5948e-2(1.28e-3)</b>	3.6385e-1(2.23e-3)≈	<b>3.6438e-1(2.22e-3)</b>
UF5	2	1.7427e + 0(2.08e-1) –	<b>1.6084e + 0(2.00e-1)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
UF6	2	7.2907e-1(1.04e-1)≈	<b>7.1366e-1(9.24e-2)</b>	<b>4.4645e-3(7.79e-3)</b> ≈	3.0084e-3(5.91e-3)
UF7	2	<b>1.2292e-1(1.62e-2)</b> ≈	1.2309e-1(1.67e-2)	3.9977e-1(2.31e-2)≈	<b>4.0005e-1(2.28e-2)</b>
UF8	3	3.3039e-1(7.20e-2) –	<b>2.9240e-1(6.51e-2)</b>	3.3135e-1(9.11e-3)≈	<b>3.3336e-1(1.36e-2)</b>
UF9	3	<b>1.5427e-1(1.38e-2)</b> ≈	1.5506e-1(1.74e-2)	5.9683e-1(1.74e-2)≈	<b>5.9686e-1(2.08e-2)</b>
UF10	3	8.7244e-1(1.82e-1) –	<b>8.3000e-1(2.24e-1)</b>	1.0846e-1(5.62e-2)≈	<b>1.1310e-1(5.08e-2)</b>
DTLZ1	3	2.1186e + 1(4.28e + 0)≈	<b>1.9836e + 1(4.59e + 0)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
DTLZ2	3	2.7146e-1(1.79e-2)≈	<b>2.5435e-1(3.95e-2)</b>	3.2724e-1(1.20e-2)≈	<b>3.3691e-1(2.23e-2)</b>
DTLZ3	3	1.4596e + 2(3.36e + 1)≈	<b>1.4550e + 2(2.93e + 1)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
DTLZ4	3	2.9494e-1(3.76e-2)≈	<b>2.9143e-1(3.58e-2)</b>	<b>3.3416e-1(4.82e-2)</b> ≈	3.3379e-1(5.06e-2)
DTLZ5	3	<b>1.1611e-2(1.75e-3)</b> ≈	1.2344e-2(1.98e-3)	<b>1.9163e-1(2.21e-3)</b> ≈	1.9102e-1(1.92e-3)
DTLZ6	3	3.8207e-3(4.08e-4)≈	<b>3.7187e-3(3.39e-4)</b>	<b>2.0039e-1(2.76e-4)</b> ≈	2.0034e-1(3.19e-4)
DTLZ7	3	1.2107e-1(1.00e-1)≈	<b>8.5440e-2(5.57e-2)</b>	1.3916e-1(2.36e-2)≈	<b>1.4812e-1(9.46e-3)</b>
+ / - /≈		0/4/18	—	0/1/21	—
Best/all		7/22	15/22	6/22	13/22

ZDT1, ZDT2, and ZDT3. However, CHLMOPSO is slightly superior to CMOPSO on ZDT1 and ZDT3 from the IGD values in Table 2. On the most of other problems, the gaps between CHLMOPSO and other algorithms are clear at a glance. The IGD values of CHLMOPSO on the most test problems are smaller, the abnormal values of experimental data are less, and the gaps between the upper and lower quartiles of CHLMOPSO are smaller, that is, the boxes are flatter. These findings demonstrate that CHLMOPSO outperforms the ten algorithms in terms of both solution outcomes and algorithm stability.

### Comparison of Pareto front

For the sake of displaying the optimization results of all algorithms intuitively, the distribution of Pareto optimal solution sets of CHLMOPSO and other ten comparison algorithms on ZDT3, UF9, and DTLZ6 is shown in Figs. 3, 4, 5, respectively. The distribution of Pareto optimal solution sets can clearly demonstrate the convergence and diversity of algorithms. It is obvious from Figs. 3, 4, 5 that the majority of

the selected comparison algorithms obviously lack convergence and diversity on ZDT3, UF9, and DTLZ6, while the performance of CHLMOPSO is relatively satisfactory.

