

## Research Article

# A Multiobjective Particle Swarm Optimization Algorithm Based on Grid Technique and Multistrategy

Kangge Zou,<sup>1</sup> Yanmin Liu ,<sup>2</sup> Shihua Wang,<sup>1</sup> Nana Li ,<sup>3</sup> and Yaowei Wu<sup>4</sup>

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

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

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

<sup>4</sup>School of Mathematics and Computational Statistics, Wuyi University, Jiangmen 529000, China

Correspondence should be addressed to Yanmin Liu; [yanmin7813@163.com](mailto:yanmin7813@163.com)

Received 24 September 2021; Accepted 12 November 2021; Published 7 December 2021

Academic Editor: Nan-Jing Huang

Copyright © 2021 Kangge Zou 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.

When faced with complex optimization problems with multiple objectives and multiple variables, many multiobjective particle swarm algorithms are prone to premature convergence. To enhance the convergence and diversity of the multiobjective particle swarm algorithm, a multiobjective particle swarm optimization algorithm based on the grid technique and multistrategy (GTSMOPSO) is proposed. The algorithm randomly uses one of two different evaluation index strategies (convergence evaluation index and distribution evaluation index) combined with the grid technique to enhance the diversity and convergence of the population and improve the probability of particles flying to the real Pareto front. A combination of grid technology and a mixed evaluation index strategy is used to maintain the external archive to avoid removing particles with better convergence based only on particle density, which leads to population degradation and affects the particle exploitation ability. At the same time, a variation operation is proposed to avoid rapid degradation of the population, which enhances the particle search capability. The simulation results show that the proposed algorithm has better convergence and distribution than CMOPSO, NSGAI, MOEAD, MOPSOCD, and NMPSO.

## 1. Introduction

Most of today's scientific and engineering problems are characterized by the fact that they usually have multiple conflicting objectives [1], and decision makers need to simultaneously optimize multiple objectives as best as possible within a given range, namely, multiobjective optimization problems (MOPs). The optimization result of such problems is not single, and there exists a Pareto optimal solution set consisting of a set of compromise solutions [2]. The goal of solving such problems is to obtain well-distributed Pareto fronts in the objective space [3–6].

With the expansion of human existence and the widening of the scope of understanding and transforming the world, the complex optimization problems encountered in reality are characterized by complex, multipolar, nonlinear, strongly constrained, and difficult modeling, which cannot

be solved in polynomial time by classical algorithms, such as simplex algorithm and conjugate gradient method, or even cannot be solved effectively with the model. With the development of information technology, swarm intelligence algorithms are widely used, and such algorithms simulate the centralized learning process of a group composed of individuals. Such algorithms can effectively overcome the models that cannot be solved by classical algorithms. As a kind of swarm intelligence algorithm, particle swarm optimization (PSO) algorithm has the characteristics of simple operation and fast convergence and has good solution potential in solving MOPs.

When PSO deals with MOPs, it is called multiobjective particle swarm optimization (MOPSO). PSO needs to solve at least three problems when dealing with MOPs. The first problem is how to preserve a set of noninferior solutions generated by the algorithm at each iteration, especially when

the number of iterations is large, and how to control the number of solutions to ensure that the noninferior solutions obtained by the algorithm can meet the solution quality requirements. The second problem is how to select the learning samples of particles among the many noninferior solutions. Since there is no truly optimal solution in MOPs, the global optimal particle and the individual optimal particle of MOPSO need to be selected by a specific strategy. The third problem is how to maintain the population diversity and prevent the population from falling into local optimal solutions. Due to the faster convergence speed of MOPSO, the resulting set of noninferior solutions is prone to lose diversity rapidly during the algorithm learning process.

The problems of maintaining the size of external archives, selecting learning samples of particles among many noninferior solutions, maintaining the diversity of the population, and preventing the population from falling into local optimal solutions are the core of MOPSO. In [7], MOPSO<sub>hv</sub>, a new hypervolume-based multiobjective particle swarm optimizer, is proposed. The algorithm uses the hypervolume contribution of the archiving solution to select global and individual leaders for each particle in the master cluster and is used to update the noninferior solution of the external archive. To increase the diversity of particles, the algorithm introduces a mutation operator. In [8], R2-MOPSO, a multiobjective particle swarm optimizer based on R2 indicator and decomposition, is proposed. The algorithm uses one strategy of R2 contribution rate to select the global best leader and update the individual best leader. Also, the algorithm uses an elite learning strategy and a Gaussian learning strategy to improve the diversity of particles. In [9], balancing convergence and diversity is a key problem in high-dimensional target spaces, and handling many objective optimization problems with R2 indicator and decomposition-based particle swarm optimizer is proposed to solve this problem. To balance convergence and diversity, a two-level archiving maintenance method based on r2 metrics and a target space decomposition strategy are designed. The algorithm selects the best global leader based on the r2 metric, while the selection of the best individual leader is based on Pareto dominance. Also, the target space decomposition leader selection used feedback from a two-level profile. A new velocity update method improves the exploitation ability of particles. Also, an elite learning strategy and an intelligent Gaussian learning strategy are embedded in R2-MaPSO to improve the ability of particles to jump out of local optimal solutions.

The current improved MOPSO improves the convergence and diversity of MOPSO to a certain extent, but there are still shortcomings. To further improve the performance of MOPSO, this paper proposes a multiobjective particle swarm algorithm based on grid technology and multi-strategy. The main improvements to MOPSO are as follows: (1) for the maintenance of external archives, a grid technique and mixed evaluation index are used to remove noninferior solutions in the external archives to improve the quality of candidate solutions; (2) to enhance the diversity and convergence of the population, one of the two different

evaluation index strategies is randomly used to select the global optimal sample based on the grid technique; and (3) to further increase the population diversity, a variation operation strategy is used to vary the particle positions. The simulation experimental results show that the proposed algorithm has certain advantages over the other five multiobjective particle swarm algorithms.

The rest of this paper is organized as follows. Section 2 introduces the multiobjective optimization problem, related work, and the concept of particle swarm algorithm. Section 3 presents the details of the algorithm in this paper. Section 4 shows the comparison results and analysis of GTMSMOPSO with other 5 multiobjective particle swarm algorithms. Finally, the conclusion of the algorithm in this paper is given in Section 5.

## 2. Background

*2.1. Multiobjective Optimization Problem.* The definition of the general multiobjective optimization problem is described as follows:

$$\begin{aligned} \min F(x) &= (f_1(x), f_2(x), \dots, f_k(x)), \\ \text{s.t. } g_i(x) &\leq 0, \quad i = 1, 2, \dots, m, \\ h_j(x) &= 0, \quad j = 1, 2, \dots, p, \end{aligned} \quad (1)$$

where  $x = (x_1, x_2, \dots, x_n)^T$  is the  $n$ -dimensional decision vector,  $f_i: R^n \rightarrow R, i = 1, 2, \dots, k$  is the objective function, and  $g_i, h_j: R^n \rightarrow R, i = 1, 2, \dots, m, j = 1, 2, \dots, p$  is the inequality constraint and the equality constraint, respectively.

Definition of multiobjective optimization is given as follows.

*Definition 1* (Pareto dominance). A vector  $m = (m_1, m_2, \dots, m_k)$  is dominant  $n = (n_1, n_2, \dots, n_k)$ , if and only if

$$\forall i \in \{1, 2, \dots, k\}, \quad m_i \leq n_i \wedge \exists i \in \{1, 2, \dots, k\}, \quad m_i < n_i. \quad (2)$$

*Definition 2* (Pareto optimal). A solution  $x' \in \Omega$  is a Pareto optimal or a noninferior solution, if and only if

$$\exists x \in \Omega, \quad x < x'. \quad (3)$$

*Definition 3* (Pareto optimal solution set). The set  $S$  including all Pareto optimal solutions is a Pareto optimal solution set, which is defined as

$$S = \left\{ x' \in \Omega \mid \exists x^* \in \Omega: F(x^*) < F(x') \right\}. \quad (4)$$

*Definition 4* (Pareto frontier). The set  $PF$  of all objective function values with the Pareto optimal solution set as the feasible region is called the Pareto frontier and is defined as

$$PF = \{y = (f_1(x), f_2(x), \dots, f_k(x)) | x \in S\}. \quad (5)$$

*2.2. Related Work.* PSO is a kind of population intelligence optimization algorithm with simple operation and fast convergence, which is widely used in single objective optimization problems. It is the simplicity of operation and fast convergence of particle swarm algorithm that has attracted increasing scholars to study it. Since MOPs have no truly optimal solutions, MOPSO generally selects the global optimal samples in the set of noninferior solutions, and because MOPSO has a fast convergence rate in the learning process, it is easy to make the obtained set of noninferior solutions lose diversity rapidly, leading to a fall into local optimal solutions.

In order to solve the above problems, increasing scholars have proposed many improvement methods for MOPSO. Coello et al. [10] proposed a multiobjective particle swarm optimization algorithm based on adaptive grid, which uses adaptive grid technology to maintain external archives, guide particle updates, and implement mutations on the particle and particles flight area. Although the algorithm shows some advantages over traditional multiobjective evolutionary algorithms for MOPs, it has difficulties in solving complex MOPs with multiple local fronts and poor diversity of nondominated solutions. Li et al. [11] proposed grid search-based multipopulation particle swarm optimization algorithm for multimodal multiobjective optimization, which uses a multiple cluster algorithm based on the k-means clustering method to locate more equivalent PSs in the decision space and uses a grid in GSMPSO-MM to explore high-quality solutions in the decision space, but the algorithm still suffers from insufficient convergence.

