



Research article

Multi-objective particle swarm optimization with reverse multi-leaders

Fei Chen¹, Yanmin Liu^{2,*}, Jie Yang², Meilan Yang¹, Qian Zhang¹ and Jun Liu³

¹ School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China

² School of Mathematics, Zunyi Normal College, Zunyi 563002, China

³ School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550025, China

* **Correspondence:** Email: yanmin7813@163.com.

Abstract: Despite being easy to implement and having fast convergence speed, balancing the convergence and diversity of multi-objective particle swarm optimization (MOPSO) needs to be further improved. A multi-objective particle swarm optimization with reverse multi-leaders (RMMOPSO) is proposed as a solution to the aforementioned issue. First, the convergence strategy of global ranking and the diversity strategy of mean angular distance are proposed, which are used to update the convergence archive and the diversity archive, respectively, to improve the convergence and diversity of solutions in the archives. Second, a reverse selection method is proposed to select two global leaders for the particles in the population. This is conducive to selecting appropriate learning samples for each particle and leading the particles to quickly fly to the true Pareto front. Third, an information fusion strategy is proposed to update the personal best, to improve convergence of the algorithm. At the same time, in order to achieve a better balance between convergence and diversity, a new particle velocity updating method is proposed. With this, two global leaders cooperate to guide the flight of particles in the population, which is conducive to promoting the exchange of social information. Finally, RMMOPSO is simulated with several state-of-the-art MOPSOs and multi-objective evolutionary algorithms (MOEAs) on 22 benchmark problems. The experimental results show that RMMOPSO has better comprehensive performance.

Keywords: multi-objective particle swarm optimization; global ranking; mean angular distance; reverse selection; information fusion

1. Introduction

In most practical application problems, there are widely multiple conflicting objectives that need to be optimized simultaneously, such as job shop scheduling, wastewater treatment, path planning, etc., which are often called multi-objective optimization problems (MOPs) [1]. Due to the conflict between the objectives, the improvement of one objective may cause the degradation of other objectives. Therefore, a set of compromise solutions are usually obtained in MOPs, which are called Pareto optimal solutions [2]. The set of all Pareto optimal solutions is called a Pareto optimal set (PS). Their mapping in the objective space is called a Pareto front (PF) [3]. When solving MOPs, it is critical to obtain a set of solutions that are as close to the true PF as possible and well distributed on it.

In recent years, a variety of meta-heuristic algorithms inspired by natural and physical phenomena have vigorously developed, such as genetic algorithm (GA) [4], ant colony optimization (ACO) [5], differential evolution (DE) [6], firefly algorithm (FA) [7] and particle swarm optimization (PSO) [8]. Among them, PSO is a type of classical swarm intelligence algorithm, which was proposed by Kennedy and Eberhart in 1995. It is a bionic evolutionary swarm intelligence algorithm inspired by the foraging behavior of birds, and has been widely used to solve single objective optimization problems because of its simple mechanism and fast convergence speed. After some research, it was found that PSO had great potential to be extended to solve MOPs. Coello et al. [9] proposed a multi-objective particle swarm optimization (MOPSO) in 2002. In MOPSO, the optimal position of each particle during the historical search process is called the personal best (*pbest*), and the optimal position currently searched by the whole population is called the global best (*gbest*), which guide particles to fly towards the true PF. In addition, MOPSO introduced an external archive to store the non-dominated solutions found by the algorithm in the search process, which provided good candidate solutions for the selection of *gbest*. It can be seen that the maintenance of external archives and the selection of two optimal solutions are extremely important for the convergence and diversity of the algorithm. However, in the process of optimization, the global search ability of traditional MOPSO is still weak, which can easily lead to premature convergence of algorithm. Additionally, since different *pbest* and *gbest* will guide the particles to approach the true PF in different directions, it will have a certain impact on the balance between convergence and diversity.

In order to effectively balance convergence and diversity, researchers put forward many improvement strategies to improve the performance of MOPSO. Primarily, choosing different leaders will guide particles to fly in different directions. If properly selected, the convergence and diversity of the algorithm can be effectively improved. For example, Cui et al. [10] introduced a two-archive mechanism in MOPSO, using crossover and mutation operators for particles in the two archives to improve the quality of global leaders, which can take convergence and diversity into account. Li et al. [11] proposed a newly defined virtual generational distance indicator to select the appropriate *gbest* and designed an adaptive *pbest* selection strategy based on different evolutionary states to enhance the exploitation and exploration abilities of particles. Sharma et al. [12] assigned solutions with minimum penalized bounded intersection fitness as global leaders in each baseline based cluster after updating the external archive by a reference line-based diversity preference method. Second, maintaining an external archive is one of the most commonly used techniques. Hu et al. [13] proposed fuzzy crowding distance to update elite archives, and this method successfully

extended traditional crowding distance to the fuzzy objective environment. Li et al. [14] used the method based on dominant difference and crowding density estimation to update external archives, and the experimental results showed that the algorithm had improved performance in terms of convergence and diversity. Yang et al. [15] designed a new external archive maintenance method based on vector angle, which effectively balanced the convergence and diversity of the algorithm, and the additional time cost was relatively low. Wu et al. [16] proposed an improved archive maintenance strategy based on predefined reference points, and in order to improve the diversity of solutions in the external archive, associated other solutions with some crowded reference points as a second criterion. In addition, many scholars also adjusted flight parameters to improve the overall performance of the algorithm. Han et al. [17] proposed an adaptive flight parameter adjustment mechanism, which used diversity information and population spacing information to obtain particle distribution with an appropriate diversity and convergence, so as to balance local exploitation ability and global exploration ability of the particles. The experimental results show that compared with some non-adaptive MOPSOs, the algorithm has certain improvements in balancing convergence and diversity. Huang et al. [18] also designed an adaptive strategy to adjust flight parameters, and used the increment of the ratio of dissipated energy to mass to detect evolution environment of the algorithm. The flight parameters could be dynamically adjusted according to the time-varying evolution environment, so as to effectively balance convergence and diversity. Generally speaking, the above-mentioned improved MOPSOs can improve convergence and diversity of the algorithm when solving most MOPs, but it still need to be further improved in terms of balancing convergence and diversity and improving overall performance of the algorithm.

