

## Research Article

# AMOBH: Adaptive Multiobjective Black Hole Algorithm

**Chong Wu,<sup>1,2</sup> Tao Wu,<sup>1,2</sup> Kaiyuan Fu,<sup>1,2</sup> Yuan Zhu,<sup>1,2</sup> Yongbo Li,<sup>1,2</sup>  
Wangyong He,<sup>1,2</sup> and Shengwen Tang<sup>3</sup>**

<sup>1</sup>School of Automation, China University of Geosciences, Wuhan 430074, China

<sup>2</sup>Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China

<sup>3</sup>State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Wuhan 430070, China

Correspondence should be addressed to Tao Wu; wutao@cug.edu.cn

Received 4 June 2017; Revised 1 October 2017; Accepted 22 October 2017; Published 23 November 2017

Academic Editor: José Alfredo Hernández-Pérez

Copyright © 2017 Chong Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a new multiobjective evolutionary algorithm based on the black hole algorithm with a new individual density assessment (cell density), called “adaptive multiobjective black hole algorithm” (AMOBH). Cell density has the characteristics of low computational complexity and maintains a good balance of convergence and diversity of the Pareto front. The framework of AMOBH can be divided into three steps. Firstly, the Pareto front is mapped to a new objective space called parallel cell coordinate system. Then, to adjust the evolutionary strategies adaptively, Shannon entropy is employed to estimate the evolution status. At last, the cell density is combined with a dominance strength assessment called cell dominance to evaluate the fitness of solutions. Compared with the state-of-the-art methods SPEA-II, PESA-II, NSGA-II, and MOEA/D, experimental results show that AMOBH has a good performance in terms of convergence rate, population diversity, population convergence, subpopulation obtention of different Pareto regions, and time complexity to the latter in most cases.

## 1. Introduction

Most problems can be considered as multiobjective optimization problems (MOPs) [1, 2] in the fields of social production [3], engineering design [4], path planning [5], product design [6], motor design [7], mechanics design [8], and so forth. For example, to design a new toll plaza of a highway, at least two objectives should be considered, which are traffic efficiency and land cost. Hence, the study of MOPs is very meaningful in many research fields. Unlike the single-objective optimization problems (SOPs), there are multiple global optimum solutions which are called Pareto optimal solutions in MOPs. And objectives are often conflicting with each other [1]. Some traditional approaches use the weight function to transform MOPs into SOPs. However, they require some prior knowledge and can only get one solution of the Pareto optimal solution set.

Over the past decades, bioinspired computation and swarm intelligence based algorithms have been introduced into solving MOPs called evolutionary multiobjective optimization. Bioinspired computation is motivated by the natural and social behavioral phenomena and can start with a set

of initial variables and then evolve to find multiple optima simultaneously [9]. Therefore, bioinspired computation is suitable for solving MOPs. The most popular multiobjective evolutionary algorithms (MOEAs) are Pareto dominance based algorithms [2], such as nondominated sorting genetic algorithm II (NSGA-II) [10] and strength Pareto evolutionary algorithm II (SPEA-II) [11]. Besides the criterion of Pareto dominance, they also adopted a diversity related secondary criterion to promote a good distribution of the solutions [12].

The BH algorithm was first proposed to solve the complex, hard optimization and clustering problem which is NP-hard [13]. It simulates the phenomenon of the black hole absorption. It has evolved from the PSO algorithm with new mechanisms. The BH algorithm searches the entire space of solutions (stars) and finds the global optimum solution (black hole). In the PSO algorithm, particles cannot disappear, which results in a premature convergence problem. To overcome the weakness of the PSO algorithm, in the BH algorithm, a star will be reborn randomly in the search space if it comes too close to the black hole. So, compared with the PSO algorithm, the BH algorithm has a better performance in avoiding the premature convergence phenomenon and a

less time complexity than PSOs and GAs. What is more, the BH algorithm has only one controlling parameter which is the radius of the black hole.

A lot of MOEAs have two main problems: (1) high selection pressure or low selection pressure and (2) how to balance the diversity and convergence. Therefore, in this paper, we proposed a new multiobjective evolutionary algorithm which is based on the black hole algorithm (BH algorithm) [13] to solve these problems. A new individual density criterion called cell density is proposed to improve the convergence and diversity of Pareto optimal solutions. In this paper, the global optimum selection strategy is based on the Shannon entropy and will be adjusted adaptively. The Shannon entropy is used to analyze the status of evolution. To calculate the entropy, the approximate Pareto front is mapped to PCCS [14]. The elite mutation strategy referred to in [14, 15] is used for improving the local search ability of the proposed algorithm. We rank solutions in an archive based on the cell density and the concepts of strength of cell dominance [14, 16], respectively. And we introduce adaptive evolution strategies to adjust the elite learning rate and the global optimum selection strategy according to evolution status adaptively. A simulation using seven test problems with different degrees of difficulty which are ZDT1, ZDT3, ZDT4 [17], DLTZ2, DLTZ4, DLTZ5, and DLTZ7 [18], respectively, is demonstrated to investigate the scalability of AMOBH. The simulation results showed that AMOBH has a good performance in both diversity and convergence during solving different multiobjective optimization problems. Furthermore, we perform a comparison with four well-known MOEAs—SPEA-II [11], PESA-II [19], NSGA-II [10], and MOEA/D [20]—for evaluation. Besides, we use the metric called inverted generation distance [20] which can evaluate convergence and diversity at the same time to analyze the results. In order to further verify the diversity or uniformity of the results, we use the metric called spacing [21–23] to corroborate it. As will be shown in this study, the proposed algorithm outperforms the other four algorithms in terms of diversity and convergence in most cases. And it has a good time complexity.

## 2. Related Works and Motivation

After [24] first introduced vector evaluated genetic algorithms (VEGAs) to solve multiobjective optimization problems, a lot of MOEAs based on GAs have been proposed. Among them, the most representative algorithms are NSGA-II [10], SPEA-II [11], and MOEA/D [20]. And, recently, a lot of MOEAs based on GAs for solving many-objective problems were also proposed [1, 2, 12, 25, 26]. But all of them suffer from a slow convergence rate and a lot of time on generating new offspring, which are the main problems of GAs [27].

Compared with GAs, the particle swarm optimization (PSO) algorithm has the advantages of simplified formula, rather quick convergence rate, global optimization performance, fewer controlling parameters, and so forth. Based on the successful experience in the field of MOEAs using GAs, a multiobjective particle swarm optimization algorithm (MOPSO) was proposed soon [28]. Compared to some

previous MOEAs, MOPSO has a faster convergence rate and it can get a rather satisfied and fully covered approximate Pareto front. Reference [29] introduced Shannon entropy [30] to analyze the MOPSO dynamics along the algorithm execution. Reference [14] introduced parallel coordinates in MOPSO to calculate entropy. However, the main problem of PSO algorithm is that it is easily trapped into a local optimum [31].

At present, a lot of MOEAs adopted the global optimum selection strategy based on the nondominated sorting [10, 11] or Pareto dominance [32] or hypervolume [33, 34] or niche [35, 36] and so on. But they all have some problems of high selection pressure or low selection pressure. In evolutionary multiobjective optimization, maintaining a good balance between convergence and diversity is particularly crucial to the performance of the evolutionary algorithms [37]. And most of the MOEAs face the problem of balancing the convergence and diversity. The evaluation of individual density is always based on one metric (such as Pareto dominance or crowding density), which results in the algorithm being unable to take into account both convergence and diversity [14].

Bearing these ideas and motivations in mind, an adaptive multiobjective black hole algorithm is proposed, investigated, and discussed in the following sections.

## 3. Definitions and Some Concepts

*3.1. Multiobjective Optimization Problem.* A minimum continuous unconstrained multiobjective optimization problem (in the optimization field, the maximization problems and minimization problems are dual problems) can be defined as follows:

$$\min F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]^T, \quad \mathbf{x} \in R^n, \quad (1)$$

in which  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in C$ ,  $n$  is the number of decision variables,  $f$  is the objective function,  $m$  is the number of objectives, and  $C$  is an  $n$ -dimensional search space.

*3.1.1. Pareto Optimality.* Normally, objectives are restricted or conflicting with each other [38], which results in the global optimum solution being not unique. There cannot be found a solution that is superior to all. But noninferior solutions exist; the so-called noninferior solutions are solutions that cannot be optimized for at least one objective; at the same time, other objectives will not deteriorate. And these solutions are called Pareto optimal.

Here are some definitions of Pareto optimality.

*Definition 1* (Pareto optimality). In search space,  $\mathbf{x}$  and  $\mathbf{x}^*$  are the decision vectors; if the following conditions are satisfied:

$$f_i(\mathbf{x}) < f_i(\mathbf{x}^*), \quad \forall i \in M, \quad \nexists \mathbf{x} \in C, \quad (2)$$

$\mathbf{x}^*$  is said to be Pareto optimal.

*Definition 2* (Pareto dominance). There are two objective vectors  $\mathbf{u}, \mathbf{v} \in W$ , where  $W$  is an  $m$ -dimensional objective space; if the following conditions are met:

$$\begin{aligned} \mathbf{v}_i &< \mathbf{u}_i, \quad \exists i \in M, \\ \mathbf{v}_j &\leq \mathbf{u}_j, \quad \forall j \in M, \quad j \neq i, \end{aligned} \quad (3)$$

$\mathbf{v}$  is said to dominate  $\mathbf{u}$ .

*Definition 3* (Pareto front). The set of all Pareto optimal solutions is called Pareto optimal set. The space composed of Pareto optimal objective vectors is called Pareto front.

*3.2. Parallel Cell Coordinate System.* Parallel coordinates are a common way of visualizing high-dimensional geometry and analyzing multivariate data [39]. Reference [16] proposed the GrEA which mapped the approximate Pareto front to grid coordinates. Inspired by these, [14] proposed a concept called parallel cell coordinate system (PCCS) which mapped the approximate Pareto front to a two-dimensional plane from Cartesian coordinates to integer coordinates. The formula of the transformation is defined as follows:

$$S_{n,m} = \begin{cases} \left\lceil N(t) \frac{f_{n,m} - f_m^{\min}}{f_m^{\max} - f_m^{\min}} \right\rceil, & f_{n,m} \neq f_m^{\min}; \\ 1, & f_{n,m} = f_m^{\min}, \end{cases} \quad (4)$$

where  $S$  is a two-dimensional plane grid composed of  $N(t) \times M$  cells,  $\lceil x \rceil$  is a top integral function,  $n = 1, 2, \dots, N(t)$ ,  $N(t)$  is the number of members in archive in the current iteration  $t$ ,  $m = 1, 2, \dots, M$ ,  $M$  is the number of objectives to be optimized,  $f_m^{\min}$  and  $f_m^{\max}$  are the minimum and maximum of objective  $m$  in the current approximate Pareto optimal set, respectively, and  $S_{n,m}$  is the integral coordinate or integral label of  $f_{n,m}$  in PCCS.

To rank the solutions in archive, we need to introduce the following definitions [14, 16].

*Definition 4* (cell dominance). If  $S_{i,m}$  and  $S_{j,m}$  ( $i \neq j$ ) are integral coordinates of any two solutions in an archive, at the same time, the following conditions are satisfied:

$$\begin{aligned} S_{i,m_1} &\leq S_{j,m_1}, \quad \forall m_1 \in M, \\ S_{i,m_2} &< S_{j,m_2}, \quad \exists m_2 \in M. \end{aligned} \quad (5)$$

Solution  $i$  is said to cell-dominate solution  $j$ .

*Definition 5* (strength of cell dominance). The total number of solutions which are cell-dominated by solution  $i$  is said to be the strength of cell dominance of solution  $i$ .

*3.3. Shannon Entropy.* In 1949, Shannon et al. in their paper [30] introduced a concept which is called Shannon entropy, which was proposed as a measure of the amount of information that is missing before reception.

Shannon entropy  $H$  is defined as follows:

$$H(X) = -K_e \sum_{x \in X} p_i(x) \log_2 p_i(x), \quad (6)$$

where  $p_i(x)$  is the probability of occurrence of the  $i$ th possible value of the source symbol,  $K_e$  is a positive constant, and  $X$  is the collection of events  $x$ .

