Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **6**, Number **3**, August **2021**

Multi-Objective Firefly Algorithm with Hierarchical Learning

Li Lv*

Jiangxi Province Key Laboratory of Water Information
Cooperative Sensing and Intelligent Processing,
Nanchang Institute of Technology,Nanchang 330099, China
lvli623@163.com

Xiao-Dong Zhou

School of Information Engineering,
Nanchang Institute of Technology,Nanchang 330099, China
zhouxiaodong767@163.com

Ping Kang

School of Information Engineering,
Nanchang Institute of Technology,Nanchang 330099, China
kangshuiping0606@163.com

Xue-Feng Fu

School of Information Engineering,
Nanchang Institute of Technology,Nanchang 330099, China
fxf@nit.edu.cn

Xiu-Mei Tian

School of Information Engineering,
Nanchang Institute of Technology,Nanchang 330099, China
simeng3676@163.net

*Corresponding author: Li Lv(lvli623@163.com)
Received April 2021; revised June 2021

Abstract. *In the optimization process of multi-objective firefly algorithm, the population is easy to fall into local optimum, which leads to poor population distribution and convergence. In order to solve this problem, this paper proposes a multi-objective firefly algorithm with hierarchical learning (MOFA-HL). Firstly, the hierarchical learning method is proposed, population is layered by non-dominant sorting to obtain individuals at each level, and the dominant individuals at the front level guide the individuals at the back level to learn, which is beneficial to search more high-quality solutions. Then, mutation operation is carried out on the population to enhance the local search ability of the algorithm. Finally, the mutated population is merged with the previous generation population, and the excellent individuals with the same population size as the previous generation are selected to enter the next generation by non-dominant sorting and crowding distance calculation. Eight test functions in the series of ZDT and DTLZ are used to test MOFA-HL. MOFA-HL is compared with four classical algorithms and four new algorithms. The results show that MOFA-HL can better improve the distribution and convergence of the population.*
Keywords: Multi-objective optimization, Firefly algorithm, Non-dominant sorting, Hierarchical learning, Crowding distance calculation.

1. **Introduction.** In the real world, there are many multi-objective optimization problems (MOPs). For example, flexible job shop scheduling requires minimum production costs and total electricity costs [1], optimal allocation of water resources requires maximum environmental and economic benefits [2], vehicle routing planning requires minimum operating cost and customer dissatisfaction [3], block-chain network security requires improving the performance of block chain network and reducing the possibility of malicious nodes clustering [4], etc. MOPs may involve multiple conflicting objectives to be optimized simultaneously, and the performance improvement of one sub-problem may lead to the performance degradation of other sub-problems. In order to balance the performance of each sub-problem, a set of non-dominated and equivalent solution sets are usually solved.

In the past thirty years, Multi-objective Evolutionary Algorithms (MOEAs) have been widely favored in the field of multi-objective optimization due to its population-based searching characteristics and the effectiveness of MOPs solving. Depending on how they evolve, MOEAs can be divided into the following categories:

(1) MOEAs based on the Pareto dominance relationship. Such as NSGA [5], NSGA-II [6], SPEA [7], SPEA2 [8], NSGA-III [9], etc. This kind of algorithms take the Pareto dominance relationship as the solution selection mechanism, few parameters, simple structure and easy implementation. However, due to their single selection mechanism, they are not effective when dealing with many-objective optimization problems, so other mechanisms need to be added to enhance algorithms performance.

(2) MOEAs based on objective decomposition. Such as MOEA/D [10], MOEA/D-DE [11], MOEA-D-AWA [12], etc. They decompose a multi-objective optimization problem (MOP) into a set of scalar sub-problems using uniformly distributed aggregation weight vectors. Once a high quality solution of a sub-problem is found, their superior information will quickly spread to other individuals in the neighborhood, thus speeding up the rate of population convergence. Nevertheless, the effect is not good when dealing with some irregular Pareto front. Generally, adaptive weight adjustment technique is used to improve algorithm performance.

(3) MOEAs based on evaluation metrics. Such as IBEA [13], SMS-EMOA [14], HypE [15], etc. This kind of algorithms use evaluation metrics as fitness evaluation function to be embedded into algorithm to guide the optimization process of MOEAs. Their purposes are clear, but with the addition of evaluation metrics, MOEAs will become more complex and time-consuming. Therefore, it is necessary to consider the efficiency and complexity of algorithm when designing such algorithms.

(4) MOEAs based on swarm intelligence. Such as MOPSO [16], MOPSO-D [17], NMPSO [18], CMOPSO [19], MOFA [20], CFMOFA [21], etc. This kind of algorithms extend some swarm intelligence techniques with good performance to the field of multi-objective optimization, and swarm intelligence algorithms have unique optimization methods and strong adaptability, such as the common particle swarm optimization algorithm [22]. In reality, there are many application cases of swarm intelligence technology, such as complex optimization problem [23], watermarking technology optimization problem [24], clustering problem [25]. MOEAs based on swarm intelligence not only needs to combine the basic idea of swarm intelligence algorithm, but also needs to combine the decomposition idea of multi-objective optimization field or the Pareto dominance relationship of multi-objective optimization field, as well as other strategies to achieve the purpose of optimization.

Multi-objective firefly algorithm (MOFA) [20] is a swarm intelligence algorithm for solving multi-objective optimization problems by simulating the firefly flight mode. The firefly flight direction is determined by comparing the Pareto dominance relationship of

objective values. MOFA is an extension of the single objective firefly algorithm (FA), and some improved algorithms [26–28] on FA are of great reference significance for improving MOFA. MOFA has few parameters, simple structure and is easy to implement, but the population is easy to fall into the local optimum in the optimization process, which leads to poor distribution and convergence of the population. For this reason, this paper proposes a multi-objective firefly algorithm with hierarchical learning (MOFA-HL).The main contributions of this paper can be summarized as follows:

(a) Hierarchical learning mechanism. We abandon the traditional random attraction model and propose a hierarchical learning mechanism: the non-dominated sorting algorithm [6] is adopted to sort the population, so that the high-ranking fireflies can guide the low-ranking fireflies to fly, so as to avoid blind flight, improve flight efficiency and improve the convergence and distribution of the population.

(b) Mutation operation. By mutating the population, the local search ability of the algorithm is enhanced, so that the algorithm can obtain more high-quality solutions close to the real Pareto front, and the distribution and convergence of the population are also improved.

(c) In order to fully verify the performance of MOFA-HL, MOFA-HL is used to solve the ZDT and DTLZ series test problems and good results is obtained.

2. **Relevant Knowledge.**

2.1. **Multi-objective optimization problem.**

The mathematical definition of multi-objective optimization problem is as follows [29]:

$$\begin{cases} \min\limits_{X} & F(X) = (f_1(X), f_2(X), \ldots, f_M(X))^T \\ \text{s.t.} & g_i(X) \le 0, i = 1, 2, \ldots, p \\ & h_j(X) = 0, j = 1, 2, \ldots, q \end{cases} \tag{1}$$

where $X = (x_1, x_2, \ldots, x_D)^T \in \Omega \subset R^D$ is the decision vector, Ω is known as the decision space, D is the number of decision variables, $F(X) \subset R^M$ is the objective vector, M is the number of objectives. All $F(X)$ constitute the unknown objective space, $g_i(X)$ is inequality constraints, $h_j(X)$ is equality constraints.

Definition 2.1. ***(Pareto dominance relationship)*** *The solution X dominates the solution Y, if and only if :*

$$\begin{cases} f_i(X) \le f_i(Y), \forall i = 1, 2, \ldots, D \\ f_j(X) < f_j(Y), \exists j = 1, 2, \ldots, D \end{cases} \tag{2}$$

Let's call $X \prec Y$, and say that X is a better solution than Y.

Definition 2.2. ***(Non-dominant relationship)*** *If the following conditions are satisfied, the solution X and Y are not dominated by each other :*

$$\begin{cases} f_i(X) < f_i(Y), \exists i = 1, 2, \ldots, D \\ f_j(X) > f_j(Y), \exists j = 1, 2, \ldots, D, j \ne i \end{cases} \tag{3}$$

At this point, it is considered that the solution X and Y are equivalent, and they belong to the non-dominant relation.

Definition 2.3. ***(Pareto optimal solution set)*** *Pareto optimal solution set (PS) is the set constituted by all Pareto optimal solutions on the decision space.*

Definition 2.4. ***(Pareto front)*** *Pareto front (PF) is the point set corresponding to the optimal solution set of Pareto in the objective space.*

2.2. **The basic multi-objective firefly algorithm.** Multi-objective firefly algorithm (MOFA) [20] regards points in the search space as fireflies, and simulates the process of attracting fireflies to high-brightness fireflies as optimization and location iterative updating process.