It is different from the abovementioned MOPSO which determines the search mechanism through the dominance relationship. In [12], a multiobjective particle swarm algorithm with random migration is proposed. The algorithm sets an age threshold. When a particle does not improve its individual position, it will increase its age, and when the particle age exceeds this age threshold, the particles will be reinitialized to increase the diversity of the population and avoid falling into local extremes. However, because decomposition replaces the dominant relationship, the algorithm cannot cover the entire Pareto frontier when solving some complex MOPs. To solve this problem, Han et al. [13] proposed multiobjective particle swarm optimization with adaptive strategy for feature selection, which mainly uses the PBI decomposition method to select the optimal solution and adaptively provides different penalty values for each weight vector and further improves the ability of MOPSO to solve MOPs. Some scholars also proposed some improved MOPSO. In [14], a modified particle swarm optimization for multimodal multiobjective optimization was proposed. The algorithm introduces a community-based dynamic learning strategy to replace the global learning strategy, enhances the population diversity, and introduces a competition mechanism to improve the

performance of the particle swarm algorithm. In [15], a self-organized speciation based multiobjective particle swarm optimizer for multimodal multiobjective problems is proposed, which uses a species formation strategy to establish multiple stable ecological niches to increase the probability of species flying to the true Pareto front. In [16], a simplified multiobjective particle swarm optimization algorithm was proposed. The algorithm uses an adaptive penalty mechanism for the PBI parameters, which can adaptively adjust the penalty value to enhance the selection pressure of the archive and improve the selection pressure of each particle. In [17], multiobjective reservoir operation using particle swarm optimization with adaptive random inertia weights is proposed, which combines the ARIW algorithm with the traditional PSO, while using a triangular probability density function, randomly generates inertia weights, and automatically adjusts the probability distribution function as it evolves. In [18], based on penalty function theory and particle swarm optimization (PSO) algorithm, an improved multiobjective particle swarm optimization algorithm based on archive management is proposed, while multiple swarm coevolution of crowded distance archive management is used to improve the search ability and diversity of the population.

Most of the current improved MOPSO rely on only one strategy to select the learning samples of particles and maintain external archives, without considering the performance of particles at different stages, resulting in insufficient convergence and diversity of the algorithms when solving complex MOPs. In this paper, a multiobjective particle swarm algorithm based on grid technology and multistrategy is proposed, and experimental simulations show that the algorithm effectively improves the performance of MOPSO.

*2.3. Particle Swarm Optimization.* The particle swarm optimization [19] is an optimization algorithm based on an iterative model, proposed by Dr. Eberhart and Dr. Kennedy in 1995, and originated from the study of the behavior of bird flocks foraging. The group searches for the global optimal solution within a range, and each particle has a fitness and speed to adjust its flight direction. During the flight, all particles in the group have a memory function, and each particle continuously learns from its own optimal position and the optimal particle position in the group. The particle velocity and position update formula are as follows:

$$v(t+1) = \omega v(t) + c_1 r_1 (p(t) - x(t)) + c_2 r_2 (g(t) - x(t)), \quad (6)$$

$$x(t+1) = x(t) + v(t+1). \quad (7)$$

Among them,  $\omega$  is the inertia weight, and its size controls the size of the search ability.  $c_1$  and  $c_2$  are the acceleration factors to make the particles have the ability to self-summarize and learn from the outstanding individuals in the group.  $r_1$  and  $r_2$  are random numbers in the range [0,1].  $p$  and  $g$ , respectively, represent the individual optimal solution and the global optimal solution.

### 3. The Proposed GTMSMOPSO Algorithm

The problems of maintaining the size of external archives, selecting learning samples of particles among many noninferior solutions, maintaining the diversity of the population, and preventing the population from falling into local optimal solutions are the core of MOPSO. In this paper, we propose a grid technique and a multistrategy approach in the part of maintenance of external archives and selection of globally optimal particles, to further improve the performance of MOPSO by a variational operation on the positions of particles.

**3.1. Grid Technique.** In MOPSO, each iteration generates a set of noninferior solutions. The grid technique divides the grid area by the information of all noninferior solution function values in the external archive, so that the grid location to which the noninferior solution belongs can be found according to the function value of the noninferior solution. The definition of grid technology is introduced as follows.

**3.1.1. Coordinates of the Grid Boundary.** In the target space, the maximum and minimum values of the noninferior solution of the  $m$ th objective function are  $\min f_m(x)$  and  $\max f_m(x)$ , respectively, and then the upper and lower boundaries of the  $m$ th target grid are given by

$$L_m = \min f_m(x) - \frac{\max f_m(x) - \min f_m(x)}{b}, \quad (8)$$

$$U_m = \max f_m(x) + \frac{\max f_m(x) - \min f_m(x)}{b},$$

where  $b$  is the number of mesh divisions.

**3.1.2. Grid Coordinates ( $G_m$ ).** let  $f_m(x)$  be the function value of particle  $x$  as the  $m$ th dimensional target, then the corresponding grid coordinates are

$$G_m(x) = \frac{f_m(x) - L_m}{d_m}, \quad (9)$$

where  $d_m = (U_m - L_m/b - 1) = ((\max f_m(x) - \min f_m(x)) + 2ep/b - 1)$  and  $ep = (\max f_m(x) - \min f_m(x)/b)$  is the grid width

**3.2. Global Optimal Sample Selection Based on Grid Technique and Multistrategy.** The global optimal particle selection in MOPSO is a key factor affecting the diversity and convergence of the algorithm. To enhance the convergence and diversity of MOPSO, this paper uses two different evaluation indices (convergence evaluation index and distribution evaluation index) to select noninferior solutions in each grid based on the roulette wheel strategy.

**3.2.1. Selection Method of Convergence Evaluation Index and Distribution Evaluation Index.** To improve the convergence of the algorithm, the noninferior solution with the largest

contribution to the Pareto solution set is selected as the global optimal sample in the same grid. In this paper, the inflection point distance of particles is used as the evaluation index of particle convergence.

The inflection point distance (IPD) is to determine an extreme straight line through the two extreme noninferior solutions in the external archive of two-objective functions and then calculate the distance from the particle to this straight line. If there are three or more goal functions, it is necessary to calculate the distance from each solution in the noninferior solution set to the extreme "hyperplane."

The equation of the extreme straight line  $L$  is as follows:

$$Ax + By + C = 0, \quad (10)$$

where  $A$ ,  $B$ , and  $C$  are all real numbers.

Supposing that the coordinate of the noninferior solution  $x$  is  $(x_0, y_0)$ , then the distance from the noninferior solution  $x$  to the straight line  $L$  is

$$\frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}. \quad (11)$$

The hyperplane in  $n$ -dimensional space is determined by the following equation:

$$Ax + b = 0. \quad (12)$$

Among them,  $A$  is an  $n$ -order square matrix,  $x$  is an  $n$ -dimensional column vector, and  $b$  is a real number, representing the distance from the hyperplane to the origin. The distance from each solution in the noninferior solution is set to the extreme "hyperplane":

$$\frac{|Ax + b|}{\|A\|}. \quad (13)$$

The inflection point of the curve is the "most concave" point on the front surface of Pareto, which contributes the most to the Pareto solution set. IPD is an index reflecting the convergence of particles in the same grid. The smaller the IPD index of the particle, the better the convergence of the particle. Taking the dual objective function as an example, as shown in Figure 1, the black noninferior solution has the best convergence, the red noninferior solution is the extreme noninferior solution, and the noninferior solution above the extreme straight line has the worst convergence.

To increase the diversity of the population, the most dispersed noninferior solution in the same grid is selected as the global optimal sample. In this paper, the grid density of particles is used as the evaluation standard of particle distribution.

The average of the sum of Euclidean distances between particle  $x_i$  and all other particles in the same grid is defined as the grid density of particle  $x_i$ , which is defined as follows:

$$gd(x_i) = \frac{1}{S} \sum_{s=1}^S \sum_{m=1}^M \sqrt{(f_m(x_i) - f_m(x_j))^2}, \quad (14)$$

where  $S$  is the number of particles in the same grid and  $M$  is the number of objective functions.

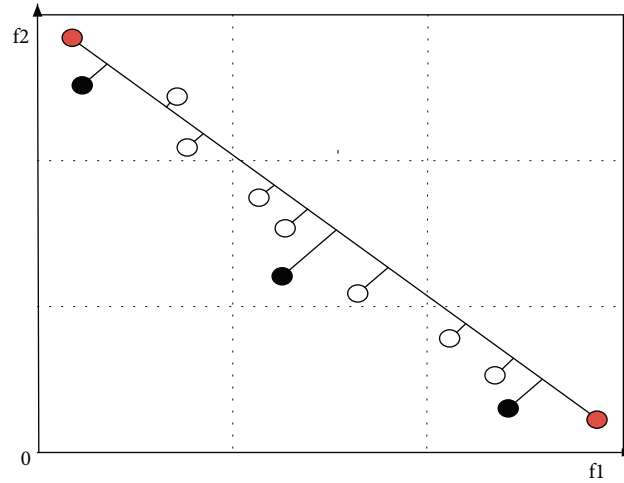


FIGURE 1: Selection of the global optimal sample.