In this paper, we refer to [14, 29] to calculate  $H$  as follows:

$$\begin{aligned} H(t) &= -N(t) \sum_{n=1}^{N(t)} \sum_{m=1}^M p_i(t) \log_2 p_i(t); \\ p_i(t) &= \frac{c \text{Num}_{n,m(t)}}{N(t) M}, \end{aligned} \quad (7)$$

in which  $c \text{Num}_{n,m(t)}$  is the number of objective vectors which are mapped to PCCS in the cell grid with indexes  $n$  and  $m$ ,  $N(t)$  is the number of solutions in archive in the current iteration  $t$  and it will be changed with the change of the solution's number in archive, and  $M$  is the number of objectives.

The update formula of entropy is as follows:

$$\Delta H(t+1) = H(t+1) - H(t). \quad (8)$$

It is a measure to evaluate the status of evolution. It was mentioned in [29] that entropy is able to capture the convergence rate of the algorithm. Entropy represents uniformity and diversity of approximate Pareto optimal solutions. Larger entropy means better uniformity and diversity. The evolution status update algorithm is as shown in Algorithm 1 [14], where  $N(t+1)$  and  $N(t)$  are the number of solutions in archive in iterations  $t+1$  and  $t$ , respectively, and  $N$  is the size of the archive. The initial entropy is  $H(0)$  and the initial delta entropy is  $\Delta H(0) = H(0)$ .

*3.4. Black Hole Algorithm.* Inspired by the black hole phenomenon, the BH algorithm was first put forward in [13].

In the BH algorithm, the best candidate at each iteration is selected as a black hole and all the other candidates form the normal stars. The creation of the black hole is not random and it is one of the real candidates of the population. Then, all the candidates move towards the black hole based on their current location and a random number.

The details of the BH algorithm are as follows. Like other population-based algorithms, in the proposed BH algorithm, a randomly generated population of candidate solutions (the stars) is placed in the search space of some problems or functions. After the initialization, the fitness values of the population are evaluated and the best candidate in the population, which has the best fitness value, is selected to be the initial black hole; at the same time, the rest form the normal stars. The black hole has the ability to absorb the stars that surround it. After initializing the black hole and stars, the black hole starts absorbing the stars around it and all the stars start moving towards the black hole. The absorption of stars by the black hole is formulated as follows:

$$\begin{aligned} X_i(t+1) &= X_i(t) + \text{rand} \cdot (X_{\text{bh}} - X_i(t)), \\ i &= 1, 2, \dots, K. \end{aligned} \quad (9)$$

Here,  $X_i(t)$  and  $X_i(t+1)$  are the locations of the star  $i$  in iterations  $t$  and  $t+1$ , respectively,  $X_{\text{bh}}$  is the location of the

**Require:** Entropy  $H$ , delta entropy  $\Delta H$ , and the size of archive  $N$ ;  
(1) **if**  $|\Delta H(t + 1)| > (2/N(t + 1)) \log_2(2) \parallel |N(t + 1) - N(t)| > 0$  **then**  
(2)      $evolutionStatus = 'convergence'$ ;  
(3) **elseif**  $(2/MN) \log_2(2) < |\Delta H(t + 1)| < (2/N(t + 1)) \log_2(2)$  &&  $N(t + 1) == N(t) == N$   
(4)      $evolutionStatus = 'diversity'$ ;  
(5) **elseif**  $|\Delta H(t + 1)| < (2/MN) \log_2(2)$  &&  $N(t + 1) == N(t)$   
(6)      $evolutionStatus = 'stagnation'$ ;  
(7) **end if**  
(8) **return**  $evolutionStatus$ ;

ALGORITHM 1: Evolution status update.

black hole in the search space,  $rand$  is a random number in the interval  $[0, 1]$ , and  $K$  is the number of stars (candidate solutions) or the size of population.

Every star (candidate solution) that crosses the event horizon of the black hole will be absorbed by the black hole. The radius of the event horizon in the BH algorithm is calculated using the following equation:

$$R = \left| \frac{F_{bh}}{\sum_{i=1}^K F_i} \right|, \quad (10)$$

where  $F_{bh}$  is the fitness value of the black hole,  $F_i$  is the fitness value of the star  $i$ ,  $K$  is the number of stars (candidate solutions), and  $R$  is the radius of the black hole.

When the distance between a candidate solution and the black hole (best candidate) is less than  $R$ , the candidate is collapsed and meanwhile a new candidate is created and distributed randomly in the search space.

## 4. Adaptive Multiobjective Black Hole Algorithm

*4.1. Multiobjective Black Hole Algorithm with Mutation.* When the BH algorithm is extended from single-objective optimization to multiobjective optimization, the radius formula will be a little different. At the same time, the criterion of whether a star crosses the event horizon of the black hole or not is also changed. In this paper, the radius of the event horizon is defined as follows:

$$R_{j,k} = \left| \frac{Fb_{j,k}}{\sum_{i=1}^K F_{i,k}} \right|, \quad k = 1, 2, \dots, M, \quad (11)$$

in which  $R_{j,k}$  is the radius of the  $k$ th objective of the black hole  $j$ ;  $Fb_{j,k}$  is the fitness value of the  $k$ th objective of the black hole  $j$ ,  $j = 1, 2, \dots, N_b$ ;  $N_b$  is the number of black holes; we set it to  $2M$ ; the reason will be explained in Section 4.4; and  $F_{i,k}$  is the fitness value of the  $k$ th objective of star  $i$ .

Therefore, the criterion of whether a star crosses the event horizon of the black hole or not is as follows: if  $\exists i \in 1, 2, \dots, K$ ,  $j \in 1, 2, \dots, 2M$ ,  $\forall k \in 1, 2, \dots, M$ ,  $|F_{i,k} - Fb_{j,k}| \leq R$ ,

$$X_i(t + 1) = (X_{\text{boundmax}} - X_{\text{boundmin}}) \cdot rand + X_{\text{boundmin}}. \quad (12)$$

Here,  $X_{\text{boundmax}}$  is the upper bound of search space,  $X_{\text{boundmin}}$  is the lower bound of search space,  $rand$  is a random number in the interval  $[0, 1]$ , and  $X_i(t + 1)$  is the location of the  $i$ th star in iteration  $t + 1$ .

*4.1.1. Star Mutation.* Almost all PSO algorithms adopt the mutation to improve the prevention of precocious puberty and avoid the local optimum [40]. Similarly, this paper introduces this strategy to the BH algorithm. We introduce a random mutation into the stars' location update. If a random number is smaller than  $p$ ,  $p$  is the mutation rate which will affect the degree of random oscillation, and then the star's location will be updated in search space randomly (the larger the value of  $p$ , the greater the random oscillation). This paper set  $p = 0.3$ .

The mutation of a star's location is shown as follows:

$$X_i(t + 1) = (X_{\text{boundmax}} - X_{\text{boundmin}}) \cdot rand + X_{\text{boundmin}}, \quad i = 1, 2, \dots, K. \quad (13)$$

This formula will enhance the proposed algorithm's ability of global search and improve the searching accuracy by increasing random oscillation.

*4.2. Cell Density.* When the archive of approximate Pareto optimal solutions is full, both new solutions and old solutions are not superior to each other. To get well-distributed diversity, the algorithm needs to estimate the impact of replacing one old solution with a new solution on the individual density. This paper presents a new individual density concept called cell density in PCCS which uses the total number of solutions in the cell grid occupied by one solution as its individual density. The formula of cell density is defined as follows:

$$D_i = \frac{\sum_{m=1}^M \sum_{j=1}^{N_s} (S_{i,m} == S_{j,m})}{V}, \quad (14)$$

where  $D_i$  is the cell density of solution  $i$ ,  $M$  is the number of objectives,  $N_s = N(t) + 1$  is the number of solutions (when the archive is full,  $N_s = N + 1$ ),  $N$  is the size of the archive,  $S_{i,m}$  and  $S_{j,m}$  are the integral labels of solution  $i$  and solution  $j$  of objective  $m$  in PCCS, respectively, and  $V$  is the area; in this paper, it is set to value 1.

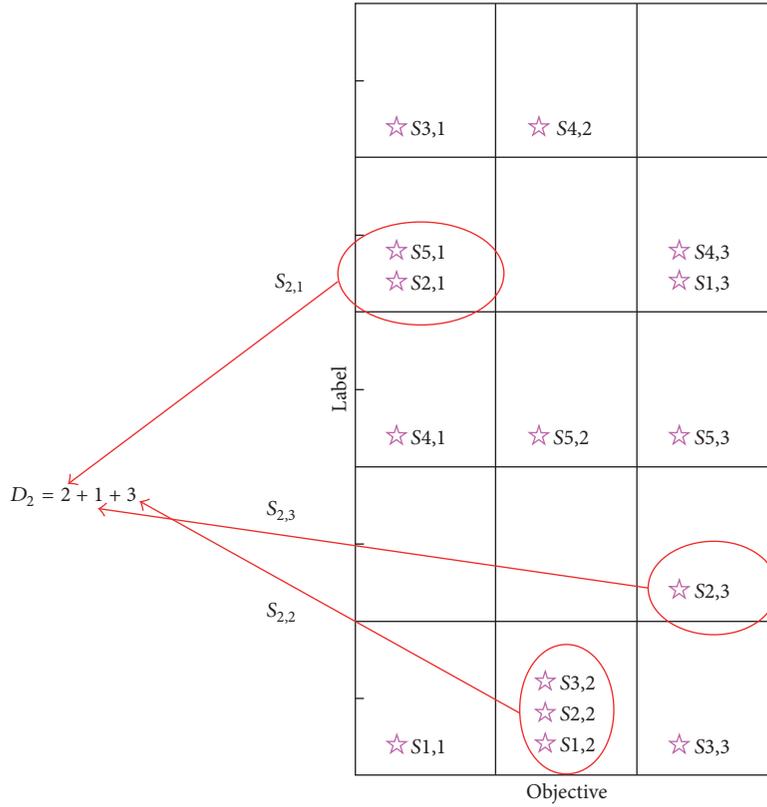


FIGURE 1: Approximate Pareto front in PCCS.

TABLE 1: The cell density.

Solution 1	Solution 2	Solution 3	Solution 4	Solution 5
6	6	5	4	4

Take Figure 1 as an example; in this example,  $M = 3, N = 5$ .  $S_{1,1}$  means solution 1 of objective 1. The cell density of all solutions is shown in Table 1.

As can be seen from Table 1, solution 1 and solution 2 have the max cell density, which is also proved by Figure 1. The example of how to compute the cell density of solution 2 is shown in Figure 1. If a new solution's cell density is lower than the max cell density of all old solutions, the new solution replaces one old solution of the max cell density and updates the archive. Otherwise, it should be refused.

**4.3. Archive Maintenance Strategy.** The algorithm begins with a fixed size ( $N$ ) of archive. The archive is used for storing elite solutions. It needs to be updated when new solutions come in. The strategy of the archive's update is based on the cell density. When a new solution comes, there will be five situations as follows.

- (1) When the new solution and all old solutions are nondominant to each other and the archive is not full, accept the new solution into the archive.
- (2) When the new solution is inferior to some old solutions, refuse the new solution.

- (3) When the new solution is dominant to some old solutions, replace all inferior old solutions with the new solution.
- (4) When the new solution and all old solutions are nondominant to each other and the archive is full, we need to evaluate the cell density of the new solution and all old solutions. And if the cell density of the new solution is not the largest one, replace one old solution which has the largest cell density with the new solution.
- (5) When the new solution and all old solutions are nondominant to each other and the archive is full, we need to evaluate the cell density of the new solution and all old solutions. And if the cell density of the new solution is the largest one, refuse the new solution.

This strategy maintains the diversity of solutions in population evolution by pruning the large density solutions from the archive. It will maintain the high quality solutions and improve the convergence rate. The algorithm of the strategy is shown in Algorithm 2.