(1) Attraction. For any given two fireflies i and j, the attraction between firefly i and j can be expressed as:

$$\beta_{ij}\left(r_{ij}\right) = \beta_0 e^{-\gamma r_{ij}^2} \tag{4}$$

where β_0 is the attraction at $r = 0$, usually take $\beta_0 = 1$. γ is the light absorption coefficient, $\gamma \in [0.01, 100]$. r_{ij} is the Euclidean distance between firefly i and j.

(2)Firefly position update. In MOFA, firefly position update is divided into two situations.

In one case, if firefly j dominates i, the position of firefly i is updated as:

$$x_i(t+1) = x_i(t) + \beta_{ij}\left(r_{ij}\right) \cdot \left(x_j(t) - x_i(t)\right) + \alpha_t \cdot \varepsilon_i^t \tag{5}$$

where t is the number of current iterations. x_i and x_j are the spatial positions of firefly i and j respectively. $\beta_{ij}(r_{ij})$ is the attraction between firefly i and j. α_t is a constant, $\alpha \in [0, 1]$. ε_i^t is a vector of random numbers drawn from a Gaussian distribution or uniform distribution.

The other case is that no individual is found to dominate a firefly i, the firefly i moves:

$$x_i^{t+1} = g_*^t + \alpha_t \varepsilon_i^t \tag{6}$$

where g_*^t is the best solution found for a given set of random weights in the current population. For a minimization problem, in order to do random walks more efficiently, we can find the current best g_*^t which minimizes a combined objective $\psi(x)$ via the weighted sum using the following methods:

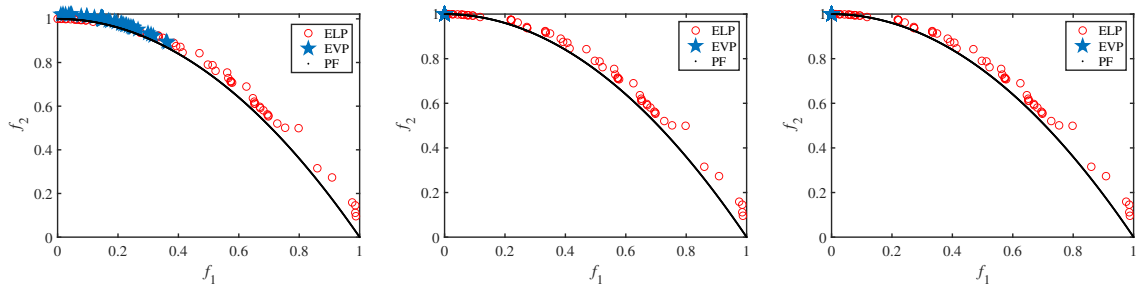
$$\psi(x) = \sum_{m=1}^{M} w_m f_m, \quad \sum_{m=1}^{M} w_m = 1 \tag{7}$$

here $w_m = p_m/M$, where p_m are random numbers that follow a uniform distribution, $p_m \in [0, 1]$. M is the objective number. A random weight assignment operation is performed to generate M uniformly distributed w_m. In order to make g_*^t not unique, the weight w_m should be selected at random at each iteration, so that the obtained g_*^t can make fireflies fly in different directions.

2.3. **MOFA defect analysis.** When a firefly does not have a non-dominant individual, MOFA adopts the global optimal individual g_*^t guidance, and theoretically the selected g_*^t can make the population evolve in different directions. However, for a minimization problem, when the m-th objective value f_m of an individual i is far less then other individual objective values, its f_m still occupies a large proportion in the combined $\psi(x)$ even if the weight w_m is biggest. The $\psi(x)$ of individual i might be much smaller than that of other individuals. Although g_*^t may be selected differently at each iteration, g_*^t will get closer and closer to the position of individual i, resulting in local optimization of the algorithm and ultimately poor distribution and convergence of the population.

Figure 1 shows the test results of MOFA on the ZDT2 [30] test function with different iterations. With 300 iterations as the termination condition, the Elite Population (ELP), Evolution Population (EVP) and real Pareto front (PF) of 30, 120 and 300 iterations were retained respectively, and their distribution maps are drawn. By observing Figure 1, with the increase of iterations, it can be found that the blue EVP gradually converged towards a certain place in the evolutionary process, and the solution of other regions could not

be found, so that the remaining ELP gradually show non-uniform distribution and poor convergence.



(a) $t = 30$ (b) $t = 120$ (c) $t = 300$

FIGURE 1. Population distribution of MOFA on the ZDT2 test function

3. **The proposed MOFA-HL.** In the process of MOFA optimization, the population is easy to fall into local optimum, which leads to poor distribution and convergence of the population. Therefore, this paper proposes a multi-objective firefly algorithm with hierarchical learning (MOFA-HL), which adopts three strategies of hierarchical learning, mutation operation and crowding distance calculation to improve the distribution and convergence of the population.

3.1. **Hierarchical learning.** In order to enable the algorithm to search for more high-quality solutions with better distribution and convergence, MOFA-HL adopts hierarchical learning method to replace the global optimal individual g_*^t guidance method. Hierarchical learning consists of two steps: 1) The population is layered. 2) Identifying which individuals guide and learn from it. The specific process is as follows:

(1) Hierarchy. The main method of hierarchy is non-dominant sorting, which is used to divide the population into different levels. Adopting the fast non-dominated sorting algorithm proposed by Deb et al. [6]: 1) Adding all the first ranking non-dominant individuals in the solution set P to the solution set F_1. 2) Removing the individuals belonging to the solution set F_1 from the solution set P, and now adding all the first ranking non-dominant individuals in the solution set P to the solution set F_2. Repeating the above steps until every individual in P is added to a certain set F_i, and finally obtaining set $F_1, F_2, \ldots, F_n$. As shown in Figure 2, the fireflies are placed in a non-dominant sorting to obtain Pareto front of different levels. The main purpose of hierarchy is to obtain the layered individuals by using the non-dominated sorting, to find the non-dominated individuals in a targeted way and to make reasonable use of the computing resources, which is different from the mode of MOFA to randomly find the non-dominated individuals.

(2) Learning. After the above hierarchical steps, the superior upper layer individuals are the main learning objects of the lower layer individuals. There may be multiple individuals in each layer, in order to determine which individual $x_p^{F_i}$ in F_i layer is specifically guided by the individual $x_q^{F_{i-1}}$ in the F_{i-1} layer. Firstly, formula 4 shows that the nearest distance between individuals generates the greatest attraction in between layers, so it is most appropriate for $x_q^{F_i}$ in the F_i layer to learn from the nearest individual $x_p^{F_{i-1}}$ in the F_{i-1} layer. Then, by calculating the Euclidean distance between individuals in the F_i and F_{i-1} layer, the individual $x_p^{F_{i-1}}$ in the F_{i-1} layer , which is closest to $x_q^{F_i}$ in the F_i layer,

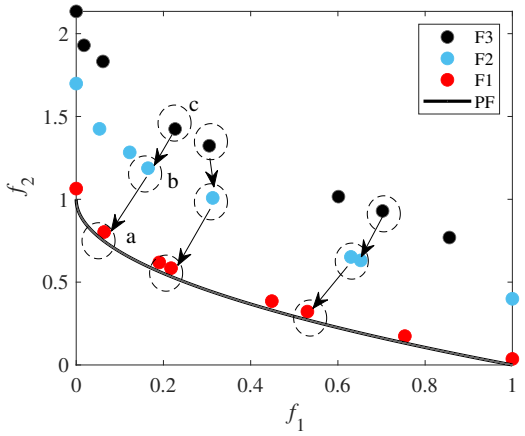


FIGURE 2. Schematic diagram of hierarchical learning in fireflies. In the schematic diagram, the fireflies are placed in a non-dominated sorting to obtain the F_1, F_2 and F_3 layer. Fireflies learning in the way that individual c at the F_3 layer learns from individual b at the F_2 layer, and individual b at the F_2 layer learns from individual a at the F_1 layer

is selected as the learning object, as shown in Figure 2. And formula 8 is used to update the position of firefly $x_q^{F_i}$ in the F_i layer:

$$x_q^{F_i}(t+1) = x_q^{F_i}(t) + \beta_{qp}\left(r_{qp}\right) \cdot \left(x_p^{F_{i-1}}(t) - x_q^{F_i}(t)\right) + \alpha_t \cdot \varepsilon_q^t \tag{8}$$

where ε_q^t is a random number vector with a normal distribution, $\varepsilon_q^t \in [-1, 1]$. $\beta_{qp}\left(r_{qp}\right)$ is the attraction between $x_q^{F_i}$ and $x_p^{F_{i-1}}$. r_{qp} is the Euclidean distance between the nearest previous layer individual $x_p^{F_{i-1}}$ to $x_q^{F_i}$. α_t is a constant, $\alpha_t \in [0, 1]$.