Based on the above review and analysis, in order to further balance convergence and diversity and to improve the overall performance of the algorithm, a multi-objective particle swarm optimization with reverse multi-leaders (RMMOPSO) is proposed. The main contributions of the proposed RMMOPSO are as follows:

1) In RMMOPSO, the convergence strategy of global ranking and the diversity strategy of mean angular distance are proposed. It is used to update the convergence and diversity archives, respectively, to improve the convergence of solutions in the archive and select solutions with a better diversity. It improves the quality of candidate solutions of the global leader and enhances the convergence and diversity of the algorithm.

2) A reverse selection method is proposed to select the global best with a good convergence ($gbest_c$) and the global best with a good diversity ($gbest_d$). This method is beneficial to select the appropriate learning samples for each particle and lead particles to quickly fly to the true PF. In addition, an information fusion strategy is proposed to update $pbest$, which enhances the information interaction between particles in the population and improves convergence of the algorithm.

3) After redefining the selection of $gbest_c$ and $gbest_d$ in RMMOPSO, a new particle velocity updating method is proposed, in which the two global leaders selected are used to cooperatively guide the flight of particles in the population. In order to promote the exchange of social information, it is conducive to improving the quality of particles in the population and achieving a better balance between convergence and diversity.

The remainder of this paper is organized as follows. Some related work is described in Section 2. Section 3 introduces the details of the proposed RMMOPSO. In Section 4, the performance of RMMOPSO is verified by simulation experiments with existing MOPSOs and multi-objective

evolutionary algorithms (MOEAs) on three suites of benchmark problems. Finally, the conclusion of the algorithm is given in Section 5.

2. Related work

2.1. Multi-objective optimization problem

The general multi-objective optimization problem [19] can be formulated as follows:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s.t.} \quad & g_i(x) \geq 0, i = 1, 2, \dots, q \\ & h_j(x) = 0, j = 1, 2, \dots, r \end{aligned} \quad (1)$$

where $x = (x_1, x_2, \dots, x_D)$ is a D -dimensional decision vector in decision space $X \subset R^D$, $F(x)$ represents the m -dimensional objective vector, $f_i(x)$ represents the objective function of the i -th dimension, $g_i(x) \geq 0 (i = 1, 2, \dots, q)$ is an inequality constraint, and $h_j(x) = 0 (j = 1, 2, \dots, r)$ is an equality constraint.

Because MOPs has multiple conflicting objectives, almost no absolute optimal solution can optimize them all at the same time. Therefore, the concept of Pareto dominance [20] can be used to evaluate the quality of solutions in MOPs. That is, a solution x dominates the other solution y , denoted as $x \prec y$, if and only if $\forall i \in \{1, 2, \dots, m\}: f_i(x) \leq f_i(y)$ and $\exists j \in \{1, 2, \dots, m\}: f_j(x) < f_j(y)$.

2.2. Particle swarm optimization

PSO [8] is a traditional swarm intelligence algorithm. It is an optimization algorithm inspired by the foraging behavior of birds, which is easily implemented and has a fast convergence speed. In PSO, each particle updates its own velocity by learning the *pbest* and *gbest*. If the velocity vector of the i -th particle is $V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$, the position vector is $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$, where $i = 1, 2, \dots, N$, N is the population size, D is the dimension of the decision space. Then, the velocity and position of the i -th particle at the $t+1$ -th iteration are updated as follows:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (pbest_i(t) - X_i(t)) + c_2 r_2 (gbest_i(t) - X_i(t)) \quad (2)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (3)$$

where t is the number of iterations, ω is the inertia weight, c_1 and c_2 are learning factors, and r_1 and r_2 are two random numbers uniformly generated in $[0, 1]$.

The formula of particle velocity update is mainly composed of three parts. The first part is called the memory term, which represents the influence of magnitude and direction of previous velocity on the current velocity and is the inheritance of the previous velocity of the particle. The second part is called the self-cognition term, which means the particles learn from their previous experience. The third part is called the group cognition term, which represents the ability of particles to learn from the whole population and reflects knowledge sharing and cooperation between particles.

3. The proposed RMMOPSO

In order to better balance the convergence and diversity of the algorithm and to improve the comprehensive performance of the algorithm, this paper proposed a MOPSO with reverse multi-leaders. The proposed RMMOPSO will be described in detail in this section. The algorithm consists of five parts, as shown in Figure 1. First, in order to improve the quality and diversity of the initial population, the quasi-reflection-based learning (QRBL) mechanism [21] is adopted to initialize the population. Second, the RMMOPSO proposed a convergence strategy of the global ranking and the diversity strategy of the mean angular distance and the convergence archive (CA) and the diversity archive (DA) are updated, respectively. Third, global leaders $gbest_c$ and $gbest_d$ are selected for each particle from CA and DA, respectively, through the reverse selection method. In this way, the selected leaders can lead particles to quickly fly to the true PF, and can better play the role of each non-dominated solution than the roulette wheel selection method. Meanwhile, an information fusion strategy is proposed to update $pbest$ to improve the convergence of the algorithm. Finally, global leaders $gbest_c$ and $gbest_d$ cooperate to guide particles flight, which achieves a balance between the convergence and diversity and improves the overall performance of the algorithm.