**4.4. Adaptive Global Optimum Selection Strategy.** In SOPs, the BH algorithm uses the global optimum solution as the

```

Require: The archive  $Ar$  and the new solution  $X(j)$ .
(1) if  $isFull(Ar)$  then
(2)   if  $isAnyDominant(X(j), Ar)$  then
(3)     %Remove all the inferior solutions.
(4)      $Remove(isInferior(Ar, X(j)), Ar)$ ;
(5)      $Accept(X(j), Ar)$ ;
(6)   else
(7)     if  $isAllNondominant(X(j), Ar)$  then
(8)       %Evaluate the cell density.
(9)        $CD = CellDensity([X(j), Ar])$ ;
(10)      if  $CD(X(j)) < max(CD)$  then
(11)         $Replace(Ar[max(CD)], X(j))$ ;
(12)      else
(13)        end if
(14)      else
(15)        end if
(16)      end if
(17) else
(18)   if  $isAnyDominant(X(j), Ar)$  then
(19)      $Remove(isInferior(Ar, X(j)), Ar)$ ;
(20)      $Accept(X(j), Ar)$ ;
(21)   else
(22)     if  $isAllNondominant(X(j), Ar)$  then
(23)        $Accept(X(j), Ar)$ ;
(24)     else
(25)       end if
(26)     end if
(27) end if
(28) return  $Ar$ ;

```

ALGORITHM 2: Archive maintenance.

black hole to update the location of stars (candidate solutions) and determine the direction of evolution. When the BH algorithm is extended to MOPs, the global optimum solution is not unique. If all the approximate Pareto optimal solutions are adopted as the black holes, this will cause a small selection pressure, hindering the effective promotion ability of the evolution process. What is more, it will slow down the convergence rate of the BH algorithm and make it premature and lose diversity. Accordingly, a reasonable strategy of global optimum selection which balances the diversification and intensification of the algorithm is necessary.

To get an approximate Pareto optimal solution set of good diversity and convergence, we need to select some nondominant solutions from the archive which can represent the diversity and convergence as the global optimum solutions. Because the convergence and diversity of solutions are generally conflicting, using different metrics to evaluate the convergence and diversity of the approximate Pareto optimal solutions in the archive, respectively, is very necessary.

This paper adopts the strength of the cell dominance as the convergence ranking metric and the cell density as the diversity ranking metric for selecting the global optimum solutions. It was mentioned in [14] that the whole population can share a common global optimum solution, but this will make the algorithm search with a single point which produces a high selection pressure, or each individual in the population

uses different global optimum solutions, but this will lead to a scatter search which produces a low selection pressure. To control the selection pressure, we adopt the strategy of selecting global optimum solutions referred to in [14]. The strategy selects a total of  $2M$  global optimum solutions from the archive for each iteration. And the selection process is based on the evolution status which can be adjusted with the change of evolution environment dynamically. Here, we give the algorithm of this strategy as shown in Algorithm 3.

*4.5. Adaptive Elite Mutation Strategy.* Inspired by [15], we adopt the strategy of elite mutation to increase the oscillation of local search. As mentioned earlier, all the solutions in the archive can be regarded as elite solutions. But here, we use this strategy to update the black holes (global optimal solutions' set which is got from Algorithm 3). Referred to in [14], the learning rate update algorithm is as shown in Algorithm 4, where  $\Delta H(t)$  is the delta entropy of iteration  $t$ ,  $T$  is the maximum of iteration,  $t/T$  is the update step size, and  $l_{max}$  and  $l_{min}$  are the maximum and minimum of learning rate, respectively; in this paper, we set  $l_{max} = 0.6$  and  $l_{min} = 0.1$ .

The learning rate update algorithm will increase the oscillation of local search to skip the local optimum solutions quickly when the evolution status is "stagnation." And if evolution status is "diversity," the learning rate will decrease for improving the searching accuracy. It is a feedback control for balancing the diversification and intensification dynamically. And the elite mutation formula is as follows [15]:

$$x_i(t+1) = (x_i^{max} - x_i^{min}) \cdot \text{Gaussian}(0, \text{rand}^2) + x_i(t), \quad (15)$$

where  $x_i(t+1)$  and  $x_i(t)$  are the  $i$ th decision variables of an elite solution in iterations  $t+1$  and  $t$ , respectively,  $x_i^{max}$  is the maximum of the  $i$ th decision variable of all elite solutions,  $x_i^{min}$  is the minimum of the  $i$ th decision variable of all elite solutions,  $\text{rand}$  is a random number uniformly distributed in the interval  $[0, 1]$ , and  $\text{Gaussian}(0, \text{rand}^2)$  is a random number of a Gaussian function with a 0 mean and a standard deviation  $\text{rand}$ .

*4.6. The Whole Algorithm.* Through the previous analysis, the whole algorithm description of AMOBH is shown in Algorithm 5, where  $APF$  is the approximate Pareto optimal front,  $APS$  is the approximate Pareto optimal solutions, and  $oV$  is the objective vectors.

*4.7. Computation Analysis.* In Section 4.6, we can see that Algorithm 2 is in the innermost loop. And we need to calculate  $N+1$  solutions' cell density in the worst case when the archive is full. At the same time, to calculate cell density, the parallel cell coordinates of each solution need to be calculated. And this computation needs to be executed  $M \times (N+1)$  times in the worst case. So, the total computation of Algorithm 2 is  $O(MN^2)$  which determines the computation of the full algorithm.

**Require:** The archive  $Ar$ .

- (1) %Calculate the cell density of solutions in archive.
- (2)  $CD = CellDensity(Ar)$ ;
- (3) %Calculate the strength of cell dominance of solutions in archive.
- (4)  $SCD = StrCellDominance(Ar)$ ;
- (5) %Sort the solutions with cell density in ascending order.
- (6)  $SC = Sort(CD, 'ascend')$ ;
- (7) %Sort the solutions with strength of cell dominance in descending order.
- (8)  $SS = Sort(SCD, 'descend')$ ;
- (9) %Select 2M solutions from archive as the global optimum solutions.
- (10) **if**  $evolutionStatus == 'convergence'$  **then**
- (11)      $gBholes = [Ar[FindFirst(SC, M - 1)], Ar[FindFirst(SS, M + 1)]]$ ;
- (12)     %Select the first  $M - 1$  solutions from SC.
- (13)     %And select the first  $M + 1$  solutions from SS.
- (14) **elseif**  $evolutionStatus == 'stagnation'$
- (15)      $gBholes = [Ar[FindFirst(SC, M)], Ar[FindFirst(SS, M)]]$ ;
- (16) **elseif**  $evolutionStatus == 'diversity'$
- (17)      $gBholes = [Ar[FindFirst(SC, M + 1)], Ar[FindFirst(SS, M - 1)]]$ ;
- (18) **end if**
- (19) **return**  $gBholes$ ;

ALGORITHM 3: Adaptive global optimum selection.

**Require:** The elite learning rate  $l$ .

- (1) **if**  $evolutionStatus == 'convergence'$  **then**
- (2)      $l(t) = l(t - 1)$ ;
- (3) **elseif**  $evolutionStatus == 'stagnation'$
- (4)      $l(t) = l(t - 1) + 2(1 + \Delta H(t))(l_{\max} - l_{\min})\frac{t}{T}$ ;
- (5) **elseif**  $evolutionStatus == 'diversity'$
- (6)      $l(t) = l(t - 1) - \Delta H(t)(l_{\max} - l_{\min})\frac{t}{T}$ ;
- (7) **end if**
- (8) **return**  $l$ ;

ALGORITHM 4: Elite learning rate update.

## 5. Results and Discussion

In this section, the paper uses seven functions with 2 or 3 objectives to be optimized of different degrees of difficulty to test the scalability of AMOBH. The dimensions of test functions range from 10 to 30. The seven test functions are ZDT1, ZDT3, ZDT4 [17], DLTZ2, DLTZ4, DLTZ5, and DLTZ7 [18]. These functions will test the proposed algorithm's performance in the different characteristics of the Pareto front: convexity, concaveness, discreteness, nonuniformity, and multimodality. All functions are to be optimized using AMOBH with iterations  $T = 1000$ , population size of stars  $K = 300$ , and archive size  $N = 50$ . To compare the performance of the proposed algorithm (AMOBH) with other multiobjective evolutionary algorithms, in this paper, we select four well-known multiobjective evolutionary algorithms (SPEA-II, PESA-II, NSGA-II, and MOEA/D) as the comparative algorithms. And the results of these functions' optimization and the performance analysis of comparative experiments are shown in the following sections.

All experiments are carried out on a personal computer with an Intel Core 2.3 GHz CPU, 4 GB memory, and Windows 10 OS. The test functions are as follows.

ZDT1:

$$\begin{aligned} \min \quad & f_1(\mathbf{x}) = x_1; \\ \min \quad & f_2(\mathbf{x}) = g(\mathbf{x}) \left( 1 - \sqrt{\frac{x_1}{g(\mathbf{x})}} \right); \\ \text{s.t.} \quad & g(\mathbf{x}) = 1 + 9 \frac{\sum_{i=2}^n x_i}{n-1}, \\ & x_i \in [0, 1], \quad i = 1, 2, \dots, n, \quad n = 30. \end{aligned} \quad (16)$$

ZDT3:

$$\begin{aligned} \min \quad & f_1(\mathbf{x}) = x_1; \\ \min \quad & f_2(\mathbf{x}) \\ & = g(\mathbf{x}) \left( 1 - \sqrt{\frac{x_1}{g(\mathbf{x})}} - \frac{x_1 \sin(10\pi x_1)}{g(\mathbf{x})} \right); \\ \text{s.t.} \quad & g(\mathbf{x}) = 1 + 9 \frac{\sum_{i=2}^n x_i}{n-1}, \\ & x_i \in [0, 1], \quad i = 1, 2, \dots, n, \quad n = 30. \end{aligned} \quad (17)$$

ZDT4:

$$\begin{aligned} \min \quad & f_1(\mathbf{x}) = x_1; \\ \min \quad & f_2(\mathbf{x}) = g(\mathbf{x}) \left( 1 - \sqrt{\frac{x_1}{g(\mathbf{x})}} \right); \\ \text{s.t.} \quad & g(\mathbf{x}) \end{aligned}$$

**Require:** The stars' population  $X$ , the archive  $Ar$ , the variable bound  $V_b$ , the max iteration  $T$ , and the elite learning rate  $l$ .

- (1)  $oV = \text{fitnessFcn}(X)$ ;
- (2)  $[APF, APS] = \text{getInitAP}(oV, X)$ ;
- (3) %Put the APS into  $Ar$ .
- (4)  $Ar[1 : \text{length}(APS)] = APS$ ;
- (5) %Map the APF to PCCS and calculate the  $H$  and  $\Delta H$ .
- (6)  $PCCS = \text{MapToPCCS}(APF)$ ;
- (7)  $[H, \Delta H] = \text{CalSE}(PCCS)$ ;
- (8) %Update the evolution status.
- (9) **Algorithm 1**;
- (10) %Get the candidate black holes.
- (11) **Algorithm 3**;
- (12) **for**  $i = 1$  **to**  $T$
- (13)   **for**  $j = 1$  **to**  $K$
- (14)    %Randomly choose a black hole for star update.
- (15)     $bBhole = gBholes(\text{randperm}(\text{length}(gBholes), 1))$ ;
- (16)    **if**  $\text{rand} < l$  **then**
- (17)      %Elite mutation.
- (18)       $bBhole = EM(bBhole)$ ;
- (19)    **end if**
- (20)     $X[j] = X[j] + \text{rand} \cdot (bBhole - X[j])$ ;
- (21)     $oV(j) = \text{fitnessFcn}(X(j))$ ;
- (22)    **if**  $\text{rand} < p$  **then**
- (23)       $X[j] = V_b.\text{min} + \text{rand} \cdot (V_b.\text{max} - V_b.\text{min})$ ;
- (24)       $oV(j) = \text{fitnessFcn}(X(j))$ ;
- (25)    **end if**
- (26)    %Determine whether accept the star  $j$ .
- (27)    **Algorithm 2**;
- (28)   **end for**
- (29)    $[APF, APS] = \text{getAP}(Ar)$ ;
- (30)    $PCCS = \text{MapToPCCS}(APF)$ ;
- (31)    $[H, \Delta H] = \text{CalSE}(PCCS)$ ;
- (32)   **Algorithm 1**;
- (33)   %Update the learning rate  $l$ .
- (34)   **Algorithm 4**;
- (35)   %Whether a star crosses the event horizon of the black holes or not.
- (36)   **for**  $j = 1$  **to**  $K$
- (37)     **if**  $\text{any}(\text{distance}(X(j), gBholes) \leq R)$  **then**
- (38)        $X[j] = V_b.\text{min} + \text{rand} \cdot (V_b.\text{max} - V_b.\text{min})$ ;
- (39)     **end if**
- (40)   **end for**
- (41)   **Algorithm 3**;
- (42) **end for**
- (43) **return**  $APF, APS$ ;

