Research Article

Multi/Many-Objective Particle Swarm Optimization Algorithm Based on Competition Mechanism

Wusi Yang,¹ Li Chen ,¹ Yi Wang,¹ and Maosheng Zhang²

¹School of Information Technology and Software, Northwest University, Xi'an 710127, China

²Key Laboratory for Geo-Hazards in Loess Area, MNR, Xi'an Center of Geological Survey, China Geological Survey, Xi'an 710054, China

Correspondence should be addressed to Li Chen; chenli@nwu.edu.cn

Received 11 September 2019; Revised 7 December 2019; Accepted 16 January 2020; Published 20 February 2020

Academic Editor: Daniele Bibbo

Copyright © 2020 Wusi Yang 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.

The recently proposed multiobjective particle swarm optimization algorithm based on competition mechanism algorithm cannot effectively deal with many-objective optimization problems, which is characterized by relatively poor convergence and diversity, and long computing runtime. In this paper, a novel multi/many-objective particle swarm optimization algorithm based on competition mechanism is proposed, which maintains population diversity by the maximum and minimum angle between ordinary and extreme individuals. And the recently proposed θ -dominance is adopted to further enhance the performance of the algorithm. The proposed algorithm is evaluated on the standard benchmark problems DTLZ, WFG, and UF1-9 and compared with the four recently proposed multiobjective particle swarm optimization algorithms and four state-of-the-art many-objective evolutionary optimization algorithms. The experimental results indicate that the proposed algorithm has better convergence and diversity, and its performance is superior to other comparative algorithms on most test instances.

1. Introduction

Multiobjective optimization problem is generated from engineering application [1], controller optimization [2], economic scheduling [3], etc. When the number of objective is more than 3, it is defined as many-objective optimization problem (MaOP). The two main goals of solving multi/many-objective optimization problems are convergence and diversity of solutions. The purpose of convergence is to find a set of solutions as close as possible to the true Pareto front (PF) and diversity is to find a set of solutions as diverse as possible. Convergence to the true PF and maintaining a good diversity between solutions are two major difficulties in solving problems of multi/many-objective optimization. However, when solving the many-objective optimization problem, the number of multiple objectives needs to be optimized simultaneously and both the objective search space and the Pareto nondominance solutions increase exponentially. Moreover, the complexity of nondominated sorting increases dramatically. It is easy to find that the following problems commonly exist in solving many-

objective optimization problems, such as poor performance, high-computational complexity, obtained approximate solution cannot approximate the true PF, uneven distribution, incomplete coverage, and poor stability. In recent years, many scholars have proposed many effective methods to solve many-objective optimization problems and improve the performance of the algorithm. The research of many-objective optimization algorithms can be summarized as follows.

The decomposition strategy-based algorithm mainly uses the aggregation method to transform the multi/many-objective problem into a single-objective optimization problem, such as MOEA/D [4], MOEA/D-AED [5], and MPSOD [6]. In the performance indicator-based algorithm, performance indicators of solution quality measurement are adopted as selection criteria in the environmental selection, such as HypE [7], MOMBI-II [8], and MaOEA-IGD [9]. The reference point guidance-based algorithm uses a set of reference points to guide evolution, evaluate the quality of the solution, and control the distribution of the population in the objective space, such as NSGA-III [10] and RVEA [11]. The many-

objective optimization algorithm based on reduction of the objective number improves the optimization ability of the algorithm by reducing the objective dimension [12]. The loose Pareto dominance-based algorithm enhances the selection pressure of the algorithm by relaxing the dominance relationship in the selection of environment and strengthening the dominance relationship among individuals. Representatives of this type include α -dominance [13], fuzzy-dominance [14], r -dominance [15], and ε -dominance [5, 16].

In recent years, with the development of metaheuristic optimization algorithms based on different swarms, a large number of multiobjective optimization algorithms have been developed, such as firefly algorithm [17], bat searching algorithm [18], chicken swarm optimization algorithm [19], and artificial bee colony algorithm [20]. Particle swarm optimization (PSO) is an important metaheuristic optimization technology, which has been used to tackle multiobjective optimization problems and shows promising performance [21–23]. However, the multiobjective optimization algorithm based on particle swarm optimization is easy to fall into the local optimum. It is very important for the performance of multiobjective particle swarm optimization to achieve a reasonable balance between convergence and diversity. Generally, the optimal individual search of multiobjective particle swarm optimization is based on the assumption that no particle can achieve the best results on all the objectives of the multiobjective optimization problem. For this premise, there are several ways to update the speed and position of the PSO. In addition to the traditional methods of updating speed and location, there are other efficient methods. The double search strategy [24] is adopted to improve the velocity of updating particles, which are aimed at accelerating the convergence speed and maintaining the population diversity, respectively. As a variant of PSO, the competitive swarm optimization operator [25] is different from the traditional updating method mainly in which the competitors in the current group lead the search process rather than the historical location. And the competitive swarm optimization algorithm can achieve better results between convergence and diversity too. In addition, comprehensive learning particle swarm optimization uses the best historical information of particles to update the speed of particles, and this strategy can keep the diversity of the swarm and prevent premature convergence. The multi-swarm comprehensive learning particle swarm optimization (MSCLPSO [26]) algorithm uses this strategy and combines multiple swarms to update the position of the particles, achieving better results.

Many papers have proposed different multiobjective particle swarm optimization algorithms, but most algorithms perform poorly while tackling MaOP. It is mainly due to the insufficient selection pressure of the algorithm, which cannot effectively balance the relationship between convergence and diversity, and makes the algorithm fall into the local area, far away from the true Pareto front. For example, the CMOPSO algorithm proposed in [27], which updates the population based on competitive-mechanism learning, can effectively tackle multiobjective optimization problems. However, experiments show that CMOPSO performs poorly

in tackling many-objective optimization problems, and the algorithm generally runs for a long time on test problems. For the shortcomings of CMOPSO algorithm, we adopted a new environment selection strategy to maintain the diversity of population and employed θ -dominance sorting [28] and reference points regeneration strategy [11] to further enhance the performance of algorithm. A multi/many-objective particle swarm optimization algorithm based on competition mechanism is proposed, which is denoted as CMaPSO. The main contribution of this paper is summarized as follows:

- (1) In this paper, the recently proposed competitive mechanism-based learning is adopted to deal with many-objective optimization problems for the first time.
- (2) A new environment selection strategy to maintain the convergence and diversity of the algorithm and the optimal individuals are selected by using the maximum and minimum angle between the ordinary individuals and the extreme points.
- (3) The proposed algorithm CMaPSO are compared with four recently published multi/many-objective particle swarm optimization algorithms MMOPSO [24], MPSOD [6], CMOPSO [27], and dMOPSO [29]. Moreover, CMaPSO algorithm is compared with four state-of-the-art many-objective optimization algorithms, MOMBIII [8], NSGA-III [10], RPEA [30], and MyODEMR [31] on the standard benchmark problems to evaluate the performance of algorithms. Experimental results show that the performance of the proposed algorithm is better than the comparative algorithms on most standard benchmark problems.

The remainder of this paper is organized as follows. Section 2 introduces the related background. The proposed CMaPSO is described in detail in Section 3. Section 4 presents the benchmark MaOPs, performance metrics, and algorithm settings used for performance comparison. Finally, conclusion and future work are presented in Section 5.

2. Related Work

In this section, we introduce some concepts, including the definition of multiobjective optimization and particle swarm optimization. And we give a brief overview of the recently proposed algorithm θ -DEA [28]. Finally, several recently proposed multi/many-objective particle swarm optimization algorithms are summarized.

2.1. Multiobjective Optimization Problems. A general multiobjective optimization minimization problems can be formulated as follows [32]:

$$\begin{aligned} &\text{minimize} && F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ &\text{subject to} && g_i(x) \leq 0, \quad i = 1, 2, \dots, p \\ &&& h_i(x) = 0, \quad i = 1, 2, \dots, q, \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \dots, x_n) \in X \subset R^n$ is an n -dimensional decision vector bounded in the decision space X , $F(x)$ represents the m -dimensional objective vector, $f_i(x)$ represents the i th minimized objective function, and $g_i(x)$ and $h_i(x)$ are the constraint function of the problem. Several definitions related to multiobjective optimization are shown as follows.

Definition 1 (Pareto dominance). A decision vector x is said to dominate another decision vector y , denoted by $x < y$, if and only if

$$\begin{aligned} \forall i \in \{1, 2, \dots, m\}: f_i(x) &\leq f_i(y) \wedge, \\ \exists j \in \{1, 2, \dots, m\}: f_j(x) &< f_j(y). \end{aligned} \quad (2)$$

Definition 2 (Pareto optimal). A vector $x^* \in X$ is Pareto optimal if there is no other $x \in X$, such that $x < x^*$.

Definition 3 (Pareto optimal set). For a given multiobjective optimization problem, the Pareto set is defined by the following equation:

$$PS = \{x \in X \mid \nexists z \in X, z < x\}. \quad (3)$$

Definition 4 (Pareto front). For a given multiobjective optimization problem, the Pareto front is defined as follows:

$$PS = \{F(X) \mid x \in PS\}. \quad (4)$$

2.2. Particle Swarm Optimization. A stochastic algorithm based on bionic swarm intelligence named particle swarm optimization (PSO) [33] is proposed by Kennedy and Eberhart, which simulates the predation behavior of birds and fish. It seeks the optimal solution through the cooperation and information sharing among individuals in the swarm. At present, it has been widely used in function optimization, neural network training, fuzzy system control, and other application fields. Each particle represents a potential solution for the optimization problem, which is influenced by position and moving velocity. The velocity and position of the k th particle are represented as $v_k = (v_{k1}, \dots, v_{km})$ and $x_k = (x_{k1}, \dots, x_{km})$, respectively. The velocity and position of the k th particle are updated by the following function:

$$\begin{aligned} v_{kj}(t+1) &= wv_{kj}(t) + c_1r_1(pbest_{kj}(t) - x_{kj}(t)) \\ &\quad + c_2r_2(gbest_{kj}(t) - x_{kj}(t)), \\ x_{kj}(t+1) &= v_{kj}(t+1) + x_{kj}(t), \end{aligned} \quad (5)$$

where $j(j = 1, \dots, n)$, t is the generation, w is the inertia weight which is used to balance local and global search, c_1 and c_2 are two learning factors from the personal and global best particles, which are called the acceleration coefficients too, and r_1 and r_2 are elements from two uniform random sequences in the range of $[0, 1]$.

2.3. θ -DEA. The newly proposed evolutionary algorithm (θ -DEA) [28] is based on a new dominance relation θ -dominance, which improves the convergence of NSGA-III [10] by referring to the procedure of the MOEA/D [4] algorithm, and still inherits the advantage of NSGA-III in maintaining diversity. In the θ -dominance, the solution is divided into different groups by giving a set of well-distributed reference points, where each group is represented by a reference point. A fitness function similar to PBI is defined to select a set of optimal solutions. The algorithm framework of θ -DEA [28] is summarized as follows.

Firstly, a group of uniformly distributed reference points is generated in the objective space and population P is initialized. Ideal point z^* and the nadir point z^{nad} can be easily obtained. Then, the recombination operator is used to recombine P to generate offspring population, which is combined with the current population P to compose a new population S . Next, a new population Q is obtained based on the Pareto nondominated level of the population S . The population Q is normalized with the help of z^* and z^{nad} , and the population Q is clustered by the clustering operator, where each reference point represented a cluster. Finally, the nondominated sorting based on θ -dominance is employed to classify Q into different θ -dominance levels. The algorithm selects the solutions according to the different levels of θ -dominance until the termination condition is reached.

2.4. Some Current Multi/Many-Objective Particle Swarm Optimization. Particle swarm optimization is often used to solve multiobjective optimization problems because of its fast convergence and easy implementation. At present, there are many research literatures on multiobjective particle swarm optimization, such as NMPSO [34], MPSO-D [6], and D²MOPSO [21]. The existing multiobjective particle swarm optimization algorithms can be generally summarized into the following types. The first type is formed by the multiobjective particle swarm optimization algorithm dominated by Pareto, which determines the personal best and global best particles based on the Pareto ranking. Through Pareto ranking and finite iteration updating of the nondominance solution, the algorithm exploits and explores the global best particles that are close to the entire true PF. The classic multiobjective optimization problems include MOPSO [22], OMOPSO [23], SMPSO [35], CMOPSO [27], and AMOPSO [36]. The second category adopts the decomposition method, which decomposes the MOP into a group of single-objective optimization problems and solves each single objective optimization problem directly by PSO. This approach has been proved to be effective in solving complex MOPs, and its representative algorithms are dMOPSO [29], D²DMOPSO [21], MOPSO/D [37], MPSO/D [6], AgMOPSO [7], and MMOPSO [24]. We select a few of the latest algorithms in these algorithms for a brief overview.

MPSO-D [6] employs a set of direction vectors to decompose the target space into a set of subspaces and ensures that each subregion has a solution. Moreover, the crowding distance and adjacent particles are used to determine the global best historical position of the particles.

MMOPSO [24] takes the decomposition approach to transform the multiobjective optimization problem into a group of aggregation problems and adopts the double search strategy to improve the velocity of updating particles. The weakness of PSO search can be effectively improved by evolutionary search on the nondominant solutions stored in external archives.

COMPSON [27] adopts Pareto dominance and competition-based learning strategy which are used to update particles, where each particle learns from the winner of each pair completion. However, competition is only between elite particles selected in current population, and there is no external archive to save global and personal best particles.

AgMOPSO [38] also transforms the multiobjective optimization problems into a set of aggregation subproblems by the decomposition method and optimizes them with the allocated particles. Archive-guided velocity update method is employed to guide the swarm for exploration, and the Pbest, lbest, and gbest particles are selected from the external archive, which is evolved by the immune evolution strategy.

In addition, there are recently proposed MOPSO methods based on indicators, reference points, and balanceable fitness estimation, such as NMPSO [34], MaOPSO [39], and R2-MOPSO [40], which are used to solve high-dimensional multiobjective optimization problems. NMPSO [34] uses a balanced fitness estimation method, which combines convergence and diversity distance to solve many-objective optimization problems. In order to enhance the performance of the algorithm, evolutionary search is used in external archives and a new PSO speed update equation is applied. The algorithm MaOPSO [39] adopts a set of reference points that are dynamically determined based on the search process and imposes the necessary selection pressure on the algorithm to make it converge to the true PF, while maintaining the diversity of the PF. R2-MOPSO [40] combines R2 performance metrics with particle swarm optimization and guides the search through a well-designed interactive process for solving many-objective optimization problems.