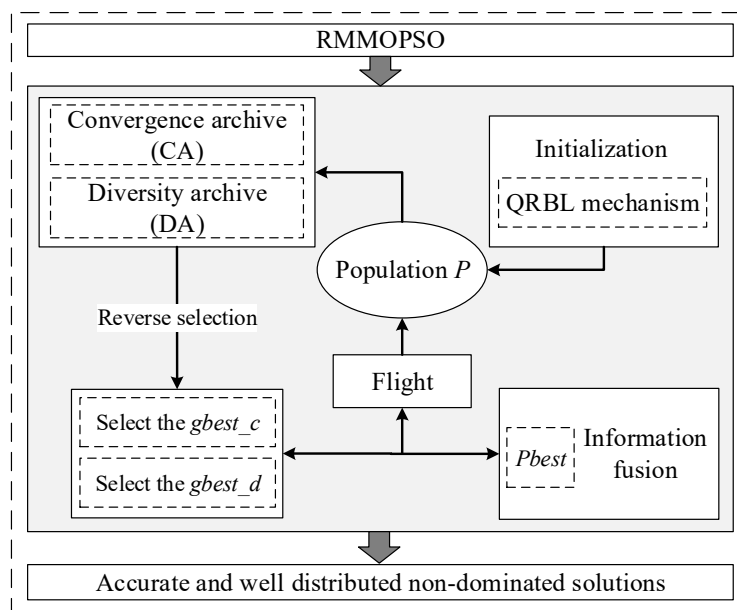


Figure 1. Framework of the proposed RMMOPSO.

3.1. Population initialization strategy

At present, the initial population of most MOPSOs is generated by random distribution, which will lead to random errors in the optimization results, thus affecting the optimization accuracy and the speed of the algorithm to a certain extent. Ergezer et al. [21] proved that the quasi-reflected point is the candidate point that is most likely to approach optimal solutions, and some scholars have applied it to other algorithms and achieved good results.

Therefore, this paper introduces the QRBL mechanism into the initial population of MOPSO to

generate a new quasi-reflected population. It not only ensures the randomness of the initial population of the algorithm, but also improves the operation efficiency and search speed of the algorithm. The specific details of the QRBL mechanism to initialize population are shown in Algorithm 1. After RMMOPSO randomly generates a population P in the decision space, it conducts the QRBL mechanism on population P to generate a new quasi-reflected population P^{qr} . Then, each particle is compared with the new particle obtained after quasi-reflected learning. Here, the sums of fitness values $fit(x)$ and $fit(x^{qr})$ on all objectives of particles before and after learning are compared, which can roughly reflect the convergence information of particles [22]. Finally, the optimal particles are retained as the final initial population P .

Algorithm 1: Initialize the population by the QRBL mechanism

Input: N (population size)

Output: P (initial population)

```

1: Generate uniformly distributed population  $P$  of  $N$  particles;
2: Generate a quasi-reflected population  $P^{qr}$  by conducting the QRBL mechanism on  $P$ ;
3: for  $i = 1 : N$ 
4:   if  $fit(x_i^{qr}) < fit(x_i)$    %  $x_i^{qr}$  is the quasi-reflected point of  $x_i$ .
5:      $x_i = x_i^{qr}$  ;
6:   end
7: end
8: return  $P$ .

```

3.2. Update the archives

The external archive is an important feature of MOPSO, which is used to store all non-dominated solutions found by the algorithm during the search process to provide good candidate solutions for the selection of $gbest$. Therefore, the quality of non-dominated solutions in the archive is particularly important. However, with the increase of the number of iterations, the number of non-dominated solutions substantially increases, which leads to a gradual increase in computational complexity. Therefore, it is increasingly necessary to evaluate non-dominated solutions in an appropriate way. Therefore, in this study, the convergence strategy of global ranking (GR) and the diversity strategy of mean angular distance (MAD) are proposed to update the convergence archive CA and the diversity archive DA of a predefined size, respectively. In addition, the archive sizes of CA and DA are similar to those set in literature [23]. The CA and DA play important roles in their respective fields, storing non-dominated solutions of different properties. Updating CA by GR can guide the population to quickly converge to the true PF, and updating DA by MAD can improve the distribution of the population in the objective space. The proposed two strategies effectively improve the convergence and diversity of the algorithm, and the specific analysis is as follows.

GR can effectively select the non-dominated solutions with a good convergence, thus improving the convergence of the algorithm. The smaller the GR value, the better the convergence of the non-dominated solutions. The specific description is as follows:

$$GR(x_i) = MR(x_i) + GD(x_i) \quad (4)$$

$$MR(x_i) = \frac{1}{L} \min_{m=1}^M (Rank(f_m(x_i))) \quad (5)$$

$$GD(x_i) = \frac{1}{L \cdot M} \sum_{\substack{j=1, \\ x_i \neq x_j}}^L \sum_{m=1}^M \frac{\max(f_m(x_i) - f_m(x_j), 0)}{f_{m,\max} - f_{m,\min}} \quad (6)$$

where x_i is the i -th non-dominated solution in the external archive, L is the number of non-dominated solutions in the external archive, M is the number of objectives, $f_m(\cdot)$ is the fitness function of the m -th objective, $Rank(\cdot)$ is the rank function, and $f_{m,\max}$ and $f_{m,\min}$ are the maximum and minimum values on the m -th objective, respectively. Here, using the maximum ranking (MR) [24] is preferred to select the solution with the best performance for some objectives, while the global detriment (GD) [25] is a method to consider how significant the differences between solutions are. The solution selected by GD is easy to eliminate the extreme value. Therefore, GR integrates the characteristics of MR and GD to evaluate the solutions in the archive. Here, we adopt the normalization method for formulas (5) and (6) to ensure the additivity of MR and GD.