$gd$  is an indicator that reflects the distribution of particles in the same grid. The larger the  $gd$  index of the particles, the more dispersed the particle distribution.

**3.2.2. The Gbest Selection Method of Two Different Evaluation Indicators on Roulette.** To ensure the diversity of the population, the noninferior solutions of each grid are selected as the global optimal sample. Due to the different number of noninferior solutions in the grid, the more noninferior solutions are selected by the roulette strategy for the grid with more noninferior solutions. To further balance the convergence and diversity of the algorithm and enable the population to explore more regions, one of the convergence evaluation index and diversity evaluation index is selected according to the current iteration number of the algorithm to select noninferior solutions in the same grid. Suppose that the noninferior solution with the largest inflection point distance in the  $j$ -th grid is  $\max(IPD_j)$ , and the noninferior solution with the largest grid density in the  $j$ -th grid is  $\max(gd_j)$ . Then, in the  $j$ -th grid, the particles select  $\max(IPD_j)$  or  $\max(gd_j)$  as the gbest probability:

$$gbest_i = \begin{cases} \max(gd_j), & \text{rand} < k \times \frac{\text{iter}}{\text{maxiter}}, \\ \max(IPD_j) & \text{otherwise,} \end{cases} \quad (15)$$

where  $\text{rand}$  is a random number between  $[0,1]$ ,  $\text{iter}$  is the current number of iterations,  $\text{itermax}$  is the maximum number of iterations, and  $k$  is a real number. Through experimental simulation, the best effect is obtained when  $k=1.3$ . From equation (15), it can be seen that the good particle diversity but poor convergence in the early iteration of the algorithm makes the particles have a higher probability of choosing  $\max(IPD)$  as the gbest to guide the flight as a way to improve the particle exploitation. Particles converge better but are less diverse in the late iterations of the algorithm, allowing particles to have a higher probability

of selecting  $\max(gd)$  as the gbest to guide their flight as a way to improve particle exploration.

**3.3. External Archive Maintenance Based on Mixed Evaluation Indicators.** In MOPSO, each iteration of the algorithm will generate a set of noninferior solutions, and these noninferior solutions will be stored in the external archive. As the algorithm runs, the number of noninferior solutions will increase. The external archive needs to be maintained when the maximum size of the external archive is reached. Most MOPSO delete highly crowded particles to maintain external archives. However, although this method guarantees the uniform distribution of noninferior solutions in the external archive, it is possible to delete noninferior solutions with better convergence. Therefore, this paper adopts a hybrid evaluation index of convergence evaluation index and distribution evaluation index to delete noninferior solutions in external archives to improve the quality of candidate solutions.

It is deleted by finding the grid with the largest number of particles. The deletion method is a hybrid evaluation index strategy of the convergence evaluation index and the distribution evaluation index of the particles. The specific formula is as follows:

$$ME(x_j) = IPD(x_j) \times gd(x_j). \quad (16)$$

The larger the  $IPD$  index of the particle, the greater the contribution of the particle to the Pareto solution set, and the better its convergence. From formula (14), it can be seen that the larger the  $gd$  index of the particle, the more dispersed the particle distribution. It can be seen from formula (16) that the particle  $ME$  index can reflect not only the degree of convergence of the particle but also the distribution of the particle. The smaller the particle  $ME$  index, the better the overall performance of the particle. Therefore, when the external archive reaches the maximum limit, the grid with the largest number of particles is found, and the particles with the smaller  $ME$  index in the grid are deleted to improve the quality of the candidate solutions.

**3.4. Mutation Operation.** MOPSO has strong exploration capabilities in the early iterations and can continuously search for new areas. However, MOPSO has a fast convergence effect, which makes the algorithm very likely to search around the local optimal solution in the later iterations, which makes the algorithm converge prematurely. Therefore, to make the algorithm maintain good population diversity in the early iterations and increase population diversity in the later iterations, mutation operations are used to increase the degree of position mutation. The position variation of particles increases linearly with the increase in the number of iterations. At the beginning of the iteration, the diversity of the population is good, so that the variation of the particles is small, so that the diversity of the population remains stable. In the later stage of the iteration, the diversity of the population is insufficient, which makes the particles vary greatly to increase the diversity of the population. In GTMSMOPSO, the update formula of particle velocity is given in equation (6), and the update formula of particle position is as follows:

$$x(t+1) = \frac{e^{-(n/N)}}{1 + e^{-(n/N)}} x(t) + v(t+1), \quad (17)$$

where  $itermax$  is the maximum number of iterations,  $iter$  is the current iteration number, and  $e$  is the natural logarithm.

**3.5. The Main Flow of the Algorithm.** The main flow of GTMSMOPSO is as follows.

*Step 1.* Set the number of populations, the maximum number of external archives, the dimension of particles, the number of grids, and the maximum number of iterations. Initialize the position and velocity randomly, set the initial value of each particle position as the best value of the individual particle, and create an external archive and set it to an empty set.

*Step 2.* Calculate the fitness of each particle and store the noninferior solutions in an external archive.

*Step 3.* Establish grids for the target space, and calculate the noninferior solutions in each grid with the inflection point distance formula and the grid density formula.

*Step 4.* The best sample of an individual is the best between the current position and the best position in the history of the individual. If the rankings are equal, one of them is randomly selected. First use the roulette method to determine the number of noninferior solutions that need to be selected in each grid, and then use equation (15) to determine the global optimal sample for the same grid.

*Step 5.* Use flight formulas (6) and (17) to update the speed and position of each particle.

*Step 6.* If the external archive does not reach saturation, continue to add noninferior solutions. If the external archive reaches a saturated state, find the grid with the largest

number of grid particles, use formula (14) to delete the particles with a smaller  $ME$  index, and then introduce noninferior solutions into the external archive to update the external archive.

*Step 7.* If the current number of iterations is less than the maximum number of iterations, return to the second step; otherwise, output the optimal solution set.

The algorithm flowchart of GTMSMOPSO is presented in Figure 2.

## 4. Experimental Simulation Analysis

**4.1. Performance Evaluation Index.** To evaluate the performance of each algorithm, this paper uses inverse generation distance (IGD) [20] and super volume (HV) [21] to evaluate the algorithm respectively. IGD is a measure of the distance between the true Pareto front and the approximate Pareto front. The lower the IGD value, the better the convergence and diversity of the approximate Pareto frontier obtained by the algorithm, and the closer it is to the true Pareto frontier. Its calculation formula is as follows:

$$IGD(P, Q) = \frac{\sum_{v \in P} d(v, Q)}{|P|}, \quad (18)$$

where  $P$  is the set of solutions uniformly distributed over the true PF and  $|P|$  is the number of individuals in the set of solutions distributed over the true PF.  $Q$  is the set of Pareto optimal solutions obtained by the algorithm, and  $d(v, Q)$  is the minimum Euclidean distance from a single  $v$  in  $P$  to the population  $Q$ .

The HV measures the noninferiority distribution of the algorithm in the space, and a larger HV indicates a better noninferiority distribution in the space.

**4.2. Selection of Parameters.** The parameter settings of GTMSMOPSO are as follows: the external archive size is set to 200, the population size is set to 200, the maximum number of iterations is 2000, the number of grids is 50,  $c_1=c_2=2$ , and  $\omega = 0.4$ . The parameter settings of the other four comparisons of multiobjective particle swarm optimization are consistent with the original literature.

**4.3. Experimental Results and Data Analysis.** To verify the effectiveness of GTMSMOPSO, we selected typical multiobjective test function sets ZDT [22], UF [23], and DTLZ [24], representative 14 multiobjective functions (ZDT1-4, ZDT6, UF2-5, UF8-10, DTLZ1, and DTLZ6), and comparative algorithms including CMOPSO [25], NSGAI [26], MOEAD [27], MOPSOCD [28], and NMPSO [29]. All algorithms, except GTMSMOPSO, run on the platform [30]. The mean (Mean) and standard deviation (Std.) of the IGD metrics and HV metrics for GTMSMOPSO and the five multiobjective intelligence algorithms on the 14 tested functions are given in Tables 1 and 2, respectively. The bolded data in the table represent the best values.