3. The Proposed Algorithm

In this section, the proposed algorithm CMaPSO is mainly described in detail, since the competitive swarm operator can achieve a better balance between convergence and diversity than the traditional particle swarm optimization algorithm. In this paper, the particle swarm optimization algorithm based on the competitive mechanism is combined with another environment selection mechanism, which is different from the algorithm CMOPSO [27]. In order to improve the convergence and diversity of the proposed algorithm in solving many-objective optimization problems, the recently proposed θ -dominance is adopted to further enhance the selection pressure for nondominated solutions. However, experiments show that simply using θ -dominance can cause the algorithm to perform poorly on some degenerate problems. Therefore, a reference points regeneration strategy is used to improve

this deficiency. We will describe the main components of the algorithm CMaPSO, including competitive mechanism for velocity update and position, environmental selection based on extreme points, and θ -dominance are described. Finally, we will give the complete framework of the algorithm CMaPSO.

3.1. Competitive Mechanism for Velocity Update and Position. Similar to the algorithm CMOPSO, the population is sorted by nondominated sorting and crowding distance ranking, but the difference is that according to the current population size, different numbers of outstanding individuals are obtained from the sorted population to form a competitive group. Crowding distance [41] is an index to describe the degree of crowding between individuals in the same nondominated front and their neighbors. A pair of particles are randomly selected from the current competitive group, where the winners will be used to guide the moving directions of particles in the current swarm.

The whole process of how to generate elite particles is called competition mechanism. Figure 1 shows the process of generating elite particles. For each pairwise competition, given a particle w in the swarm, two elite particles a and b are randomly selected from the competitive group. The elite particle with a smaller angle wins the competition after the angles between a , b , and w are calculated, respectively. It is easy to find from the figure that the angle θ_1 of a to w is significantly smaller than the angle θ_2 of b to w , so a as an elite particle guides the evolution of the population.

The updated velocity $\bar{v}_i$ and position $\bar{p}_i$ of the i th particle are calculated using the following equations as suggested in the competitive swarm optimizer:

$$\bar{v}_i = c_1 R_1 v_i + c_2 R_2 (p_w - p_i), \quad (6)$$

$$\bar{p}_i = p_i + \bar{v}_i, \quad (7)$$

where $R_1, R_2 \in [0, 1]$ are two randomly generated vectors and p_w is the position of the winner, and c_1 and c_2 are learning factors. The pseudocode for modified competition mechanism-based learning strategy is described in Algorithm 1. Finally, the polynomial mutation are widely adopted in the multiobjective optimization algorithm [24, 27] is used to mutate the population to enhance the diversity. The expression of polynomial variation is as follows:

$$v_k^2 = v_k + \delta \times (u_k - l_k), \quad (8)$$

where

$$\delta = \begin{cases} 2u + (1 - 2u)(1 - \delta_1)^{1/(\eta_m + 1)}, & \text{if } u \leq 0.5, \\ 1 - [2(1 - u) + (u - 0.5)(1 - \delta_2)^{\eta_m + 1}]^{1/(\eta_m + 1)}, & \text{if } u > 0.5, \end{cases} \quad (9)$$

with $\delta_1 = (v_k - l_k)/(u_k - l_k)$, $\delta_2 = (u_k - v_k)/(u_k - l_k)$, u is a random number in $[0, 1]$, η_m is the distribution index, and v_k is a parent individual.

FIGURE 1: An example of elite particle generation.

3.2. Generation of Reference Points. In this paper, θ -dominance ordering is adopted to increase the selection pressure of the algorithm, and the selection of nondominated solution is implemented by the guidance of reference points. Therefore, it is necessary to introduce the generation of reference points. Reference point methods are often used to improve MOEAs' diversity and uniformity capacity in solving the multi/many-objective optimization problem, such as NSGA-III [10] and θ -EDA [28]. These algorithms first generate a set of predefined or user-preferred reference points based on the decomposition. This paper adopts a two-layer (boundary and inside layers) approach [10, 28], which uses the systematic approach to generate two reference direction sets: one set on the boundary layer and the other on the inside layer. Suppose the divisions of boundary and inner layers is H_1 and H_2 , respectively, then number of reference points is computed as follows:

$$P = \binom{H_1 + m - 1}{H_1} + \binom{H_2 + m - 1}{H_2}, \quad (10)$$

where m is the dimension of objective space and H_1 and H_2 are user-defined integers.

3.3. θ -Dominance Sorting. θ -dominance adopts the PBI (penalty-based boundary intersection) method of MOEA/D [4]. And the PBI scalar function is defined as follows:

$$\min_{x \in \Omega} g^{\text{PBI}}(x | \lambda, z^*) = d_1 + \theta d_2, \quad (11)$$

where

$$d_1 = \frac{\|(F(x) - z^*)^T \lambda\|}{\|\lambda\|}, \quad (12)$$

$$d_2 = \left\| (F(x) - z^*) - d_1 \frac{\lambda}{\|\lambda\|} \right\|, \quad (13)$$

$\theta > 0$ is a preset penalty parameter and generally set to $\theta = 5$. $z^* = (z_1^*, z_2^*, \dots, z_m^*)$ is the reference point, i.e., $z_i^* = \min\{f_i(x) | x \in \Omega\}$ for each $i = 1, 2, \dots, m$.

The normalized population is clustered by PBI scalar function, and the θ -dominance ordering is performed according to the value of PBI, where the extreme solutions are controlled by the size of θ . The definition of θ -dominance is as follows: given two solutions $x, y \in S_t$, x is said to θ -dominate y , denoted by $x \prec_\theta y$, iff $x \in C_i$, $y \in C_i$ and

$F_i(x) < F_i(y)$, where $i \in \{1, 2, \dots, N\}$. The purpose of θ -dominance is to obtain a set of nondominated solutions nearest to PF and well distributed along PF. The pseudocode of the θ -dominance sorting process is shown in Algorithm 2.

3.4. The Reference Point Regeneration Strategy. The uniformly distributed reference point is based on the general assumption that PF has a regular geometric structure, that is, smooth, continuous, and well distributed. However, there are various multiobjective optimization problems in practical applications. The geometric structure of PF may be very irregular, such as discontinuous, degenerate, and multimodal. For these problems, if uniformly distributed reference vectors are still employed and some reference vectors may not be associated with individuals, then the density of Pareto optimal solutions is insufficient. Based on such problems, we will adopt the reference vector regeneration strategy proposed in RVEA [11]. This strategy mainly regenerates reference points and replaces those that are not used and wasted. The pseudocode of the reference points regeneration strategy is shown in Algorithm 3.

3.5. Environmental Selection Based on Extreme Points. In this paper, we discuss the extreme points, which mainly refer to the extreme solutions in the normalized solution set. Let $X = (x_1, x_2, \dots, x_M)$ be a solution in the normalized solution set. If solution X has a component x_i , $i = 1, 2, \dots, M$, which is the maximum or minimum value of the corresponding position component in all solutions, then this solution is defined as an extreme solution here. Firstly, the extreme solutions are selected to enter the archive. Then, the angle between the unselected solution and the selected solutions are calculated, and the solution entering the archive is selected by the size of angle. The angle between solutions x_i and y_j is defined as follows:

$$\text{angle}(x_i, y_j) \triangleq \arccos \left| \frac{F'(x_i) \cdot F'(y_j)}{x_i \cdot y_j} \right|. \quad (14)$$

Obviously, $\text{angle}(x_i, y_j) \in [0, \pi/2]$.

The process of archives updating based on extreme points is illustrated in Figure 2. Suppose there is a population in the objective space, including five individuals X_1, X_2, X_3, Y_1 , and Y_2 . Firstly, the population is normalized, and the two extreme points Y_1 and Y_2 are selected by calculation. Then, the angles between X_1, X_2 , and X_3 and Y_1 and Y_2 are calculated, respectively, and the maximum is obtained after minimizing the results. X_1 is first selected to enter into the optimal population, and in this way, X_2 and X_3 are sequentially selected into the optimal population. It can be seen from the individual selection process that the individual who is away from the two extreme points and in the middle of the two extreme points is preferentially selected, which can ensure the diversity and uniformity of the optimal population. Algorithm 4 shows pseudocode for archives updating based on extreme points.

Input: Initial population P^* ; elite particle set Q ; size of elite particle set K
Output: new positions P

- (1) $Q \leftarrow$ select K particles from P^* according to nondominance sort and crowding distance;
- (2) for $i = 1:N$
- (3) randomly choose two elite particles a, b from Q
- (4) the elite particle with a smaller angle wins the competition
- (5) update the velocity of particle using equation (6)
- (6) update the position of particle p using equation (7)
- (7) update the position P according to P^* and p
- (8) end
- (9) $P \leftarrow$ Polynomial_Mutation(P) using equation (8)

ALGORITHM 1: Competition-Based Learning.

Input: population P
Output: new population P

- (1) $GDnum \leftarrow$ GetParetoDominanceNum(P);
- (2) Calculate d_1 using equation (12);
- (3) Calculate d_2 using equation (13);
- (4) $k \leftarrow \min(d_2)$;
- (5) for $i = 1$ to N
- (6) $D \leftarrow$ Calculate $d_1(k, i) + \theta * d_2(k, i)$ using equation (11);
- (7) rank $\leftarrow$ sort(D);
- (8) end for
- (9) $P \leftarrow P(\text{rank})$;

ALGORITHM 2: θ -dominance sorting.

Input: population P , reference points K
Output: new reference points K

- (1) associate $\leftarrow$ Population association with reference points (P, K);
- (2) $K_{\text{unused}} \leftarrow$ Find the unused reference point;
- (3) $K_{\text{used}} \leftarrow$ Find the used reference point;
- (4) $Z_{\text{max}} \leftarrow$ Calculate maximal objective values;
- (5) $K_{\text{unused}} \leftarrow$ Generate random values between $[0, K_{\text{max}}]$;
- (6) $K \leftarrow K_{\text{used}} \cup K_{\text{unused}}$;

ALGORITHM 3: Reference point regeneration strategy.

3.6. The Complete CMaPSO Algorithm. The complete framework of the proposed CMaPSO method is described in Algorithm 5. Firstly, the population is initialized, and the corresponding population number is obtained. A set of uniformly distributed reference points K are generated using equation (8). Then, the external archives are updated with the existing information. In the while loop, the competition mechanism of PSO algorithm (Algorithm 1) is used to update the population P to obtain a new population P . The next step is to update the external archives with the new population P , and the elite population A is obtained. In order to avoid the algorithm falling into local optimum, the genetic operator is used to cross and mutate external archives A so that the information between elite individuals can be fully exchanged, and new population S can be generated. The external archives are updated by combining the new

population S and elite population A , and the new elite population A is gained. Finally, under certain conditions, the reference point regeneration strategy is adopted to obtain new reference points to replace the previous according to the distribution of individuals in archives A . The algorithm continues the while-loop until the termination condition is satisfied.

3.7. Computational Complexity of Proposed Algorithm CMaPSO. We consider the main steps of the proposed algorithms and analyzed the algorithm complexity of the algorithm CMaPSO. The algorithm CMaPSO is mainly composed of learning based on competition mechanism, updating elite archives, θ -dominance sorting, and reference points regeneration strategy. The time complexity of these

FIGURE 2: Illustration of individual selection by angle size.

Input: population P ; population number N

Output: new archives A

- (1) $Q \leftarrow$ Pareto-dominance sorting of population P and selecting individuals with rank 1
- (2) $Zmin \leftarrow \min(Q)$;
- (3) $Zmax \leftarrow \max(Q)$;
- (4) $Q = (Q - Zmin) / (Zmax - Zmin)$;
- (5) $Q \leftarrow \theta$ -dominance sorting of population Q and selecting individuals with rank 1 (Algorithm 2);
- (6) $N_1 \leftarrow \text{size}(Q)$;
- (7) Angle $\leftarrow$ Calculating population Q and Q angle using equation (13)
- (8) choose $\leftarrow \text{false}(1, N_1)$;
- (9) Extreme1 $\leftarrow \min(Q)$;
- (10) Extreme2 $\leftarrow \max(Q)$;
- (11) choose $\leftarrow \text{choose}(\text{Extreme1})$;
- (12) choose $\leftarrow \text{choose}(\text{Extreme2})$;
- (13) **while** $\text{sum}(\text{choose}) < \min(N, N_1)$
- (14) Select $\leftarrow \text{choose}$;
- (15) Unselect $\leftarrow \sim \text{choose}$;
- (16) $k \leftarrow \max(\min(\text{Angle}(\text{Select}, \text{Unselect})))$;
- (17) choose $\leftarrow \text{choose}(k) = \text{true}$;
- (18) **end while**
- (19) $A \leftarrow P(\text{choose})$;

ALGORITHM 4: Archives updating.

Input: initialization population P ; Population number N ;

Output: updated Archives A

- (1) $K \leftarrow$ Generating uniformly distributed reference points
- (2) $A \leftarrow$ Initialize the archives(P, N, K) Algorithm 4;
- (3) **While** the termination criterion is not met **do**
- (4) $P \leftarrow$ Competition Learning (P) Algorithm 1;
- (5) $A \leftarrow$ Update Archives($[P, A], N, K$) Algorithm 4;
- (6) $S \leftarrow$ Cross-mutation of A by genetic operator;
- (7) $A \leftarrow$ Update Archives($[A, S], N, K$) Algorithm 4;
- (8) $K \leftarrow$ Reference points regeneration strategy Algorithm 3;
- (9) **End While**

ALGORITHM 5: Framework of the proposed CMapSO.

main parts is $O(mN^2)$ in the worst case. Therefore, the overall worst-case time complexity of a generation in CMaPSO is $O(mN^2)$.