When the positions of all the individuals in the F_i layer are updated, the individuals in the F_{i-1} layer repeat the above learning process and continue to learn from better individuals in the previous layer until the end of the learning process in the F_2 layer, and new individuals in $F_2', F_3', \ldots, F_n'$ layer are obtained.

The hierarchical learning method of MOFA-HL has the following advantages: 1) It makes the population converge to a better front, and makes rational use of computing resources without blindly searching for non-dominant individuals to learn. 2) Choosing a better Pareto front as the guide layer can enable the population to search for more high-quality solutions close to the real Pareto front, which ensures the distribution and convergence of the population.

3.2. **Mutation operation.** Hierarchical learning can only reach the F_2 layer, because the individuals of F_1 layer do not have guidance level, so the individuals at F_1 layer have no learning layer. In order to make the individuals in the F_1 layer have the ability of optimizing, and the individuals in $F_2', F_3', \ldots, F_n'$ layer have the local random searching ability after hierarchical learning, so as to maintain the diversity of the population and prevent the premature convergence of the population, the individuals in the F_1 layer and $F_2', F_3', \ldots, F_n'$ layer are merged, and the mutation operation is carried out on the merged population.

Mutation operation in genetic algorithm is used for reference [31]. Let the value range of the k-th dimension variable of individual x_i $(x_{i1}, x_{i2}, \ldots, x_{id})$ be $[Var\_\min_{ik}, Var\_\max_{ik}]$, l dimensions of individual x_i are randomly selected for mutation. The proportion between l and the total dimension d (mutation rate, mu) should not be too large. In order to avoid

large differences between solutions before and after mutation, which would destroy the tendency of approaching the optimal solution, mu is set to $1/d$ after many experiments. The mutation formula for individual x_i is as follows:

$$x_{ik}(t+1) = x_{ik}(t) + \text{sigma} \cdot (Var\_\max_{ik} - Var\_\min_{ik}) \cdot \varepsilon_{ik}^{t} \tag{9}$$

where k represents the k-th dimension of the i-th individual, sigma is the step size factor, sigma $\in (0, 1]$. ε_{ik} is a random number that fits a normal distribution, $\varepsilon_{ik}^{t} \in [-1, 1]$. Figure 3 shows a schematic diagram of local mutation in fireflies. Figure 3 shows a schematic diagram of local mutation in fireflies. In the diagram, firefly a locally mutates into fireflies b, c, d, and f. The local mutation operation maintains the population diversity and improves the tendency of firefly to approach the optimal solution.

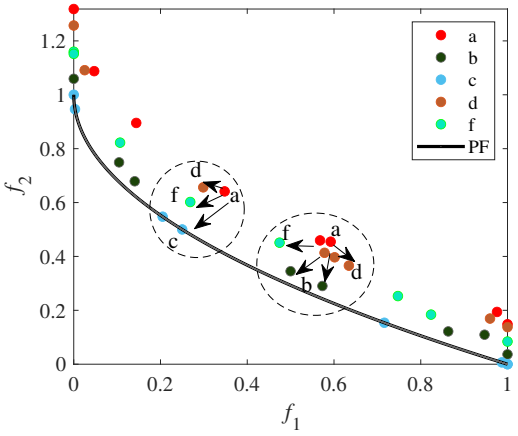


FIGURE 3. Schematic diagram of local mutation in fireflies

3.3. **Crowding distance calculation.** In order to screen out excellent individuals with a population size of N to enter the next generation evolution, the population Q_t after the above mutation is merged with the previous generation population P_t into $2N$ population R_t, and then N excellent individuals are selected from R_t for the next generation by non-dominant sorting.

When $K(K < N)$ individuals are screened from $F_1, F_2, \ldots, F_{l-1}$ layer, the number of the remaining $N-K$ individuals is less than that of the F_l layer. If $N-K$ individuals are randomly screened from the F_l layer, it is possible to make the population distribution worse. Therefore, the crowding distance calculation (CD) method proposed by Deb et al. [6] is used as a reference in this paper. By calculating the crowding distance of all the individuals in the F_l layer, $N-K$ individuals with better crowding distance are selected to maintain the distribution of the population. Note that CD is only calculated for a certain layer of individuals, not for all individuals. The CD of the i-th individual $x_i^{F_l}$ in the F_l layer is calculated as follows:

$$CD_{x_i^{F_l}} = \sum_{m=1}^{M} \left(\frac{f^m_{x_{i+1}^{F_l}} - f^m_{x_{i-1}^{F_l}}}{f_{\max}^{F_l,m} - f_{\min}^{F_l,m}} \right) \tag{10}$$

where M is the number of objectives. The m-th objective value of all individuals in F_l layer is respectively arranged in ascending order. $f^m_{x_{i+1}^{F_l}}$ is the objective value of individual $x_{i+1}^{F_l}$, $f^m_{x_{i-1}^{F_l}}$ is the m-th objective value of individual $x_{i-1}^{F_l}$, $f_{\max}^{F_l,m}$ and $f_{\min}^{F_l,m}$ are respectively the maximum and minimum value of the m-th objective of all individuals in the F_l layer.

Obviously, if the difference between the objective values of $x_{i-1}^{F_l}$ and $x_{i+1}^{F_l}$ is larger, then correspondingly, the CD of $x_i^{F_l}$ is going to be larger, and the probability of $x_i^{F_l}$ being selected is going to be higher.

3.4. **The basic steps of MOFA-HL.**

Input: N (population size), mu (mutation rate), sigma (step factor), β_0 (initial attraction), γ (light absorption coefficient), α (disturbance factor).
Output: P (final population).

$P_t \leftarrow$ Random Initialization(N);
while termination criterion not fulfilled **do**
  $F \leftarrow$ Calculate the non-dominated front number of each solution in P_t;
  $P_1 \leftarrow \phi$;
  for $i = 1$ to $Max(F) - 1$ **do**
    $x^{F_{i+1}} \leftarrow$ The solution $x^{F_{i+1}}$ for the $(i+1)$-th front learns from the solution x^{F_i} for the nearest neighbor of the i-th front by (8);
    $P_1 \leftarrow P_1 \cup \left\{x^{F_{i+1}}\right\}$;
  end for
  $P_1 \leftarrow P_1 \cup \left\{x^{F_1}\right\}$;
  $Q_t \leftarrow$ Mutation(P_1, mu, sigma);
  $R_t \leftarrow Q_t \cup P_t$;
  $P_{t+1} \leftarrow N$ elite individuals are selected by non-dominated sorting and crowding distance calculation in R_t;
end whlie
return P;

4. **Experiment and analysis.**

4.1. **Experiment settings.**

MOFA-HL is compared with four classical MOEAs (NSGA-II [6], SPEA2 [8], MOEA/D [10], MOPSO [16]) and four new MOEAs (NMPSO [18], CMOPSO [19], MOFA [20], CFMOFA [21]). We obtained some experimental data of the algorithm through Platemo [32] platform. Each algorithm is run 30 times independently to save the mean value and standard deviation of each evaluation metric. Specifically, the population size is set to 100 for 2-objectives MOPs and the number of function evaluations is 30000; the population size is set to and 200 for 3-objectives MOPs, and the number of function evaluations is 120000. Parameter Settings of each algorithm are shown in Table 1.

Table 1. Parameter Settings of each algorithm

Algorithm	**Parameter setting**
NSGA-II, MOEA/D, SPEA2, MOPSO, NMPSO, CMOPSO	Parameter setting using PlatEMO platform
MOFA	$\alpha = 0.2, \beta_0 = 1, \gamma = 1$
CFMOFA	$\alpha = 0.2, \beta_0 = 1, \gamma = 1, m = 2$
MOFA-HL	$\alpha = 0.2, \beta_0 = 1, \gamma = 1, \text{sigma} = 0.2, mu = 0.1$

4.2. **Test function.** In order to test the performance of algorithms on MOPs, eight test functions in the ZDT series [30] (2-objectives MOPs) designed by Zitzler et al. and the DTLZ series [33] (3-objectives MOPs) designed by Deb et al. are used for testing. The eight test functions are defined in Table 2.