ALGORITHM 5: Adaptive multiobjective black hole algorithm.

$$= 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)),$$

$$x_1 \in [0, 1], x_i \in [-5, 5], i = 2, 3, \dots, n, n = 10.$$
(18)

DLTZ2:

$$\min f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos\left(\frac{x_1\pi}{2}\right) \cos\left(\frac{x_2\pi}{2}\right);$$

$$\min f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos\left(\frac{x_1\pi}{2}\right) \sin\left(\frac{x_2\pi}{2}\right);$$

$$\min f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin\left(\frac{x_1\pi}{2}\right);$$

$$\text{s.t. } g(\mathbf{x}) = \sum_{i=3}^n (x_i - 0.5)^2,$$

$$x_i \in [0, 1], i = 1, 2, \dots, n, n = 10.$$

(19)

DLTZ4:

$$\min f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos\left(\frac{x_1^\alpha\pi}{2}\right) \cos\left(\frac{x_2^\alpha\pi}{2}\right);$$

$$\begin{aligned}
\min \quad & f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos\left(\frac{x_1^\alpha \pi}{2}\right) \sin\left(\frac{x_2^\alpha \pi}{2}\right); \\
\min \quad & f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin\left(\frac{x_1^\alpha \pi}{2}\right); \\
\text{s.t.} \quad & g(\mathbf{x}) = \sum_{i=3}^n (x_i - 0.5)^2, \\
& x_i \in [0, 1], \quad i = 1, 2, \dots, n, \quad n = 10, \quad \alpha = 100.
\end{aligned} \tag{20}$$

DLTZ5:

$$\begin{aligned}
\min \quad & f_1(\mathbf{x}) \\
& = (1 + g(\mathbf{x})) \cos\left(\frac{x_1 \pi}{2}\right) \cos\left(\frac{\pi(1 + 2g(\mathbf{x})x_2)}{4(1 + g(\mathbf{x}))}\right);
\end{aligned}$$

$$\begin{aligned}
\min \quad & f_2(\mathbf{x}) \\
& = (1 + g(\mathbf{x})) \cos\left(\frac{x_1 \pi}{2}\right) \sin\left(\frac{\pi(1 + 2g(\mathbf{x})x_2)}{4(1 + g(\mathbf{x}))}\right); \\
\min \quad & f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin\left(\frac{x_1 \pi}{2}\right); \\
\text{s.t.} \quad & g(\mathbf{x}) = \sum_{i=3}^n (x_i - 0.5)^2,
\end{aligned}$$

$$x_i \in [0, 1], \quad i = 1, 2, \dots, n, \quad n = 10. \tag{21}$$

DLTZ7:

$$\begin{aligned}
\min \quad & f_1(\mathbf{x}) = x_1; \\
\min \quad & f_2(\mathbf{x}) = x_2; \\
\min \quad & f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \left( 3 - \frac{f_1(\mathbf{x})}{1 + g(\mathbf{x})} (1 + \sin(3\pi f_1(\mathbf{x}))) - \frac{f_2(\mathbf{x})}{1 + g(\mathbf{x})} (1 + \sin(3\pi f_2(\mathbf{x}))) \right); \\
\text{s.t.} \quad & g(\mathbf{x}) = 1 + \frac{9}{n-3} \sum_{i=3}^n x_i, \quad x_i \in [0, 1], \quad i = 1, 2, \dots, n, \quad n = 20.
\end{aligned} \tag{22}$$

**5.1. Results of ZDT1 Optimization.** The first optimization function is ZDT1. ZDT1 is very simple and is only used for the validation of feasibility of the proposed algorithm's application in multiobjective optimization problems. It has a convex Pareto optimal front. Figure 2(a) illustrates the true and approximate Pareto front of ZDT1 in one of the experiments. Figure 2(b) shows the entropy evolution during the optimization of ZDT1. As mentioned earlier, the entropy curve can be a measure to capture the convergence rate of the algorithm.

As can be seen from Figure 2(b), there is an ascending period which takes about 100 iterations. After that, the curve becomes smooth. Hence, AMOBH has a good convergence rate. It can be seen that the initial delta entropy is quite large. This is because the initial approximate Pareto optimal solutions that we select are  $M$  solutions which have the best fitness in at least one objective. As mentioned earlier, the entropy will change with the change of solutions' number in archive and the initial delta entropy is equal to the entropy in iteration 0. So, the initial cell density is very small, making the initial entropy and delta entropy quite large. And from Figure 2(a), we can see that AMOBH can get an approximate Pareto front which is quite close to the true Pareto front and has a good diversity.

**5.2. Results of ZDT3 Optimization.** To test the performance of the proposed algorithm in discontinuous problems, the second optimization function that we choose is ZDT3. It has

a total of 30 decision variables. ZDT3 represents the discreteness features [17]. Its Pareto front has several noncontiguous convex parts. Figure 3(a) illustrates the true and approximate Pareto front of ZDT3 in one of the experiments. From Figure 3(a), we can see that the convergence and diversity of the approximate Pareto front which is obtained by the proposed algorithm are quite good. Entropy evolution during the optimization of ZDT3 is represented in Figure 3(b). It can be seen that the convergence rate does not largely change. And it has been proved that the proposed algorithm has a good performance in some disconnected problems.

**5.3. Results of ZDT4 Optimization.** The third optimization function is ZDT4. The Pareto front of ZDT4 is convex and it contains many local solutions. Hence, we use this function to test the proposed algorithm's ability to deal with multimodality. Figure 4(a) illustrates Pareto front of ZDT4 in one of the experiments. In this function, we set the elite learning rate to 0.4 to get a better balance of diversity and convergence rate. We can see that the proposed algorithm converges very close to the true Pareto optimal front and it maintains a very diverse approximate Pareto optimal solution set. The entropy evolution during the optimization of ZDT4 is represented in Figure 4(b).

In Figure 4(b), we can see that the entropy curve is very different from that of previous functions' optimization. It has gone through several drastic changes. This is because it contains many local Pareto fronts. To skip these local

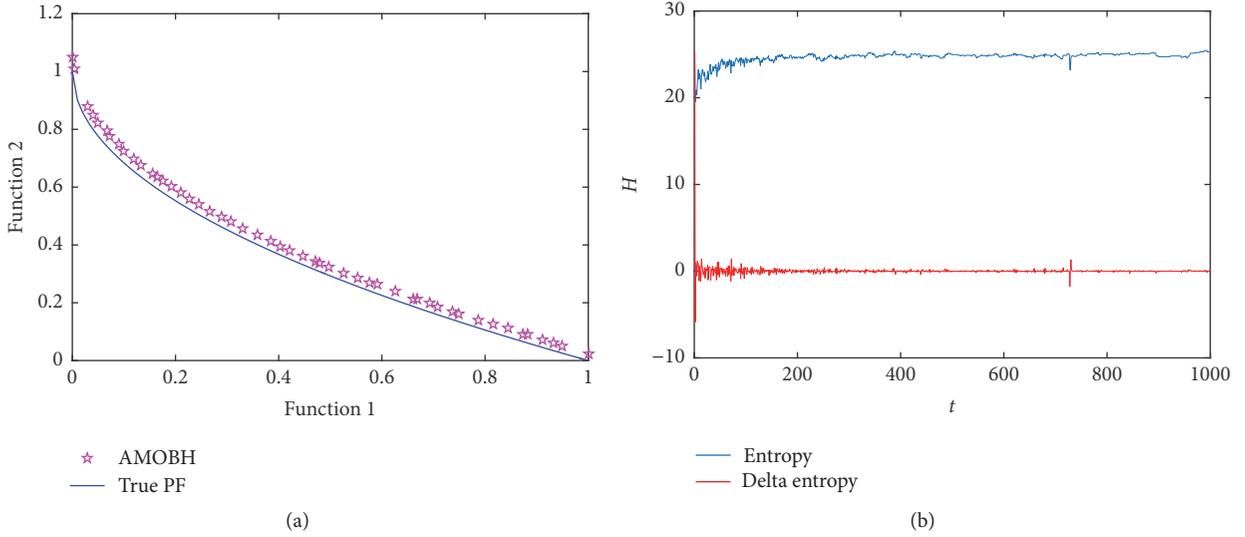


FIGURE 2: Simulation results of ZDT1. (a) The final approximate Pareto front of ZDT1. (b) Entropy during the AMOBH evolution for ZDT1.

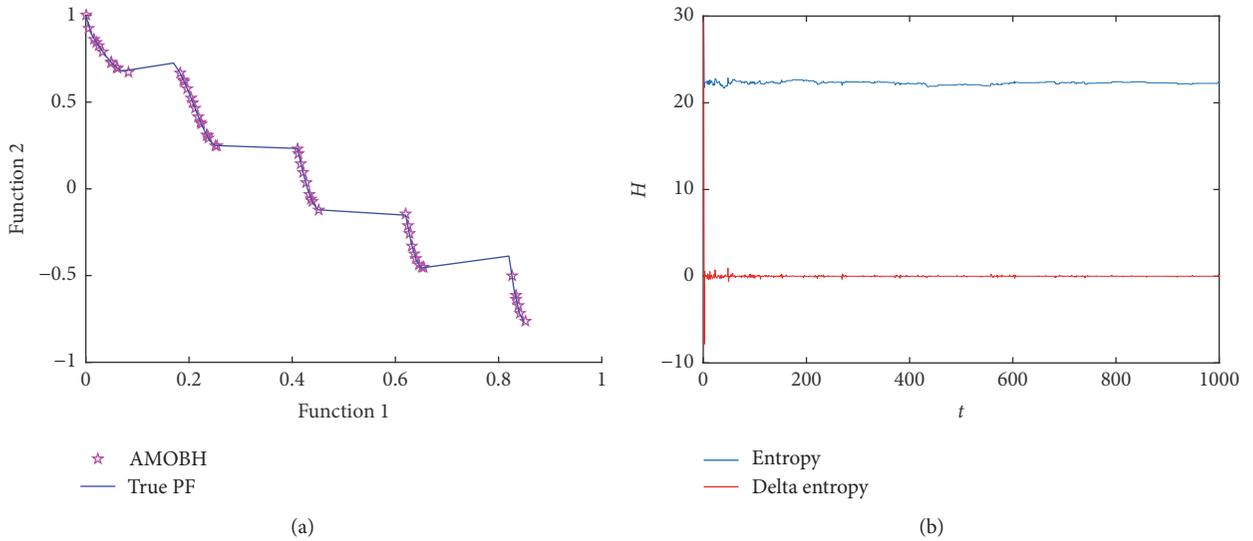


FIGURE 3: Simulation results of ZDT3. (a) The final approximate Pareto front of ZDT3. (b) Entropy during the AMOBH evolution for ZDT3.

Pareto fronts, many local solutions will be replaced from the archive which makes the drastic oscillation of the entropy curve. Therefore, it takes more time to reach convergence. It takes about 600 iterations to be stable. It is seen that the convergence rate of the proposed algorithm during solving some multimodal problems is not very ideal. It needs to be improved.