MAD can better measure the similarity of solutions in the archive and effectively evaluate the diversity of non-dominated solutions, thereby increasing the diversity of the population. Additionally, the larger the value of MAD, the better the diversity of non-dominated solutions. Assuming that the two adjacent solutions of solution x_i are x_{i-1} and x_{i+1} , respectively, MAD can be described as:

$$MAD_i = (d(x_i, x_{i-1}) + d(x_i, x_{i+1})) / 2 + (\text{angle}(x_i, x_{i-1}) + \text{angle}(x_i, x_{i+1})) / 2 \quad (7)$$

$$d(x_i, x_j) = \sum_{m=1}^M |f_m(x_i) - f_m(x_j)| \quad (8)$$

$$\text{angle}(x_i, x_j) = \arccos \left| \frac{F(x_i) \cdot F(x_j)}{\text{norm}(F(x_i)) \cdot \text{norm}(F(x_j))} \right| \quad (9)$$

where $d(x_i, x_j)$ is the Manhattan distance between non-dominated solutions x_i and x_j , $\text{angle}(x_i, x_j)$ is the vector angle between the solutions x_i and x_j , $F(x_i)$ and $F(x_j)$ represent the fitness values of solutions x_i and x_j respectively, $F(x_i) \cdot F(x_j)$ is the inner product of $F(x_i)$ and $F(x_j)$, and $\text{norm}(\cdot)$ is the norm function. The Manhattan distance and vector angle [26] are combined to ensure the distribution and uniformity of the final solution set, so that the solutions with good diversity are retained in DA.

3.3. Gbest selection based on reverse selection

In most meta-heuristic algorithms, how to effectively balance convergence and diversity of algorithms is a crucial problem. In most MOPSOs, a particle chooses only one global leader in its social learning part. In this case, the particles cannot learn more useful experiences from various samples, thus reducing the diversity of the population. In fact, in human society, people always learn from multiple samples in the process of social learning. An individual will gain more experiences from multiple learning samples than from a single sample.

In RMMOPSO, each particle balances its convergence and diversity by selecting two global leaders. The two global leaders are selected one by one through the reverse selection method from

archives CA and DA, respectively, denoted as $gbest_c$ and $gbest_d$. In [27], researchers used a one-by-one selection strategy to select individuals, which can effectively improve the performance of the algorithm. In this paper, the reverse selection method proposed here can match the appropriate leader for each particle. The method guides particles to the nearest non-dominated solutions so that the particles can approach the PF in a few iterations. Additionally, every non-dominated solution is fully utilized to maximize the utilization of limited resources and to ensure a good distribution of the population. The method is specifically described as follows.

Algorithm 2: *Gbest* selection based on reverse selection

Input: CA (Convergence archive), P (population), N (population size), N_{CA} (the number of non-dominated solutions in CA)

Output: $gbest_c$ (Global best with good convergence)

1: The population is randomly divided into $h = \text{ceil}(N/N_{CA})$ groups;

 %The ceil function represents the integer in the direction of positive infinity.

2: **for** $k=1$ to h

3: temp_CA=CA;

4: **while** isempty(P_k)=0 % P_k represents the k -th group of particles.

5: Compute distance $D_{k,1}$ between non-dominated solution NS_1 in temp_CA and all the particles in P_k ;

6: Find the shortest distance $d_{k,1}$ in $D_{k,1}$ and the corresponding particle $x_{k,i}$;

7: Compute distance $D_{k,2}$ between $x_{k,i}$ and remaining non-dominated solutions in temp_CA;

8: Find the shortest distance $d_{k,2}$ in $D_{k,2}$ and the corresponding non-dominated solution NS_j ;

9: **if** $d_{k,1} \leq d_{k,2}$

10: $gbest_c_{k,i} = NS_1$; %Select NS_1 as the leader $gbest_c$ of particle $x_{k,i}$.

11: **else**

12: $gbest_c_{k,i} = NS_j$; %Select NS_j as the leader $gbest_c$ of particle $x_{k,i}$.

13: **end**

14: Empty the particle of the selected leader in the P_k and the corresponding selected non-dominated solution;

15: **end**

16: **end**

17: **return** $gbest_c$.

Assuming that the population size is N and the number of non-dominated solutions in the archive is N_A , the population is first randomly divided into $h = \text{ceil}(N/N_A)$ groups, where the ceil function represents the integer in the direction of positive infinity. Then, the leaders are selected for particles of each group in turn. Primarily, for example, taking the particles in the k -th, the distance $D_{k,1}$ between the first non-dominated solution NS_1 in the archive and all particles in this group is calculated in the objective space. Additionally, the shortest distance $d_{k,1}$ in $D_{k,1}$ and the corresponding particle $x_{k,i}$ are found. Then, the distance $D_{k,2}$ between particle $x_{k,i}$ and the remaining non-dominated solutions in the archive is calculated, and the shortest distance $d_{k,2}$ in $D_{k,2}$ and the corresponding non-dominated solution NS_j is found. Finally, the sizes of $d_{k,1}$ and $d_{k,2}$ are compared. If $d_{k,1} \leq d_{k,2}$,

the non-dominated solution NS_1 is selected as the $gbest$ of particle $x_{k,i}$. Otherwise, the non-dominated solution NS_j is selected as the $gbest$ of particle $x_{k,i}$. Additionally, the particles that have chosen leaders and the corresponding non-dominated solutions are emptied. In particular, when all the non-dominated solutions have been matched, it is necessary to reuse these solutions as candidate leaders of particles that have not yet chosen leaders. Algorithm 2 is the pseudo-code of taking CA as an example to select $gbest_c$ by using the reverse selection. The pseudo-code for selecting $gbest_d$ from DA by the reverse selection is similar to Algorithm 2.