Table 2. Test function definition

Problem	Definition	Constraints
ZDT1	$f_1(x) = x_1, f_2(x) = g(x) * \left(1 - \sqrt{x_1/g(x)}\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i/(n-1)$	$n = 30, 0 \le x_i \le 1$ $i = 1, \ldots, n$
ZDT2	$f_1(x) = x_1, f_2(x) = g(x) * \left(1 - (x_1/g(x))^2\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i/(n-1)$	$n = 30, 0 \le x_i \le 1$ $i = 1, \ldots, n$
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x) * \left(1 - \sqrt{x_1/g(x)} - x_1 \sin(10\pi x_1)/g(x)\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i/(n-1)$	$n = 30, 0 \le x_i \le 1$ $i = 1, \ldots, n$
ZDT6	$f_1(x) = 1 - e^{(-4x_1)} \sin^6(6\pi x_1)$ $f_2(x) = g(x) * \left(1 - (f_1(x)/g(x))^2\right)$ $g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i/(n-1)\right)^{0.25}$	$n = 10, 0 \le x_i \le 1$ $i = 1, \ldots, n$
DTLZ1	$f_1(X) = \frac{1}{2} x_1 x_2 \cdots x_{M-1}(1 + g(X_M))$ $f_2(X) = \frac{1}{2} x_1 x_2 \cdots (1 - x_{M-1})(1 + g(X_M))$ $\vdots$ $f_{M-1}(X) = \frac{1}{2} x_1 (1 - x_2)(1 + g(X_M))$ $f_M(X) = \frac{1}{2}(1 - x_1)(1 + g(X_M))$ $g(X_M) = 100 * [\lvert X_M \rvert + \sum_{x_i \in X_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))]$ $\lvert X_M \rvert = n - M + 1$	$n = 7$ $0 \le x_i \le 1$ $M = 3$ $i = 1, \ldots, n$
DTLZ2	$f_1(X) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \cos(x_{M-2}\pi/2) \cos(x_{M-1}\pi/2)$ $f_2(X) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \cos(x_{M-2}\pi/2) \sin(x_{M-1}\pi/2)$ $f_3(X) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \sin(x_{M-2}\pi/2)$ $\vdots$ $f_M(X) = (1 + g(X_M)) \sin(x_1\pi/2)$ $g(X_M) = \sum_{x_i \in X_M} (x_i - 0.5)^2$	$n = 12$ $0 \le x_i \le 1$ $M = 3$ $i = 1, \ldots, n$
DTLZ3	$f_1(X) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \cos(x_{M-2}\pi/2) \cos(x_{M-1}\pi/2)$ $f_2(X) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \cos(x_{M-2}\pi/2) \sin(x_{M-1}\pi/2)$ $f_3(x) = (1 + g(X_M)) \cos(x_1\pi/2) \cdots \sin(x_{M-2}\pi/2),$ $\vdots$ $f_M(X) = (1 + g(X_M)) \sin(x_1\pi/2)$ $g(X_M) = 100 * [\lvert X_M \rvert + \sum_{x_i \in \mathbb{X}_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))]$ $\lvert X_M \rvert = n - M + 1$	$n = 12$ $0 \le x_i \le 1$ $M = 3$ $i = 1, \ldots, n$
DTLZ4	$f_1(X) = (1 + g(X_M)) \cos(x_1^\alpha\pi/2) \cdots \cos(x_{M-2}^\alpha\pi/2) \cos(x_{M-1}^\alpha\pi/2)$ $f_2(X) = (1 + g(X_M)) \cos(x_1^\alpha\pi/2) \cdots \cos(x_{M-2}^\alpha\pi/2) \sin(x_{M-1}^\alpha\pi/2)$ $f_3(X) = (1 + g(X_M)) \cos(x_1^\alpha\pi/2) \cdots \sin(x_{M-2}^\alpha\pi/2)$ $\vdots$ $f_M(X) = (1 + g(X_M)) \sin(x_1^\alpha\pi/2)$ $g(X_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2$	$n = 12$ $0 \le x_i \le 1$ $M = 3$ $i = 1, \ldots, n$ $a = 100$

4.3. **Evaluation metrics.** In order to evaluate the optimization performance of algorithms, SP is used to evaluate the distribution of algorithms, and IGD is used to evaluate the distribution and convergence of algorithms.

(1) Spacing (SP). SP [34] refers to the standard deviation of the distance between each individual in the approximate Pareto front (PF_{known}) and its nearest individual. Its

calculation formula is as follows:

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2} \tag{11}$$

$$d_i = \min_{j \in PF_{known}} \|i - j\|, (i, j = 1, 2, \ldots, n) \tag{12}$$

where $\bar{d}$ is the average of all d_i, d_i is the minimum Euclidean distance between individual i and j in the approximate Pareto front (PF_{known}),n is the number of approximate Pareto front solutions obtained, M is the objective number. When SP value is 0, it means that all individuals are equidistant from each other. The smaller the SP value, the better the distribution of the algorithm.

(2) Inverted Generational Distance (IGD). IGD [34] measures the distance between the real PF and the approximate PF obtained by the algorithm. Its calculation formula is as follows:

$$IGD = \frac{\sum_{j \in PF_{true}} d'_j}{n} \tag{13}$$

$$d'_j = \min_{i \in PF_{known}} \|j - i\|, (i, j = 1, 2, \ldots, n) \tag{14}$$

where d'_j represents the minimum Euclidean distance from individual j in the real Pareto front (PF_{true}) to individual i in the approximate Pareto front (PF_{known}). This metric can reflect not only the convergence of the algorithm but also the distribution of the algorithm. The smaller the IGD value, the better the convergence and distribution of the algorithm.

4.4. **Experimental results and analysis.** The proposed MOFA-HL is compared with four classical algorithms (NSGAII, MOEA/D, SPEA2, MOPSO) and four new algorithms (NMPSO, CMOPSO, MOFA, CFMOFA) on the MOPs of ZDT series (2-objectives) and DTLZ series (3-objectives), respectively.

Table 3 presents the mean and standard deviation of SP values and the "+/-/≈" results obtained by nine MOEAs on the eight MOPs. The Wilcoxon rank sum test with a significance level of 0.05 is performed on the SP values of each MOEA and the proposed MOFA-HL. The symbol '+' indicates that the corresponding algorithm is better than MOFA-HL, '-' indicates that the corresponding algorithm is worse than MOFA-HL, and '≈' indicates that the corresponding algorithm is statistically similar to MOFA-HL.

Table 3 shows that MOFA-HL and NSGA-II, MOEA/D, SPEA2, MOPSO, MOFA, NMPSO, CMOPSO and CFMOFA have an advantageous ratio of 8/8, 6/8, 4/8, 7/8, 8/8, 8/8 respectively on the eight MOPs. MOFA-HL has an advantage ratio of 6/9 among the nine algorithms, which is equivalent to SPEA2 and CMOPSO in terms of distribution quality.

In order to make full use of the obtained data, Friedman test is carried out on the experimental results of 30 runs of the nine algorithms, and the performance of the nine algorithms is further analyzed. Table 4 shows the rank-mean results and rankings of the nine algorithms on SP. As can be seen from Table 4, MOFA-HL ranks the first, and its rank-mean is 2.25. In Table 3, through Wilcoxon rank sum test, MOFA-HL is found to have similar quality with SPEA2 and CMOPSO in terms of distribution, while in Table 4, Friedman test shows a gap between MOFA-HL and SPEA2 and CMOPSO, and the results show that MOFA-HL is superior.