In relation to the two-objective test problem ZDT3, the approximate PF distribution obtained by all algorithms is showcased in Fig. 3, only CMOPSO and CHLMOPSO can approximate the true PF, and the distribution is relatively uniform. Only a few solutions of NMPSO converge, while other algorithms do not achieve satisfactory convergence. Compared with the two-objective problems, the three-objective problems have higher requirements for the selected algorithms. In Fig. 4, comparing the distribution of Pareto optimal solution set of each algorithm on UF9, CHLMOPSO is well distributed near the true PF, while other algorithms show obvious insufficient convergence and diversity. In Fig. 5, although CMOPSO, NMPSO, MOPSOCD, and CHLMOPSO all have good convergence on DTLZ6, NMPSO and MOPSOCD still have the problem of insufficient diversity, and their Pareto optimal solution sets are

**Table 10** The comparison results of IGD and HV for CHLMOPSO and variant CHLMOPSO-III on 22 test problems

Problem	M	IGD		HV	
		CHLMOPSO-III	CHLMOPSO	CHLMOPSO-III	CHLMOPSO
ZDT1	2	3.0291e-3(1.09e-4)≈	<b>3.0056e-3(1.18e-4)</b>	<b>7.2128e-1(1.33e-4)≈</b>	7.2124e-1(1.44e-4)
ZDT2	2	3.3083e-3(1.59e-4)≈	<b>3.2928e-3(1.71e-4)</b>	4.4576e-1(3.30e-4)≈	<b>4.4582e-1(1.26e-4)</b>
ZDT3	2	3.8822e-3(2.34e-4)≈	<b>3.8435e-3(2.43e-4)</b>	5.9996e-1(3.47e-4)≈	<b>6.0007e-1(6.33e-4)</b>
ZDT4	2	<b>7.7262e-1(1.30e-1)≈</b>	7.9258e-1(1.37e-1)	<b>1.4919e-1(1.03e-1)≈</b>	1.3220e-1(1.05e-1)
ZDT6	2	2.4294e-3(1.22e-4)≈	<b>2.4231e-3(9.74e-5)</b>	3.8934e-1(1.72e-4)≈	<b>3.8935e-1(1.64e-4)</b>
UF1	2	1.6275e-1(1.81e-2) -	<b>1.4149e-1(1.42e-2)</b>	4.6944e-1(2.43e-2) -	<b>5.0017e-1(2.17e-2)</b>
UF2	2	9.1963e-2(6.39e-3) -	<b>8.0615e-2(5.63e-3)</b>	6.1493e-1(7.41e-3) -	<b>6.2844e-1(5.74e-3)</b>
UF3	2	3.2171e-1(1.15e-2) -	<b>3.0547e-1(1.77e-2)</b>	3.0984e-1(1.59e-2) -	<b>3.3451e-1(2.23e-2)</b>
UF4	2	5.7146e-2(3.56e-3)≈	<b>5.5948e-2(1.28e-3)</b>	3.6341e-1(4.72e-3)≈	<b>3.6438e-1(2.22e-3)</b>
UF5	2	1.7074e + 0(1.28e-1) -	<b>1.6084e + 0(2.00e-1)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
UF6	2	7.1481e-1(6.32e-2)≈	<b>7.1366e-1(9.24e-2)</b>	3.8660e-4(1.08e-3) -	<b>3.0084e-3(5.91e-3)</b>
UF7	2	1.4580e-1(1.88e-2) -	<b>1.2309e-1(1.67e-2)</b>	3.6642e-1(2.68e-2) -	<b>4.0005e-1(2.28e-2)</b>
UF8	3	<b>2.8275e-1(5.27e-2)≈</b>	2.9240e-1(6.51e-2)	3.1063e-1(2.10e-2) -	<b>3.3336e-1(1.36e-2)</b>
UF9	3	1.7955e-1(1.49e-2) -	<b>1.5506e-1(1.74e-2)</b>	5.5802e-1(2.07e-2) -	<b>5.9686e-1(2.08e-2)</b>
UF10	3	<b>6.5069e-1(2.22e-1) +</b>	8.3000e-1(2.24e-1)	<b>1.4199e-1(6.99e-2)≈</b>	1.1310e-1(5.08e-2)
DTLZ1	3	2.1044e + 1(4.85e + 0)≈	<b>1.9836e + 1(4.59e + 0)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
DTLZ2	3	<b>1.0540e-1(1.29e-2) +</b>	2.5435e-1(3.95e-2)	<b>4.4415e-1(1.89e-2) +</b>	3.3691e-1(2.23e-2)
DTLZ3	3	1.5338e + 2(3.18e + 1)≈	<b>1.4550e + 2(2.93e + 1)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
DTLZ4	3	3.3727e-1(4.43e-2) -	<b>2.9143e-1(3.58e-2)</b>	2.5120e-1(6.22e-2) -	<b>3.3379e-1(5.06e-2)</b>
DTLZ5	3	2.8324e-2(2.67e-3) -	<b>1.2344e-2(1.98e-3)</b>	1.7506e-1(3.59e-3) -	<b>1.9102e-1(1.92e-3)</b>
DTLZ6	3	3.8226e-3(3.87e-4)≈	<b>3.7187e-3(3.39e-4)</b>	2.0031e-1(2.74e-4)≈	<b>2.0034e-1(3.19e-4)</b>
DTLZ7	3	1.3562e-1(7.13e-2) -	<b>8.5440e-2(5.57e-2)</b>	1.3197e-1(1.43e-2) -	<b>1.4812e-1(9.46e-3)</b>
+ / - /≈		2/9/11	—	1/10/11	—
Best/all		4/22	18/22	4/22	15/22