The multiobjective test function sets ZDT (ZDT1-4 and ZDT6) are all biobjective test functions. ZDT1 has a convex

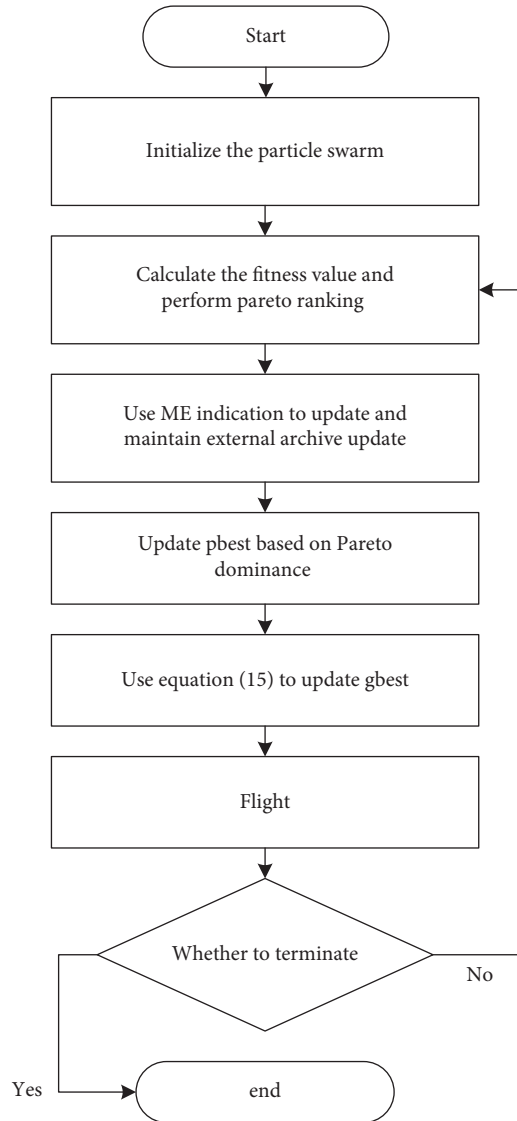


FIGURE 2: Algorithm flowchart of GTMSPSO.

TABLE 1: The IGD value of the algorithm on 14 test functions.

Function	IGD	GTMSMOPSO	CMOPSO	NSGAI	MOEAD	MOPSOCD	NMPSO
ZDT1	Mean	5.65E-03	<b>3.25E-03</b>	3.87E-02	1.89E-01	2.05E-02	3.16E-02
	Std.	6.97E-04	5.13E-04	8.49E-03	7.89E-02	6.41E-02	9.92E-03
ZDT2	Mean	5.99E-03	<b>2.80E-03</b>	1.09E-01	5.66E-01	1.43E-01	6.25E-02
	Std.	9.06E-04	3.78E-04	9.68E-02	7.72E-02	2.24E-01	1.32E-01
ZDT3	Mean	2.05E-01	<b>3.56E-03</b>	3.27E-02	1.73E-01	3.83E-02	9.79E-02
	Std.	6.89E-03	6.42E-04	8.31E-03	6.56E-02	4.76E-02	7.02E-03
ZDT4	Mean	<b>5.90E-03</b>	1.63E+02	3.14E+01	4.77E-01	1.57E+02	1.37E+02
	Std.	6.89E-04	3.42E+01	5.00E+00	1.92E-01	3.07E+01	2.71E+01
ZDT6	Mean	<b>2.20E-03</b>	2.72E-01	2.99E+00	8.37E-02	1.39E+00	2.21E-03
	Std.	1.92E-04	2.65E-01	2.10E-01	2.12E-02	1.80E+00	1.60E-04
UF2	Mean	8.26E-02	<b>6.38E-02</b>	6.63E-02	2.17E-01	1.38E-01	8.19E-02
	Std.	7.73E-03	4.76E-03	6.48E-03	7.33E-02	1.45E-02	7.06E-03
UF3	Mean	<b>3.18E-01</b>	3.97E-01	4.40E-01	3.47E-01	3.65E-01	3.64E-01
	Std.	3.12E-02	2.56E-02	1.82E-02	2.85E-02	5.30E-02	5.59E-02

TABLE 1: Continued.

Function	IGD	GTMSMOPSO	CMOPSO	NSGAI	MOEAD	MOPSOCD	NMPSO
UF4	Mean	<b>5.55E-02</b>	1.09E-01	8.07E-02	1.15E-01	7.82E-02	6.39E-02
	Std.	4.55E-03	1.12E-02	2.33E-03	4.84E-03	7.18E-03	8.78E-03
UF5	Mean	1.93E+00	8.45E-01	<b>8.20E-01</b>	1.36E+00	3.92E+00	1.69E+00
	Std.	1.70E-01	1.75E-01	2.98E-01	2.73E-01	4.67E-01	4.19E-01
UF8	Mean	4.26E-01	5.54E-01	<b>3.12E-01</b>	5.01E-01	7.61E-01	4.48E-01
	Std.	8.11E-02	1.05E-01	4.58E-02	2.53E-01	1.83E-01	1.18E-01
UF9	Mean	<b>4.34E-01</b>	8.45E-01	4.49E-01	5.36E-01	9.05E-01	4.70E-01
	Std.	2.67E-02	1.14E-01	6.02E-02	1.05E-01	1.29E-01	6.12E-02
UF10	Mean	<b>4.88E-01</b>	4.51E+00	1.44E+00	7.04E-01	5.23E+00	1.50E+00
	Std.	1.91E-01	4.79E-01	5.08E-01	9.49E-02	8.63E-01	3.41E+00
DTLZ1	Mean	<b>6.34E-02</b>	7.46E+01	5.38E+00	5.79E+00	4.29E+01	4.25E+01
	Std.	1.86E-02	1.03E+01	1.53E+00	4.08E+00	1.56E+01	6.56E+00
DTLZ6	Mean	2.62E-02	1.47E-01	<b>4.05E-03</b>	1.87E-01	5.76E-02	1.29E-02
	Std.	3.61E-02	3.32E-01	1.46E-03	3.34E-01	1.65E-01	2.18E-03

TABLE 2: The HV value of the algorithm on 14 test functions.

Function	HV	GTMSMOPSO	CMOPSO	NSGAI	MOEAD	MOPSOCD	NMPSO
ZDT1	Mean	7.18E-01	<b>7.20E-01</b>	6.69E-01	5.41E-01	7.00E-01	6.87E-01
	Std.	7.16E-04	6.96E-04	1.38E-02	5.36E-02	7.54E-02	1.18E-02
ZDT2	Mean	4.41E-01	<b>4.45E-01</b>	3.29E-01	9.20E-02	3.30E-01	4.03E-01
	Std.	1.31E-03	6.18E-04	5.30E-02	2.53E-02	1.68E-01	9.11E-02
ZDT3	Mean	<b>6.55E-01</b>	6.00E-01	5.77E-01	5.58E-01	5.82E-01	5.69E-01
	Std.	2.89E-03	1.31E-03	4.85E-03	7.46E-02	3.33E-02	4.15E-03
ZDT4	Mean	<b>7.18E-01</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	1.42E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
ZDT6	Mean	3.84E-01	1.86E-01	0.00E+00	0.00E+00	1.82E-01	<b>3.90E-01</b>
	Std.	1.19E-03	1.49E-01	0.00E+00	0.00E+00	1.51E-01	1.37E-04
UF2	Mean	6.28E-01	<b>6.45E-01</b>	6.37E-01	5.58E-01	5.45E-01	6.22E-01
	Std.	6.92E-03	6.62E-03	5.41E-03	3.05E-02	1.85E-02	7.51E-03
UF3	Mean	<b>6.30E-01</b>	2.62E-01	2.11E-01	2.91E-01	2.66E-01	2.75E-01
	Std.	6.75E-03	2.78E-02	1.19E-02	3.91E-02	4.02E-02	5.43E-02
UF4	Mean	3.35E-01	2.85E-01	3.34E-01	2.85E-01	3.34E-01	<b>3.59E-01</b>
	Std.	2.56E-02	9.46E-03	3.71E-03	5.11E-03	9.82E-03	1.32E-02
UF5	Mean	<b>3.70E-01</b>	1.67E-02	1.06E-02	0.00E+00	0.00E+00	6.91E-05
	Std.	5.95E-03	2.89E-02	2.42E-02	0.00E+00	0.00E+00	3.79E-04
UF8	Mean	<b>3.88E-01</b>	1.21E-02	2.82E-01	1.72E-01	8.58E-03	2.85E-01
	Std.	1.94E-02	1.72E-02	3.36E-02	6.56E-02	1.44E-02	4.65E-02
UF9	Mean	2.84E-01	2.37E-02	2.95E-01	2.80E-01	2.18E-02	<b>3.14E-01</b>
	Std.	3.00E-02	2.75E-02	5.75E-02	6.53E-02	2.29E-02	5.54E-02
UF10	Mean	<b>2.66E-01</b>	0.00E+00	0.00E+00	4.02E-02	0.00E+00	0.00E+00
	Std.	6.06E-02	0.00E+00	0.00E+00	2.66E-02	0.00E+00	0.00E+00
DTLZ1	Mean	<b>7.60E-01</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	3.43E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
DTLZ6	Mean	1.63E-01	1.69E-01	<b>1.99E-01</b>	1.44E-01	1.90E-01	1.98E-01
	Std.	4.01E-02	7.15E-02	3.10E-03	7.38E-02	2.77E-02	6.18E-04

PF and ZDT2 has a nonconvex PF. From Tables 1 and 2, it can be seen that both GTMSMOPSO and CMOPSO have smaller IGD values and larger HV values on ZDT1 and ZDT2 test functions, so they have better overall performance. The performance of NSGAI and MOEAD is slightly worse. ZDT3 has a broken Pareto front and is nonconvex. It can be seen from Tables 1 and 2 that GTMSMOPSO has the

best HV value on the ZDT3 test function, although it has a poor IGD value. ZDT4 has many locally optimal solutions and has convex PFs, and ZDT6 has nonconvex and non-uniformly spaced PFs. GTMSMOPSO showed the best IGD and HV values on the ZDT4 test function. GTMSMOPSO showed the best IGD value on the ZDT6 test function, and its HV value was slightly lower than the HV value of