4. Experimental Studies

This section is devoted to the research of simulation experiments by selecting appropriate comparative algorithms, standard benchmark problems, performance metrics, etc. The performance of the proposed algorithm CMaPSO is evaluated by comparison experiment results. The background of simulation experiment is introduced, including standard benchmark problems, performance metrics, experimental environment, and some parameters setting. In order to evaluate the performance of CMaPSO, the recently published advanced algorithms are compared with the algorithm. Four recently proposed multi/many-objective particle swarm optimization algorithms, such as CMOPSO [27], MMOPSO [24], MPSOD [6], and dMOPSO [29] and four recently proposed many-objective evolutionary optimization algorithms, such as NSGA-III [10], MOBIII [8], RPEA [30], and MyODEMR [31] are employed to validate the performance of the proposed algorithm. In the simulation experiments, we use the parameters proposed in the original paper to set the comparative algorithms.

4.1. Standard Benchmark Problems. In order to verify the performance of the proposed algorithm CMaPSO, a total of 25 benchmark MOPs with 3, 5, 8, 10, and 15 objectives from two test suits are used to evaluate the performance of the proposed algorithm CMaPSO, including DTLZ [42], WFG [43], and UF [44], where DTLZ1 to DTLZ7, WFG1 to WFG9, and UF1 to UF9 are considered. The Pareto front characteristics of test problems adopted in simulation experiments are summarized, as shown in Table 1. The mathematical expressions of the test sets DTLZ, WFG, and UF are presented in Supplementary materials (available (here)).

We have adopted the proposals for setting decision variables of test suites DTLZ and WFG in [34, 45]. For DTLZ1 to DTLZ7, the number of decision variables is set as $d = m + k - 1$, where n is the number of decision variables, m is the number of objective, and k is the fixed parameter. For WFG1 to WFG9, the number of decision variables is set as $d = k + l$, where n is the number of decision variables, k is set as $2 * (m - 1)$, m is the objective number, and l is the fixed parameter. The number of objectives and decision variables in the UF test set are arranged according to the recommendations in [44], where the test functions UF1 to UF7 are with the objective number $m = 2$, the number of decision scalars $d = 30$, and UF8 to UF9 with the objective number $m = 3$ and $d = 30$. The detailed setting information is shown in Table 2.

4.2. Parameter Setting. In the simulation experiment of this paper, different population numbers are set for different dimensions, and the number of evaluations is uniformly set to 150000, as shown in Table 3. For each test case, all

TABLE 1: The features of the test problems.

Problems	Features
DTLZ1	Linear, multimodal
DTLZ2	Concave
DTLZ3	Concave, multimodal
DTLZ4	Concave, biased
DTLZ5	Concave, degenerate
DTLZ6	Concave, degenerate, biased
DTLZ7	Mixed, disconnected, multimodal, scaled
WFG1	Mixed, biased, scaled
WFG2	Convex, disconnected, multimodal, nonseparable, scaled
WFG3	Linear, degenerate, nonseparable, scaled
WFG4	Concave, multimodal, scaled
WFG5	Concave, deceptive, scaled
WFG6	Concave, nonseparable, scaled
WFG7	Concave, biased, scaled
WFG8	Concave, biased, nonseparable, scaled
WFG9	Concave, biased, multimodal, deceptive, nonseparable, scaled
UF1	Complex Pareto set
UF2	Complex Pareto set
UF3	Complex Pareto set
UF4	Complex Pareto set, concave
UF5	Complex Pareto set, discrete
UF6	Complex Pareto set, discontinuous
UF7	Complex Pareto set
UF8	Complex Pareto set, concave
UF9	Complex Pareto set, discontinuous

TABLE 2: Test sets DTLZ and WFG decision variables settings.

Problems	m	d	Parameter
DTLZ1	3, 5, 8, 10, 15	$m + k - 1$	$k = 5$
DTLZ2- DTLZ6	3, 5, 8, 10, 15	$m + k - 1$	$k = 10$
DTLZ7	3, 5, 8, 10, 15	$m + k - 1$	$k = 20$
WFG	3, 5, 8, 10, 15	$k + l$	$k = 2 * (m - 1); l = 20$
UF1-UF7	2	30	
UF8-UF9	2	30	

TABLE 3: Set different population sizes for different dimensions and a uniform number of evaluations.

Dimension number	Population number	Evaluation
3	100	150000
5	150	150000
8	200	150000
10	250	150000
15	300	150000

employed algorithms run independently for 30 times on a PC with Intel Core i7-6500u @ 2.50 GHz dual-core CPU and Microsoft Windows 10 operating system. The simulation experiment was carried out on the recently proposed PlatEMO-2.0 [46]. The average value and standard deviation of the simulated data of all the comparison algorithms are compared to illustrate the performance of the algorithm, where the best performance value on each test instance is highlighted with a bold background. Moreover, the Wilcoxon rank sum test is adopted at a significance level of 0.05,

TABLE 4: The mean and standard deviation of the IGD value of the proposed algorithms and the four recently comparative algorithms CMOPSO, MMOPSO, MPSOD, and dMOPSO on DTLZ for 3, 5, 8, 10, and 15 objective problems, where the best value for each test case is highlighted with a bold background.

Problem	M	CMOPSO	dMOPSO	MMOPSO	MPSOD	CMApSO
DTLZ1	3	3.2557e-2(1.79e-2)-	7.8004e-1(6.93e-1)-	5.4064e-2(8.59e-2)-	1.9100e-2(5.02e-5)+	2.2047e-2(2.92e-4)
	5	7.3425e+0(3.95e+0)-	7.6675e-1(7.32e-1)-	1.9539e+0(1.99e+0)-	6.7643e-2(3.40e-2)-	6.4419e-2(5.43e-3)
	8	3.9661e+1(1.55e+1)-	4.6021e-1(1.83e-1)-	1.1739e+1(6.20e+0)-	3.5917e-1(7.11e-2)-	1.0696e-1(5.96e-3)
	10	9.1496e+1(2.26e+1)-	1.2893e+0(1.71e+0)-	1.3584e+1(3.95e+0)-	7.6499e-1(1.39e-1)-	1.3077e-1(1.58e-2)
	15	1.0964e+2(1.26e+1)-	9.2687e-1(9.42e-1)-	1.6867e+1(7.47e+0)-	1.2428e+0(4.81e-1)-	1.3692e-1(8.31e-3)
DTLZ2	3	5.7035e-2(3.05e-4)=	1.3362e-1(7.19e-3)-	6.9615e-2(1.86e-3)-	5.4516e-2(1.23e-5)+	6.0866e-2(1.22e-3)
	5	3.6973e-1(3.88e-2)-	2.9030e-1(4.75e-3)-	2.4978e-1(1.01e-2)-	1.9659e-1(6.55e-5)=	1.9525e-1(1.20e-3)
	8	2.3305e+0(2.70e-2)-	5.1260e-1(3.01e-2)-	7.3105e-1(6.33e-2)-	3.1787e-1(3.28e-4)+	3.5837e-1(1.79e-3)
	10	2.3737e+0(1.60e-2)-	6.0004e-1(3.10e-2)-	1.1142e+0(9.31e-2)-	4.4485e-1(5.84e-4)-	4.2631e-1(2.22e-3)
	15	2.4967e+0(1.68e-2)-	8.2668e-1(3.60e-2)-	1.4850e+0(1.96e-1)-	5.3230e-1(7.13e-4)-	4.9904e-1(1.88e-3)
DTLZ3	3	3.4943e+1(1.33e+1)-	3.7248e+0(4.47e+0)-	7.3776e-2(8.27e-3)-	7.6661e+0(3.46e+0)-	6.2885e-2(6.68e-3)
	5	1.4470e+2(1.53e+1)-	1.2027e+2(5.87e+1)-	5.0745e+1(5.00e+1)-	1.8499e+1(3.69e+0)-	2.1712e-1(1.90e-2)
	8	4.3773e+2(1.09e+2)-	1.6852e+2(5.82e+1)-	1.6750e+2(3.70e+1)-	3.3532e+1(4.90e+0)-	6.2463e-1(2.48e-1)
	10	5.3570e+2(1.46e+2)-	1.7975e+2(1.55e+1)-	1.9185e+2(2.35e+1)-	4.8642e+1(5.54e+0)-	1.9282e+0(1.17e+0)
	15	7.3422e+2(6.36e+0)-	1.8714e+2(1.26e+1)-	1.8323e+2(4.26e+1)-	6.1343e+1(8.37e+0)-	1.1990e+0(7.43e-1)
DTLZ4	3	6.2273e-2(3.06e-3)+	2.2990e-1(4.42e-2)+	7.0481e-2(1.90e-3)+	5.4519e-2(1.24e-5)+	3.3993e-1(3.21e-1)
	5	4.1558e-1(5.73e-2)-	4.4812e-1(2.54e-2)-	2.3859e-1(9.52e-3)-	1.9596e-1(1.04e-4)=	1.9522e-1(9.00e-4)
	8	1.5137e+0(1.31e-1)-	5.8493e-1(1.32e-2)-	7.8410e-1(1.11e-1)-	3.4118e-1(1.38e-3)=	3.5373e-1(1.59e-3)
	10	1.3490e+0(8.33e-2)-	6.5214e-1(1.78e-2)-	1.3628e+0(1.18e-1)-	4.5450e-1(5.10e-4)-	4.2690e-1(4.12e-3)
	15	1.5053e+0(7.64e-2)-	7.5203e-1(8.43e-3)-	1.9333e+0(3.48e-1)-	5.7137e-1(2.72e-3)-	5.3276e-1(4.13e-3)
DTLZ5	3	5.6927e-3(8.30e-4)+	4.0628e-2(5.35e-3)-	6.1386e-3(4.10e-4)=	3.2522e-2(1.17e-3)-	6.3719e-3(4.23e-4)
	5	3.1111e-1(2.57e-2)-	2.9915e-2(4.86e-3)+	7.1303e-2(1.83e-2)-	8.5261e-2(3.41e-3)-	3.6560e-2(6.96e-3)
	8	1.8356e+0(4.73e-1)-	2.9288e-2(5.87e-3)+	1.7165e-1(5.24e-2)-	1.2868e-1(9.95e-3)-	5.4626e-2(9.29e-3)
	10	1.7715e+0(7.11e-1)-	2.3843e-2(5.25e-3)+	1.2075e-1(3.69e-2)-	1.7173e-1(1.78e-2)-	4.8186e-2(8.22e-3)
	15	1.7061e+0(2.95e-1)-	2.7307e-2(5.83e-3)+	1.7347e-1(9.17e-2)-	1.9137e-1(3.52e-2)-	4.7407e-2(4.20e-3)
DTLZ6	3	4.2165e-3(4.46e-5)+	3.3835e-2(1.32e-5)-	6.9594e-3(9.60e-4)=	3.3198e-2(3.59e-4)-	6.8460e-3(6.24e-4)
	5	3.8651e+0(1.57e+0)-	3.2017e-2(6.18e-6)+	2.9603e-1(2.36e-1)-	8.6575e-2(1.18e-3)+	1.9242e-1(4.37e-2)
	8	9.6427e+0(1.81e-1)-	2.4404e-2(2.53e-4)+	4.0488e-1(2.42e-1)-	1.2025e-1(4.51e-4)+	2.5237e-1(5.01e-2)
	10	9.7001e+0(2.54e-1)-	2.4908e-2(5.63e-4)+	5.4698e-1(4.21e-1)-	1.1575e-1(2.34e-4)+	2.2817e-1(1.31e-1)
	15	9.7485e+0(1.53e-1)-	5.2210e-2(5.22e-6)+	6.6979e-1(5.08e-1)-	1.4965e-1(3.22e-2)+	2.2625e-1(2.45e-1)
DTLZ7	3	6.4645e-2(1.71e-3)+	1.4252e-1(2.36e-2)+	8.5361e-2(5.39e-3)+	1.3306e-1(1.85e-3)+	1.5317e-1(8.45e-2)
	5	5.2563e-1(2.25e-2)+	5.4125e-1(6.41e-2)+	3.6133e-1(8.47e-3)+	6.2844e-1(9.43e-3)+	2.0956e+0(7.85e-1)
	8	5.9164e+0(2.86e+0)-	1.3807e+0(1.35e-1)+	8.2129e-1(3.09e-2)+	1.7184e+0(3.57e-1)+	2.8717e+0(1.14e+0)
	10	1.6003e+1(5.01e+0)-	1.6656e+0(1.67e-1)+	1.5255e+0(1.27e-1)+	2.1238e+0(6.55e-1)+	9.5705e+0(3.19e+0)
	15	6.5560e+1(6.40e+0)-	2.9213e+0(3.20e-1)+	2.4919e+0(3.91e-1)+	2.8822e+0(4.36e-1)+	1.5214e+1(3.32e+0)
+/-/=		6/29/0	14/21/0	7/28/0	14/19/2	

where the symbols “+,” “-,” and “=” indicate that the result is significantly better, significantly worse, and statistically similar to that obtained by the proposed algorithm.

4.3. Performance Metrics. Performance evaluation indicators are used to measure the convergence and distribution performance of the solution set obtained by the algorithm. The commonly used performance evaluation indicators are as follows.

4.3.1. Generation Distance. Generation distance (GD) [47] is used to evaluate the convergence performance of the algorithm. The smaller the GD value of the solution set obtained by an algorithm, the better the convergence performance of the algorithm. In particular, when $GD = 0$, all the optimal solutions obtained by the algorithm are on the true Pareto front.

4.3.2. Spacing. Spacing (SP) [47] is used to reflect the uniformity of the solution set distribution obtained by the algorithm. The smaller the SP value, the more uniform the solution set distribution.

4.3.3. Inverted Generational Distance. The inverted generational distance (IGD) [48] is one of the most widely used indicators. It could provide comprehensive information about convergence and distribution performance of the algorithm by calculating the minimum distance sum between each point on the true PF and the calculated individual set. The smaller the IGD value, the better the quality of the solution set obtained by the algorithm. Particularly, if IGD is equal to 0, the obtained PF contains every point of the true PF.

In this paper, IGD is employed to evaluate the performance of the algorithm. Let S be a uniformly distributed subset selected from the true PF and S' is the optimal Pareto

TABLE 5: The mean and standard deviation of the IGD value of the proposed algorithm and the four recently comparative algorithms CMOPSO, MMOPSO, MPSOD, and dMOPSO on WFG for 3, 5, 8, 10, and 15 objective problems, where the best value for each test case is highlighted with a bold background.