Notably, the reverse selection method is similar to the matched pattern, in that the non-dominated solutions that have already been selected will not be used as candidate solutions when selecting leaders for the remaining particles in each group. In order to intuitively illustrate the process of selecting $gbest$ by the reverse selection method, Figure 2 shows the process of selecting leaders for six particles from the six non-dominated solutions in a two-dimensional objective space.

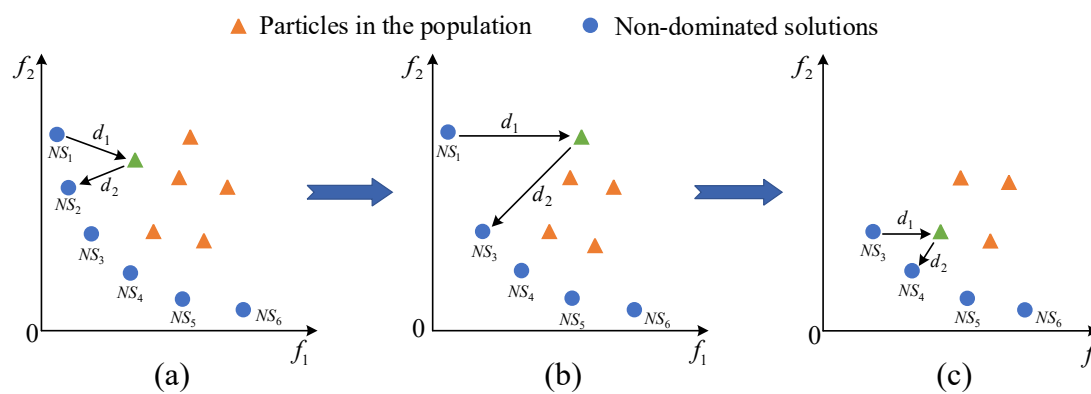


Figure 2. (a) $d_2 < d_1$, NS_2 is selected as the $gbest$ of the green particle; (b) $d_1 < d_2$, NS_1 is selected as the $gbest$ of the green particle; (c) $d_2 < d_1$, NS_4 is selected as the $gbest$ of the green particle.

3.4. Information fusion for $pbest$

In RMMOPSO, in order to improve the convergence of the algorithm, an information fusion strategy is proposed to update $pbest$ after selecting the global leaders for the particles, namely:

$$pbest_i^d = r^d \cdot pbest_i^d + (1 - r^d) \cdot gbest_c_i^d \quad (10)$$

where r^d is a random number uniformly distributed in $[0,1]$, and $gbest_c_i^d$ is the d -th dimension of the global leader of particle i .

Figure 3 shows the details of the $pbest$ update. During $pbest$ updates, if the previous position is dominated by the current position, then the current position will replace the previous position. Otherwise, $pbest$ will be updated using the information fusion strategy. In order to increase the diversity of the population, a random number r is first generated when updating each dimension of $pbest$. If the random number r is larger than the predefined parameter p , the corresponding dimension is updated by using Eq (10). Otherwise, a random number in the search range $[L^d, U^d]$ is initialized. Here we simply set p to $1/N$. This is because a relatively small p can protect the structure of $pbest$ and $gbest_c$ to a certain extent, and to make the particles learn more experiences from $gbest_c$ and to

jump out of the local optimum more easily.

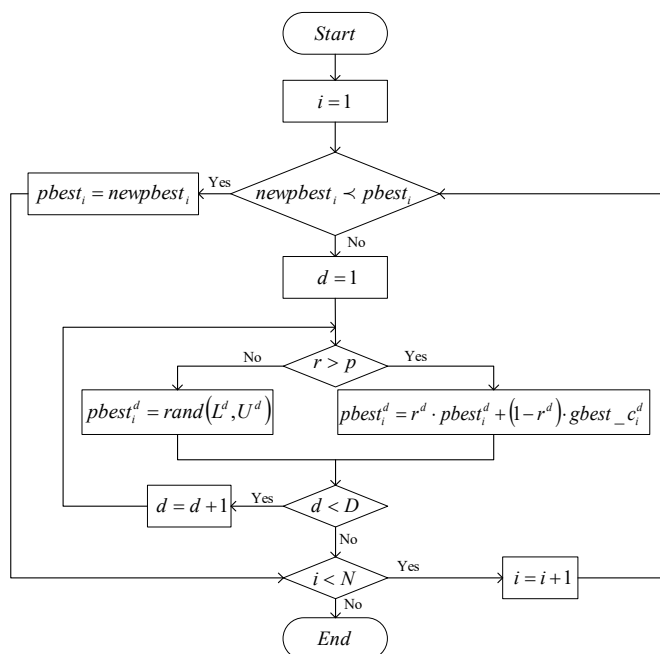


Figure 3. The update process of *pbest*.