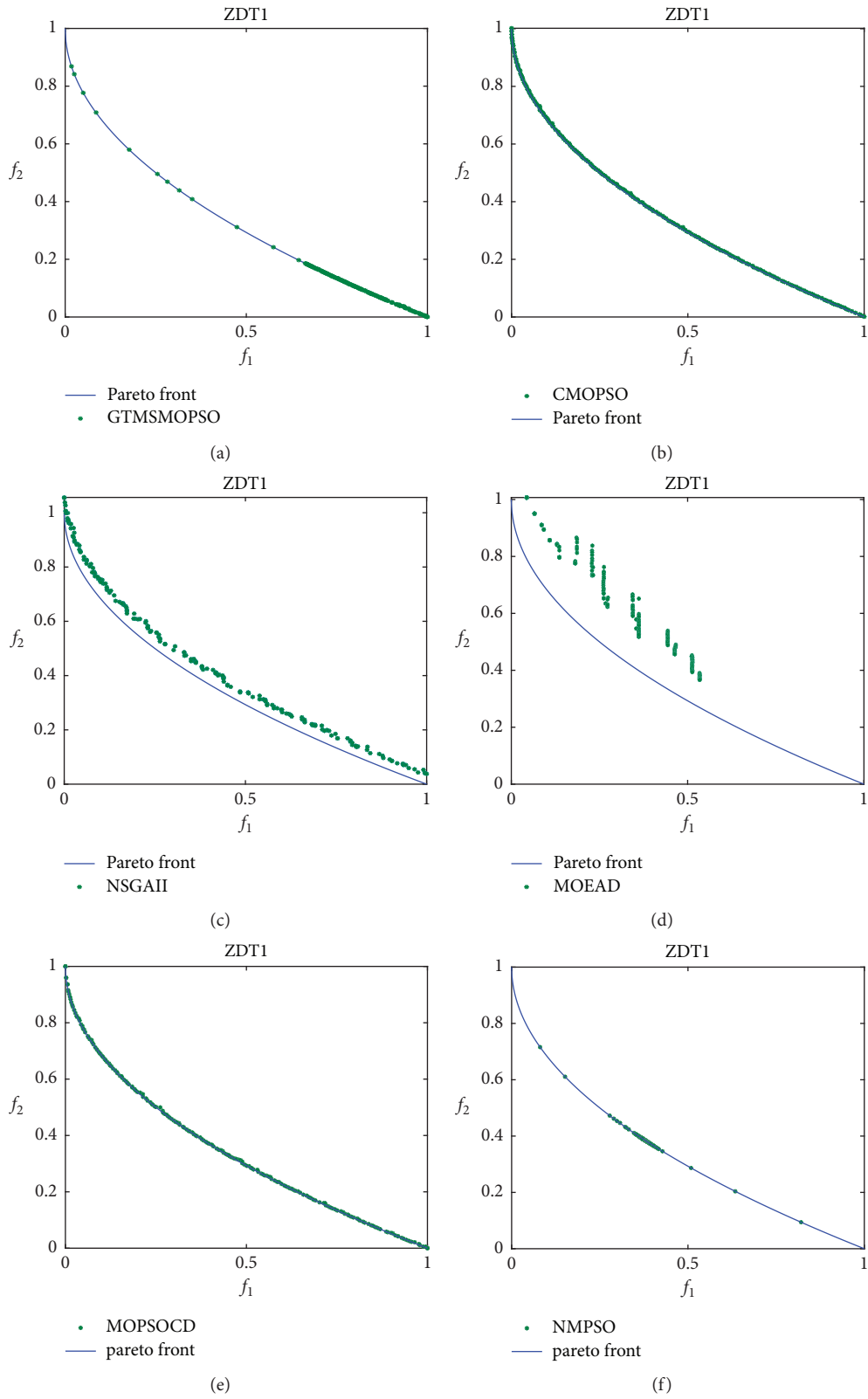


FIGURE 3: Pareto frontier effect of three different algorithms for ZDT1 function.

NMPSO. In summary, it can be seen that GTMSMOPSO has the best performance on the multiobjective test function set ZDT series.

The multiobjective test functions UF2-UF5 are bi-objective test functions and UF8-10 are three-objective test function. As can be seen from Tables 1 and 2, GTMSMOPSO

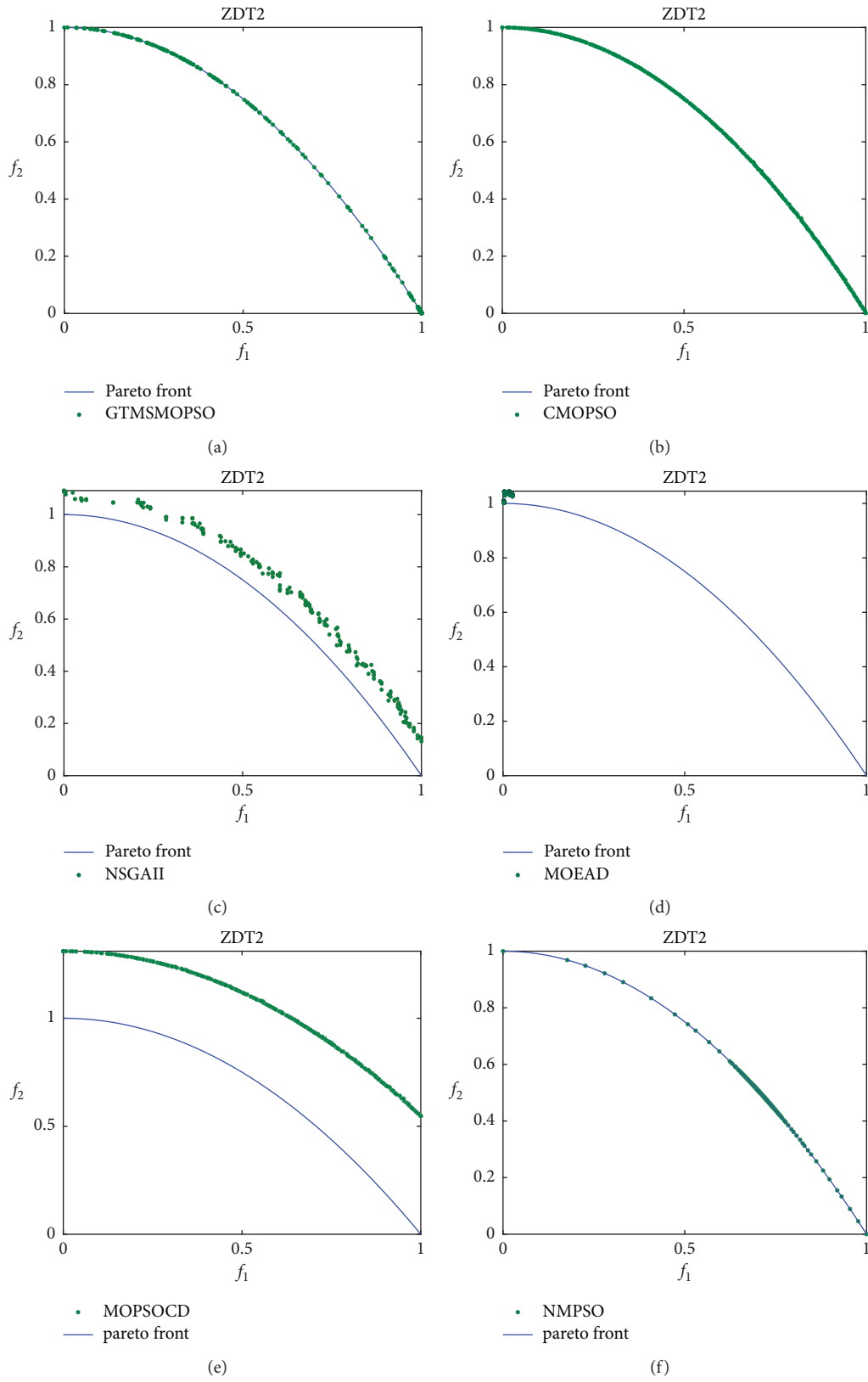


FIGURE 4: Pareto frontier effect of three different algorithms for ZDT2 function.