Then, we compare the convergence rates of the proposed MOFA-HL with the eight MOEAs. Due to space constraints, Figure 4 plots the convergence curves of the SP of

TABLE 3. Mean and standard deviation of the nine algorithms on SP, where gray indicates the best, and the smaller SP, the better. The more '-', the better the distribution of MOFA-HL

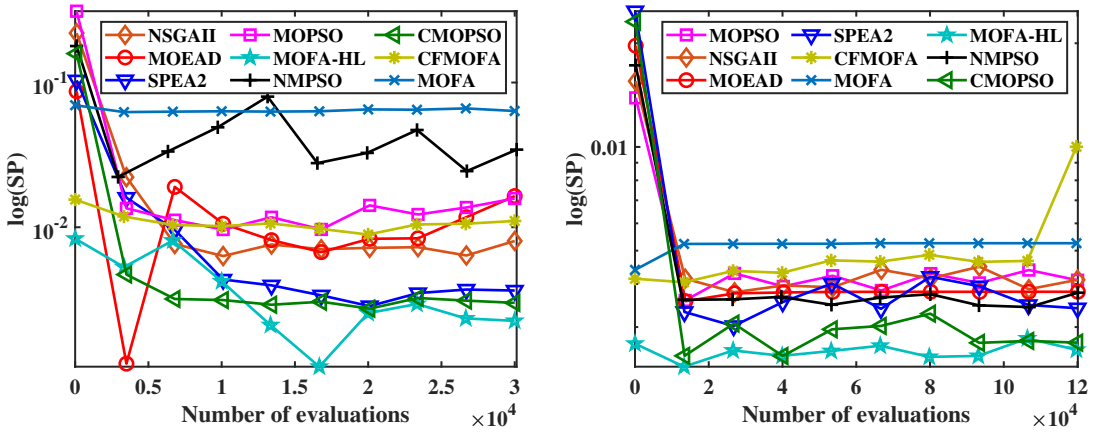
Problem		NSGA-II	MOEA/D	SPEA2	MOPSO	MOFA	NMPSO	CMOPSO	CFMOFA	MOFA-HL
ZDT1	mean	6.9752e-3	7.0958e-3	4.0474e-3	1.2626e-2	3.9693e-2	4.3766e-2	3.9046e-3	1.1568e-2	6.8124e-3
	(std.)	(6.25e-4)	(1.87e-3)	(3.29e-4)	(2.19e-3)	(7.94e-2)	(1.18e-2)	(2.04e-4)	(1.64e-3)	(1.52e-3)
	rank sum	-	-	+	-	-	-	+	-	
ZDT2	mean	7.2598e-3	1.4090e-2	6.2935e-3	8.4062e-3	3.2835e-2	3.2078e-2	3.2331e-3	1.4066e-2	6.2474e-3
	(std.)	(6.36e-4)	(1.03e-2)	(2.11e-4)	(5.52e-3)	(1.93e-2)	(4.44e-3)	(2.35e-4)	(3.96e-3)	(2.38e-4)
	rank sum	-	-	-	-	-	-	+	-	
ZDT3	mean	8.7830e-3	2.0984e-2	7.0569e-3	1.2738e-2	6.3703e-2	1.0682e-1	6.5923e-3	1.7453e-2	6.2357e-3
	(std.)	(8.26e-4)	(3.35e-3)	(3.54e-4)	(3.75e-3)	(6.11e-2)	(4.36e-3)	(2.93e-4)	(2.53e-3)	(3.24e-4)
	rank sum	-	-	-	-	-	-	-	-	
ZDT6	mean	7.8285e-3	5.7362e-3	3.2450e-3	5.8783e-2	3.2749e-2	8.5436e-3	5.4792e-2	1.4070e-1	4.4598e-3
	(std.)	(4.62e-4)	(3.81e-4)	(2.12e-4)	(8.63e-2)	(2.38e-2)	(5.12e-3)	(7.97e-2)	(2.24e-1)	(2.51e-5)
	rank sum	-	-	+	-	-	-	-	-	
DTLZ1	mean	2.8490e-2	2.4506e-4	4.9566e-2	2.6709e+01	5.5243e+1	2.9575e-2	2.6453e-1	1.0522e+1	2.5003e-2
	(std.)	(5.35e-3)	(5.37e-5)	(3.42e-1)	(1.69e+0)	(3.55e+0)	(3.24e-3)	(1.10e+0)	(1.43e-1)	(3.23e-4)
	rank sum	-	+	-	-	-	-	-	-	
DTLZ2	mean	4.5259e-2	4.2185e-2	2.8425e-2	4.7728e-2	4.9441e-2	4.2096e-2	2.8760e-2	4.3571e-2	4.1702e-2
	(std.)	(4.23e-3)	(2.13e-5)	(1.52e-3)	(4.38e-3)	(2.71e-3)	(2.42e-3)	(2.34e-3)	(2.14e-4)	(1.56e-3)
	rank sum	-	-	+	-	-	-	+	-	
DTLZ3	mean	1.5567e-1	5.3726e-2	1.8039e+01	9.9408e+12	.7586e+1	5.5523e-2	3.8894e+0	1.1645e+1	5.2084e-2
	(std.)	(5.87e-1)	(1.54e-4)	(3.47e+0)	(1.60e+1)	(9.02e+0)	(2.42e-2)	(3.41e+0)	(2.38e+0)	(2.15e-3)
	rank sum	-	-	-	-	-	-	-	-	
DTLZ4	mean	4.0149e-2	3.8042e-2	2.0542e-2	3.5994e-2	1.1508e-1	3.8243e-2	2.2085e-2	4.0094e-2	3.5221e-2
	(std.)	(2.26e-3)	(2.16e-2)	(1.22e-3)	(3.25e-2)	(2.59e-2)	(2.41e-3)	(2.15e-3)	(8.04e-4)	(3.26e-3)
	rank sum	-	≈	+	≈	-	-	+	-	
+/-/≈		0/8/0	1/6/1	4/4/0	0/7/1	0/8/0	0/8/0	4/4/0	0/8/0	

TABLE 4. Rank-mean results and ranking of nine algorithms on SP

Average ranking	Algorithm	Rank-mean value
1	MOFA-HL	2.25
2	SPEA2	2.63
3	CMOPSO	3.38
4	MOEA/D	4.38
5	NSGA-II	4.75
6	NMPSO	6.00
7	MOPSO	6.38
8	CFMOFA	6.88
9	MOFA	8.38

the nine algorithms on ZDT1 and DTLZ2. In order to facilitate observation, logarithm of the vertical axis value is taken to enlarge the graph. On the ZDT1 with 2-objectives, MOFA-HL shows a fast convergence rate. On the DTLZ2 of the 3-objectives, MOFA-HL has the fastest convergence speed and good stability, and it shows the best distribution performance.

Table 5 shows the mean and standard deviation of the nine algorithms on IGD and the results of "+/-/≈". As can be seen from Table 5, MOFA-HL compared with the other eight algorithms. MOFA-HL perform best in five of the eight MOPs, follow by MOEA/D with two, and finally CMOPSO with one. MOFA-HL and NSGA-II, MOEA/D, SPEA2, MOPSO, MOFA, NMPSO, CMOPSO and CFMOFA have an advantageous ratio of 6/8, 6/8, 6/8, 8/8, 8/8, 6/8, 7/8, 8/8 respectively on the eight MOPs.



(a) ZDT1 (b) DTLZ2

FIGURE 4. SP convergence curve of nine algorithms

TABLE 5. Mean and standard deviation of the nine algorithms on IGD, where gray indicates the best, and the smaller IGD, the better. The more '-', the better the distribution and convergence of MOFA-HL