**5.4. Results of DLTZ2 Optimization.** From this section, we extend test functions from two-objective problems to three-objective problems to test the scalability of AMOBH. The first three-objective optimization function is DLTZ2. It has 10 decision variables. The true Pareto front of DLTZ2 is a surface  $F_1^2 + F_2^2 + F_3^2 = 1$ , as the results illustrated in Figure 5(a). Figure 5(a) also illustrates the approximate Pareto front of DLTZ2 in one of the experiments. It can be seen

that the proposed algorithm converges to an approximate Pareto front which is very close to the true Pareto front and has a good uniformity on the surface. Figure 5(b) shows entropy evolution during the optimization of DLTZ2. We can see that the entropy curve is quite similar to ZDT3. Both of them have a very short dynastic initial period and a stable period with slight oscillation. And it has been proved that the proposed algorithm has a high convergence rate in some concave problems and a good scalability when the number of objectives is expanded from 2 to 3.

**5.5. Results of DLTZ4 Optimization.** To further investigate the proposed algorithm's performance in three-objective optimization problems, the next optimization function we choose is DLTZ4. The function of DLTZ4 is very similar to DLTZ2, but all  $x_i$  in objectives of DLTZ2 are replaced by  $x_i^a$

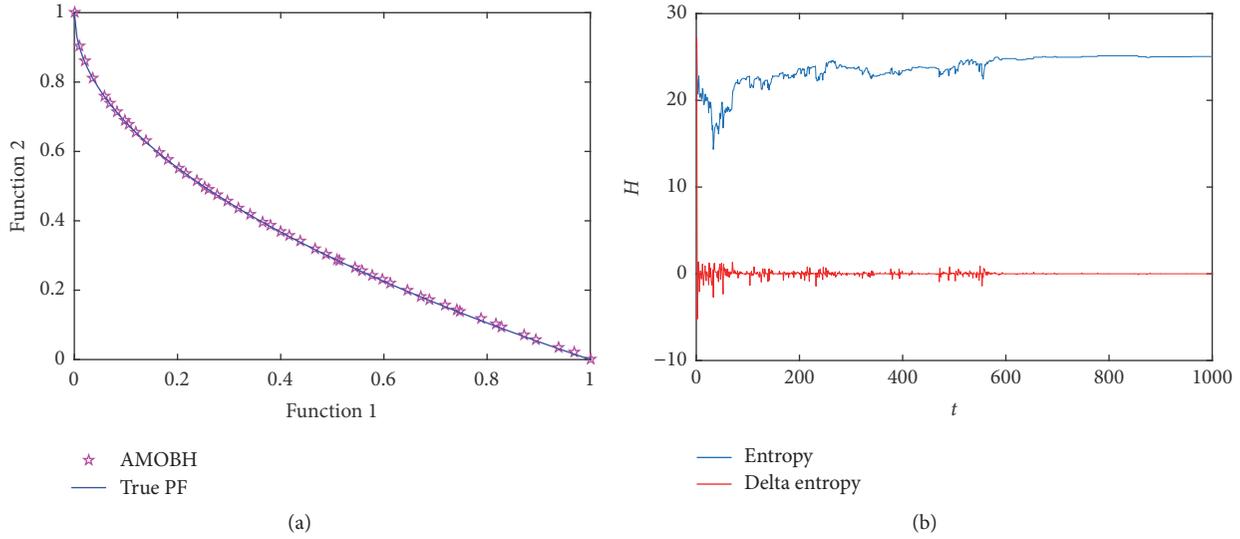


FIGURE 4: Simulation results of ZDT4. (a) The final approximate Pareto front of ZDT4. (b) Entropy during the AMOBH evolution for ZDT4.

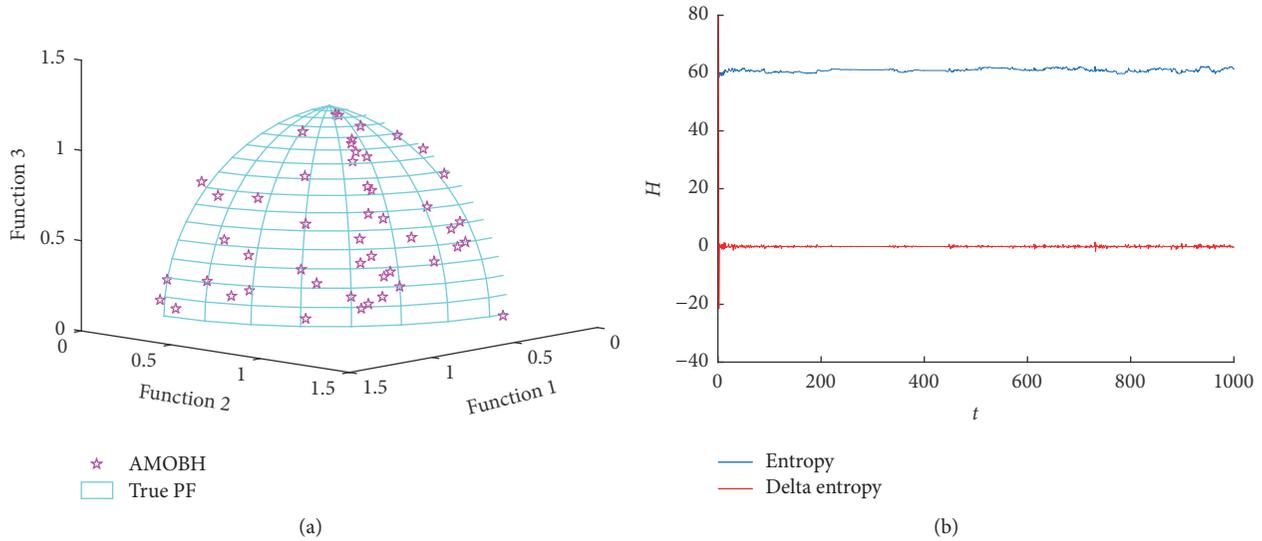


FIGURE 5: Simulation results of DLTZ2. (a) The final approximate Pareto front of DLTZ2. (b) Entropy during the AMOBH evolution for DLTZ2.

in objectives of DLTZ4. And  $a = 100$ . Therefore, the larger value of  $\alpha$  makes the feature of Pareto front nonuniform. DLTZ4 is used to further test the proposed algorithm's ability to maintain a good distribution of solutions. Figure 6(a) illustrates the true Pareto front and approximate Pareto front of DLTZ4 in one of the experiments. We can see that the true Pareto front of DLTZ4 is a surface same as DLTZ2. What is more, the proposed algorithm also converges to an approximate Pareto front which is very close to the true Pareto front and has a good distribution. Figure 6(b) illustrates the entropy evolution during the optimization of DLTZ4. We can see the difference of entropy curve between DLTZ2 and DLTZ4. The initial period of DLTZ4 is an ascending curve and it takes much longer time to stabilize than DLTZ2. This is because of the nonuniform Pareto front of DLTZ4. However, it takes only 100 iterations to

converge. These results confirm that the proposed algorithm has a quite high convergence rate and an ability to obtain good distribution solutions in some concave and nonuniform problems.

**5.6. Results of DLTZ5 Optimization.** The next optimization function is DLTZ5. DLTZ5 is used to test the proposed algorithm's ability to converge to a degenerated curve. Figure 7(a) illustrates the true and approximate Pareto front of DLTZ5 in one of the experiments. The true Pareto front of DLTZ5 is a part of a circle. Figure 7(b) shows entropy evolution during the optimization of DLTZ5. It is seen that the entropy curve is very similar to DLTZ2, which means that the proposed algorithm has a good convergence rate and it can also be proved by the good convergent and uniform approximate Pareto front shown in Figure 7(a).

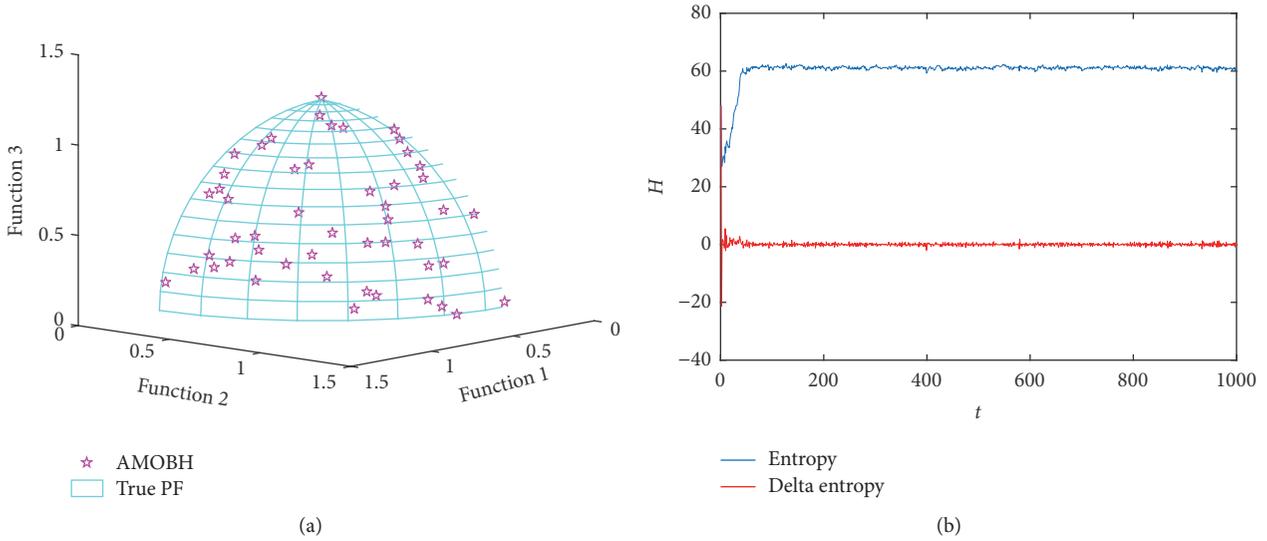


FIGURE 6: Simulation results of DLTZ4. (a) The final approximate Pareto front of DLTZ4. (b) Entropy during the AMOBH evolution for DLTZ4.

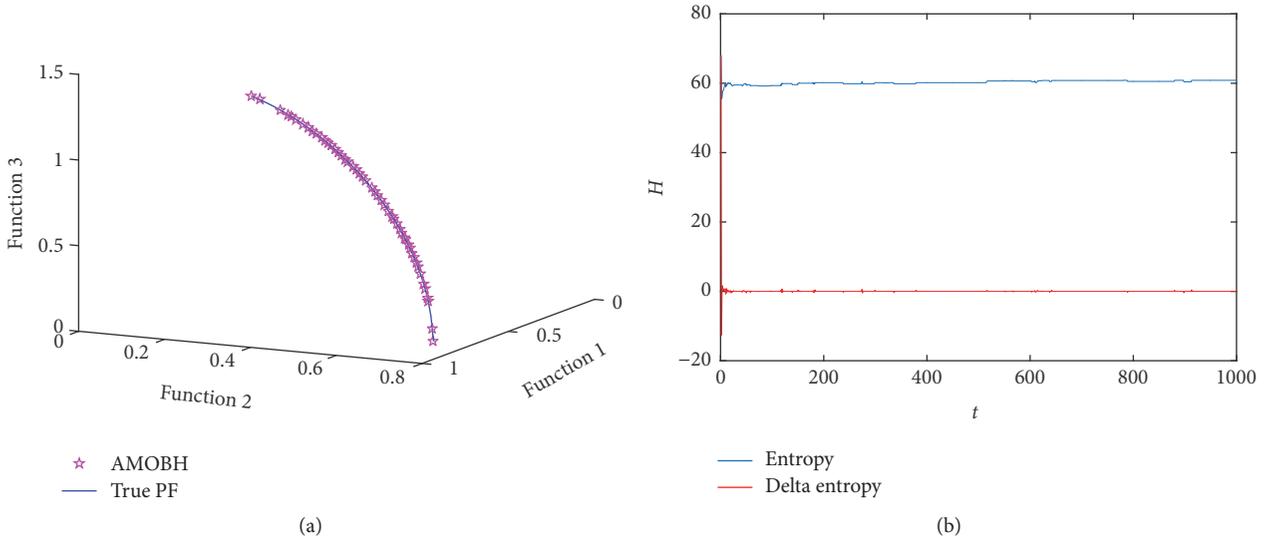


FIGURE 7: Simulation results of DLTZ5. (a) The final approximate Pareto front of DLTZ5. (b) Entropy during the AMOBH evolution for DLTZ5.