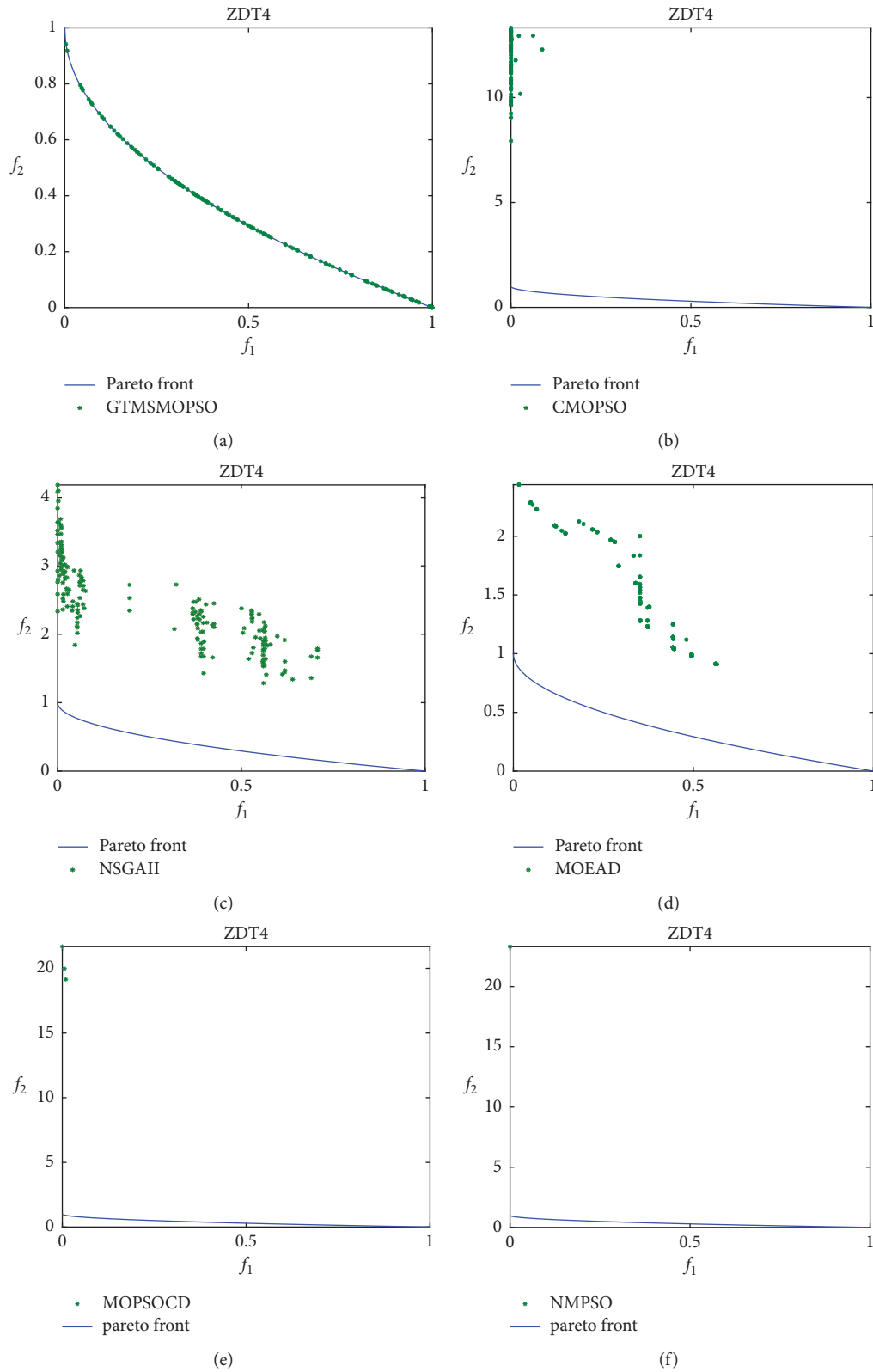


FIGURE 5: Pareto frontier effect of three different algorithms for ZDT4 function.

has four optimal IGD averages and three optimal HV averages in the UF series of test functions. GTMSMOPSO performs suboptimally for the IGD value in the UF8 test

function. In summary, it can be seen that GTMSMOPSO outperforms the other three multiobjective intelligent algorithms in the UF series of test functions.

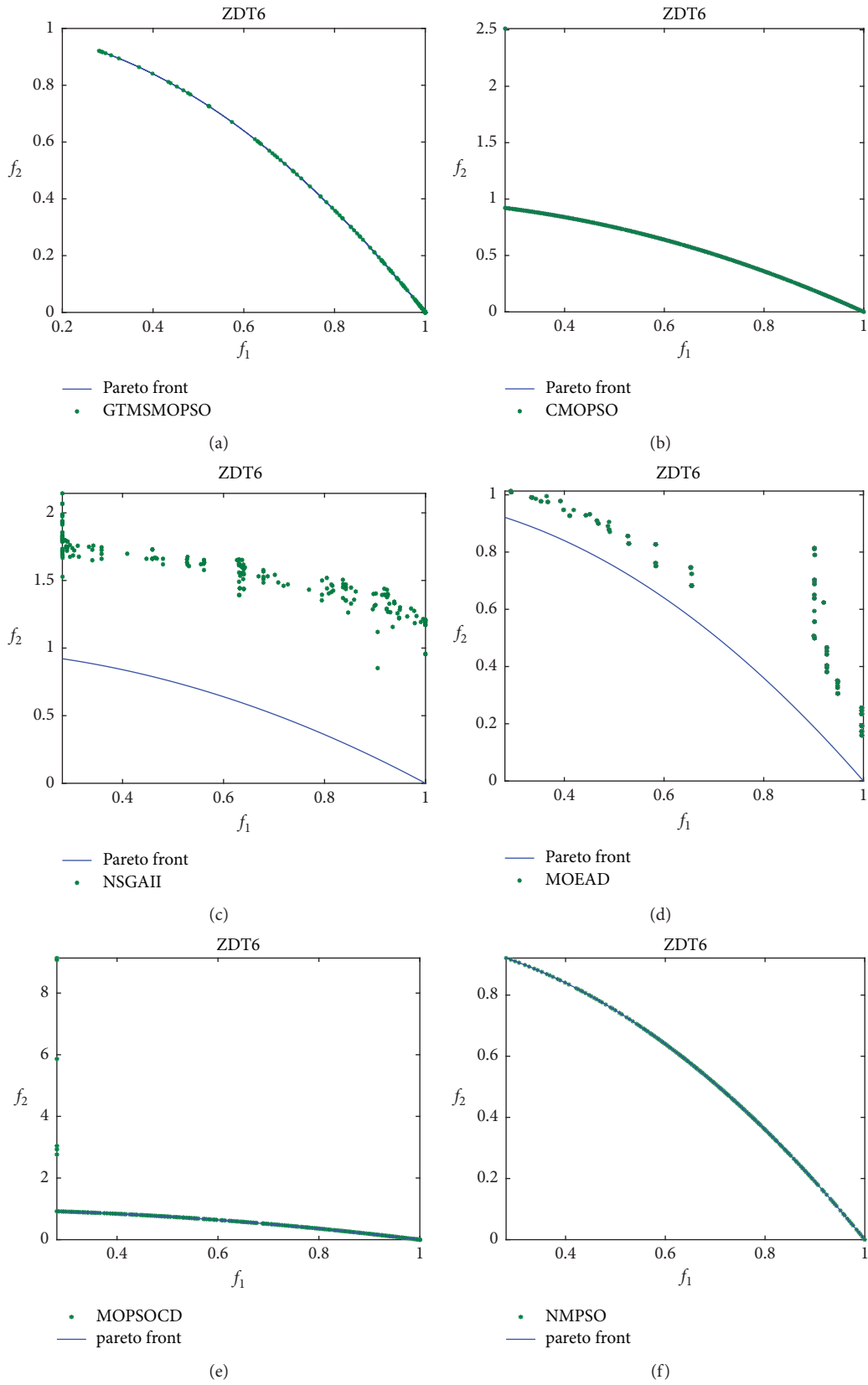


FIGURE 6: Pareto frontier effect of three different algorithms for ZDT6 function.