Problem	M	CMOPSO	dMOPSO	MMOPSO	MPSOD	CMApSO
WFG1	3	1.5022e+0(8.42e-3)-	1.5274e+0(6.04e-3)-	4.3581e-1(4.58e-2)-	1.4777e+0(3.20e-2)-	3.6797e-1(6.06e-2)
	5	2.0348e+0(1.90e-2)-	2.0337e+0(1.12e-2)-	1.5417e+0(4.96e-2)=	1.9401e+0(2.34e-2)-	1.5343e+0(1.87e-1)
	8	2.6729e+0(5.37e-2)-	2.7149e+0(1.02e-2)-	2.3994e+0(4.93e-2)-	2.6159e+0(1.69e-2)-	1.9306e+0(9.28e-2)
	10	3.0866e+0(2.20e-2)-	3.1181e+0(2.13e-2)-	2.8852e+0(5.91e-2)-	3.0500e+0(2.39e-2)-	2.6358e+0(1.22e-1)
	15	4.0310e+0(3.21e-2)-	4.2140e+0(3.10e-2)-	3.8640e+0(4.63e-2)-	3.9929e+0(2.72e-2)-	3.7283e+0(7.91e-2)
WFG2	3	1.8125e-1(8.54e-3)-	3.7120e-1(1.97e-2)-	2.3566e-1(1.26e-2)-	1.9821e-1(4.74e-3)-	1.6900e-1(2.59e-3)
	5	6.4846e-1(2.79e-2)-	6.9837e-1(3.03e-2)-	7.8138e-1(4.76e-2)-	6.3753e-1(3.77e-2)-	4.6603e-1(6.91e-3)
	8	1.2562e+0(3.65e-2)-	1.4097e+0(5.78e-2)-	1.4180e+0(6.18e-2)-	1.1999e+0(3.65e-2)-	1.0521e+0(4.00e-2)
	10	1.5717e+0(5.30e-2)-	1.6534e+0(2.39e-2)-	1.5513e+0(3.78e-2)-	1.4113e+0(5.47e-2)-	1.1252e+0(3.35e-2)
	15	2.4815e+0(2.93e-1)-	2.6033e+0(6.95e-2)-	2.0643e+0(9.59e-2)-	1.9621e+0(4.47e-2)-	1.6533e+0(7.00e-2)
WFG3	3	1.9304e-1(1.56e-2)-	5.0349e-1(7.21e-2)-	1.0427e-1(1.24e-2)=	2.7890e-1(2.65e-2)-	1.0706e-1(7.68e-3)
	5	1.0736e+0(9.85e-2)-	7.9881e-1(5.77e-2)-	5.7070e-1(8.62e-2)-	8.3271e-1(4.10e-2)-	3.8881e-1(7.07e-2)
	8	2.2141e+0(1.14e-1)-	2.3638e+0(6.11e-1)-	1.2723e+0(1.52e-1)-	1.4738e+0(3.54e-2)-	5.1806e-1(8.01e-2)
	10	2.8920e+0(1.42e-1)-	4.1413e+0(2.61e-1)-	1.5555e+0(2.18e-1)-	1.8289e+0(5.14e-2)-	6.7278e-1(9.57e-2)
	15	4.9914e+0(6.64e-1)-	8.9915e+0(2.49e+0)-	2.8363e+0(7.44e-1)-	2.3692e+0(5.44e-2)-	9.6840e-1(7.37e-2)
WFG4	3	2.7079e-1(2.97e-3)-	3.2880e-1(1.19e-2)-	2.9431e-1(1.69e-2)-	2.6878e-1(3.91e-3)-	2.4216e-1(3.15e-3)
	5	1.0795e+0(2.11e-2)=	1.5872e+0(1.58e-1)-	1.2103e+0(3.12e-2)-	1.4754e+0(1.97e-2)-	1.1111e+0(1.17e-2)
	8	2.8804e+0(2.52e-2)=	7.4798e+0(3.60e-1)-	3.2326e+0(4.10e-2)-	4.7715e+0(9.97e-2)-	3.0092e+0(2.50e-2)
	10	4.2454e+0(6.57e-2)-	1.0133e+1(1.55e-1)-	4.5236e+0(3.79e-2)-	6.5960e+0(1.21e-1)-	4.2453e+0(2.07e-2)
	15	7.9002e+0(1.11e-1)-	1.5922e+1(5.08e-1)-	8.2037e+0(7.88e-2)-	9.0792e+0(3.48e-1)-	7.8890e+0(3.35e-1)
WFG5	3	2.6458e-1(8.21e-3)-	3.1650e-1(3.26e-2)-	2.8914e-1(1.08e-2)-	2.5515e-1(9.62e-4)-	2.3862e-1(3.81e-3)
	5	1.0560e+0(9.16e-3)+	1.3479e+0(3.99e-2)-	1.2652e+0(2.90e-2)-	1.5195e+0(7.72e-3)-	1.1126e+0(1.35e-2)
	8	2.9384e+0(3.29e-2)+	4.6484e+0(2.25e-1)-	3.4768e+0(7.43e-2)-	4.0103e+0(1.64e-1)-	3.1077e+0(2.64e-2)
	10	4.1166e+0(3.84e-2)=	6.8581e+0(1.97e-1)-	4.7322e+0(6.47e-2)-	5.4337e+0(1.23e-1)-	4.2940e+0(5.03e-2)
	15	7.6413e+0(1.14e-1)-	1.0395e+1(2.98e-1)-	8.3772e+0(1.16e-1)-	8.1415e+0(9.91e-2)-	7.1937e+0(9.64e-2)
WFG6	3	2.5950e-1(8.18e-3)-	2.9186e-1(1.76e-2)-	3.2614e-1(1.46e-2)-	2.8432e-1(1.11e-2)-	2.4022e-1(4.65e-3)
	5	1.1636e+0(3.08e-2)=	2.7199e+0(1.35e-1)-	1.2643e+0(1.98e-2)-	1.4793e+0(5.45e-2)-	1.1512e+0(7.89e-3)
	8	3.1267e+0(4.98e-2)-	8.4369e+0(4.82e-1)-	3.4341e+0(4.46e-2)-	4.6085e+0(3.11e-1)-	3.1219e+0(3.29e-2)
	10	4.3428e+0(3.50e-2)=	1.0817e+1(2.06e-1)-	4.6880e+0(4.17e-2)-	6.3171e+0(2.85e-1)-	4.3853e+0(5.38e-2)
	15	7.7888e+0(9.19e-2)-	1.6862e+1(2.16e-1)-	8.2301e+0(8.32e-2)-	8.9450e+0(2.52e-1)-	7.4678e+0(1.54e-1)
WFG7	3	2.3377e-1(3.70e-3)=	4.1408e-1(1.16e-2)-	2.8324e-1(1.06e-2)-	2.6031e-1(2.03e-3)-	2.3088e-1(4.14e-3)
	5	1.1356e+0(2.42e-2)=	2.1718e+0(1.67e-1)-	1.2244e+0(3.73e-2)-	1.4645e+0(1.92e-2)-	1.1329e+0(1.50e-2)
	8	2.9881e+0(2.59e-2)+	8.0059e+0(5.46e-1)-	3.4421e+0(4.91e-2)-	4.4225e+0(1.60e-1)-	3.1161e+0(2.14e-2)
	10	4.2298e+0(2.95e-2)+	1.0927e+1(2.66e-1)-	4.6792e+0(4.96e-2)-	6.0794e+0(8.89e-2)-	4.3651e+0(3.94e-2)
	15	7.6036e+0(8.04e-2)-	1.6080e+1(2.70e-1)-	8.2605e+0(7.27e-2)-	8.1167e+0(1.85e-1)-	7.2438e+0(1.31e-1)
WFG8	3	3.0896e-1(8.31e-3)-	5.2524e-1(1.86e-2)-	3.4639e-1(8.27e-3)-	2.9225e-1(5.07e-3)-	2.8088e-1(9.10e-3)
	5	1.2601e+0(2.36e-2)-	1.7185e+0(1.05e-1)-	1.2889e+0(2.74e-2)-	1.3809e+0(1.20e-2)-	1.1628e+0(6.08e-3)
	8	3.3381e+0(1.61e-2)-	7.7007e+0(3.06e-1)-	3.6005e+0(1.07e-1)-	3.8116e+0(4.78e-2)-	3.1612e+0(2.83e-2)
	10	4.5975e+0(8.74e-2)=	1.0607e+1(2.12e-1)-	4.8154e+0(9.00e-2)-	5.4392e+0(1.08e-1)-	4.4489e+0(3.53e-2)
	15	8.1480e+0(1.03e-1)-	1.5893e+1(3.03e-1)-	8.4762e+0(1.36e-1)-	8.8058e+0(2.00e-1)-	7.8886e+0(2.45e-1)
WFG9	3	2.2829e-1(2.82e-3)+	2.8053e-1(7.44e-3)-	2.9606e-1(1.42e-2)-	2.6147e-1(1.36e-2)-	2.4682e-1(9.13e-3)
	5	1.0789e+0(1.58e-2)=	1.8778e+0(1.26e-1)-	1.3602e+0(1.05e-1)-	1.4301e+0(2.11e-2)-	1.1105e+0(7.62e-3)
	8	3.2033e+0(5.01e-2)-	6.7459e+0(1.26e-1)-	3.7661e+0(8.50e-2)-	3.9769e+0(1.86e-1)-	3.0804e+0(2.19e-2)
	10	4.4527e+0(5.42e-2)-	9.5509e+0(2.31e-1)-	5.0222e+0(4.49e-2)-	5.3584e+0(1.73e-1)-	4.1480e+0(4.35e-2)
	15	7.9157e+0(4.31e-2)-	1.3970e+1(1.14e+0)-	8.5996e+0(1.19e-1)-	8.4536e+0(2.16e-1)-	7.4027e+0(9.57e-2)
+/-/=		8/29/8	0/45/0	0/43/2	0/45/0	

TABLE 6: Significance test between the proposed algorithm CMApSO and the comparative algorithms on the DTLZ test problems.

	CMOPSO	dMOPSO	MMOPSO	MPSOD	CMApSO
Rank first	3	8	4	5	15
Better (+)	5	14	6	13	
Same (=)	1	0	2	3	
Worse (-)	29	21	27	19	

TABLE 7: Significance test between the proposed algorithm CMaPSO and other comparison algorithms on the WFG test problems.

	CMOPSO	dMOPSO	MMOPSO	MPSOD	CMaPSO
Rank first	10	0	1	0	34
Better (+)	5	0	0	0	
Same (=)	12	0	2	0	
Worse (-)	28	45	43	45	

TABLE 8: Significance test between the proposed algorithm CMaPSO and the comparison algorithms on all test problems.

	CMOPSO	dMOPSO	MMOPSO	MPSOD	CMaPSO
Rank first	13	8	5	5	49
Better (+)	10	14	6	13	
Same (=)	13	0	4	3	
Worse (-)	57	66	70	64

FIGURE 3: Average IGD performance scores for different objectives and different test instances. (a) Average performance scores over all dimensions for different test cases. (b) Average performance scores of the test instances DTLZ for different number of objectives. (c) Average performance scores of the test instances WFG for different number of objectives. (d) Average performance scores over all test instances for different number of objectives.

FIGURE 4: All comparison algorithms have average performance scores for IGD on (a) DTLZ, (b) WFG, and (c) two test suites.

solution set obtained by the current algorithm. The IGD is defined as follows:

$$\text{IGD}(S, S') = \frac{\sum_{i=1}^{|S|} d(S, S'_i)}{|S|}, \quad (15)$$

where $|S|$ returns the number of solutions in set S and $d(S, S')$ computes the minimum Euclidean distance from $S_i \in S$ to the solutions of S' in the objective space.

In addition, to intuitively quantify the overall performance of the algorithm on different test problems, the

FIGURE 5: All comparison algorithms have average performance scores for IGD on (a) DTLZ, (b) WFG, and (c) two test suites.

performance score method is introduced. Suppose there are $\mathcal{L}$ algorithms $A_1, A_2, \dots, A_{\mathcal{L}}$, if algorithm A_i is significantly better than algorithm A_j according to IGD, δ_{ij} is

set as 1, and 0 otherwise. Then, for each algorithm A_i , the performance score $P(A_i)$ is computed by the following equation:

$$P(A_i) = \sum_{j=1, j \neq i}^{\mathcal{L}} \delta_{ij}. \quad (16)$$

The performance score indicates how many other algorithms are better than the corresponding algorithm on the test problems, and smaller values represent better performance than other algorithms.

4.4. Performance Comparison between CMaPSO and Four Multi/Many-Objective Particle Swarm Optimization Algorithms. This section we compare the proposed algorithm CMaPSO with four recently published multi/many-objective particle swarm optimization algorithms CMOPSO, MMOPSO, dMOPSO, and MPSOD on the 3, 5, 8, 10, and 15 objectives of DTLZ and WFG test suites, respectively. The comprehensive performance index IGD is used to evaluate the simulation results, and the experimental results can illustrate the performance of the algorithm on different test cases. Performance score is used to intuitively illustrate the performance of each algorithm. Tables 4 and 5 show the simulation results of the proposed algorithms and other comparative algorithms on DTLZ and WFG test suites, where the experimental results are reported by the mean and standard deviation of IGD values. Tables 6 and 7 lists the number of the best, better, similar, and worse performance of CMaPSO and comparative algorithms on DTLZ and WFG, respectively. Table 8 summarizes the experimental results in Tables 6 and 7 to illustrate the total significance test of each comparative algorithms on all test issues. Figure 3 depicts the trend of the average IGD performance scores of all comparative algorithms for different objectives and different test instances and links the trends of the proposed algorithm with blue polylines. Figure 4 shows the average performance scores of all comparative algorithms on DTLZ and WFG test suites using histogram.