**5.7. Results of DLTZ7 Optimization.** The final optimization function that we choose for three-objective optimization is DLTZ7; it has 4 disconnected Pareto front regions in search space. We use it to have a look at the performance of the proposed algorithm to obtain a subpopulation in different Pareto optimal regions. Figure 8(a) illustrates the approximate Pareto front and true Pareto front of DLTZ7 in one of the experiments. As can be seen from Figure 8(a), the proposed algorithm gets an approximate Pareto front which covers the whole Pareto optimal regions and converges very close to the true Pareto front. Figure 8(b) illustrates the entropy evolution during the optimization of DLTZ7. In the first 50 iterations, there is an obvious ascending phase, which means the diversity of the population has acute changes. This is because some new solutions replace some old solutions which

have large cell density in the archive and make the entropy curve violently change. This also means that the algorithm frequently jumps out of one Pareto optimal region to another and keeps the diversity of population. And then, the entropy curve stabilizes in a certain range. As can be seen from these figures, the proposed algorithm has an ability to maintain the subpopulation in different Pareto optimal regions. It is also seen that the proposed algorithm can converge to an approximate Pareto front which is quite close to the true Pareto front, and it was further proved that AMOBH has a good scalability.

**5.8. Comparative Experiment.** In this section, we choose four well-known multiobjective evolutionary algorithms—SPEA-II, PESA-II, NSGA-II, and MOEA/D—for comparison with

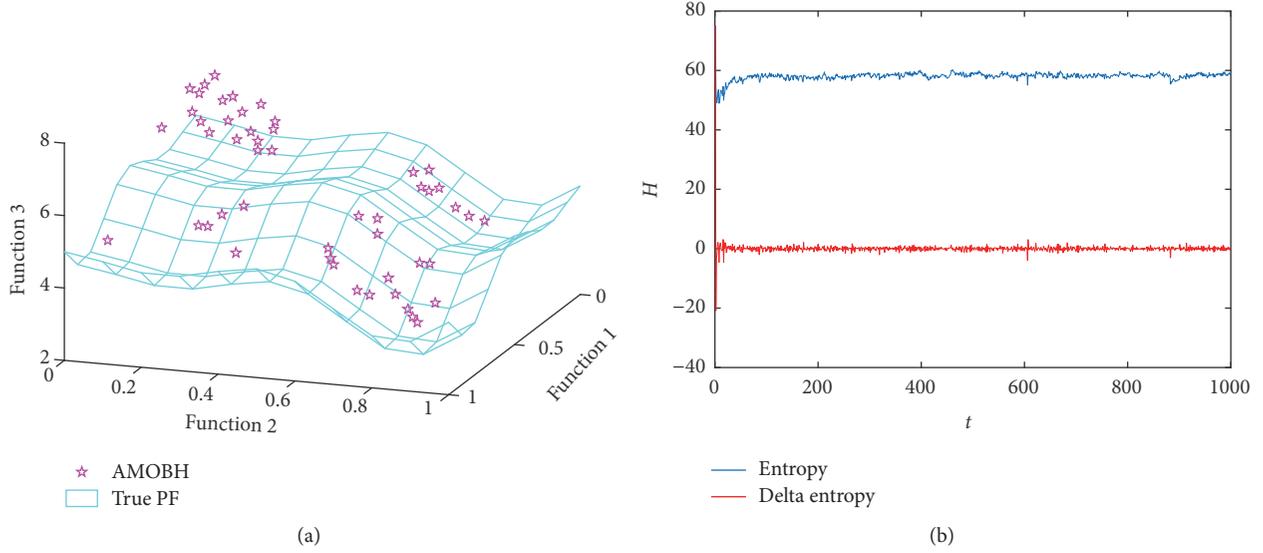


FIGURE 8: Simulation results of DLTZ7. (a) The final approximate Pareto front of DLTZ7. (b) Entropy during the AMOBH evolution for DLTZ7.

AMOBH. Before analyzing the results, we first describe the test metrics used for the experiments. In multiobjective optimization, there are mainly two goals to achieve regarding the obtained approximate Pareto optimal solution set [41]. The algorithm should converge as close to the true Pareto optimal front as possible and it should maintain as diverse an approximate Pareto optimal solution set as possible. Hence, the first test metric we choose is inverted generation distance (IGD) referred to in [20], which can evaluate convergence and diversity at the same time. Smaller IGD value means better diversity and convergence. The formula of IGD is shown as follows:

$$\text{IGD} = \frac{1}{|S_N|} \sum_{i=1}^{|S_N|} \text{Dist}_i; \quad (23)$$

$$\text{Dist}_i = \min_{j=1}^{|A_N|} \sqrt{\sum_{m=1}^M \left( \frac{f_m(S_i) - f_m(A_j)}{f_m^{\max} - f_m^{\min}} \right)^2},$$

where  $|S_N|$  is the sample number of the real Pareto optimal solutions,  $\text{Dist}_i$  is the minimum normalized Euclidean distance,  $|A_N|$  is the number of the approximate Pareto optimal solutions,  $f_m^{\max}$  and  $f_m^{\min}$  are the maximum and minimum of objective  $m$  in the real Pareto optimal solution set, respectively,  $S_i$  is the sample  $i$  of the real Pareto optimal solution set, and  $A_j$  is the solution  $j$  of the approximate Pareto optimal solution set. The testing results and  $|S_N|$  of each test function are shown in Table 2.

To further verify the uniformity of the results, the second test metric we choose is spacing [21–23]; the formula of spacing is as follows:

$$S_p \cong \sqrt{\frac{1}{A_N - 1} \sum_{i=1}^{A_N} (\bar{d} - d_i)^2}; \quad (24)$$

$$\bar{d} = \frac{\sum_{i=1}^{A_N} d_i}{A_N},$$

where  $A_N$  is the number of the approximate Pareto solutions and  $d_i$  is the Euclidean distance between the  $i$ th solution and its nearest neighbor in approximate Pareto solution set. Smaller  $S_p$  means better uniformity of the approximate Pareto solutions.

And to compare the time complexity of the proposed algorithm, we record the CPU runtime of all the comparative experiments and give the average CPU runtime in Table 2.

For the sake of fairness, all algorithms use the same max iteration  $T = 1000$  and the same population size  $N = 50$ . For AMOBH, SPEA-II, PESA-II, and MOEA/D, the archive's size  $K = 50$ . In Tables 2 and 3, the Best, Worst, Mean, and STD represent the best, worst, mean, and standard deviation of IGD metric or spacing metric using the same algorithm which runs 25 times independently on the same test problem, respectively. And the boldface data are the best among all the data; the AT is the average CPU runtime.

In Table 2, we can see that the proposed algorithm achieves the best mean and STD values of IGD metric on the performance of all 7 problems. That is to say, the proposed algorithm has a good accuracy and stability on the performance of these problems. There is no significant difference between the proposed algorithm and SPEA-II on the performance of ZDT1 and ZDT3, which also can be seen in Figures 9 and 10. For the two objectives and multimodal problem, ZDT4, although NSGA-II gets the best STD value of IGD metric, there is no significant difference between NSGA-II and the proposed algorithm. However, the mean, best, and worst values of IGD metric on ZDT4 obtained by NSGA-II are significantly inferior to the proposed algorithm. Hence, the rank of performance on ZDT4 is AMOBH  $\gg$  MOEA/D  $>$  SPEA-II  $>$  NAGA-II  $>$  PESA-II, and this is also evident from Figure 11. For the performance on nonuniform problem DLTZ4, the proposed algorithm gets the best balance of convergence and uniformity as Figures 12(b), 12(f), 12(j), 12(n), and 12(r) show. And for the performance on the disconnected multimodal problem DLTZ7, the proposed algorithm

TABLE 2: Experimental results in terms of IGD and AT of five algorithms on seven test instances.

Test instance	SPEA-II	PESA-II	NSGA-II	MOEA/D	AMOBH	$S_N$
<b>ZDT1</b>						
Mean	2.53E-03	4.80E-03	4.85E-02	8.22E-03	<b>2.51E-03</b>	
Best	1.80E-03	1.80E-03	4.24E-02	<b>1.16E-03</b>	1.73E-03	
Worst	<b>3.50E-03</b>	1.22E-02	6.20E-02	4.63E-02	3.60E-03	1001
STD	4.88E-04	2.43E-03	4.56E-03	1.14E-02	<b>4.71E-04</b>	
AT(s)	213	148	202	211	<b>33</b>	
<b>ZDT3</b>						
Mean	6.72E-03	7.20E-03	7.53E-03	7.44E-03	<b>6.48E-03</b>	
Best	6.20E-03	6.60E-03	7.20E-03	6.50E-03	<b>6.10E-03</b>	
Worst	7.50E-03	7.70E-03	8.50E-03	1.02E-02	<b>6.77E-03</b>	1001
STD	3.40E-04	3.62E-04	2.49E-04	9.01E-04	<b>1.48E-04</b>	
AT(s)	215	149	188	209	<b>34</b>	
<b>ZDT4</b>						
Mean	1.60E-02	2.17E-02	2.00E-02	1.41E-02	<b>3.09E-03</b>	
Best	1.15E-02	8.60E-03	1.42E-02	8.50E-03	<b>4.39E-04</b>	
Worst	2.39E-02	3.43E-02	2.37E-02	1.79E-02	<b>8.80E-03</b>	501
STD	2.34E-03	6.29E-03	<b>1.92E-03</b>	2.09E-03	2.61E-03	
AT(s)	212	172	199	214	<b>32</b>	
<b>DLTZ2</b>						
Mean	4.28E-03	4.75E-03	6.33E-03	4.09E-03	<b>3.00E-03</b>	
Best	3.40E-03	3.73E-03	3.90E-03	3.50E-03	<b>2.70E-03</b>	
Worst	5.20E-03	6.20E-03	2.18E-02	4.80E-03	<b>3.30E-03</b>	1126
STD	4.56E-04	6.45E-04	3.58E-03	4.20E-04	<b>1.27E-04</b>	
AT(s)	224	265	202	222	<b>38</b>	
<b>DLTZ4</b>						
Mean	7.84E-03	5.20E-03	4.02E-02	1.75E-02	<b>3.12E-03</b>	
Best	3.60E-03	3.60E-03	3.11E-02	3.60E-03	<b>2.80E-03</b>	
Worst	3.31E-02	7.90E-03	9.46E-02	3.88E-02	<b>3.50E-03</b>	1126
STD	8.41E-03	9.66E-04	1.44E-02	1.19E-02	<b>1.79E-04</b>	
AT(s)	217	275	201	220	<b>40</b>	
<b>DLTZ5</b>						
Mean	1.59E-03	1.48E-03	1.33E-03	1.70E-03	<b>4.65E-04</b>	
Best	4.95E-04	9.13E-04	<b>2.21E-04</b>	6.81E-04	3.34E-04	
Worst	1.70E-02	2.50E-03	4.60E-03	2.62E-03	<b>6.04E-04</b>	1001
STD	3.23E-03	4.03E-04	1.17E-03	5.25E-04	<b>7.20E-05</b>	
AT(s)	214	243	190	216	<b>34</b>	
<b>DLTZ7</b>						
Mean	4.39E-03	4.47E-03	3.89E-02	1.92E-02	<b>4.30E-03</b>	
Best	<b>3.10E-03</b>	3.30E-03	3.14E-02	9.40E-03	3.70E-03	
Worst	1.08E-02	6.70E-03	4.57E-02	2.62E-02	<b>4.70E-03</b>	2601
STD	1.87E-03	1.06E-03	4.06E-03	4.68E-03	<b>2.79E-04</b>	
AT(s)	216	252	197	166	<b>37</b>	

can converge to all 4 Pareto regions and has the best mean and STD values of IGD metric, which means the proposed algorithm can get a good balance of diversity and convergence on DLTZ7 although some solutions cannot converge to the real Pareto front, which also can be seen from Figure 12. According to previous analysis, it can be concluded that the proposed algorithm outperforms the other four algorithms on IGD metric, especially on the performance of ZDT4,

DLTZ4, and DLTZ5. Comparison results show that the proposed algorithm can get a good balance of diversification and intensification in dealing with the problems of more objectives, nonuniformity, and multimodality.