not evenly distributed. However, the convergence and distribution of other algorithms are not ideal, so only CMOPSO and CHLMOPSO have better distribution and convergence.

The above experiments demonstrate that CHLMOPSO has more stable and significant performance than the selected comparison algorithms. This shows that the proposed competitive hybrid learning strategy can achieve significant convergence while ensuring the uniform distribution of solutions.

### Comparison of IGD indicator convergence

Along with the quality of solutions, a vital measurement of performance is the speed at which algorithms convergence. Figure 6 depicts the convergence trajectories of the IGD values generated through all algorithms following 10,000 evaluations on ZDT3, UF9, and DTLZ6. In contrast to the ten comparison algorithms, CHLMOPSO has a swift convergence speed. Additionally, it can be seen from the termination state of each algorithm in the convergence process that CHLMOPSO has the smallest IGD value, which more clearly

emphasizes that CHLMOPSO has remarkable convergence performance. As can be recognized, the competitive hybrid learning strategy proposed by CHLMOPSO can enhance the convergence and search efficiency of the algorithm.

### Ablation study

In CHLMOPSO, four key strategies are proposed to improve the overall performance of the algorithm. With the aim of investigating their respective contributions, we set up the experiments to compare the performance of CHLMOPSO with its four variants: (1) the CHLMOPSO without the derivative treatment strategy of pbest (CHLMOPSO-I); (2) the CHLMOPSO without adaptive flight parameters (CHLMOPSO-II); (3) the CHLMOPSO without competitive hybrid learning strategy (CHLMOPSO-III); (4) the CHLMOPSO without optimal angle distance to maintain external archive (CHLMOPSO-IV). The effectiveness of these four strategies will be discussed in depth in this subsection.