3.5. New speed update method

The RMMOPSO improves the updating method of particles, that is, when each particle in the population is updated, two global leaders *gbest_c* and *gbest_d* are selected for each particle from the archives CA and DA, respectively, by the reverse selection method, which are used to guide the particle flight in the population cooperatively, so as to promote the exchange of social information. This is beneficial to improve the quality of particles in the population and to better balance the convergence and diversity of the algorithm. The specific update method are as follows:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (pbest_i(t) - X_i(t)) + c_2 r_2 ((gbest_c_i(t) + gbest_d_i(t))/2 - X_i(t)) \tag{11}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{12}$$

where *t* is the number of iterations, ω is the inertia weight, c_1 and c_2 are learning factors, and r_1 and r_2 are two random numbers uniformly generated in $[0,1]$.

3.6. General framework of RMMOPSO

The main components of RMMOPSO are introduced in detail above, namely initialization by the quasi-reflection-based learning mechanism, the archives update, *gbest* selection based on reverse selection, information fusion for *pbest*, and a new speed update method. The pseudo-code of RMMOPSO is shown in Algorithm 3, where *G* and G_{max} represent the current iteration number and the maximum iteration number, respectively.

At first, RMMOPSO initializes the population in line 1 by introducing the QRBL mechanism, as

described in Algorithm 1. After that, RMMOPSO starts the loop of the population search process in lines 2–8. The archives CA and DA are updated by the global ranking and mean angular distance, respectively (in line 3). Additionally, the global leaders $gbest_c$ and $gbest_d$ are selected from the archives CA and DA, respectively, by the reverse selection of Algorithm 2 (in line 4). At the same time, an information fusion on $pbest$ is performed, as described in Section 3.4. Finally, the velocity and position of the particles are updated using Eqs (11) and (12) (in lines 6 and 7). The above evolution process is repeated until the maximum number of iterations is reached, and the final external archive A is output.

Algorithm 3: General framework of RMMOPSO

Input: N (population size), CA (Convergence archive), DA (Diversity archive)

Output: A (final external archive A)

1: Initialization P (**Algorithm 1**);

2: **while** $G \leq G_{\max}$

3: Update CA and DA ;

4: Select $gbest_c$ and $gbest_d$ by the reverse selection (**Algorithm 2**);

5: Update $pbest$;

6: Update the velocity by Eq (11);

7: Update the position by Eq (12);

8: **end**

9: **return** A .

3.7. Computational complexity of RMMOPSO

In order to analyze the computational complexity of RMMOPSO, the main components of the algorithm in a generation are considered. Algorithm 3 shows that the computational steps of RMMOPSO mainly include initialization by the quasi-reflection-based learning mechanism, an update of archives CA and DA, selection of $gbest$ based on reverse selection, information fusion for $pbest$, and a new speed update method. Assuming that the population size is N , the sizes of CA and DA are both $N/2$, the number of objectives is M , and the dimension of the decision variables is D . Then, the complexity of initialization by the quasi-reflection-based learning mechanism in Algorithm 1 is $O(ND + N)$. The computational complexity of the update of archives CA and DA includes two cases, and when the archives are not full, the computational complexity is $O(1)$. However, when CA and DA overflow, the particles in the archives need to first be sorted by non-dominated sorting. In the worst case, the computational complexity of non-dominated sorting is $O(MN^2)$. In addition, the worst-case complexity of updating CA with GR convergence strategy is $O(N + N^2)$, and the worst-case complexity of updating DA with MAD diversity strategy is $O(N)$. Therefore, in the worst case, the update of CA and DA costs $O(MN^2)$. After that, the computational complexity of selecting $gbest_c$ and $gbest_d$ by reverse selection is $O(N)$. Finally, in the update of $pbest$, the Pareto dominance relationship between the previous position and the current position should be judged first, and then $pbest$ needs to be updated with an information fusion strategy. Therefore, in the worst case, updating $pbest$ costs $O(ND)$ calculations. The computational complexity of the new speed update

method is $O(1)$. Therefore, the total complexity of the proposed RMMOPSO in the worst case of a generation is $O(MN^2)$.

4. Experimental studies

4.1. Test problems

In order to evaluate the performance of RMMOPSO more objectively, a total of 22 test problems from three different benchmark suites, ZDT [28], UF [29], and DTLZ [30], are used to evaluate the performance of the algorithms. Among them, there are five two-objective test problems in ZDT benchmark suite, seven two-objective test problems and three three-objective test problems in UF benchmark suite, and seven three-objective test problems in DTLZ benchmark suite. These test problems have different characteristics and complex PFs, such as concavity and convexity, multi-modal, disconnected, irregular PF shape, etc., which can effectively verify the reliability and efficiency of the algorithms. For the two-objective test problems, the number of decision variables of ZDT1-ZDT3 and UF1-UF7 is set to 30, and the number of decision variables of ZDT4 and ZDT6 is set to 10. For the three-objective test problems, the number of decision variables of UF8~UF10 is set to 30, that of DTLZ1 is set to 7, that of DTLZ2~DTLZ6 is set to 12, and that of DTLZ7 is set to 22. The parameter settings for each test problem are shown in Table 1, where *FES* represents the maximum number of evaluations.

Table 1. Parameter settings for the selected test problems.

Problems	N	M	D	<i>FES</i>
ZDT1~ZDT3	200	2	30	10000
ZDT4 and ZDT6	200	2	10	10000
UF1~UF7	200	2	30	10000
UF8~UF10	200	3	30	10000
DTLZ1	200	3	7	10000
DTLZ2~DTLZ6	200	3	12	10000
DTLZ7	200	3	22	10000

4.2. Performance indicators

This paper uses inverted generational distance (IGD) [31] and hypervolume (HV) [32] as the performance evaluation indicators of the algorithm. These two indicators are used to evaluate the performance of RMMOPSO with the selected MOPSOs and MOEAs.