Table 3 shows the results in terms of spacing metric of five algorithms. For the performance of ZDT1 and ZDT3, although the values of spacing metric obtained by the proposed algorithm are not the best, there is no significant

TABLE 3: Experimental results in terms of spacing of five algorithms on seven test instances.

Test instance	SPEA-II	PESA-II	NSGA-II	MOEA/D	AMOBH
<b>ZDT1</b>					
Mean	4.48E - 02	<b>4.02E - 02</b>	9.78E - 01	1.98E - 01	5.38E - 02
Best	<b>2.74E - 02</b>	3.06E - 02	8.05E - 01	3.13E - 02	4.06E - 02
Worst	6.30E - 02	<b>6.14E - 02</b>	1.23E + 00	1.09E + 00	6.98E - 02
STD	9.78E - 03	<b>7.76E - 03</b>	9.44E - 02	2.67E - 01	8.43E - 03
<b>ZDT3</b>					
Mean	<b>1.07E - 02</b>	1.98E - 02	3.67E - 02	4.57E - 02	1.40E - 02
Best	<b>7.60E - 03</b>	1.30E - 02	3.39E - 02	1.39E - 02	9.70E - 03
Worst	<b>1.39E - 02</b>	3.77E - 02	3.89E - 02	1.78E - 01	2.30E - 02
STD	1.58E - 03	6.07E - 03	<b>1.31E - 03</b>	3.19E - 02	2.97E - 03
<b>ZDT4</b>					
Mean	7.44E - 02	6.47E - 02	6.43E - 02	1.10E - 01	<b>2.09E - 02</b>
Best	5.21E - 02	1.45E - 02	5.86E - 02	4.97E - 02	<b>1.16E - 02</b>
Worst	9.93E - 02	1.03E - 01	7.46E - 02	8.47E - 01	<b>4.09E - 02</b>
STD	9.28E - 03	2.29E - 02	<b>3.61E - 03</b>	1.54E - 01	9.27E - 03
<b>DLTZ2</b>					
Mean	1.65E - 01	1.44E - 01	1.91E - 01	1.09E - 01	<b>4.50E - 02</b>
Best	8.06E - 02	5.83E - 02	6.04E - 02	2.93E - 02	<b>2.23E - 02</b>
Worst	2.76E - 01	3.33E - 01	6.99E - 01	3.40E - 01	<b>7.68E - 02</b>
STD	5.27E - 02	6.80E - 02	1.46E - 01	7.54E - 02	<b>1.33E - 02</b>
<b>DLTZ4</b>					
Mean	2.83E - 01	<b>5.82E - 02</b>	1.28E + 00	5.19E - 01	5.86E - 02
Best	<b>1.13E - 02</b>	1.96E - 02	1.17E + 00	2.82E - 02	2.54E - 02
Worst	1.18E + 00	2.21E - 01	2.54E + 00	1.25E + 00	<b>8.30E - 02</b>
STD	3.49E - 01	5.29E - 02	2.83E - 01	5.52E - 01	<b>1.76E - 02</b>
<b>DLTZ5</b>					
Mean	4.02E - 02	5.74E - 02	8.58E - 02	5.09E - 02	<b>2.94E - 02</b>
Best	<b>1.00E - 02</b>	1.80E - 02	1.01E - 02	1.36E - 02	1.09E - 02
Worst	1.30E - 01	1.85E - 01	3.06E - 01	1.77E - 01	<b>3.93E - 02</b>
STD	3.54E - 02	3.98E - 02	8.27E - 02	3.94E - 02	<b>8.70E - 03</b>
<b>DLTZ7</b>					
Mean	1.76E - 01	<b>1.65E - 01</b>	1.59E + 00	9.18E - 01	2.13E - 01
Best	1.22E - 01	<b>1.20E - 01</b>	1.32E + 00	4.55E - 01	1.65E - 01
Worst	5.12E - 01	<b>2.17E - 01</b>	1.84E + 00	1.20E + 00	2.46E - 01
STD	9.74E - 02	2.37E - 02	1.45E - 01	1.89E - 01	<b>1.93E - 02</b>

difference between the proposed algorithm and the best one. It can be seen that PESA-II gets the best mean and STD values of spacing metric on ZDT1. SPEA-II gets the best mean value of spacing metric on ZDT3. Although NSGA-II gets the best STD value of spacing metric on ZDT3, there is no significant difference between NSGA-II and SPEA-II. However, the mean value of spacing metric on ZDT3 obtained by NSGA-II is significantly inferior to the SPEA-II. Hence, SPEA-II gets the best performance on ZDT3 in spacing metric. NSGA-II gets the best STD value of spacing metric on the performance of ZDT4 and the proposed algorithm is second to NSGA-II. However, the mean, best, and worst values of spacing metric on ZDT4 obtained by NSGA-II are significantly inferior to the proposed algorithm. That is to say, the proposed algorithm performs better on ZDT4 than NSGA-II in spacing metric. For the performance of DLTZ4, PESA-II

gets the best mean value of spacing metric, but there is no significant difference between PESA-II and the proposed algorithm in the mean value of spacing metric, whereas the proposed algorithm is significantly superior to PESA-II in the STD value of spacing metric. That is to say, the proposed algorithm is more stable than PESA-II on the performance of DLTZ4 in spacing metric. The proposed algorithm gets the best mean and STD values of spacing metric on DLTZ2 and DLTZ5. Hence, the proposed algorithm is significantly superior to the other four algorithms on the performance of DLTZ2 and DLTZ5 which also can be seen in Figure 12. For the performance of DLTZ7, the rank of performance in mean spacing metric is PESA-II > SPEA-II > AMOBH >> MOEA/D > NAGA-II and the rank of performance in STD spacing metric is AMOBH > PESA-II > SPEA-II >> NAGA-II > MOEA/D. However, the difference between PESA-II

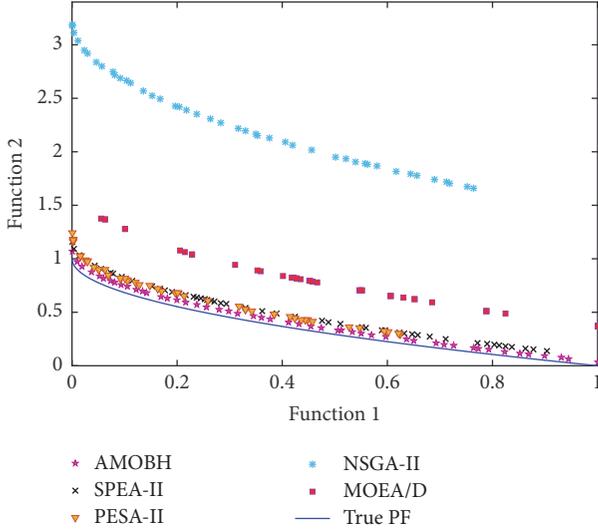


FIGURE 9: The approximate Pareto front of ZDT1 obtained by AMOBH, SPEA-II, PESA-II, NSGA-II, and MOEA/D, respectively, in one of the experiments.

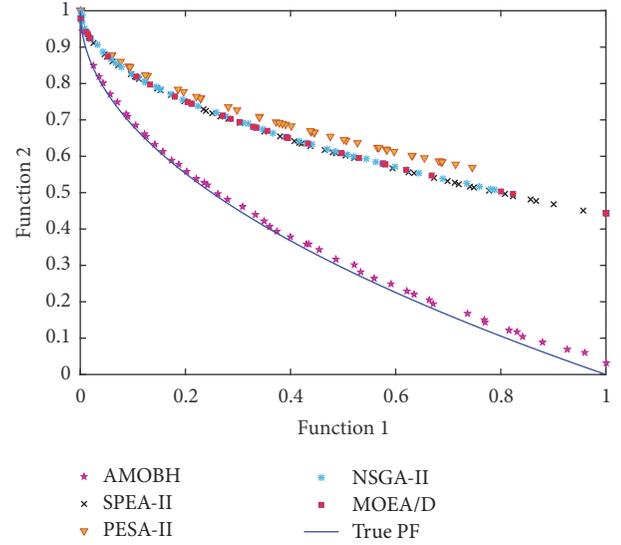


FIGURE 11: The approximate Pareto front of ZDT4 obtained by AMOBH, SPEA-II, PESA-II, NSGA-II, and MOEA/D, respectively, in one of the experiments.

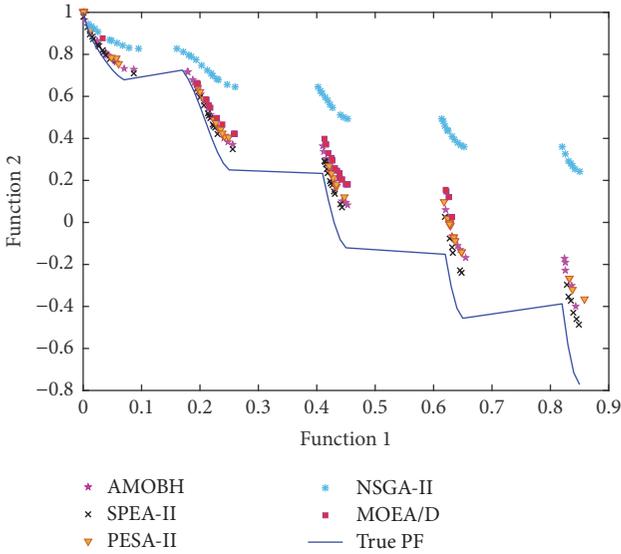


FIGURE 10: The approximate Pareto front of ZDT3 obtained by AMOBH, SPEA-II, PESA-II, NSGA-II, and MOEA/D, respectively, in one of the experiments.

and AMOBH is not so significant. This result has further corroborated the good scalability of the proposed algorithm. At the same time, in terms of the average CPU runtime, the proposed algorithm is significantly superior to the other four algorithms, as Table 2 shows.

## 6. Conclusions

This paper proposes a new multiobjective evolutionary algorithm based on an improved BH algorithm which adopts the mutation into the stars location update. In the proposed algorithm, we use Shannon entropy for adaptive evolution

strategies' control. In order to get a uniform Pareto front distribution, we propose a new individual density concept which is called cell density to calculate the individual density of approximate Pareto optimal solutions in parallel cell coordinate system. In AMOBH, the entropy represents the uniformity and diversity of the approximate Pareto optimal solution set and it can be a measure to capture the convergence rate of the algorithm. Larger entropy means better uniformity and diversity. And the interpretation of cell density is to express the uniformity of approximate Pareto optimal solutions in parallel cell coordinate system. It is used for screening the solution which has max individual density to keep the diversity of the population. The simulations have confirmed its efficiency in screening. The optimization of seven functions was carried out to test the scalability of AMOBH. AMOBH gets a good balance of diversity and convergence in almost all test functions, although its convergence rate will be affected a lot on the performance of some multimodal problems. Through the comparative experiments with another four well-known multiobjective evolutionary algorithms, AMOBH outperforms the other four algorithms in most cases especially in some complex high-dimensional problems, which has further proved that AMOBH has a good scalability. And in terms of the time complexity, AMOBH is also much better than the other four algorithms. Hence, AMOBH is a new powerful multiobjective evolutionary algorithm.