**Table 11** The comparison results of IGD and HV for CHLMOPSO and variant CHLMOPSO-IV on 22 test problems

Problem	M	IGD		HV	
		CHLMOPSO-IV	CHLMOPSO	CHLMOPSO-IV	CHLMOPSO
ZDT1	2	4.4303e-3(1.42e-4) –	<b>3.0056e-3(1.18e-4)</b>	7.1981e-1(1.52e-4) –	<b>7.2124e-1(1.44e-4)</b>
ZDT2	2	4.5023e-3(1.59e-4) –	<b>3.2928e-3(1.71e-4)</b>	4.4397e-1(4.74e-4) –	<b>4.4582e-1(1.26e-4)</b>
ZDT3	2	5.7647e-3(3.57e-4) –	<b>3.8435e-3(2.43e-4)</b>	5.9955e-1(5.40e-4) –	<b>6.0007e-1(6.33e-4)</b>
ZDT4	2	8.1630e-1(4.42e-2)≈	<b>7.9258e-1(1.37e-1)</b>	1.1124e-1(3.07e-2)≈	<b>1.3220e-1(1.05e-1)</b>
ZDT6	2	4.3643e-3(3.84e-4) –	<b>2.4231e-3(9.74e-5)</b>	3.8729e-1(4.52e-4) –	<b>3.8935e-1(1.64e-4)</b>
UF1	2	1.4721e-1(9.26e-3) –	<b>1.4149e-1(1.42e-2)</b>	4.9253e-1(1.33e-2)≈	<b>5.0017e-1(2.17e-2)</b>
UF2	2	8.1390e-2(4.41e-3)≈	<b>8.0615e-2(5.63e-3)</b>	6.2762e-1(4.61e-3)≈	<b>6.2844e-1(5.74e-3)</b>
UF3	2	3.0575e-1(1.26e-2)≈	<b>3.0547e-1(1.77e-2)</b>	<b>3.3638e-1(1.71e-2)≈</b>	3.3451e-1(2.23e-2)
UF4	2	5.6354e-2(9.44e-4)≈	<b>5.5948e-2(1.28e-3)</b>	3.6388e-1(1.74e-3)≈	<b>3.6438e-1(2.22e-3)</b>
UF5	2	1.6639e + 0(2.00e-1)≈	<b>1.6084e + 0(2.00e-1)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
UF6	2	<b>6.8734e-1(7.03e-2)≈</b>	7.1366e-1(9.24e-2)	<b>6.7091e-1(8.90e-3)≈</b>	3.0084e-3(5.91e-3)
UF7	2	<b>1.1882e-1(1.44e-2)≈</b>	1.2309e-1(1.67e-2)	<b>4.0530e-1(2.01e-2)≈</b>	4.0005e-1(2.28e-2)
UF8	3	3.2839e-1(8.95e-2) –	<b>2.9240e-1(6.51e-2)</b>	<b>3.3913e-1(9.63e-3)≈</b>	3.3336e-1(1.36e-2)
UF9	3	<b>1.4561e-1(1.58e-2)≈</b>	1.5506e-1(1.74e-2)	<b>6.0908e-1(1.94e-2)≈</b>	5.9686e-1(2.08e-2)
UF10	3	8.5876e-1(2.07e-1)≈	<b>8.3000e-1(2.24e-1)</b>	<b>1.1374e-1(6.07e-2)≈</b>	1.1310e-1(5.08e-2)
DTLZ1	3	1.9889e + 1(4.45e + 0)≈	<b>1.9836e + 1(4.59e + 0)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
DTLZ2	3	2.5625e-1(2.66e-2)≈	<b>2.5435e-1(3.95e-2)</b>	3.3560e-1(1.86e-2)≈	<b>3.3691e-1(2.23e-2)</b>
DTLZ3	3	1.5729e + 2(2.53e + 1)≈	<b>1.4550e + 2(2.93e + 1)</b>	0.0000e + 0(0.00e + 0)≈	0.0000e + 0(0.00e + 0)
DTLZ4	3	<b>2.8682e-1(3.50e-2)≈</b>	2.9143e-1(3.58e-2)	<b>3.4366e-1(4.00e-2)≈</b>	3.3379e-1(5.06e-2)
DTLZ5	3	1.3134e-2(2.47e-3)≈	<b>1.2344e-2(1.98e-3)</b>	1.9070e-1(2.23e-3)≈	<b>1.9102e-1(1.92e-3)</b>
DTLZ6	3	6.1176e-3(1.58e-3) –	<b>3.7187e-3(3.39e-4)</b>	1.9802e-1(1.55e-3) –	<b>2.0034e-1(3.19e-4)</b>
DTLZ7	3	9.3549e-2(6.29e-2)≈	<b>8.5440e-2(5.57e-2)</b>	1.4620e-1(1.07e-2)≈	<b>1.4812e-1(9.46e-3)</b>
+ / – /≈		0/7/15	—	0/5/17	—
Best/all		4/22	18/22	7/22	12/22

In order to ensure the fairness in the comparison of algorithm performance, the parameter settings of these four variants are consistent with CHLMOPSO. The four variants are tested on the ZDT, UF, and DTLZ test suites and run independently for 30 times on each test problem to obtain statistical average and standard deviation (in parenthesis) results with respect to the IGD and HV indicators. The comparison results are presented in Tables 8, 9, 10, 11, the best results of each test problem are shown in bold, and the Wilcoxon rank sum test is performed to detect whether there are significant performance differences between CHLMOPSO and the four variants.