The IGD indicator is a comprehensive performance indicator, which is used to measure the distance between the Pareto optimal solution set obtained by the algorithm and true PF. Additionally, it can effectively test the convergence and diversity of the algorithm; the smaller the IGD value of an algorithm, the better the convergence and diversity of the algorithm. The calculation formula of IGD is:

$$IGD(P^*, S) = \frac{1}{|P^*|} \sum_{x^* \in P^*} dist(x^*, S) \quad (13)$$

$$\text{dist}(x^*, S) = \min_{j=1}^{|S|} \sqrt{\sum_{i=1}^k \left(\frac{f_i(x^*) - f_i(a_j)}{f_i^{\max} - f_i^{\min}} \right)^2} \quad (14)$$

where S represents the Pareto optimal solution set obtained by the algorithm, P^* is the solution set uniformly distributed on PF, $\text{dist}(x^*, S)$ is the shortest Euclidean distance between the solution x^* and S , and f_i^{\min} and f_i^{\max} represent the minimum value and maximum value of the i -th objective value, respectively, ($i = 1, 2, \dots, k$, k is the number of objectives), $a_j \in S$ and $j = 1, 2, \dots, |S|$.

The HV indicator is also a comprehensive performance indicator, which refers to the volume of the region in the objective space enclosed by the Pareto optimal solution set obtained by the algorithm and the reference points. This indicator can estimate the convergence and diversity of the solution set obtained by the algorithm; the larger the HV value of an algorithm, the better the overall performance of the algorithm. Assuming $Z^r = (Z_1^r, Z_2^r, \dots, Z_m^r)$ is a reference point dominated by all Pareto optimal solutions in the objective space, then HV can be calculated as follows, where δ denotes the Lebesgue measure.

$$HV(S, Z^r) = \delta \left(\bigcup_{x \in S} [f_1(x), Z_1^r] \times \dots \times [f_m(x), Z_m^r] \right) \quad (15)$$

4.3. Experimental settings

Table 2. Parameter settings of RMMOPSO and other comparison algorithms.

Algorithms	Parameter settings
NMPSO	$\omega \in [0.1, 0.5], c_1, c_2, c_3 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$
MPSOD	$\omega \in [0.1, 0.9], c_1, c_2, c_3 \in [1.5, 2.5], p_c = 0.9, p_m = 1/n, F = 0.5, CR = 0.5, \eta_m = 20, \eta_c = 20$
MMOPSO	$\omega \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.0], p_c = 0.9, p_m = 1/n, \eta_c = 20, \eta_m = 20, \delta = 0.9$
SMPSO	$\omega \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$
DGEA	$R = 10$
SPEAR	$p_m = 1/n, p_c = 1.0, \eta_m = 20, \eta_c = 20$
NSGAIII	$p_m = 1/n, p_c = 1.0, \eta_m = 20, \eta_c = 20$
MOEAD	$p_m = 1/n, p_c = 1.0, \eta_m = 20, \eta_c = 20$
RMMOPSO	$\omega = 0.4, c_1 = c_2 = 2, \text{size}(CA) = \text{size}(DA) = N/2, p = 1/N$

In this section, in order to better verify the performance of RMMOPSO, eight algorithms are selected for performance comparison, including four MOPSOs (NMPSO [33], MPSOD [34], MMOPSO [35] and SMPSO [36]) and four MOEAs (DGEA [37], SPEAR [38], NSGAIII [39] and MOEAD [40]).

In order to ensure the fairness of the algorithm performance comparison, the relevant parameters set by all the comparison algorithms are consistent with the original references. The main parameter settings of each algorithm are shown in Table 2, where ω , c_1 and c_2 are the parameters used for the update velocity in MOPSOs, and p_c and p_m represent crossover probability and mutation probability, respectively. η_c and η_m are the distribution indexes of the simulated binary crossover (SBX) and the polynomial mutation (PM), respectively. In MPSOD, F and CR are set differential evolution parameters. In MMOPSO, δ is the probability that the controlling parent solutions are

selected from the neighborhood of T . R is the number of direction vectors in DGEA. In MOEAD, T represents the size of the neighborhood between the weight coefficients. In addition, the population size of all algorithms is set to 200, and the maximum number of function evaluations is set to 10000. Each algorithm is independently run for 30 times on each test problem, and then the mean and standard deviation of IGD and HV values are recorded. Additionally, all experimental results of the algorithms are implemented by MATLAB R2021b under the Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz Windows 7 system. The source codes of all comparison algorithms are provided by PlatEMO [41].

4.4. Comparisons of performance indicators

4.4.1. Comparisons with four existing MOPSOs

Tables 3 and 4 show the mean and standard deviation of IGD and HV values of RMMOPSO and other four MOPSOs on 22 test problems, respectively, and the best IGD and HV values for each test problem are shown in bold. The Wilcoxon rank sum test [42] is adopted at the significance level of $\alpha = 0.05$ to show significant differences between test results. The symbols '+', '-', and ' \approx ' in the tables indicate that the results of other algorithms are significantly better than, significantly worse than, and statistically similar to RMMOPSO, respectively. In addition, the two-tailed t-test [43] is carried out at the significance level of $\alpha = 0.05$, and the test results of the two-tailed t-test are also shown in the tables. The t values in the tables are presented in bold to indicate that RMMOPSO is significantly superior to other algorithms, and the symbol '*' indicates that the performance of RMMOPSO is statistically similar to other algorithms. The '/' in Table 4 indicates that the standard deviations of the two compared arrays are 0, and the t values cannot be calculated.