Table 4 summarizes the simulation results of five multi/many-objective particle swarm optimization algorithms on the test set DTLZ, and the best value for each test case is highlighted with a bold background. From the table, it can be seen intuitively that the number of the best values of the proposed algorithm CMaPSO on test cases DTLZ1, DTLZ2, DTLZ3, and DTLZ4 accounts for the majority. The algorithm MMOPSO performs best on test case DTLZ7. The algorithm CMaPSO does not perform as well as the algorithm dMOPSO on the DTLZ5 and DTLZ6 test cases, and the number of best values of the algorithm dMOPSO on these two test cases is ahead of other algorithms. From Figure 3(a), the proposed algorithm CMaPSO has the best comprehensive IGD performance score on test cases DTLZ1, DTLZ2, and DTLZ3. In addition, we can find that the algorithm MPSOD has the best comprehensive IGD performance score on DTLZ2 and DTLZ4. dMOPSO achieves the best on DTLZ5 and DTLZ6. MMOPSO gains the best on DTLZ7. CMOPSO does not obtain the best performance on any test case of DTLZ. From Figure 3(b), it is obvious that the proposed algorithm CMaPSO has the best IGD performance score on the 5, 8, 10, and 15 objectives of DTLZ in all

comparative algorithms. And the performance score of the proposed algorithm on 3 objectives is not as good as the algorithm CMOPSO. The histogram of Figure 4(a) synthetically reflects IGD performance scores of all comparative algorithms on the DTLZ, which can effectively evaluate the performance of each algorithm. The proposed algorithm CMaPSO obtains the best IGD performance score.

Table 6 summarizes the significance test results of the proposed algorithm CMaPSO and the comparative algorithms on the test set DTLZ. There are a total of 35 test cases on the test set DTLZ. The proposed algorithm CMaPSO has 15 test cases ranking first, accounting for 42.86% of the total. The comparative algorithms CMOPSO, dMOPSO, MMOPSO, and MPSOD obtain 3, 8, 4, and 5 test cases ranking first, accounting for 8.57%, 22.86%, 11.43%, and 14.29% of the total, respectively. The algorithm MPSOD and dMOPSO perform best in the four employed comparative algorithms. From the results of the abovementioned data analysis, it can be concluded that the performance of the algorithm CMaPSO is better than the comparative algorithms on the DTLZ test suite.

Table 5 presents the simulation results of the five comparative algorithms on the test set WFG, and the best values of each test case is also marked in a bold background. Obviously, the algorithm CMaPSO gets the best performance on most test cases of WFG. The performance of the comparative algorithms on WFG is almost common, which is inferior to the proposed algorithm.

Figure 3(a) intuitively shows that CMaPSO obtains the best performance scores on test cases WFG1, WFG2, WFG3, WFG4, WFG5, WFG6, WFG8, and WFG9, but does not include the test case WFG7. The main reason is that although the algorithm CMaPSO achieves the best performance on the 3, 5, and 15 objectives of WFG7, the performance is not satisfactory on the 8 and 10 objectives, and the average performance score mainly emphasizes the average performance of the algorithm on test cases. Noticed that the algorithm CMOPSO only gets two the best performance on test case of WFG7, its average performance score is the best. As shown in Figure 3(c), the average performance score of the algorithm CMaPSO on the 3, 5, 8, 10, and 15 objectives of the WFG is better than the comparative algorithms adopted. The average performance score of the comparative algorithms on different test cases of WFG are obtained by comprehensive calculation, and the histogram is used to intuitively represent the scores. From Figure 4(b), algorithm CMaPSO obtained 0.94 points, which is the best score among all the comparative algorithms. Other algorithms CMOPSO, dMOPSO, MMOPSO, and MPSOD got 2.64, 3.65, 2.8, and 2.46 points, respectively.

In this paper, all the comparative algorithms are simulated on 45 test instances of WFG. From Table 7, the significance test of algorithm CMaPSO ranks first on 34 test cases, accounting for 75.56% of the total. The comparative algorithms CMOPSO, dMOPSO, MMOPSO, and MPSOD obtains 10, 0, 1, and 0 test cases ranking first, accounting for 22.22%, 0%, 2.22%, and 0% of the total, respectively. Among the four comparative algorithms, the

TABLE 9: The mean and standard deviation of the IGD value of the four advanced many-objective evolutionary optimization algorithms on the DTLZ test problems. The best value for each test case is highlighted with a bold background.

Problem	M	NSGAIII	MOMBIII	RPEA	MyODEMR	CMApSO
DTLZ1	3	2.0565e-2(1.04e-5)+	2.0619e-2(8.18e-5)+	1.6814e-1(5.04e-2)-	4.2157e-2(3.41e-2)-	2.2047e-2(2.92e-4)
	5	6.3383e-2(5.69e-5)+	6.6678e-2(9.94e-3)-	1.8383e-1(3.37e-2)-	8.7627e-2(2.86e-2)-	6.4419e-2(5.43e-3)
	8	1.1143e-1(1.83e-2)=	2.1921e-1(2.54e-2)-	2.4231e-1(2.91e-2)-	2.0962e-1(2.52e-2)-	1.0696e-1(5.96e-3)
	10	1.6571e-1(5.13e-2)-	2.1222e-1(3.29e-2)-	2.5486e-1(3.20e-2)-	3.3665e-1(6.77e-2)-	1.3077e-1(1.58e-2)
	15	1.8382e-1(1.10e-1)-	2.7085e-1(1.26e-2)-	2.6497e-1(4.63e-2)-	1.3754e-1(2.15e-2)=	1.3692e-1(8.31e-3)
DTLZ2	3	5.4464e-2(4.13e-7)+	5.4492e-2(7.81e-6)+	1.0882e-1(5.73e-3)-	6.9240e-2(3.21e-3)-	6.0866e-2(1.22e-3)
	5	1.9490e-1(8.21e-6)=	1.9522e-1(2.60e-4)=	2.8356e-1(3.09e-2)-	2.0436e-1(2.42e-3)-	1.9415e-1(1.20e-3)
	8	3.6070e-1(9.57e-2)=	3.2542e-1(7.00e-4)+	3.8254e-1(1.06e-2)-	3.5330e-1(4.43e-3)=	3.5837e-1(1.79e-3)
	10	5.0117e-1(7.63e-2)-	4.5504e-1(9.49e-4)-	4.4131e-1(8.14e-3)-	4.2515e-1(5.01e-3)=	4.2631e-1(2.22e-3)
	15	6.6601e-1(7.66e-2)-	7.9236e-1(6.58e-2)-	5.4126e-1(2.56e-3)-	5.6685e-1(1.61e-2)-	4.9904e-1(1.88e-3)
DTLZ3	3	5.4576e-2(1.14e-4)+	5.4600e-2(3.55e-5)+	2.0403e-1(2.78e-2)-	2.4784e-1(2.11e-1)-	6.2885e-2(6.68e-3)
	5	1.9545e-1(4.91e-4)+	1.9644e-1(1.09e-3)+	4.5246e-1(4.01e-2)-	6.4675e-1(3.12e-1)-	2.1712e-1(1.90e-2)
	8	6.6757e-1(1.04e+0)-	4.1902e-1(1.44e-1)+	6.8299e-1(6.66e-2)-	1.1522e+0(6.07e-2)-	6.2463e-1(2.48e-1)
	10	3.4566e+0(2.42e+0)-	6.5314e-1(1.20e-1)+	8.2396e-1(9.59e-2)+	1.2320e+0(1.66e-2)+	1.9282e+0(1.17e+0)
	15	4.5793e+1(2.46e+1)-	1.0840e+0(2.34e-2)+	1.0639e+0(2.51e-1)+	1.2840e+0(1.44e-2)-	1.1990e+0(7.43e-1)
DTLZ4	3	1.0318e-1(1.54e-1)+	1.0321e-1(1.54e-1)+	5.7672e-1(3.08e-1)-	6.9716e-2(1.65e-3)+	3.3993e-1(3.21e-1)
	5	2.4182e-1(9.90e-2)-	2.8891e-1(1.14e-1)-	3.8703e-1(2.15e-1)-	2.2383e-1(5.72e-3)-	1.9522e-1(9.00e-4)
	8	3.7941e-1(1.06e-1)-	3.6120e-1(4.19e-2)-	3.9606e-1(3.51e-2)-	3.7977e-1(5.46e-3)-	3.5373e-1(1.59e-3)
	10	4.7630e-1(4.28e-2)-	4.5749e-1(3.48e-4)-	4.4578e-1(6.26e-3)-	4.7513e-1(1.10e-2)-	4.2690e-1(4.12e-3)
	15	5.4388e-1(4.87e-2)-	6.0064e-1(6.02e-3)-	5.4767e-1(3.31e-3)-	6.2169e-1(2.24e-3)-	5.3276e-1(4.13e-3)
DTLZ5	3	1.3276e-2(1.68e-3)-	2.5140e-2(3.41e-5)-	2.0755e-2(4.05e-3)-	3.7309e-2(9.09e-3)-	6.3719e-3(4.23e-4)
	5	3.0252e-1(2.37e-1)-	2.7061e-1(5.20e-3)-	5.1907e-2(9.00e-3)-	8.8047e-1(5.26e-2)-	3.6560e-2(6.96e-3)
	8	3.7565e-1(1.07e-1)-	2.8819e-1(1.66e-1)-	8.3767e-2(3.01e-2)-	1.4984e+0(5.73e-2)-	5.4626e-2(9.29e-3)
	10	4.1017e-1(7.95e-2)-	6.7472e-1(1.10e-1)-	8.3944e-2(1.93e-2)-	1.5900e+0(9.84e-2)-	4.8186e-2(8.22e-3)
	15	9.6608e-1(2.40e-1)-	7.2006e-1(2.90e-2)-	1.1546e-1(3.40e-2)-	1.7978e+0(3.24e-1)-	4.7407e-2(4.20e-3)
DTLZ6	3	2.1027e-2(1.92e-3)-	2.5149e-2(1.70e-6)-	1.9479e-2(3.42e-3)-	6.3035e-2(5.54e-2)-	6.8460e-3(6.24e-4)
	5	3.3816e-1(1.04e-1)-	3.1434e-1(1.33e-5)-	1.3371e-1(3.60e-2)+	8.5385e-1(9.81e-2)-	1.9242e-1(4.37e-2)
	8	1.3884e+0(1.17e+0)-	5.1714e-1(1.32e-1)-	1.7717e-1(6.42e-2)+	1.1300e+0(1.61e-1)-	2.5237e-1(5.01e-2)
	10	2.9306e+0(1.62e+0)-	5.9585e-1(1.14e-1)-	2.2791e-1(6.45e-2)=	1.2125e+0(1.43e-1)-	2.2717e-1(1.31e-1)
	15	6.1288e+0(1.05e+0)-	6.7540e-1(6.14e-2)-	1.9664e-1(6.15e-2)+	1.4495e+0(6.06e-1)-	2.2625e-1(2.45e-1)
DTLZ7	3	7.6242e-2(3.24e-3)+	2.2087e-1(2.31e-1)-	2.9105e-1(2.30e-1)-	2.5739e-1(3.68e-2)-	1.5317e-1(8.45e-2)
	5	3.3576e-1(1.70e-2)+	4.4758e-1(7.53e-2)+	1.4291e+0(1.63e-1)+	8.7415e-1(6.57e-2)+	2.0956e+0(7.85e-1)
	8	8.1348e-1(4.32e-2)+	2.5140e+0(7.66e-1)+	2.3518e+0(3.34e-1)+	3.4193e+0(5.21e-1)-	2.8717e+0(1.14e+0)
	10	1.2525e+0(1.36e-1)+	4.7379e+0(7.11e-1)+	2.2184e+0(2.87e-1)+	5.3261e+0(4.22e-1)+	9.5705e+0(3.19e+0)
	15	3.5665e+0(3.34e-1)+	1.1085e+1(9.34e-2)+	2.4343e+0(8.72e-1)+	1.0521e+1(3.99e-1)+	1.5214e+1(3.32e+0)
+/-/=		11/21/3	13/21/1	9/25/1	5/27/3	

algorithm CMOPSO has the best performance on WFG. There are 5 test cases whose significance test is better than the algorithm CMApSO on the WFG test problems, 12 test cases is similar to the CMApSO, and 28 test cases is obviously inferior to the algorithm.

We have analyzed the performance of all the comparative algorithms on the DTLZ and WFG test suites, respectively. Then, we combine the two test suites and make a comprehensive analysis of the performance of all the comparative algorithms. From Table 8, the number of significant test results ranked first obtained by CMApSO, CMOPSO, dMOPSO, MMOPSO, and MPSOD on all test cases are 49, 13, 8, 5, and 5, accounting for 61.25%, 28.89%, 10%, 6.25%, and 6.25% of the total, respectively. As shown in Figure 3(d), the average performance score of CMApSO is better than the other comparative algorithms on the 3, 5, 8, 10, and 15 objectives of the test instances. The algorithm CMApSO achieves better performance than the algorithm CMOPSO which also adopts competitive mechanism. It can also be effectively illustrated from histogram Figure 4(c). Figures 5(a)–5(e) illustrates the

approximate Pareto front obtained by five comparative algorithms on the 15 objectives of DTLZ1. By comparing these figures, the result obtained by CMApSO is the closest to the true PF in the comparative algorithms.

4.5. Performance Comparison between CMApSO and Four Advanced Many-Objective Evolutionary Optimization Algorithms. In this section, the performance of the proposed algorithm CMApSO is verified by simulation experiments, which is compared with four recent evolutionary algorithms on DTLZ and WFG test suites. The comprehensive performance metric IGD is adopted to illustrate the experimental results. Tables 9 and 10 summarize the experimental results. Tables 11 and 12 statistics the number of the best, better, similar, and worse performance of CMApSO and comparative algorithms on DTLZ and WFG, respectively. Table 13 summarizes the experimental results in Tables 11 and 12 to illustrate the total significance test of each comparative algorithms on all test issues. Figure 6 illustrates the

TABLE 10: The mean and standard deviation of the IGD value of the four advanced many-objective evolutionary optimization algorithms on the WFG test problems. The best value for each test case is highlighted with a bold background.