The experimental results indicate that CHLMOPSO performs significantly better than its four variants on most of the test problems, which confirms that the combination of the four proposed strategies improves the comprehensive performance of CHLMOPSO and has an important role in this algorithm. Additionally, as can be seen from Tables 8, 9, 10, 11, the variants CHLMOPSO-III and CHLMOPSO-IV perform worse than CHLMOPSO relative to those of CHLMOPSO-I

and CHLMOPSO-II, which demonstrates that the proposed competitive hybrid learning strategy and the optimal angle distance strategy to maintain the external archive have made relatively great contributions to CHLMOPSO. In the variant CHLMOPSO-III, the performance of the competitive hybrid learning strategy is more significant on UF test suite. And in the variant CHLMOPSO-IV, the performance improvement of the optimal angle distance strategy to maintain the external archive is predominantly manifested on the ZDT test suite. However, in terms of the overall results, after removing the competitive hybrid learning strategy, the performance of variant CHLMOPSO-III is obviously worse, thus verifying that the competitive hybrid learning strategy has the greatest contribution to CHLMOPSO. This is because some studies have shown that the update of particles through the competitive mechanism in CSO can significantly increase the diversity of population, thus avoiding premature convergence. And our proposed competitive hybrid learning strategy combines the updating strategy of CSO with that of MOPSO to achieve a good equilibrium between convergence and diversity.

## Conclusion

For improvements in the comprehensive performance of algorithm, a competitive hybrid learning MOPSO is put forward. CHLMOPSO proposes an updating strategy of pbest to derive pbest of the stagnant particles without evolution for the purpose of enhancing the ability of particles to explore. Subsequently, the flight parameters are transformed adaptively in accordance with the evolutionary state of particles. In the proposed competitive hybrid learning strategy, the algorithm selects different updating strategies. Additionally, the optimal angle distance strategy is applied to effectively maintain the archive, thereby speeding up algorithm convergence.

On three suits of benchmark problems, the simulation experiments compared CHLMOPSO with the selected MOPSOs and MOEAs. The IGD and HV indicators are utilized to assess its performance as a whole, and two non-parametric statistical tests, the Wilcoxon rank sum test and the Friedman rank test, are used to process experimental results. The findings from the experiments also verify the effectiveness of CHLMOPSO, which can better equilibrate the convergence and diversity.

However, according to the experimental results, it can be observed that the performance of the proposed CHLMOPSO needs to be further improved on some MOPs (such as DTLZ) with complex PF features that are multi-modal and concave. In addition, it can also be concluded from the ablation experiments that the adaptive parameters strategy in CHLMOPSO does not make a significant contribution to the algorithm, which may be due to the fact that the proposed adaptive flight parameters strategy only takes into account the evolutionary ability of the particles but not that of the population. Therefore, in our future work, it will be meaningful to design an adaptive parameters strategy that takes into account both the evolutionary ability of the population and the particles to obtain more accurate and better diversity of solutions for the particles. This can better balance the global exploration and local exploitation abilities of the algorithm to solve concave and multi-modal MOPs. Furthermore, the effective application of CHLMOPSO to solve practical problems such as vehicle routing, network communication, and production scheduling is also one of the future research.

**Acknowledgements** This work was supported in part by Key Laboratory of Evolutionary Artificial Intelligence in Guizhou (Qian Jiaoji [2022] No. 059) and the Key Talens Program in digital economy of Guizhou Province.