To sum up, the BH algorithm can be used in multiobjective optimization problems like PSO algorithm and GA. Compared with some traditional multiobjective evolutionary algorithms, AMOBH has the advantages of solving the problems of higher dimensions and more objectives. And its time complexity is also quite good. In the future, we will be devoted to testing AMOBH in more complex multiobjective optimization problems like the many-objective optimization

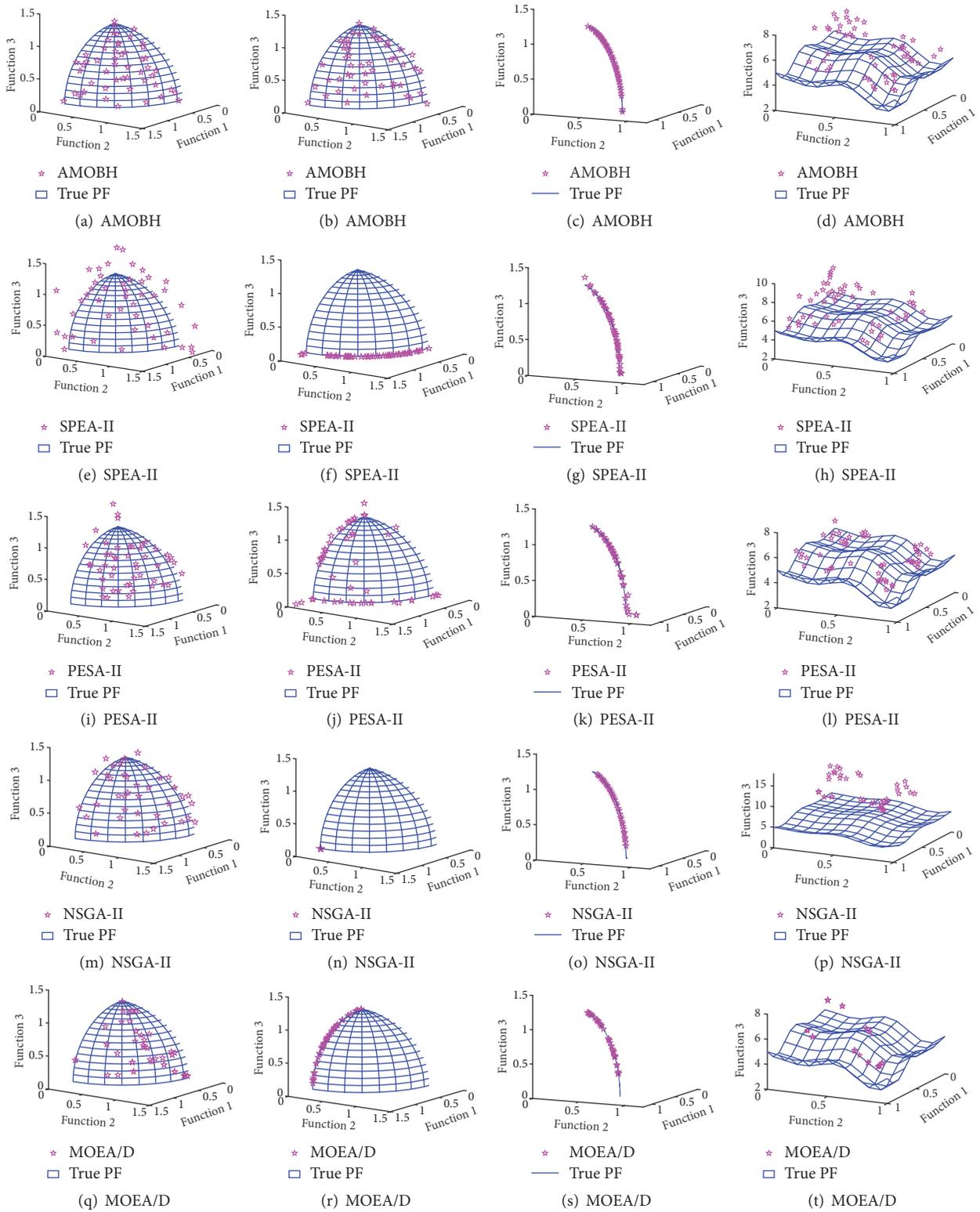


FIGURE 12: The approximate Pareto front of DLTZ2, DLTZ4, DLTZ5, and DLTZ7. (a)–(d). The approximate Pareto front obtained by AMOBH. (e)–(h). The approximate Pareto front obtained by SPEA-II. (i)–(l). The approximate Pareto front obtained by PESA-II. (m)–(p). The approximate Pareto front obtained by NSGA-II. (q)–(t). The approximate Pareto front obtained by MOEA/D.

problems, comparing it with more multiobjective evolutionary algorithms and improving its convergence rate in some multimodal problems. What is more, we are considering to apply it to a DC motor parameters optimization problem.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

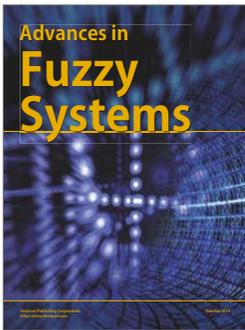
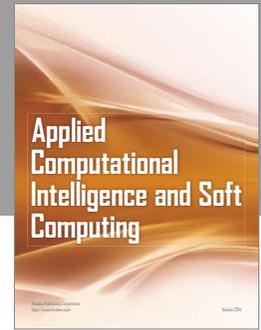
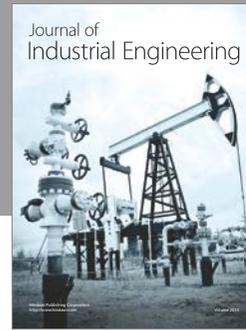
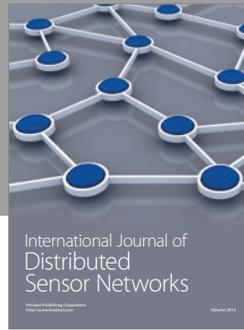
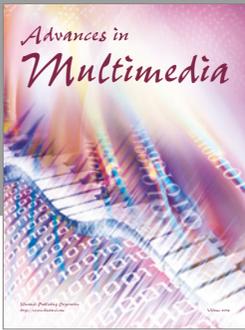
## Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China with Grant no. 11401110; the Open Research Fund of Research Center for Advanced Control of Complex Systems and Intelligent Geoscience Instrument, China University of Geosciences (Wuhan), with Grants nos. AU2015CJ018 and AU2015CJ008; the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan), with Grants nos. CUGL120238 and CUG160833; and the Natural Science Foundation Project of Hubei Province with Grants nos. 2016CFB481 and 2014CFB903.

## References

- [1] D. Gong, Z. Miao, and J. Sun, "A memetic algorithm for multi-objective optimization problems with interval parameters," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation, CEC 2016*, pp. 1674–1681, July 2016.
- [2] D. Gong, F. Sun, J. Sun, and X. Sun, "Set-based many-objective optimization guided by a preferred region," *Neurocomputing*, vol. 228, pp. 241–255, 2017.
- [3] S. Hoffenson, A. Dagman, and R. Söderberg, "A multi-objective tolerance optimization approach for economic, ecological, and social sustainability," in *Re-Engineering Manufacturing for Sustainability*, pp. 729–734, Springer, 2013.
- [4] J. Sanchis, M. A. Martnez, X. Blasco, and G. Reynoso-Meza, "Modelling preferences in multi-objective engineering design," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 8, pp. 1255–1264, 2010.
- [5] Y. Zhang, D.-W. Gong, and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, 2013.
- [6] N. S. Wasley, P. K. Lewis, C. A. Mattson, and H. J. Ottosson, "Experimenting with concepts from modular product design and multi-objective optimization to benefit people living in poverty," *Development Engineering*, vol. 2, pp. 29–37, 2016.
- [7] H. Qiu and H. Duan, "Multi-objective pigeon-inspired optimization for brushless direct current motor parameter design," *Science China Technological Sciences*, vol. 58, no. 11, pp. 1915–1923, 2015.
- [8] P. S. Bharti, S. Maheshwari, and C. Sharma, "Multi-objective optimization of die-sinking electric discharge machining," *Applied Mechanics and Materials*, vol. 110–116, pp. 1817–1824, 2012.
- [9] Ma. Zhou, L.-b. Zhang, C.-g. Zhou, M. Ma, and X.-h. Liu, "Solutions of multi-objective optimization problems based on particle swarm optimization," *Journal of Computer Research and Development*, vol. 7, article 7, 2004.
- [10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [11] E. Zitzler, M. Laumanns, L. Thiele et al., *SPEA-II: Improving the Strength Pareto Evolutionary Algorithm*, 2001.
- [12] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2689–2702, 2017.
- [13] J. Zhang, K. Liu, Y. Tan, and X. He, "Random black hole particle swarm optimization and its application," in *Proceedings of the 2008 IEEE International Conference Neural Networks and Signal Processing, ICNNSP*, pp. 359–365, Nanjing, China, June 2008.
- [14] W. Hu, G. G. Yen, and X. Zhang, "Multiobjective particle swarm optimization based on Pareto entropy," *Journal of Software*, vol. 25, no. 5, pp. 1025–1050, 2014.
- [15] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [16] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.
- [17] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, article 173, 2006.
- [18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, vol. 1, pp. 825–830, May 2002.
- [19] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: region-based selection in evolutionary multiobjective optimization," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pp. 283–290, Morgan Kaufmann Publishers, 2001.
- [20] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [21] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," *Cellular Immunology*, vol. 37, pp. 1–13, 1995.
- [22] D. A. V. Veldhuizen and G. B. Lamont, "On measuring multi-objective evolutionary algorithm performance," in *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 204–211, 2002.
- [23] Y. DongDong, J. LiCheng, G. MaoGuo, and Y. Hang, "Clone selection algorithm to solve preference multi-objective optimization," *Journal of Software*, vol. 21, no. 1, pp. 14–33, 2010.
- [24] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st international Conference on Genetic Algorithms*, pp. 93–100, L. Erlbaum Associates Inc., 1985.
- [25] L. Tang and X. Wang, "A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 20–45, 2013.
- [26] D. Gong, G. Wang, X. Sun, and Y. Han, "A set-based genetic algorithm for solving the many-objective optimization problem," *Soft Computing*, vol. 19, no. 6, pp. 1477–1495, 2015.
- [27] B. Niu, H. Wang, J. Wang, and L. Tan, "Multi-objective bacterial foraging optimization," *Neurocomputing*, vol. 116, pp. 336–345, 2013.

- [28] C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 1051–1056, May 2002.
- [29] E. J. S. Pires, J. A. T. Machado, and P. B. de Moura Oliveira, "Entropy diversity in multi-objective particle swarm optimization," *Entropy*, vol. 15, no. 12, pp. 5475–5491, 2013.
- [30] C. E. Shannon, W. Weaver, and N. Wiener, *The Mathematical Theory of Communication*, The University of Illinois Press, Urbana, Ill, USA, 1949.
- [31] S. H. Ling, H. H. C. Iu, F. H. F. Leung, and K. Y. Chan, "Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3447–3460, 2008.
- [32] S. Ganguly, N. C. Sahoo, and D. Das, "Multi-objective particle swarm optimization based on fuzzy-Pareto-dominance for possibilistic planning of electrical distribution systems incorporating distributed generation," *Fuzzy Sets and Systems*, vol. 213, pp. 47–73, 2013.
- [33] J. Bader, D. Brockhoff, S. Welten, and E. Zitzler, "On using populations of sets in multiobjective optimization," *EMO*, vol. 5467, pp. 140–154, 2009.
- [34] I. C. García, C. A. C. Coello, and A. Arias-Montaña, "MOPSOhv: A new hypervolume-based multi-objective particle swarm optimizer," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 266–273, July 2014.
- [35] W. Liu, C. Xie, J. Wen, J. Wang, and W. Wang, "Optimization of transmission network maintenance scheduling based on niche multi-objective particle swarm algorithm," *Zhongguo Dianji Gongcheng Xuebao/Proceedings of the Chinese Society of Electrical Engineering*, vol. 33, no. 4, pp. 141–148, 2013.
- [36] X. Bi and C. Wang, "A niche-elimination operation based NSGA-III algorithm for many-objective optimization," *Applied Intelligence*, pp. 1–24, 2017.
- [37] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
- [38] A. Pan, H. Tian, L. Wang, and Q. Wu, "A decomposition-based unified evolutionary algorithm for many-objective problems using particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2016, Article ID 6761545, 15 pages, 2016.
- [39] A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*, Springer, 2009.
- [40] W. Hu and Z. S. Li, "A simpler and more effective particle swarm optimization algorithm," *Journal of Software*, vol. 18, no. 4, pp. 861–868, 2007.
- [41] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms: An Introduction*, New York, NY, USA, John Wiley & Sons, 2001.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