Problem	M	NSGAIII	MOMBIII	RPEA	MyODEMR	CMApSO
WFG1	3	1.4828e-1(2.49e-3)+	1.6620e-1(5.47e-3)+	9.2744e-1(1.52e-1)-	1.3417e+0(1.82e-1)-	3.6797e-1(6.06e-2)
	5	6.4684e-1(3.78e-2)+	5.8287e-1(1.61e-1)+	1.0331e+0(2.28e-1)+	2.0874e+0(2.46e-1)-	1.5343e+0(1.87e-1)
	8	1.3976e+0(7.24e-2)+	1.3016e+0(1.42e-1)+	1.5068e+0(1.96e-1)+	3.3522e+0(4.90e-1)-	1.9306e+0(9.28e-2)
	10	2.0023e+0(7.95e-2)+	1.7141e+0(1.15e-1)+	1.6465e+0(4.47e-2)+	3.8284e+0(5.35e-1)-	2.6358e+0(1.22e-1)
	15	3.1092e+0(1.19e-1)+	2.3870e+0(1.30e-1)+	2.3827e+0(1.27e-1)+	4.6443e+0(2.96e-1)-	3.7283e+0(7.91e-2)
WFG2	3	1.6470e-1(6.72e-4)+	2.0194e-1(6.16e-3)-	3.6533e-1(3.55e-2)-	4.5895e-1(1.21e-1)-	1.6900e-1(2.59e-3)
	5	4.7149e-1(1.75e-3)-	5.7165e-1(1.42e-1)-	7.8005e-1(1.13e-1)-	8.2489e-1(1.70e-1)-	4.6603e-1(6.91e-3)
	8	1.2751e+0(3.16e-1)-	1.2626e+0(1.40e-1)-	1.2647e+0(1.04e-1)-	1.5215e+0(2.79e-1)-	1.0521e+0(4.00e-2)
	10	1.6707e+0(1.94e-1)-	2.0888e+0(4.73e-1)-	1.4623e+0(1.79e-1)-	1.5048e+0(8.71e-2)-	1.1252e+0(3.35e-2)
	15	2.1508e+0(1.51e-1)-	8.2290e+0(1.17e+0)-	2.2949e+0(4.78e-1)-	3.0521e+0(7.59e-1)-	1.6533e+0(7.00e-2)
WFG3	3	8.3210e-2(1.98e-2)+	8.7194e-2(4.77e-3)+	3.8867e-2(2.52e-3)+	4.4170e-1(8.55e-1)-	1.0706e-1(7.68e-3)
	5	4.1981e-1(7.22e-2)-	1.5880e+0(1.01e-1)-	4.7314e-2(3.24e-3)+	5.2690e+0(7.44e-2)-	3.8881e-1(7.07e-2)
	8	9.1391e-1(2.26e-1)-	8.0883e+0(1.11e-1)-	6.2806e-2(4.62e-3)+	8.3924e+0(1.29e-1)-	5.1806e-1(8.01e-2)
	10	2.6070e+0(1.52e+0)-	9.8077e+0(1.12e-1)-	7.4803e-2(1.78e-2)+	1.0188e+1(3.34e-1)-	6.7278e-1(9.57e-2)
	15	3.7180e+0(1.79e+0)-	1.5405e+1(1.42e-1)-	2.5292e-1(8.11e-2)+	1.5672e+1(1.76e-1)-	9.6840e-1(7.37e-2)
WFG4	3	2.2082e-1(3.05e-5)+	2.4583e-1(8.40e-3)=	4.2990e-1(5.53e-2)-	2.7243e-1(1.20e-2)-	2.4216e-1(3.15e-3)
	5	1.1738e+0(5.34e-4)-	1.4715e+0(2.57e-1)-	1.2874e+0(3.45e-2)-	1.1384e+0(1.71e-2)-	1.1111e+0(1.17e-2)
	8	3.1688e+0(5.54e-1)-	3.7620e+0(2.54e-1)-	3.0544e+0(5.59e-2)-	3.0160e+0(4.24e-2)=	3.0092e+0(2.50e-2)
	10	4.8174e+0(1.69e-1)-	7.1181e+0(6.87e-1)-	4.3969e+0(7.21e-2)-	4.4493e+0(8.51e-2)-	4.2453e+0(2.07e-2)
	15	8.8091e+0(4.82e-1)-	1.7970e+1(2.11e+0)-	7.8842e+0(6.43e-2)=	8.6756e+0(1.15e-1)-	7.8810e+0(3.35e-1)
WFG5	3	2.2990e-1(3.37e-5)+	2.4460e-1(6.10e-3)-	3.9181e-1(3.73e-2)-	2.9952e-1(9.80e-3)-	2.3862e-1(3.81e-3)
	5	1.1646e+0(3.35e-4)-	1.2602e+0(2.99e-2)-	1.2474e+0(4.90e-2)-	1.1418e+0(1.20e-2)=	1.1126e+0(1.35e-2)
	8	2.9408e+0(2.41e-3)+	3.5800e+0(7.16e-2)-	3.0665e+0(3.83e-2)+	2.9863e+0(2.87e-2)+	3.1077e+0(2.64e-2)
	10	4.7349e+0(8.84e-3)-	6.5255e+0(2.19e-1)-	4.4372e+0(8.84e-2)-	4.3173e+0(3.59e-2)=	4.2940e+0(5.03e-2)
	15	8.0003e+0(5.40e-2)-	2.5317e+1(1.22e+0)-	7.6767e+0(7.70e-2)-	8.0252e+0(3.24e-2)-	7.1937e+0(9.64e-2)
WFG6	3	2.2516e-1(1.33e-3)+	2.3983e-1(6.51e-3)=	4.7244e-1(8.63e-2)-	2.8904e-1(3.28e-2)-	2.4022e-1(4.65e-3)
	5	1.1664e+0(8.31e-4)-	1.4798e+0(2.15e-1)-	1.6724e+0(2.56e-1)-	1.1727e+0(2.88e-2)=	1.1512e+0(7.89e-3)
	8	2.9554e+0(4.23e-3)+	3.7423e+0(1.69e-2)-	3.4378e+0(1.56e-1)-	2.9837e+0(2.41e-2)+	3.1219e+0(3.29e-2)
	10	4.7823e+0(1.14e-2)-	7.1299e+0(5.60e-2)-	4.6581e+0(9.25e-2)-	4.4373e+0(5.43e-2)=	4.3853e+0(5.38e-2)
	15	1.1085e+1(5.47e-1)-	1.6092e+1(2.29e+0)-	7.8701e+0(9.14e-2)-	8.1375e+0(1.86e-1)-	7.4678e+0(1.54e-1)
WFG7	3	2.2106e-1(6.94e-5)+	2.3205e-1(1.00e-2)=	5.3389e-1(1.02e-1)-	2.8956e-1(1.13e-2)-	2.3088e-1(4.14e-3)
	5	1.1751e+0(4.29e-4)-	1.3428e+0(2.36e-1)-	1.6856e+0(1.46e-1)-	1.1859e+0(1.59e-2)-	1.1329e+0(1.50e-2)
	8	2.9697e+0(6.98e-3)+	3.7402e+0(1.59e-2)-	3.5080e+0(1.56e-1)-	3.0879e+0(4.12e-2)=	3.1161e+0(2.14e-2)
	10	4.8138e+0(3.06e-2)-	6.7608e+0(3.78e-1)-	4.6440e+0(1.67e-1)-	4.5280e+0(5.27e-2)-	4.3651e+0(3.94e-2)
	15	1.0458e+1(7.63e-1)-	1.4015e+1(2.08e+0)-	7.9997e+0(8.23e-2)-	8.3747e+0(3.39e-2)-	7.2438e+0(1.31e-1)
WFG8	3	2.3431e-1(1.40e-3)+	2.8959e-1(2.51e-3)-	4.6968e-1(4.60e-2)-	2.9584e-1(5.32e-3)-	2.8088e-1(9.10e-3)
	5	1.1461e+0(6.16e-4)=	2.9829e+0(2.30e-2)-	1.5205e+0(6.39e-2)-	1.1519e+0(1.79e-2)=	1.1628e+0(6.08e-3)
	8	3.4504e+0(5.59e-1)-	3.8210e+0(1.16e-1)-	3.4439e+0(1.78e-1)-	3.0529e+0(3.78e-2)+	3.1612e+0(2.83e-2)
	10	5.0070e+0(3.78e-1)-	7.2435e+0(8.51e-2)-	4.8263e+0(1.69e-1)-	5.4932e+0(2.10e-1)-	4.4489e+0(3.53e-2)
	15	9.3500e+0(5.01e-1)-	1.9697e+1(2.05e+0)-	8.0531e+0(1.17e-1)-	1.0313e+1(9.71e-2)-	7.8886e+0(2.45e-1)
WFG9	3	2.3336e-1(1.44e-2)+	2.7000e-1(1.50e-2)-	3.9451e-1(5.41e-2)-	2.9471e-1(7.23e-3)-	2.4682e-1(9.13e-3)
	5	1.1330e+0(5.26e-3)=	2.6498e+0(1.75e-1)-	1.2971e+0(1.14e-1)-	1.1141e+0(2.39e-2)=	1.1105e+0(7.62e-3)
	8	2.9408e+0(1.80e-2)+	3.7030e+0(3.45e-2)-	3.5023e+0(2.83e-1)-	2.9853e+0(3.12e-2)+	3.0804e+0(2.19e-2)
	10	4.5640e+0(1.61e-2)-	7.0156e+0(1.15e-1)-	4.9314e+0(4.81e-1)-	4.3104e+0(3.19e-2)-	4.1480e+0(4.35e-2)
	15	7.9258e+0(1.66e-1)-	2.6021e+1(8.11e-1)-	8.3650e+0(2.09e-1)-	8.1817e+0(6.41e-2)-	7.4027e+0(9.57e-2)
+/-/=		17/25/3	6/36/3	10/34/1	4/33/8	

TABLE 11: Significance test between CMApSO and other comparison algorithms on the DTLZ test problems.

	NSGAIII	MOMBIII	RPEA	MyODEMR	CMApSO
Rank first	9	3	5	2	16
Better (+)	11	13	9	5	
Same ($\approx$)	3	1	1	3	
Worse (-)	21	21	25	27	

TABLE 12: Significance test between CMaPSO and other comparison algorithms on the WFG test problems.

	NSGAIII	MOMBIII	RPEA	MyODEMR	CMaPSO
Rank first	11	2	7	3	22
Better (+)	17	6	10	4	
Same ($\approx$)	3	3	1	8	
Worse (-)	25	36	34	33	

TABLE 13: Significance test between CMaPSO and the comparison algorithms on the all test problems.

	NSGAIII	MOMBIII	RPEA	MyODEMR	CMaPSO
Rank first	20	5	12	5	38
Better (+)	28	19	19	9	
Same ($\approx$)	6	4	2	11	
Worse (-)	46	57	59	60

FIGURE 6: Continued.

FIGURE 6: The approximate Pareto fronts obtained by (a) CMaPSO, (b) CMOPSO, (c) dMOPSO, (d) MMOPSO, (e) MPSOD, (f) MOBIII, (g) MyODEMR, (h) NSGA-III, and (i) RPEA on DTLZ1 (15-objective) are shown by a parallel coordinate.

average performance scores of all comparative algorithms for different objectives and test problems. Figure 7 shows the average performance scores of all comparative algorithms on DTLZ and WFG and the two test suites, respectively.

From Table 9, the algorithm CMaPSO has the best performance on DTLZ1, DTLZ4, and DTLZ5 test problems and achieves the best performance on 8, 10, and 15 objectives. The proposed algorithm does not get most of the best performance on the 3, 5, 8, 10, and 15 objectives of the test problem DTLZ2 and only performs best on the 5 and 15 objectives, but it still achieves the best performance by a slight advantage compared with the comparative algorithms. From Figure 6(a), the average performance score of CMaPSO on test cases DTLZ1, DTLZ2, DTLZ4, and DTLZ5 is the best. And From Figure 6(b), it is obviously observed that the best average performance score of CMaPSO for different objective numbers on test suite DTLZ is 5, 8, and 10 objectives. The comparative algorithm NSGA-III has the best performance on DTLZ7 test problem and obtains the best performance on 3, 5, 8, and 10 objectives, respectively. Notice that the proposed algorithm has the worst performance on DTLZ7 compared with the other four comparative algorithms. The comparative algorithms NSGA-III and MOBIII achieve the same number of best performances on DTLZ3, where NSGA-III obtains the best performance on 3 and 5 objectives, while MOBIII gains the best performance on the 8

and 10 objectives of the test problem DTLZ3. The algorithm RPEA gets the best performance on the 5, 8, and 15 objectives of the test problem DTLZ6.

Table 11 summarizes the number of the best, better, similar, and worse performance of CMaPSO and comparative algorithms on DTLZ. From Table 9, there are a total of 35 test cases. And the proposed algorithm has 16 test cases ranking first, accounting for 45.7% of the total from Table 11. In addition, it can be clearly observed from Figure 7(a) that the proposed algorithm has the best average performance score for different objective numbers on the test suite DTLZ. The comparative algorithms NSGA-III, MOBIII, RPEA, and MyODEMR achieves 9, 3, 5, and 2 test cases ranking first, accounting for 25.7%, 8.5%, 14.2%, and 5.7% of the total, respectively. The algorithm MOBIII performs best in all the comparative algorithms. A total of 13 test cases are significantly better than the algorithm CMaPSO on the DTLZ, but 21 test instances are obviously worse than CMaPSO. From the results of data analysis, it can be concluded that the performance of the algorithm CMaPSO is better than that of the comparative algorithms on the DTLZ test suite.

From Table 10, the algorithm CMaPSO has the best performance on WFG2, WFG4, WFG5, WFG6, WFG7, and WFG9 test problems and basically achieves the best performance on 5, 10, and 15 objectives. The proposed algorithm and the comparative algorithm NSGA-III have similar performance on WFG8 test problem, where NSGA-III gets the best

FIGURE 7: Average IGD performance scores for different objectives and different test instances. (a) Average performance scores over all dimensions for different test cases. (b) Average performance scores of the test instances DTLZ for different number of objectives. (c) Average performance scores of the test instances WFG for different number of objectives. (d) Average performance scores over all test instances for different number of objectives.

performance on 3 and 5 objectives, while CMaPSO has the best performance on 10 and 15 objectives. From Figure 6(a), it can be intuitively observed that the proposed algorithm has the best performance scores on the test cases WFG2, WFG4, WFG5, WFG6, WFG7, WFG8, and WFG9. Moreover, from Figure 6(c), the average performance score of the proposed algorithm is the best on the 5, 10, and 15 objectives of WFG test suite. The performance of RPEA ranks first on the test problem WFG3, which is superior to other algorithms. Algorithms MOBIII and RPEA get the same number of best performances on test problem WFG1. Notice that the algorithm CMaPSO performs worse on the test problem WFG1 than the comparative algorithms NSGA-III, MOBIII, and RPEA.