**Data availability** All data generated or analyzed during this study are included in this published article.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Madani A, Engelbrecht A, Ombuki-Berman B (2023) Cooperative coevolutionary multi-guide particle swarm optimization algorithm for large-scale multi-objective optimization problems. *Swarm Evol Comput* 78:101262. <https://doi.org/10.1016/j.swevo.2023.101262>
- Wang Z, Mao B, Hao H, Hong W, Xiao C, Zhou A (2022) Enhancing diversity by local subset selection in evolutionary multiobjective optimization. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2022.3194211>
- Ma L, Huang M, Yang S, Wang R, Wang X (2021) An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. *IEEE Trans Cybern* 52(7):6684–6696. <https://doi.org/10.1109/TCYB.2020.3041212>
- Ru H, Huang J, Chen W, Xiong C (2023) Modeling and identification of rate-dependent and asymmetric hysteresis of soft bending pneumatic actuator based on evolutionary firefly algorithm. *Mech Mach Theory* 181:105169. <https://doi.org/10.1016/j.mechmachtheory.2022.105169>
- Luna JM, Kiran RU, Fournier-Viger P, Ventura S (2023) Efficient mining of top-k high utility itemsets through genetic algorithms. *Inf Sci* 624:529–553. <https://doi.org/10.1016/j.ins.2022.12.092>
- Sun L, Chen Y, Ding W, Xu J, Ma Y (2023) AMFSA: adaptive fuzzy neighborhood-based multilabel feature selection with ant colony optimization. *Appl Soft Comput* 138:110211. <https://doi.org/10.1016/j.asoc.2023.110211>
- Wang Y, Liu Z, Wang GG (2023) Improved differential evolution using two-stage mutation strategy for multimodal multi-objective optimization. *Swarm Evol Comput* 78:101232. <https://doi.org/10.1016/j.swevo.2023.101232>
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Icnn95-international Conference on Neural Networks* 4:1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
- Coello CAC, Lechuga MS (2002) MOPSO: a proposal for multiple objective particle swarm optimization. In *Pro. 2002 Congr Evol Comput CEC'02* (Cat. No. 02TH8600). *IEEE* 2:1051–1056. <https://doi.org/10.1109/CEC.2002.1004388>
- Nebro AJ, Durillo JJ, Garcia-Nieto J, Coello CAC, Luna F, Alba E (2009) SMPSO: a new PSO-based metaheuristic for multi-objective optimization. In: *2009 IEEE Symp Comput Intell MCDM*, pp 66–73. <https://doi.org/10.1109/MCDM.2009.4938830>
- Li L, Wang W, Xu X (2017) Multi-objective particle swarm optimization based on global margin ranking. *Inf Sci* 375:30–47. <https://doi.org/10.1016/j.ins.2016.08.043>
- Meza J, Espitia H, Montenegro C, Giménez E, González-Crespo R (2017) MOVPSO: vortex multi-objective particle swarm optimization. *Appl Soft Comput* 52:1042–1057. <https://doi.org/10.1016/j.asoc.2016.09.026>