Problem		NSGA-II	MOEA/D	SPEA2	MOPSO	MOFA	NMPSO	CMOPSO	CFMOFA	MOFA-HL
ZDT1	mean	5.0963e-3	1.2915e-2	4.5465e-3	8.0493e-1	3.1782e-2	2.9412e-2	4.8735e-3	8.9560e-3	4.4408e-3
	(std.)	(1.45e-4)	(9.61e-3)	(6.32e-5)	(2.22e-1)	(4.36e-3)	(1.30e-2)	(8.91e-5)	(9.92e-4)	(5.36e-4)
	rank sum	-	-	-	-	-	-	-	-	
ZDT2	mean	5.6344e-3	2.7171e-2	5.1285e-3	1.6829e+0	6.4161e-2	1.9362e-2	5.4237e-3	1.3116e-2	5.1004e-3
	(std.)	(3.36e-4)	(4.31e-2)	(2.56e-5)	(4.93e-1)	(2.38e-2)	(3.53e-3)	(2.36e-4)	(4.71e-3)	(2.36e-4)
	rank sum	-	-	-	-	-	-	-	-	
ZDT3	mean	7.2029e-3	2.9571e-2	6.1241e-3	8.6502e-1	4.9602e-2	1.0026e-1	5.0225e-3	1.4541e-2	6.0172e-3
	(std.)	(2.54e-3)	(2.11e-2)	(5.36e-3)	(2.23e-1)	(9.68e-3)	(4.51e-3)	(4.78e-5)	(2.61e-3)	(1.36e-4)
	rank sum	-	-	-	-	-	-	+	-	
ZDT6	mean	4.3775e-3	7.2116e-3	4.2068e-3	1.9850e-1	3.3622e-1	5.3250e-3	4.1006e-3	9.7681e-2	4.0275e-3
	(std.)	(5.12e-4)	(4.23e-3)	(1.45e-5)	(1.01e+0)	(1.07e-1)	(2.65e-4)	(5.37e-5)	(1.76e-1)	(1.32e-5)
	rank sum	-	-	-	-	-	-	-	-	
DTLZ1	mean	3.2031e-2	2.1919e-2	2.2026e-2	4.1617e+0	1.4568e+2	2.7466e-2	5.9684e-1	1.5554e+2	3.1633e-2
	(std.)	(2.45e-4)	(2.47e-5)	(5.36e-5)	(2.03e+0)	(1.42e+0)	(1.85e-4)	(7.56e-1)	(3.41e+0)	(3.63e-3)
	rank sum	≈	+	+	-	-	+	-	-	
DTLZ2	mean	5.8514e-2	4.6884e-2	4.7040e-2	7.5068e-2	5.6509e-2	6.4267e-2	4.9549e-2	6.8044e-2	4.4567e-2
	(std.)	(1.36e-3)	(2.58e-7)	(4.36e-4)	(5.45e-3)	(7.56e-4)	(2.78e-3)	(2.89e-4)	(1.36e-3)	(4.12e-4)
	rank sum	-	-	-	-	-	-	-	-	
DTLZ3	mean	5.1257e-2	4.0040e-2	4.1519e-2	5.7123e+1	4.8802e-1	9.8786e-2	4.5236e+1	7.2140e+1	4.0080e-1
	(std.)	(5.36e-3)	(4.75e-4)	(4.23e-4)	(4.39e+1)	(4.38e-1)	(1.87e-1)	(2.79e+1)	(4.50e+0)	(2.56e-2)
	rank sum	+	+	+	-	-	+	-	-	
DTLZ4	mean	5.5278e-2	1.6432e-1	6.4742e-2	2.0930e-1	6.9505e-1	8.9140e-2	5.1225e-2	9.1016e-2	5.0534e-2
	(std.)	(2.34e-3)	(2.74e-1)	(5.63e-2)	(8.18e-2)	(1.65e-1)	(1.23e-1)	(4.25e-4)	(7.95e-3)	(2.34e-3)
	rank sum	-	-	-	-	-	-	-	-	
+/-/≈		1/6/1	2/6/0	2/6/0	0/8/0	0/8/0	2/6/0	1/7/0	0/8/0	

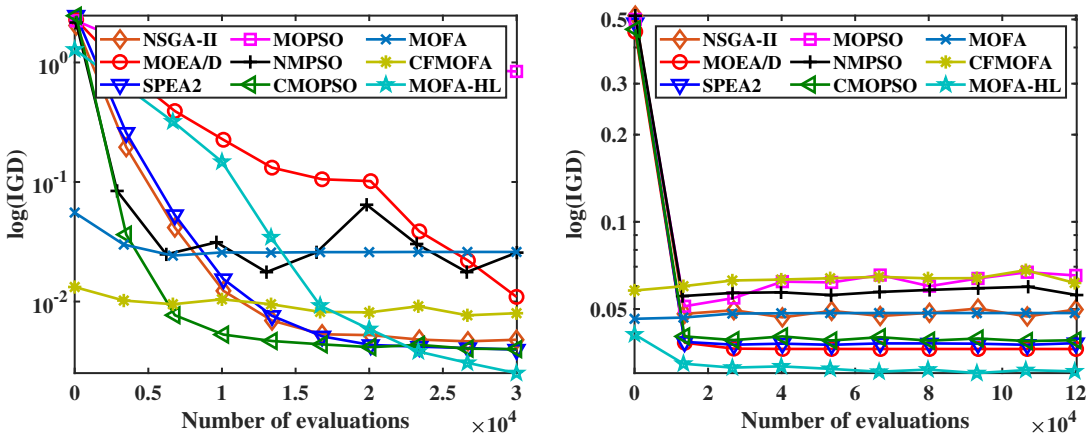
Through Friedman test, the experimental data obtained from 30 runs of the nine algorithms are deeply analyzed. Table 6 shows the rank-mean results and rankings of the nine algorithms on IGD. As can be seen from Table 6, MOFA-HL ranks the first with a rank mean of 2.13. The results show that MOFA-HL has the best distribution and convergence.

Similarly, in order to clearly know the convergence curves of the nine algorithms on the IGD, Figure 5 shows the convergence curves of the nine algorithms on ZDT1 and DTLZ2. Since MOFA-HL adopts a hierarchical learning method, its evolution is characterized by gradual progression, and its evolution process is relatively slow. Therefore,

TABLE 6. Rank-mean results and ranking of nine algorithms on IGD

Average ranking	Algorithm	Rank-mean value
1	MOFA-HL	2.13
2	SPEA2	2.63
3	CMOPSO	3.50
4	NSGA-II	4.00
5	MOEA/D	4.50
6	NMPSO	5.63
7	CFMOFA	6.75
8	MOFA	7.50
9	MOPSO	8.38

the convergence speed of MOFA-HL in the early stage is slightly worse than that of other algorithms, but with the increase of the number of evaluation times, the convergence of MOFA-HL will continue to improve. In Figure 5, the convergence speed of MOFA-HL on the ZDT1 of the 2-objectives is continuously accelerated, and the final IGD is superior to other algorithms. MOFA-HL shows unique advantages when dealing with the DTLZ problems of 3-objectives, and MOFA-HL shows the fastest and stable convergence speed on DTLZ2. In general, MOFA-HL has the best distribution and convergence.



(a) ZDT1 (b) DTLZ2

FIGURE 5. IGD convergence curve of nine algorithms

To show the fitting effect between the approximate PF and the real PF of the nine algorithms. Table 7 presents the Pareto front fitting graphs of MOFA-HL and 4 classic MOEAs on the eight MOPs. Table 8 presents the Pareto front fitting graphs of MOFA-HL and 4 new MOEAs on the eight MOPs. In each figure, the black dot set represents the real Pareto front point set, while the red dot set represents the approximate Pareto front point set obtained by each algorithm. According to Table 7 and Table 8, MOFA-HL and SPEA2 show good convergence and distribution, and the fitting effect is similar, which indirectly indicates that the Friedman test comprehensive ranking between them is similar. NSGA-II, MOEA/D, NMPSO and CMOPSO all show moderate fitting effect. MOPSO, MOFA and CFMOFA are far from close to the real Pareto front in some of the

3-objectives MOPs, showing poor fitting effect. On the whole, the fitting results of all the algorithms are consistent with the Friedman test ranking.