Table 12 statistics the number of the best, better, similar, and worse performance of CMaPSO and the comparative algorithms NSGA-III, MOBIII, RPEA, and MyODEMR on the test suite WFG. The performance of all the adopted algorithms are evaluated by 45 test cases with 3, 5, 8, 10, and 15 objectives on the WFG test problem. The proposed algorithm has won 22 first-ranked test cases, accounting for 48.9% of the total. Figure 7(b) shows the average performance score of all the comparative algorithms on the test suite WFG, where the proposed algorithm gets the best score. The algorithms NSGA-III, MOBIII, RPEA, and MyODEMR achieve 11, 2, 7, and 3 rank first test cases on the WFG test suite, accounting for 22%, 4.4%, 15.6%, and 6.7%

TABLE 14: The mean and standard deviation of the IGD value of the proposed algorithms and the four recently comparative algorithms CMOPSO, MMOPSO, MPSOD, and dMOPSO on UF1-9, where the best value for each test case is highlighted with a bold background.

Problem	M	CMOPSO	dMOPSO	MMOPSO	MPSOD	CMApSO
UF1	2	$5.7138e-2$ ($1.83e-2$) ⁺	$3.7540e-1$ ($7.68e-2$) ⁻	$5.9788e-2$ ($1.53e-2$) ⁺	$5.1749e-2$ ($1.34e-2$)⁺	$1.3331e-1$ ($3.73e-2$)
UF2	2	$2.9899e-2$ ($1.04e-2$) ⁺	$9.2730e-2$ ($1.52e-2$) ⁻	$2.4096e-2$ ($2.37e-3$) ⁺	$1.8998e-2$ ($2.46e-3$)⁺	$6.3018e-2$ ($1.76e-2$)
UF3	2	$1.2699e-1$ ($3.38e-2$) ⁺	$3.1958e-1$ ($5.65e-3$) ⁼	$1.2233e-1$ ($4.01e-2$)⁺	$1.2517e-1$ ($1.90e-2$) ⁺	$3.1369e-1$ ($2.23e-2$)
UF4	2	$8.1806e-2$ ($8.02e-3$) ⁻	$1.1693e-1$ ($5.52e-3$) ⁻	$4.5258e-2$ ($2.98e-3$) ⁻	$4.4709e-2$ ($1.32e-3$) ⁼	$4.4248e-2$ ($2.67e-3$)
UF5	2	$3.8937e-1$ ($1.25e-1$)⁺	$2.1659e+0$ ($2.83e-1$) ⁻	$5.1412e-1$ ($2.03e-1$) ⁺	$5.3157e-1$ ($1.15e-1$) ⁺	$1.1908e+0$ ($2.65e-1$)
UF6	2	$2.2563e-1$ ($1.06e-1$) ⁼	$1.3225e+0$ ($3.58e-1$) ⁻	$3.4226e-1$ ($1.48e-1$) ⁻	$2.2916e-1$ ($5.69e-2$) ⁼	$2.2525e-1$ ($1.68e-1$)
UF7	2	$7.6681e-2$ ($1.27e-1$) ⁺	$3.0330e-1$ ($7.88e-2$) ⁻	$1.3358e-1$ ($1.18e-1$) ⁺	$1.9950e-2$ ($3.37e-3$)⁺	$2.3489e-1$ ($1.84e-1$)
UF8	3	$6.8855e-1$ ($9.86e-2$) ⁻	$3.2944e-1$ ($3.23e-2$) ⁻	$2.4841e-1$ ($7.90e-2$) ⁻	$2.2823e-1$ ($6.93e-2$) ⁼	$2.2780e-1$ ($5.92e-3$)
UF9	3	$9.4259e-1$ ($1.61e-1$) ⁻	$6.2794e-1$ ($4.36e-2$) ⁻	$3.4257e-1$ ($6.32e-2$) ⁺	$2.4705e-1$ ($5.94e-2$)⁺	$5.0482e-1$ ($6.16e-2$)
+/-/=		5/3/1	0/7/2	6/3/0	6/0/3	

of the total, respectively. The algorithm NSGA-III performs best in the comparative algorithms MOBII, RPEA, and MyODEMR. A total of 17 test cases are significantly better than the algorithm CMApSO on the WFG test problem, but most of the test cases are inferior to the proposed algorithm. We can summarize the performance of the algorithm on the WFG test suite, which is obviously better than the employed comparative algorithm.

Table 13 summarizes the number of the best, better, similar, and worse performance of CMApSO and the comparative algorithms NSGA-III, MOBIII, RPEA, and MyODEMR on the test suites DTLZ and WFG. From the table, the performance of the proposed algorithm CMApSO ranks first on 38 test cases, accounting for almost half of the total. Figure 6(d) illustrates the comprehensive average performance scores of all the comparative algorithms for different objective numbers on the test suites DTLZ and WFG, where the proposed algorithm achieves the best score on 5, 8, 10, and 15 objectives. And Figure 7(c) presents the average performance scores of all the comparative algorithms on the test suites DTLZ and WFG, and the proposed algorithm obtains the best score. The performance of algorithms NSGA-III, MOBIII, RPEA, and MyODEMR has achieved 20, 5, 12, and 5 test cases ranking first, accounting for 25%, 6.25%, 15%, and 6.25% of the total. The performance of NSGA-III is the best among the employed comparative algorithms. However, the performance algorithm NSGA-III on most test cases is inferior to the proposed algorithm CMApSO. From all the abovementioned discussions, we can draw the conclusion that the performance of the proposed algorithm is better than the adopted comparative algorithms on the DTLZ and WFG test suits. Figures 5(f)–5(i) presents the approximate PF obtained by five many-objective evolutionary optimization algorithms on the 15 objectives of DTLZ1. Obviously, most of the comparative algorithms can converge to the true Pareto front, but the diversity of algorithm CMApSO is better than the other algorithms.

4.6. Experiments of All Comparison Algorithms on the Test Set UF1-9. To sum up, we have verified the performance of all comparison algorithms on the many-objective test problems of the test suites DTLZ and WFG, as well as on the 3-objective test problem. Next, we use the complex test cases UF1-9 [44] to further prove the performance of the proposed algorithm on the 2 and 3 objective test problems. Tables 14 and 15 present the mean and standard deviation of the IGD values of the proposed algorithm and four recent multi-objective particle swarm optimization algorithms and four advanced many-objective evolutionary optimization algorithms on the test set UF1-9. The best value for each test case is highlighted with a bold background.

In Table 14, the algorithm MPSOD obtained the best performance on the four test problems, and the proposed algorithm CMApSO was the second, and the best performance was gained on the three test problems. CMOPSO and MMOPSO got one. Through comprehensive comparison and analysis, the performance of MPSOD is the best, MMOPSO and CMOPSO are better, and the comprehensive performance of CMOPSO is not as good as these three algorithms. From Table 15, the algorithms NSGAIII and MyODEMR achieved the best performance on three test problems, and the proposed algorithm CMApSO obtained the best performance on two test problems. Although the algorithm MOMBIII got the best performance on one test question, its comprehensive performance was still excellent. Through the Wilcoxon rank sum test analysis, it is easy to find that NSGAIII performs best, MOMBIII performs better, and CMApSO and MyODEMR perform similarly. In order to be able to observe the experimental results more visually, Figure 8 shows the experimental results of all comparison algorithms on test case UF1-9 by adopting the box graph. The box graph of these test problems can more vividly explain the performance of all comparison algorithms. It is easy to conclude that although the performance of the algorithm CMApSO on the test problem UF1-9 is not the

TABLE 15: The mean and standard deviation of the IGD value of the four advanced many-objective evolutionary optimization algorithms on the UF1-9 test problems. The best value for each test case is highlighted with a bold background.

Problem	M	NSGAIII	MOMBIII	RPEA	MyODEMR	CMApSO
UF1	2	$1.3356e-1$ ($2.19e-2$)=	$9.4162e-2$ ($1.62e-2$)+	$1.3086e-1$ ($3.66e-2$)=	$6.2656e-2$ ($3.88e-2$)+	$1.3331e-1$ ($3.73e-2$)
UF2	2	$3.7800e-2$ ($7.34e-3$)+	$4.7754e-2$ ($1.52e-2$)+	$6.5567e-2$ ($2.89e-2$)–	$6.1227e-2$ ($4.67e-2$)=	$6.3018e-2$ ($1.76e-2$)
UF3	2	$2.1073e-1$ ($4.71e-2$)+	$3.1511e-1$ ($6.13e-2$)=	$2.9777e-1$ ($3.22e-2$)=	$1.9565e-1$ ($6.36e-2$)+	$3.1369e-1$ ($2.23e-2$)
UF4	2	$4.6032e-2$ ($1.33e-3$)–	$4.6501e-2$ ($2.61e-3$)–	$7.5297e-2$ ($1.17e-2$)–	$5.1650e-2$ ($2.55e-3$)–	$4.4248e-2$ ($2.67e-3$)
UF5	2	$2.7662e-1$ ($7.32e-2$)+	$2.9237e-1$ ($1.00e-1$)+	$3.2235e-1$ ($1.15e-1$)+	$3.1381e-1$ ($6.56e-2$)+	$1.1908e+0$ ($2.65e-1$)
UF6	2	$1.7038e-1$ ($1.09e-1$)+	$1.7013e-1$ ($9.07e-2$)+	$2.0888e-1$ ($1.11e-1$)+	$2.4345e-1$ ($1.21e-1$)–	$2.2525e-1$ ($1.68e-1$)
UF7	2	$1.3985e-1$ ($1.35e-1$)+	$2.2897e-1$ ($1.76e-1$)=	$2.4553e-1$ ($1.39e-1$)–	$5.2280e-2$ ($6.40e-2$)+	$2.3489e-1$ ($1.84e-1$)
UF8	3	$4.9871e-1$ ($1.03e-1$)–	$4.3170e-1$ ($1.26e-1$)–	$5.2668e-1$ ($4.79e-2$)–	$2.9866e-1$ ($4.61e-2$)–	$2.2780e-1$ ($5.92e-3$)
UF9	3	$3.0157e-1$ ($8.61e-2$)+	$3.6046e-1$ ($1.10e-1$)+	$5.0350e-1$ ($9.58e-2$)=	$5.8142e-1$ ($1.31e-1$)–	$5.0482e-1$ ($6.16e-2$)
+/-/=		6/2/1	5/2/2	2/4/3	4/4/1

FIGURE 8: Continued.

FIGURE 8: The performance of CMOPSO, dMOPSO, MMOPSO, MPSOD, NSGAIII, MOBIII, RPEA, MyODEMR, and CMaPSO on the test problem UF1-9 is presented by using the box plot.

best among all the comparison algorithms, and it can effectively solve most complex multiobjective optimization problems.

4.7. Comparison of Running Time between Algorithms CMaPSO and CMOPSO. The algorithm CMaPSO and CMOPSO all employ the competition-based learning method to update the speed and position of the particles. In order to evaluate the performance of the proposed algorithm more comprehensively, it is necessary to compare the

running time of the algorithm. From Table 16, it can be intuitively observed that the best values of all gray background are obtained by the algorithms CMaPSO. In this paper, CMaPSO can be regarded as the generalization of CMOPSO. The data in Table 16 shows that CMaPSO has shorter running time than CMOPSO on the multi/many-objective optimization problems. In addition, from Table 17, the significance test data of algorithm running time also shows that the proposed algorithm is more efficient than the algorithm CMOPSO.

TABLE 16: The running time of algorithm CMapSO and algorithm CMOPSO is on the test suites DTLZ and WFG test suites, and the best value is highlighted in bold background.

Problem	M	CMOPSO	CMapSO
DTLZ1	3	$2.0325e+1(6.40e+0)-$	$1.3247e+1(7.85e-1)$
	5	$9.8638e+1(4.97e-1)-$	$1.3337e+1(1.19e+0)$
	8	$1.7133e+2(8.88e+1)-$	$1.5771e+1(9.33e-1)$
	10	$9.6345e+2(8.72e+1)-$	$2.0261e+1(1.96e+0)$
	15	$1.0586e+3(3.81e+1)-$	$1.9567e+1(1.75e+0)$
DTLZ2	3	$1.0213e+2(1.16e+1)-$	$1.3558e+1(9.22e-1)$
	5	$1.1368e+2(7.83e+0)-$	$1.3941e+1(1.33e+0)$
	8	$7.2410e+2(3.37e+1)-$	$1.5809e+1(1.15e+0)$
	10	$1.3434e+3(1.17e+2)-$	$2.4512e+1(2.05e+0)$
	15	$1.1525e+3(4.03e+1)-$	$2.2854e+1(2.69e+0)$
DTLZ3	3	$1.3356e+1(3.38e+0)=$	$1.4022e+1(9.61e-1)$
	5	$1.2529e+1(2.35e+0)=$	$1.3185e+1(5.12e-1)$
	8	$4.0956e+2(1.88e+1)-$	$1.4730e+1(8.98e-1)$
	10	$1.0847e+3(7.70e+1)-$	$2.0134e+1(1.48e+0)$
	15	$1.0725e+3(2.48e+1)-$	$1.9188e+1(2.33e+0)$
DTLZ4	3	$8.7389e+1(8.57e+0)-$	$2.1440e+1(1.72e+0)$
	5	$1.8360e+2(1.82e+1)-$	$1.2812e+1(1.03e+0)$
	8	$6.9939e+2(3.26e+1)-$	$1.6309e+1(1.06e+0)$
	10	$1.3385e+3(7.43e+1)-$	$2.4079e+1(1.69e+0)$
	15	$1.0910e+3(6.83e+1)-$	$2.2716e+1(2.43e+0)$
DTLZ5	3	$5.5077e+1(2.27e+0)-$	$2.0834e+1(1.82e+0)$
	5	$9.6019e+1(9.49e+0)-$	$1.5241e+1(2.38e+0)$
	8	$6.7374e+2(4.92e+1)-$	$1.5641e+1(1.02e+0)$
	10	$1.2522e+3(5.87e+1)-$	$2.2997e+1(8.48e-1)$
	15	$1.0647e+3(6.33e+1)-$	$2.0185e+1(2.56e+0)$
DTLZ6	3	$7.4785e+1(3.08e+0)-$	$2.1424e+1(1.94e+0)$
	5	$1.7705e+2(6.23e+1)-$	$1.2330e+1(9.39e-1)$
	8	$7.4953e+2(3.44e+1)-$	$1.5401e+1(9.24e-1)$
	10	$1.3524e+3(1.51e+2)-$	$2.0092e+1(1.94e+0)$
	15	$1.1064e+3(4.43e+1)-$	$1.9240e+1(2.52e+0)$
DTLZ7	3	$4.4794e+1(6.82e+0)-$	$2.4811e+1(2.15e+0)$
	5	$1.5473e+2(1.08e+1)-$	$1.7914e+1(1.61e+0)$
	8	$5.5592e+2(3.54e+1)-$	$2.4724e+1(3.07e+0)$
	10	$1.0898e+3(7.96e+1)-$	$1.9551e+1(2.98e+0)$
	15	$1.1207e+3(1.96e+1)-$	$2.8312e+1(1.30e+1)$
WFG1	3	$7.5295e+1(5.78e+0)-$	$2.0274e+1(2.07e+0)$
	5	$1.6252e+2(1.93e+1)-$	$1.5829e+1(2.06e+0)$
	8	$2.6396e+2(2.69e+1)-$	$1.7167e+1(1.95e+0)$
	10	$4.0521e+2(8.53e+1)-$	$1.8244e+1(3.50e+0)$
	15	$5.6740e+2(1.73e+1)-$	$1.8881e+1(1.99e+0)$
WFG2	3	$3.8167e+1(2.22e+0)-$	$1.3256e+1(1.46e+0)$
	5	$2.3355e+2(1.66e+1)-$	$1.4563e+1(1.38e+0)$
	8	$5.7765e+2(1.90e+1)-$	$1.7866e+1(1.28e+0)$
	10	$1.2167e+3(3.60e+2)-$	$2.2871e+1(2.67e+0)$
	15	$9.5575e+2(2.00e+1)-$	$2.2693e+1(1.62e+0)$
WFG3	3	$4.2085e+1(1.82e+0)-$	$1.2114e+1(9.71e-1)$
	5	$2.7558e+2(4.94e+0)-$	$1.3580e+1(1.13e+0)$
	8	$7.6980e+2(2.19e+1)-$	$2.1984e+1(1.91e+0)$
	10	$1.4993e+3(2.84e+2)-$	$3.5400e+1(1.10e+1)$
	15	$1.1370e+3(2.93e+1)-$	$2.6364e+1(3.04e+0)$
WFG4	3	$1.0979e+2(6.46e+0)-$	$1.4193e+1(1.04e+0)$
	5	$3.4257e+2(1.10e+1)-$	$1.4548e+1(1.12e+0)$
	8	$8.9442e+2(5.22e+1)-$	$2.2418e+1(1.48e+0)$
	10	$1.4845e+3(1.63e+2)-$	$2.7789e+1(3.35e+0)$
	15	$1.1123e+3(5.59e+1)-$	$2.7387e+1(4.26e+0)$