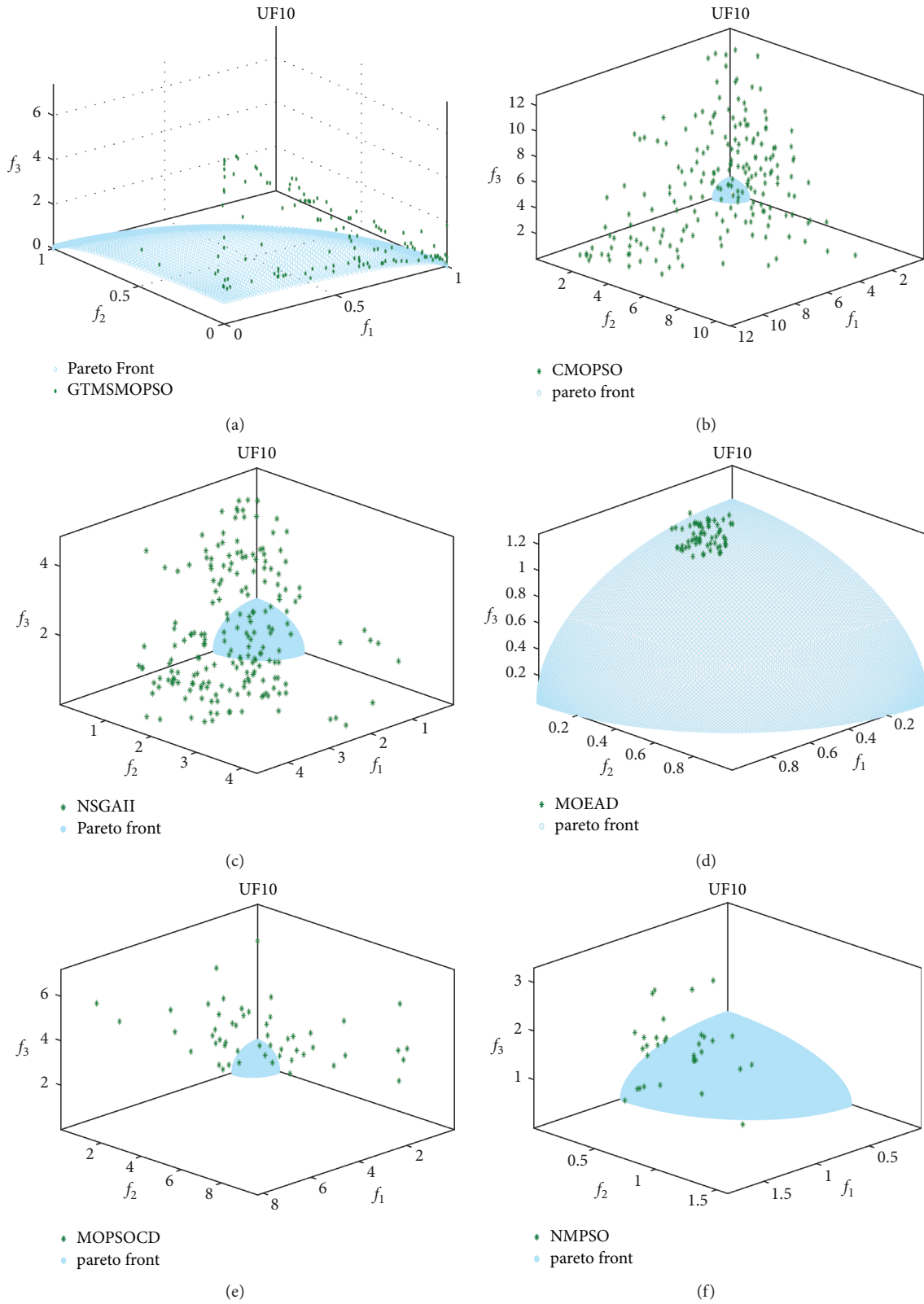


FIGURE 7: Pareto frontier effect of three different algorithms for UF10 function.

The multiobjective test functions DTLZ1 and DTLZ6 are both three-objective test functions. DTLZ1 is a multimodal function containing many local Pareto hyperplanes. The

GTMSMOPSO algorithm achieves the best IGD and HV values in the DTLZ1 function, which reflects a good performance. The GTMSMOPSO algorithm in the DTLZ6

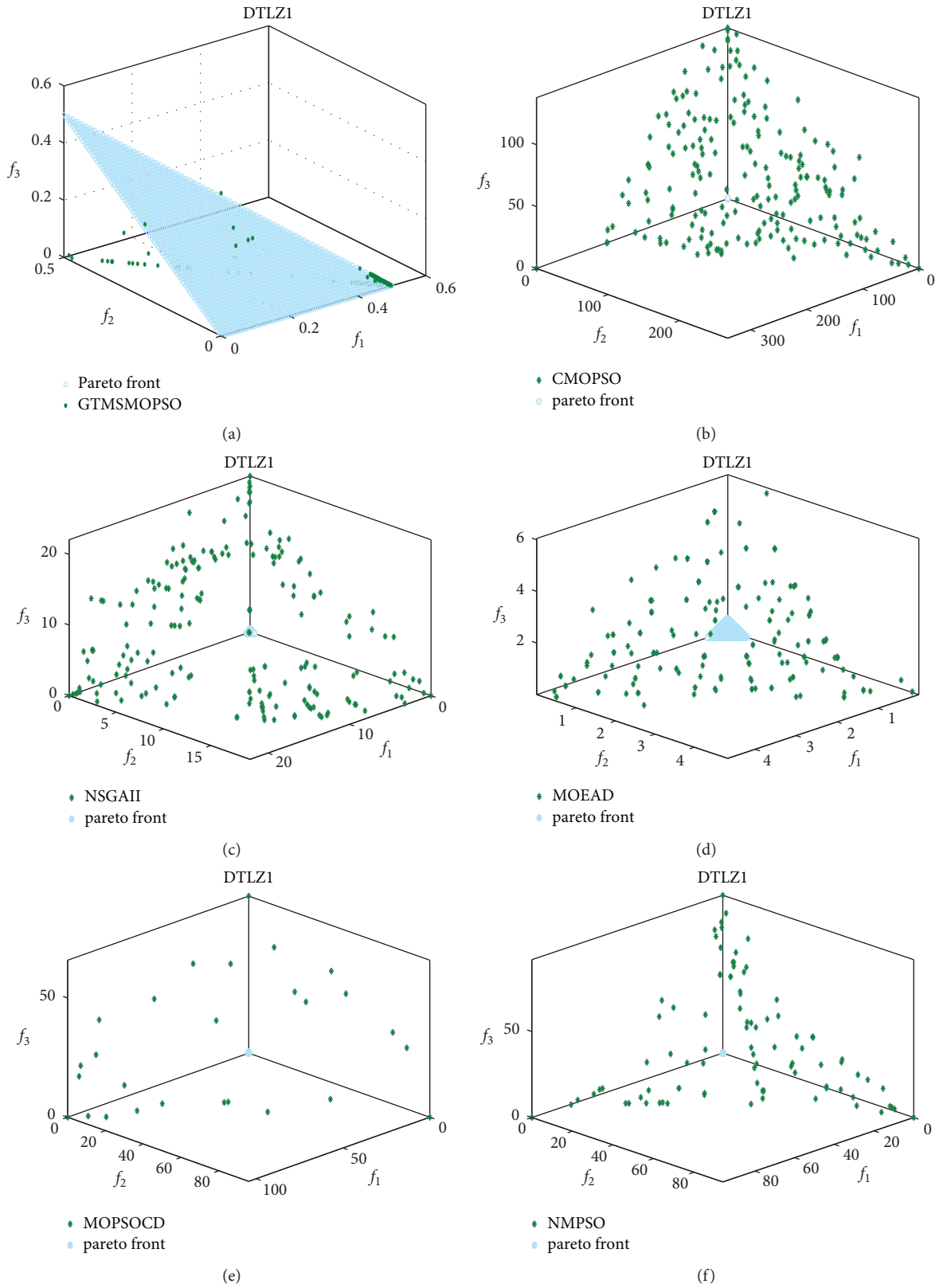


FIGURE 8: Pareto frontier effect of three different algorithms for DTLZ1 function.

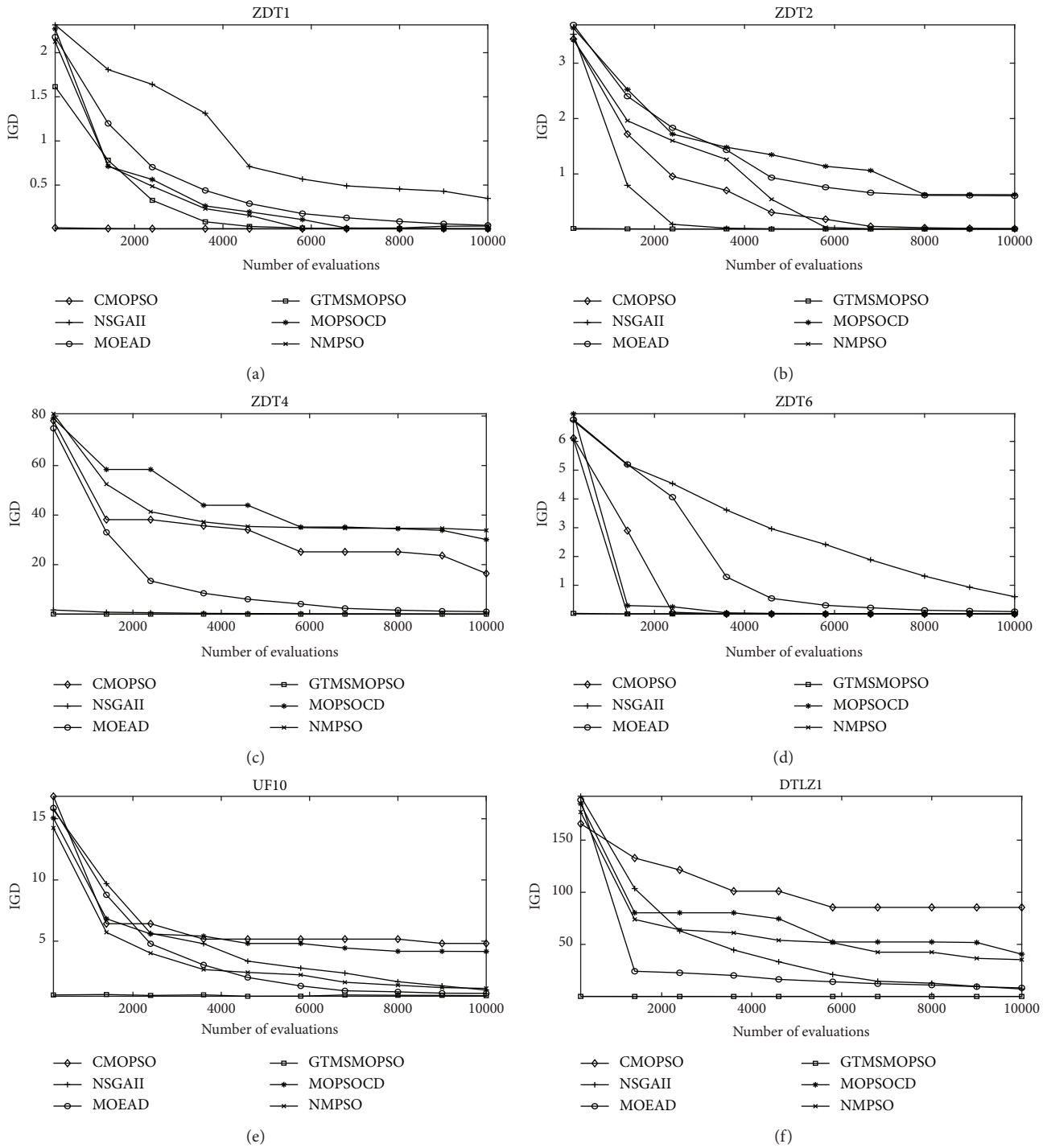


FIGURE 9: Convergence curves of different algorithms on 6 test functions.

function has suboptimal IGD values and HV values. In summary, it can be seen that the performance of the GTMSMOPSO algorithm in the test function of the DTLZ series is good.