TABLE 7. Pareto front fitting diagram of MOFA-HL and four classic MOEAs

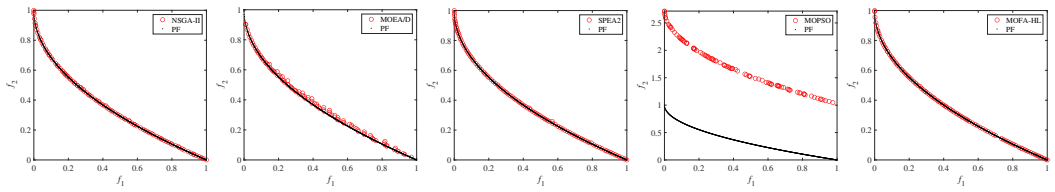
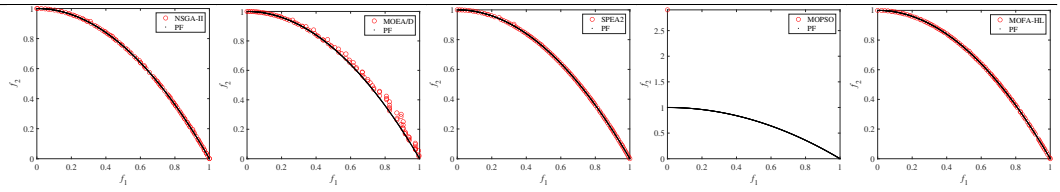
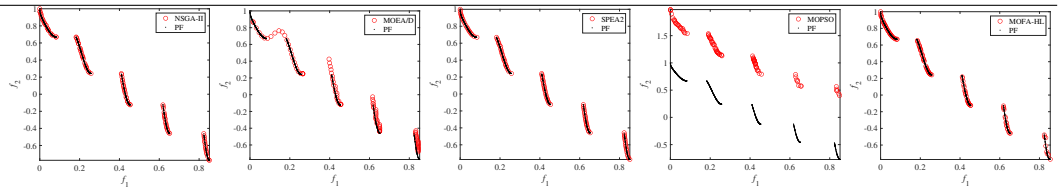
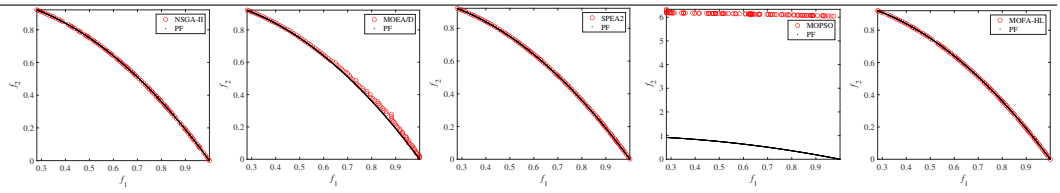
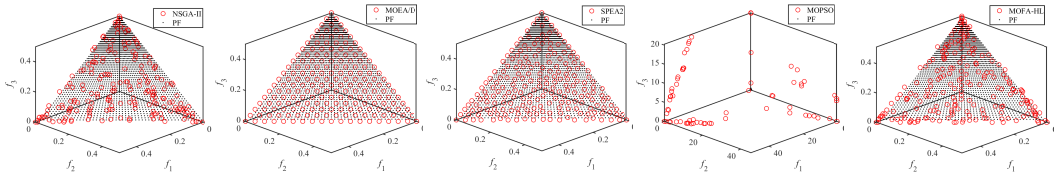
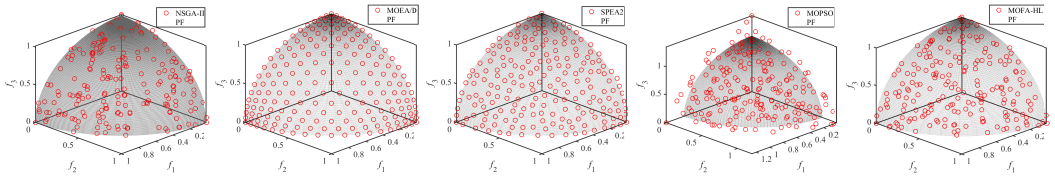
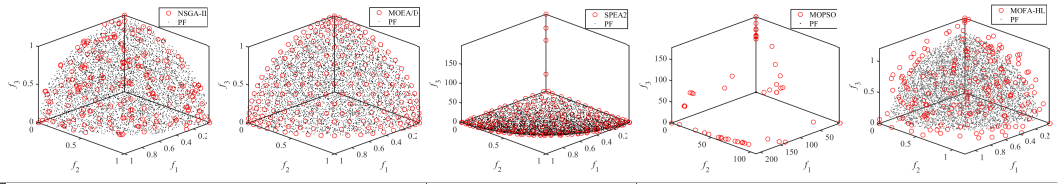
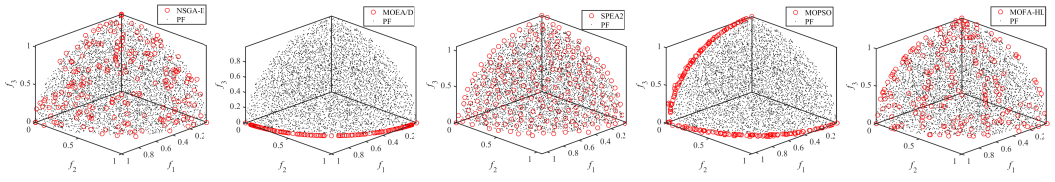
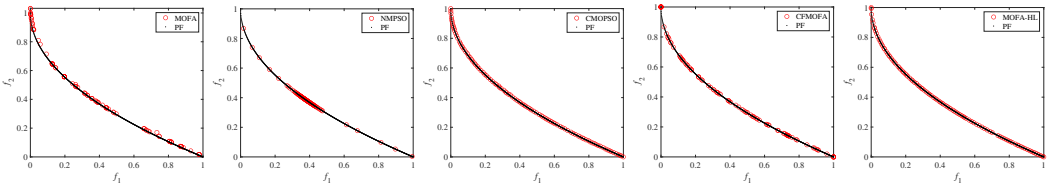
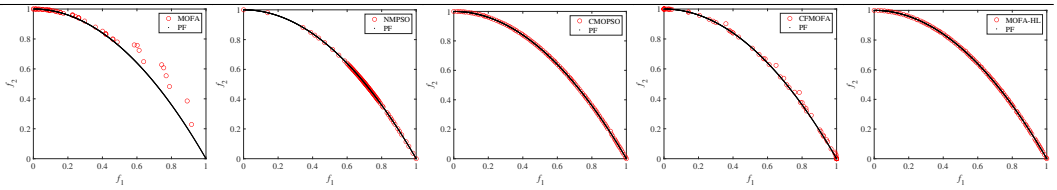
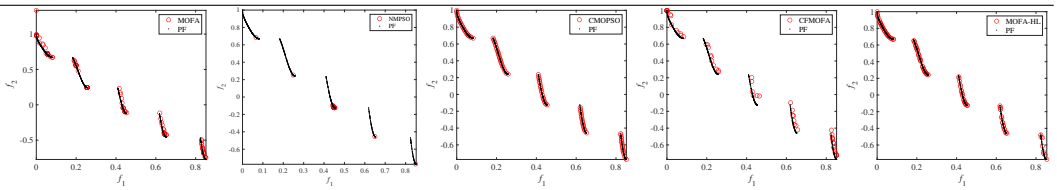
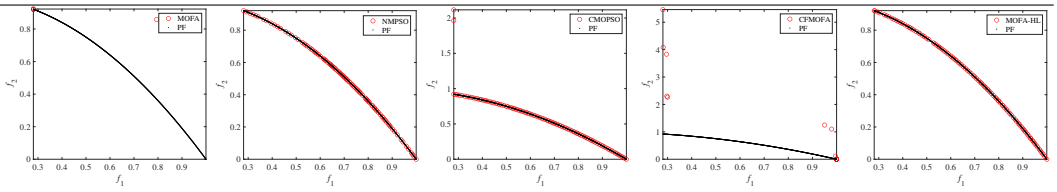
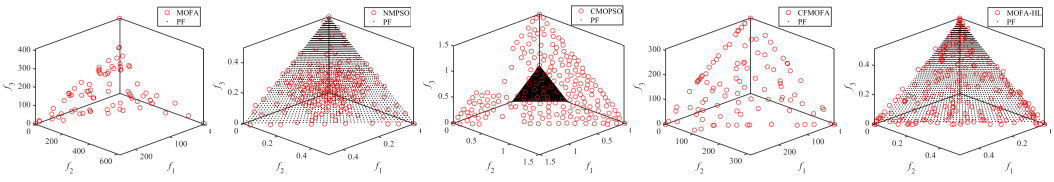
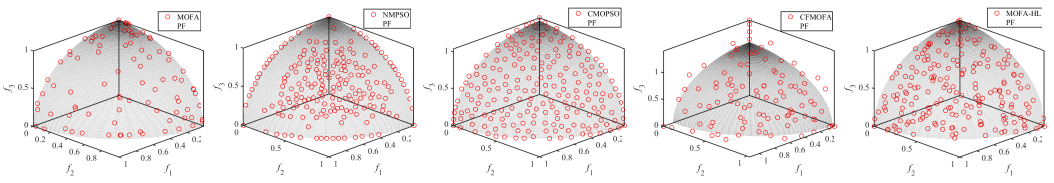
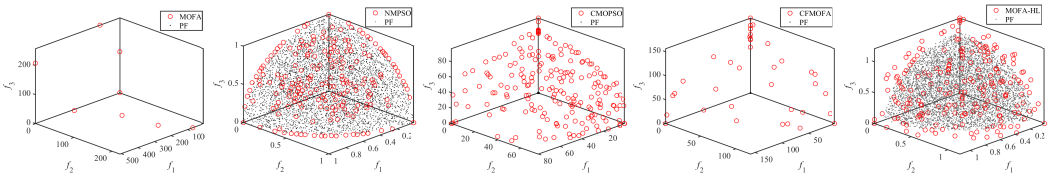
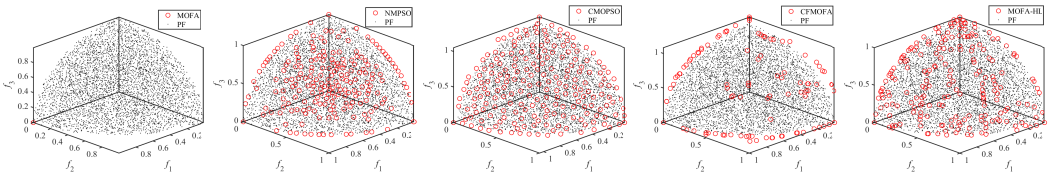
Problem	NSGA-II	MOEA/D	SPEA2	MOPSO	MOFA-HL
ZDT1					
ZDT2					
ZDT3					
ZDT6					
DTLZ1					
DTLZ2					
DTLZ3					
DTLZ4					