TABLE 16: Continued.

Problem	M	CMOPSO	CMApSO
WFG5	3	$1.3188e+2(6.75e+0)-$	$1.4279e+1(2.29e+0)$
	5	$3.7246e+2(2.76e+1)-$	$1.4887e+1(2.55e+0)$
	8	$9.2150e+2(1.14e+2)-$	$1.8684e+1(1.16e+0)$
	10	$1.4429e+3(1.25e+2)-$	$2.9439e+1(4.44e+0)$
	15	$1.0793e+3(4.72e+1)-$	$2.4731e+1(3.08e+0)$
WFG6	3	$9.3599e+1(4.71e+0)-$	$1.3490e+1(1.75e+0)$
	5	$2.9378e+2(1.98e+1)-$	$1.4815e+1(1.63e+0)$
	8	$7.7411e+2(3.71e+1)-$	$1.8815e+1(1.32e+0)$
	10	$1.4829e+3(1.28e+2)-$	$3.0236e+1(4.03e+0)$
	15	$1.0765e+3(5.49e+1)-$	$2.5458e+1(2.88e+0)$
WFG7	3	$1.1370e+2(1.30e+1)-$	$1.3340e+1(8.87e-1)$
	5	$3.2237e+2(2.06e+1)-$	$1.5460e+1(1.01e+0)$
	8	$8.9250e+2(1.36e+2)-$	$1.6997e+1(1.76e+0)$
	10	$1.3971e+3(2.05e+2)-$	$2.9195e+1(6.42e+0)$
	15	$1.1366e+3(1.01e+2)-$	$2.5141e+1(2.82e+0)$
WFG8	3	$8.7390e+1(8.17e+0)-$	$1.4061e+1(2.43e-1)$
	5	$3.0043e+2(1.38e+1)-$	$1.4979e+1(1.15e+0)$
	8	$7.1609e+2(3.23e+1)-$	$1.7472e+1(1.58e+0)$
	10	$1.3283e+3(1.52e+2)-$	$2.6407e+1(2.45e+0)$
	15	$1.1042e+3(1.92e+1)-$	$2.6856e+1(2.65e+0)$
WFG9	3	$1.0228e+2(7.44e+0)-$	$1.4616e+1(2.02e-1)$
	5	$2.8967e+2(1.54e+1)-$	$1.5815e+1(1.41e+0)$
	8	$8.5906e+2(3.19e+2)-$	$1.6439e+1(1.29e+0)$
	10	$1.2128e+3(5.55e+1)-$	$2.7584e+1(2.53e+0)$
	15	$1.2783e+3(2.52e+2)-$	$2.5133e+1(2.04e+0)$

TABLE 17: Significance test between CMApSO and algorithm CMOPSO on the DTLZ and WFG test suites.

	CMOPSO	CMApSO
Rank first	0	78
Better (+)	0	
Same ($\approx$)	2	
Worse (-)	78	

5. Conclusion

In this paper, we have proposed a new multi/many-objective particle swarm optimization algorithms, termed as CMApSO, which adopts competition-based learning to update the velocity and position of particles. The algorithm can be considered as the generalization of the algorithm CMOPSO. The main reasons are as follows: firstly, they all adopt the same learning method to update the speed and position of particles. Secondly, the algorithm CMOPSO does not perform well in solving many-objective optimization problems. Thirdly, the running time of the algorithm CMOPSO is relatively time-consuming. We mainly modify the environmental selection mechanism of CMOPSO and adopt the maximum and minimum angle between ordinary and extreme individuals to select excellent individuals to enter the next generation population. In order to tackle some complex problems more effectively, we use θ -dominance sorting and reference points regeneration strategy to further improve the algorithm.

To evaluate the competitiveness of the proposed algorithms, the proposed algorithm CMApSO is compared with four recent multi/many-objective particle swarm

optimization algorithms and compared with four many-objective evolutionary optimization algorithms, which have been empirically compared on eighty-nine standard benchmark test cases. The experimental results indicate that the algorithm CMApSO has good robustness on most test problems and can effectively deal with multi/many-objective optimization problems. However, from the simulation comparison results, we found that the performance of the algorithm CMApSO on the test case DTLZ7 is disappointing. In order to further improve the performance of the algorithm, we will continue to study the proposed algorithm CMApSO and apply it to solve engineering problems in the future.

Data Availability

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

Consent

Informed consent was obtained from all individual participants included in the study.

Conflicts of Interest

All authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key Research and Development Program Projects of China (Grant no. 2018YFC1504700) and project of Natural Science Foundation in Shaanxi Province (Grant no. 2018JM6029)

Supplementary Materials

The mathematical expressions of the test sets DTLZ, WFG, and UF are presented in supplementary materials. (*Supplementary Materials*)

References

- [1] M. A. Tawhid and V. Savsani, "A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems," *Applied Intelligence*, vol. 48, no. 10, pp. 3762–3781, 2018.
- [2] S.-Z. Zhao, M. W. Iruthayarajan, S. Baskar, and P. N. Suganthan, "Multi-objective robust pid controller tuning using two lbests multi-objective particle swarm optimization," *Information Sciences*, vol. 181, no. 16, pp. 3323–3335, 2011.
- [3] A. Parisio and L. Glielmo, "Multi-objective optimization for environmental/economic microgrid scheduling," in *Proceedings of the IEEE International Conference on Cyber Technology in Automation*, IEEE, Bangkok, Thailand, May 2012.
- [4] 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, 2008.
- [5] H. Li, J. Deng, Q. Zhang, and J. Sun, "Adaptive epsilon dominance in decomposition-based multiobjective evolutionary algorithm," *Swarm and Evolutionary Computation*, vol. 45, pp. 52–67, 2019.
- [6] C. Dai, Y. Wang, and M. Ye, "A new multi-objective particle swarm optimization algorithm based on decomposition," *Information Sciences*, vol. 325, pp. 541–557, 2015.
- [7] J. Bader and E. Zitzler, "HyPE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [8] R. Hernández Gómez and C. A. Coello Coello, "Improved metaheuristic based on the r2 indicator for many-objective optimization," in *Proceedings of the 2015 Genetic and Evolutionary Computation Conference—GECCO'15*, pp. 679–686, Madrid, Spain, 2015.
- [9] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2018.
- [10] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [11] 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.
- [12] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.
- [13] D. Cai, Y. Wang, and L. Hu, "An improved α -dominance strategy for many-objective optimization problems," *Soft Computing*, vol. 20, no. 3, pp. 1105–1111, 2016.
- [14] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 269–285, 2014.
- [15] R. Liu, X. Song, L. Fang, and L. Jiao, "An r-dominance-based preference multi-objective optimization for many-objective optimization," *Soft Computing*, vol. 21, no. 17, pp. 5003–5024, 2017.
- [16] L. S. Batista, F. Campelo, F. G. Guimarães, and J. A. Ramírez, "Pareto cone ϵ -dominance: improving convergence and diversity in multiobjective evolutionary algorithms," in *Proceedings of the 2011 International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Ouro Preto, Brazil, April 2011.
- [17] I. Fister, M. Perc, S. M. Kamal, and I. Fister, "A review of chaos-based firefly algorithms: perspectives and research challenges," *Applied Mathematics and Computation*, vol. 252, pp. 155–165, 2015.
- [18] H. Zhang and Q. Hui, "Many objective cooperative bat searching algorithm," *Applied Soft Computing*, vol. 77, pp. 412–437, 2019.
- [19] D. Zouache, Y. Ould Arby, F. Nouioua, and F. Ben Abdelaziz, "Multi-objective chicken swarm optimization: a novel algorithm for solving multi-objective optimization problems," *Computers & Industrial Engineering*, vol. 129, pp. 377–391, 2019.
- [20] Amarteet and J. K. Chhabra, "Many-objective artificial bee colony algorithm for large-scale software module clustering problem," *Soft Computing*, vol. 22, no. 19, pp. 6341–6361, 2018.
- [21] N. A. Moubayed, A. Petrovski, and J. McCall, "D²MOPSO: mopso based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evolutionary Computation*, vol. 22, no. 1, pp. 47–77, 2014.
- [22] 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.
- [23] M. R. Sierra and C. A. Coello Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance," *Lecture Notes in Computer Science*, pp. 505–519, Springer, Berlin, Germany, 2005.
- [24] Q. Lin, J. Li, Z. Du, J. Chen, and Z. Ming, "A novel multi-objective particle swarm optimization with multiple search strategies," *European Journal of Operational Research*, vol. 247, no. 3, pp. 732–744, 2015.
- [25] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 191–204, 2015.
- [26] Y. Xiang, Z. Xueqing, and D. Wen-Bo, "Multiswarm comprehensive learning particle swarm optimization for solving multi-objective optimization problems," *PLoS One*, vol. 12, no. 2, Article ID e0172033, 2017.
- [27] 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.
- [28] Y. Yuan, H. Xu, B. Wang, and X. Yao, “A new dominance relation-based evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.
 - [29] S. Z. Martinez and C. A. Coello Coello, “A multi-objective particle swarm optimizer based on decomposition,” in *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference*, pp. 69–76, Dublin, Ireland, July 2011.
 - [30] Y. Liu, D. Gong, X. Sun, and Y. Zhang, “Many-objective evolutionary optimization based on reference points,” *Applied Soft Computing*, vol. 50, pp. 344–355, 2017.
 - [31] R. Denysiuk, L. Costa, and I. Espirito Santo, “Many-objective optimization using differential evolution with variable-wise mutation restriction,” in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation Conference—GECCO’13*, pp. 591–598, Amsterdam, Netherlands, 2013.
 - [32] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc, New York, NY, USA, 2001.
 - [33] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, WA, Australia, November 1995.
 - [34] 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.
 - [35] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. CoelloCoello, F. Luna, and E. Alba, “SMPPO: a new PSO-based meta-heuristic for multi-objective optimization,” in *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pp. 66–73, IEEE, Nashville, TN, USA, March 2009.
 - [36] G. T. Pulido and C. A. Coello Coello, “Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer,” in *Proceedings of the Genetic and Evolutionary Computation Conference—GECCO 2004*, vol. 3102, pp. 225–237, Seattle, WA, USA, June 2004.
 - [37] W. Peng and Q. Zhang, “A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems,” in *Proceedings of the 2008 IEEE International Conference on Granular Computing*, IEEE, Hangzhou, China, August 2008.
 - [38] Q. Zhu, Q. Lin, W. Chen et al., “An external archive-guided multiobjective particle swarm optimization algorithm,” *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2794–2808, 2017.
 - [39] E. M. N. Figueiredo, T. B. Ludermir, and C. J. A. Bastos-Filho, “Many objective particle swarm optimization,” *Information Sciences*, vol. 374, pp. 115–134, 2016.
 - [40] A. Díaz-Manríquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal, “R2-based multi/many-objective particle swarm optimization,” *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 1898527, 10 pages, 2016.
 - [41] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
 - [42] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multiobjective optimization test problems,” in *Proceedings of 2002 IEEE Congress on Evolutionary Computation*, pp. 825–830, IEEE, Honolulu, HI, USA, May 2002.
 - [43] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
 - [44] Q. Zhang, A. Zhou, S. Zhaoy et al., “Multiobjective optimization test instances for the cec 2009 special session and competition,” Technical report CES-487, University of Essex, Nanyang Technological University, Clemson University, Singapore, 2008.
 - [45] B. Li, K. Tang, J. Li, and X. Yao, “Stochastic ranking algorithm for many-objective optimization based on multiple indicators,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 924–938, 2016.
 - [46] 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.
 - [47] D. A. Van Veldhuizen and G. B. Lamont, “On measuring multi-objective evolutionary algorithm performance,” in *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 204–211, IEEE, La Jolla, CA, USA, July 2000.
 - [48] H. Li and Q. F. Zhang, “Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.