To visualize the convergence and diversity of the noninferior solution sets obtained by each algorithm, Figures 3–8 show the noninferior solution sets obtained by each algorithm on the two-objective test functions (ZDT1, ZDT2, ZDT4, and ZDT6) and the three-objective test

functions (UF10 and DTLZ1) with the true Pareto front shown. Experimental simulations show that NSGAI and MOEAD both suffer from multiple underdiversity and underconvergence on the test functions in the ZDT series, and CMOPSO suffers from severe underdiversity and underconvergence on the ZDT4 test function, as seen in Figures 3–6. However, GTMSMOPSO not only approximates the true Pareto front on the test functions in the ZDT series, but also has better distributivity. Therefore,

GTMSMOPSO has better convergence and distribution of the test functions in the ZDT series. As can be seen from Figures 7 and 8, GTMSMOPSO is closer to the true Pareto front in the three-objective test functions (UF10 and DTLZ1), and the other five algorithms show multiple underconvergence or underdistribution. In summary, it can be seen that GTMSMOPSO has good convergence and distributivity compared with the other five compared algorithms.

From Figure 9, it can be concluded that GTMSMOPSO can converge faster to a smaller and stable IGD value with ZDT1, ZDT2, ZDT4, ZDT6, UF10, and DTLZ1 test functions among the 6 multiobjective test functions. Therefore, the algorithm in this paper outperforms the other five comparison algorithms.

## 5. Conclusion

In this paper, we propose a multiobjective particle swarm algorithm based on grid technology and multistrategy. The algorithm is improved by the maintenance of external archives and the selection of global optimal samples, and the variational operation of positions is proposed. For the maintenance of the external archive, the particles in the grid with the highest number of particles are removed by using a mixed evaluation index strategy, which provides a better quality of particles for selecting the global optimal solution. In the selection of globally optimal samples, the particles in the same grid are selected according to the current number of iterations using one of the inflection distance strategies and the grid density strategy to achieve a balance between algorithm exploration and exploitation, thus improving the diversity and convergence of the algorithm. To further enhance the diversity of the population, a linear incremental variation of the position of the particles is performed to enhance the exploration capability of the algorithm. To verify the effectiveness of the algorithm in this paper, simulation experiments are performed on 14 multiobjective functions (ZDT1-ZDT4, ZDT6, UF2-UF5, UF8-UF10, DTLZ1, and DTLZ6) of this paper's algorithm and five other multiobjective particle swarm algorithms. The experimental simulation results show that the algorithm in this paper has good convergence and diversity and has a good spatialization effect.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] R. M. Rizk-Allah, A. E. Hassanien, and A. Slowik, "Multi-objective orthogonal opposition-based crow search algorithm for large-scale multi-objective optimization," *Neural Computing & Applications*, vol. 32, no. 17, pp. 13715–13746, 2020.
- [2] J. Luo, A. Gupta, and YS. Ong, "Evolutionary optimization of expensive multi-objective problems with co-sub-Pareto front Gaussian process surrogates," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1708–1721, 2018.
- [3] X. Wang, L. Ma, S. Yang et al., "An aggregated pairwise comparison-based evolutionary algorithm for multi-objective and many-objective optimization," *Applied Soft Computing*, vol. 96, Article ID 106641, 2020.
- [4] W. Peng, L. Bo, and Z. Wen, "Adaptive region adjustment to improve the balance of convergence and diversity in MOEA/D," *Applied Soft Computing*, vol. 70, pp. 797–813, 2018.
- [5] C. Bao, L. Xu, and E. D. Goodman, "A new dominance-relation metric balancing convergence and diversity in multi- and many-objective optimization," *Expert Systems with Applications*, vol. 134, pp. 14–27, 2019.
- [6] G. Chen and J. Li, "A diversity ranking based evolutionary algorithm for multi-objective and many-objective optimization," *Swarm and Evolutionary Computation*, vol. 48, no. 1, pp. 274–287, 2019.
- [7] G. I. Chaman, C. A. Coello Coello, and A. A.-M. MOPSOHv, "A new hypervolume-based multi-objective particle swarm optimizer," *Evolutionary Computation IEEE*, pp. 266–273, 2014.
- [8] F. Li, J. Liu, and S. Tan, "R2-MOPSO: a multi-objective particle swarm optimizer based on R2-indicator and decomposition," *Evolutionary Computation IEEE*, pp. 3148–3155, 2015.
- [9] J. Liu, F. Li, and X. Kong, "Handling many-objective optimisation problems with R2 indicator and decomposition-based particle swarm optimiser," *International Journal of Systems Science*, vol. 50, no. 1-4, pp. 320–336, 2019.
- [10] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.
- [11] G. Li, W. Wang, W. Zhang, Z. Wang, H. Tu, and W. You, "Grid search based multi-population particle swarm optimization algorithm for multimodal multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 62, Article ID 100843, 2021.
- [12] A. N. N. G. Kayakutlu, "Multi-objective particle swarm optimization with random immigrants," *Complex Intelligent Systems*, vol. 6, no. 3, pp. 635–650, 2020.
- [13] F. Han, W.-T. Chen, Q.-H. Ling, and H. Han, "Multi-objective particle swarm optimization with adaptive strategies for feature selection," *Swarm and Evolutionary Computation*, vol. 62, no. 6, Article ID 100847, 2021.
- [14] X. Zhang, H. Liu, and L. Tu, "A modified particle swarm optimization for multimodal multi-objective optimization," *Engineering Applications of Artificial Intelligence*, vol. 95, Article ID 103905, 2020.
- [15] B. Qu, C. Li, and J. Liang, "A self-organized speciation based multi-objective particle swarm optimizer for multi-modal multi-objective problems," *Applied Soft Computing*, vol. 86, Article ID 105886, 2019.
- [16] V. Trivedi, P. Varshney, and M. Ramteke, "A simplified multi-objective particle swarm optimization algorithm," *Swarm Intelligence*, vol. 14, pp. 1–34, 2020.
- [17] H.-T. Chen, W.-C. Wang, X.-N. Chen, and L. Qiu, "Multi-objective reservoir operation using particle swarm optimization with adaptive random inertia weights," *Water Science and Engineering*, vol. 13, no. 2, pp. 136–144, 2020.
- [18] L. Chen, Q. Li, X. Zhao, Z. Fang, F. Peng, and J. Wang, "Multi-population coevolutionary dynamic multi-objective particle



- swarm optimization algorithm for power control based on improved crowding distance archive management in CRNs,” *Computer Communications*, vol. 145, pp. 146–160, 2019.
- [19] J. Kennedy and C. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Neural Networks*, pp. 1942–1948, Washington, DC, USA, June 1995.
- [20] Q. Qingfu Zhang, A. Aimin Zhou, and Y. Yaochu Jin, “RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.
- [21] J. Guillermo Falcn-Cardona and A. Carlos, “Convergence and diversity analysis of indicator-based multi-objective evolutionary algorithms,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, Prague, Czech, July 2019.
- [22] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multi-objective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [23] Q. Zhang, A. Zhou, and S. Zhao, *Multi-objective Optimization Test Instances for the CEC 2009 Special Session and Competition*, Mechanical Engineering, New York, NY, USA, 2008.
- [24] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 825–830, Hong Kong, China, June 2002.
- [25] X. Zhang, X. Zheng, R. Cheng, J. Qiu, and Y. Jin, “A competitive mechanism based multi-objective particle swarm optimizer with fast convergence,” *Information Sciences*, vol. 427, pp. 63–76, 2018.
- [26] 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.
- [27] Q. Hui Li 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.
- [28] C. R. Raquel and P. C. Naval Jr, “An effective use of crowding distance in multi-objective particle swarm optimization,” in *Proceedings of the 7th Annual Conference On Genetic And Evolutionary Computation*, pp. 257–264, Qingdao, China, November 2005.
- [29] 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. 1, pp. 32–46, 2018.
- [30] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “Platemo: a MATLAB platform for evolutionary multi-objective optimization [educational forum],” *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.