13. Li Y, Zhang Y, Hu W (2023) Adaptive multi-objective particle swarm optimization based on virtual Pareto front. *Inf Sci* 625:206–236. <https://doi.org/10.1016/j.ins.2022.12.079>
14. Gu Q, Jiang M, Jiang S, Chen L (2021) Multi-objective particle swarm optimization with R2 indicator and adaptive method. *Complex Intell Syst* 7:2697–2710. <https://doi.org/10.1007/s40747-021-00445-3>
15. García IC, Coello CAC, Arias-Montaña A (2014) MOPSOhv: a new hypervolume-based multi-objective particle swarm optimizer. In: 2014 IEEE Congr Evol Comput (CEC). IEEE 266–273. <https://doi.org/10.1109/CEC.2014.6900540>
16. Zapotecas MS, Coello CAC (2011) A multi-objective particle swarm optimizer based on decomposition. In: Proc 13th Annu Conf Genet Evol Comput, pp 69–76. <https://doi.org/10.1145/2001576.2001587>
17. Dai C, Wang Y, Ye M (2015) A new multi-objective particle swarm optimization algorithm based on decomposition. *Inf Sci* 325:541–557. <https://doi.org/10.1016/j.ins.2015.07.018>
18. Lin Q, Li J, Du Z, Chen J, Ming Z (2015) A novel multi-objective particle swarm optimization with multiple search strategies. *Eur J Oper Res* 247(3):732–744. <https://doi.org/10.1016/j.ejor.2015.06.071>
19. Zhu Q, Lin Q, Chen W, Wong KC, Coello CAC, Li J et al (2017) An external archive-guided multiobjective particle swarm optimization algorithm. *IEEE Trans Cybern* 47(9):2794–2808. <https://doi.org/10.1109/TCYB.2017.2710133>
20. Wu B, Hu W, Hu J, Yen GG (2019) Adaptive multiobjective particle swarm optimization based on evolutionary state estimation. *IEEE Trans Cybern* 51(7):3738–3751. <https://doi.org/10.1109/TCYB.2019.2949204>
21. Han H, Lu W, Qiao J (2017) An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. *IEEE Trans Cybern* 47(9):2754–2767. <https://doi.org/10.1109/TCYB.2017.2692385>
22. Sharma D, Vats S, Saurabh S (2021) Diversity preference-based many-objective particle swarm optimization using reference-lines-based framework. *Swarm Evol Comput* 65:100910. <https://doi.org/10.1016/j.swevo.2021.100910>
23. Han H, Lu W, Zhang L, Qiao J (2017) Adaptive gradient multiobjective particle swarm optimization. *IEEE Trans Cybern* 48(11):3067–3079. <https://doi.org/10.1109/TCYB.2017.2756874>
24. Cui Y, Meng X, Qiao J (2022) A multi-objective particle swarm optimization algorithm based on two-archive mechanism. *Appl Soft Comput* 119:108532. <https://doi.org/10.1016/j.asoc.2022.108532>
25. Li L, Chang L, Gu T, Sheng W, Wang W (2019) On the norm of dominant difference for many-objective particle swarm optimization. *IEEE Trans Cybern* 51(4):2055–2067. <https://doi.org/10.1109/TCYB.2019.2922287>
26. Cheng R, Jin Y (2014) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45(2):191–204. <https://doi.org/10.1109/TCYB.2014.2322602>
27. Feng Y, Feng L, Kwong S, Tan KC (2021) A multivariation multifactorial evolutionary algorithm for large-scale multiobjective optimization. *IEEE Trans Evol Comput* 26(2):248–262. <https://doi.org/10.1109/TEVC.2021.3119933>
28. Wang X, Zhang K, Wang J, Jin Y (2021) An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multiobjective optimization. *IEEE Trans Evol Comput* 26(5):859–871. <https://doi.org/10.1109/TEVC.2021.3111209>
29. Tan KC, Lee TH, Khor EF (2001) Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Trans Evol Comput* 5(6):565–588. <https://doi.org/10.1109/4235.974840>
30. Zhang X, Zheng X, Cheng R, Qiu J, Jin Y (2018) A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf Sci* 427:63–76. <https://doi.org/10.1016/j.ins.2017.10.037>
31. Lin Q, Liu S, Zhu Q, Tang C, Song R, Chen J et al (2018) Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Trans Evol Comput* 22(1):32–46. <https://doi.org/10.1109/TEVC.2016.2631279>
32. Raquel CR, Naval Jr PC (2005) An effective use of crowding distance in multiobjective particle swarm optimization. In: Proc 7th Annu Conf Genet Evol Comput, pp. 257–264. <https://doi.org/10.1145/1068009.1068047>
33. He C, Cheng R, Yazdani D (2020) Adaptive offspring generation for evolutionary large-scale multi-objective optimization. *IEEE Trans Syst Man Cybern Syst* 52(2):786–798. <https://doi.org/10.1109/TSMC.2020.3003926>
34. Tian Y, Cheng R, Zhang X, Su Y, Jin Y (2018) A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Trans Evol Comput* 23(2):331–345. <https://doi.org/10.1109/TEVC.2018.2866854>
35. Jiang S, Yang S (2017) A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Trans Evol Comput* 21(3):329–346. <https://doi.org/10.1109/TEVC.2016.2592479>
36. Li K, Deb K, Zhang Q, Kwong S (2014) An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans Evol Comput* 19(5):694–716. <https://doi.org/10.1109/TEVC.2014.2373386>
37. Deb K, Jain H (2013) An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601. <https://doi.org/10.1109/TEVC.2013.2281535>
38. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: a matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput Intell Mag* 12(4):73–87. <https://doi.org/10.1109/MCI.2017.2742868>
39. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8(2):173–195. <https://doi.org/10.1162/106365600568202>
40. Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W, Tiwari S (2008) Multi-objective optimization test instances for the CEC 2009 special session and competition. *Mech Eng New York* 264:1–30
41. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multi-objective optimization. *Evol Mult Opt London* 105–145. [https://doi.org/10.1007/1-84628-137-7\\_6](https://doi.org/10.1007/1-84628-137-7_6)
42. Zhou AM, Jin YC, Zhang QF, Sendhoff B, Tsang E (2006) Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In: 2006 IEEE Int Conf Evol Comput 892–899. <https://doi.org/10.1109/CEC.2006.1688406>
43. While L, Hingston P, Barone L, Huband S (2006) A faster algorithm for calculating hypervolume. *IEEE Trans Evol Comput* 10(1):29–38. <https://doi.org/10.1109/TEVC.2005.851275>
44. Zhou Y, Chen Z, Huang Z, Xiang Y (2020) A multiobjective evolutionary algorithm based on objective-space localization selection. *IEEE Trans Cybern* 52(5):3888–3901. <https://doi.org/10.1109/TCYB.2020.3016426>
45. Lu J, Zhang J, Sheng J (2022) Enhanced multi-swarm cooperative particle swarm optimizer. *Swarm Evol Comput* 69:100989. <https://doi.org/10.1016/j.swevo.2021.100989>