TABLE 8. Pareto front fitting diagram of MOFA-HL and four new MOEAs

Problem	MOFA	NMPSO	CMOPSO	CFMOFA	MOFA-HL
ZDT1					
ZDT2					
ZDT3					
ZDT6					
DTLZ1					
DTLZ2					
DTLZ3					
DTLZ4					

5. **Conclusions.** In reality, multi-objective optimization problems are increasingly complicated, so it is particularly important to design more efficient and feasible multi-objective optimization algorithms. In view of the optimization process of MOFA, the population is easy to fall into local optimum, and more high-quality solutions cannot be found, which leads to poor convergence and distribution of the population. In this paper, multi-objective firefly algorithm with hierarchical learning (MOFA-HL) is proposed. In this algorithm, hierarchical learning method is proposed, which enables the population to learn towards the superior front individuals and obtain more solutions close to the real Pareto front. After hierarchical learning, the population is operated with local mutation to promote the population to search for high-quality solutions in a small range. Finally, the crowding distance calculation is used to maintain the population distribution in the process of preserving the elite solution. Under the cooperation of multi-strategy, the algorithm can find more high-quality solutions with better distribution and convergence. MOFA-HL was compared with 4 classical and 4 new MOEAs, and the experimental data are tested by Friedman test. Meanwhile, the convergence curves of nine algorithms on some MOPs and the Pareto front fitting graphs of nine algorithms are plotted. The results show that MOFA-HL has the best overall ranking, which show that MOFA-HL can effectively improve the distribution and convergence of the population.

Acknowledgment. This work is supported by Jiangxi Province key laboratory of water information cooperative sensing and intelligent processing open project fund (No. 2016WICSIP022), the National Natural Science Foundation of China (Nos. 62066030, 61663029, 61762063) and the Jiangxi Province Department of Education Science and Technology Project (Nos. GJJ201915, No. GJJ170991).

REFERENCES

[1] E. Jiang, L. Wang, Multi-objective optimization based on decomposition for flexible job shop scheduling under time-of-use electricity prices, *Knowledge-Based Systems*, vol.204, 2020.

[2] F. Rezaei, H. R. Safavi, F-MOPSO/Div: An improved extreme-point-based multi-objective PSO algorithm applied to a socio-economic-environmental conjunctive water use problem, *Environmental Monitoring and Assessment*, vol.192, no.12, 2020.

[3] Y. Niu, Y. Zhang, Z. Cao, et al, MIMOA: A membrane-inspired multi-objective algorithm for green vehicle routing problem with stochastic demands, *Swarm and Evolutionary Computation*, vol.60, 2021.

[4] X. J. Cai, S. J. Geng, J. B. Zhang, et al, A sharding scheme based many-objective optimization algorithm for enhancing security in blockchain-enabled industrial internet of things, *IEEE Transactions on Industrial Informatics*, 2021, https://doi.org/10.1109/TII.2021.3051607.

[5] N. Srinivas, K. Deb, Multi-objective optimization using non-dominated sorting in genetic algorithms, *Evolutionary Computation*, vol.2, no.3, pp.221–248,1994.

[6] K. Deb, A. Pratap, S. Agarwal, et al, NSGA-II: A fast and elitist multi-objective genetic algorithm, *IEEE Transactions on Evolutionary Computation*, vol.6, no.2, pp.182–197, 2002.

[7] E. Zitzler, L. Thiele, Multi-objective 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] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, *Zurich: Swiss Federal Institute of Technology(ETH) Zurich*, 2001.

[9] K. Deb, H. Jainm, An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, Part I: Solving problems with box constraints, *IEEE Transactions on Evolutionary Computation*, vol.18, no.4, pp.577–601, 2014.

[10] Q. Zhang, H. Li, MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, vol.11, no.6, pp.712–731, 2008.

[11] H. Li, Q. Zhang, Multi-objective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, *IEEE Transactions on Evolutionary Computation*, vol.13, no.2, pp.284–302, 2009.

[12] Y. Qi, X. Ma, F. Liu, et al, MOEA/D with adaptive weight adjustment, *Evolutionary Computation*, vol.22, no.2, pp.231–264, 2014.
[13] E. Zitzler, K. Simon, Indicator-based selection in multi-objective search, *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature*, pp.832–842, 2004.
[14] M. Emmerich, N. Beume, B. Naujoks, An EMO algorithm using the hypervolume measure as selection criterion, *Proceedings of the 3rd International Conference Evolutionary Multi-Criterion Optimization*, pp.62–76, 2005.
[15] J. Bader, E. Zitzler, HypE: An algorithm for fast hypervolume-based many-objective optimization, *Evolutionary Computation*, vol.19, no.1, pp.45–76, 2011.
[16] C. A. C. Coello, M. S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, *Proceedings of the 2002 Congress on Evolutionary Computation*, pp.1051–1056, 2002.
[17] M. S. Zapotecas, C. A. C. Coello, A multi-objective particle swarm optimizer based on decomposition, *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp.69–76, 2011.
[18] Q. Lin, S. Liu, Q. Zhu, et al, particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems, *IEEE Transactions on Evolutionary Computation*, vol.22, no.2, pp.32–46, 2016.
[19] X. Zhang, X. Zheng, R. Cheng, et al, A competitive mechanism based multi-objective particle swarm optimizer with fast convergence, *Information Sciences*, vol.427, pp.63–76, 2018.
[20] X. S. Yang, Multi-objective firefly algorithm for continuous optimization, *Engineering with Computers*, vol.29, no.2, pp.175–184, 2013.
[21] L. Lv, J. Zhao, J. Wang, et al, Multi-objective firefly algorithm based on compensation factor and elite learning, *Future Generation Computer Systems*, vol.91, no.2, pp.37–47, 2019.
[22] J. Zhao, L. Lv, H. Wang, et al, Particle swarm optimization based on vector gaussian learning, *KSII Transactions on Internet and Information Systems*, vol.11, no.4, pp.2038–2057, 2017.
[23] J. S. Pan, N. Liu, S. C. Chu, et al, An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems, *Information Sciences*, vol.561, no.2, 2020.
[24] J. S. Pan, X. X. Sun, S. C. Chu, et al, Digital watermarking with improved sms applied for QR code, *Engineering Applications of Artificial Intelligence*, vol.1, no.97, 2021.
[25] J. Zhao, J. Tang, A. Shi, et al, Improved density peaks clustering based on firefly algorithm, *International Journal of Bio-Inspired Computation*, vol.15, no.1, pp.24–42, 2020.
[26] J. Zhao, W. Chen, J. Ye, et al, Firefly algorithm based on level-based attracting and variable step size. *IEEE Access*, vol.8, pp.58700–58716, 2020.
[27] L. Lv, J. Zhao. The firefly algorithm with gaussian disturbance and local search, *Journal of Signal Processing Systems for Signal Image and Video Technology*, vol.90, no.8-9, pp.1123–1131, 2018.
[28] J. Zhao, Z. F. Xie, L. Lv, et al, Deep learning firefly algorithm, *Chinese Journal of Electronics*, vol.46, no.11, pp.2633–2641, 2018.
[29] Y. Tian, Research on some key problems of many-objective optimization algorithm, *Hefei: Anhui university*, 2015.
[30] E. Zitzler, K. Deb, Comparison of multi-objective evolutionary algorithms: Empirical results, Evolutionary Computation, vol.2, no.8, pp.173–195, 2000.
[31] S. Zheng, Industrial intelligence technology and application, *Shanghai: Shanghai Science and Technology Publishing House*, 2019.
[32] Y. Tian, R. Cheng, X. Zhang, et al, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization, *IEEE Computational Intelligence Magazine*, vol.12, no.4, pp.73–87, 2017.
[33] K. Deb, L. Thiele, M. Laumanns, et al, Scalable test problems for evolutionary multi-objective optimization, *Multi-Objective Optimization*, vol.112, no.6, pp.105–145, 2005.
[34] J. Zheng, J. Zou, Multi-objective evolutionary optimization, *Beijing: Science Press*, 2017.