It can be directly observed from Table 3 that the comprehensive performance of the proposed RMMOPSO is obviously better than the four compared MOPSOs among the 22 benchmark problems. According to the Wilcoxon rank sum test results, RMMOPSO and the comparison algorithms NMPSO, MPSOD, MMOPSO, and SMPSO perform significantly better in 14, 19, 9, and 18 of the 22 comparisons, respectively. Additionally, similar results are obtained in 2, 0, 5, and 1 of the comparisons, respectively. It can be seen from the best IGD values in the table that RMMOPSO obtains 13 best IGD values on 22 test problems, while the number of the best IGD values obtained by NMPSO, MPSOD, MMOPSO and SMPSO are 1, 0, 6, and 2, respectively. In addition, according to the statistical results of the t-test in the penultimate row of Table 3, the algorithm proposed in the paper is significantly superior to the comparison algorithms on most test problems. Therefore, we can conclude that when the IGD indicator is used to measure the comprehensive performance of the algorithm, RMMOPSO outperforms the other four algorithms. MPSOD does not perform best on any of the test problems, because based on the decomposition of the objective space, the strategy of MPSOD does not improve the performance of the algorithm to a large extent, which is slightly worse than other algorithms. However, MMOPSO has the best performance on test problems DTLZ1 and DTLZ4, because it adopted simulated binary crossover and polynomial mutation operators in addition to the original MOPSO update strategy. However, the results on ZDT benchmark suite are not very satisfactory. Therefore, from the overall analysis of IGD indicator, it can be found that the comprehensive performance of RMMOPSO on the 22 test problems is better than the other four comparison algorithms.

In addition to the IGD indicator, HV is also used to further evaluate the comprehensive performance of RMMOPSO. As can be seen from the best HV values in Table 4, RMMOPSO obtained 11 best HV values on 22 test problems, while the number of the best HV values obtained by NMPSO, MPSOD, MMOPSO, and SMPSO are 3, 0, 5, and 2, respectively. Moreover, according to the statistical results of Wilcoxon rank sum test and t-test in Table 4, RMMOPSO is significantly superior to the comparison algorithms on most test problems. Therefore, it can be seen from Table 4 that RMMOPSO and the selected MOPSOs still have a strong competitiveness when the HV indicator is used to measure comprehensive performance of the algorithms.

4.4.2. Comparisons with four existing MOEAs

Tables 5 and 6 respectively show the mean and standard deviation of the IGD and HV values of RMMOPSO and four MOEAs on 22 test problems. Similarly, the Wilcoxon rank sum test and two-tailed t-test are used at the significance level of $\alpha = 0.05$ to test for significant differences between the results. Additionally, the best IGD and HV values for each test problem are shown in bold. The ‘/’ in Table 6 indicates that the standard deviations of the two compared arrays are 0, and the t values cannot be calculated.

As can be seen from Table 5, the comprehensive performance of the proposed RMMOPSO is significantly better than the four compared MOEAs among the 22 test problems. According to the results of Wilcoxon rank sum test, RMMOPSO and comparison algorithms DGEA, SPEAR, NSGAI, and MOEAD perform significantly better than these algorithms in 19, 12, 15, and 16 of the 22 comparisons, respectively, and obtained similar results in 1, 4, 3, and 0. Similarly, from the statistical results of t-test in the penultimate row of Table 5, RMMOPSO is significantly better than the comparison algorithms DGEA, SPEAR, NSGAI, and MOEAD on 19, 12, 15, and 16 test problems, respectively. It can be seen from the best IGD values in the table that RMMOPSO obtains 14 best IGD values on 22 test problems, while the number of the best IGD values obtained by DGEA, SPEAR, NSGAI, and MOEAD are 0, 2, 0, and 6 respectively. It can be observed that when the IGD indicator is used to measure the comprehensive performance of the algorithm, RMMOPSO outperforms the four comparison algorithms. NSGAI and DGEA do not perform the best on any of the selected test problems, because the strategy of NSGAI reference-based non-dominated ranking and the strategy of DGEA adaptive offspring generation cannot improve the performance of the algorithm to a great extent, which is slightly worse than other algorithms. Therefore, from the overall analysis of the IGD indicator, it can be found that the comprehensive performance of RMMOPSO on the 22 test problems is better than the other four comparison algorithms, because the reverse multi-leaders strategy proposed in this paper significantly improves the search ability of RMMOPSO and can effectively balance the convergence and diversity of the algorithm.

As shown in Table 6, the comparison results using the HV indicator are similar to those using the IGD indicator. RMMOPSO also obtains 14 best HV values on 22 test problems, while DGEA, SPEAR, NSGAI, and MOEAD obtain 0, 1, 1, and 5 best HV values, respectively. Therefore, it can be seen from Table 6 that RMMOPSO has a better comprehensive performance than the other four MOEAs when the performance of the algorithm is measured by the HV indicator.

4.5. Intuitive comparisons of graphics

4.5.1. Comparisons with four existing MOPSOs

In order to visually compare the stability of each algorithm on three suites of benchmark problems, ZDT, UF, and DTLZ. Figure 4 shows the statistical box plots of the IGD values distribution obtained by the RMMOPSO and other four MOPSOs running independently for 30 times on some test problems, where 1, 2, 3, 4, and 5 on the horizontal coordinate of these box plots are successively represented as NMPSO, MPSOD, MMOPSO, SMPSO, and RMMOPSO, and the vertical coordinate represents the IGD value of each algorithm. The smaller the gap between the upper and lower quartiles of the box plot is, and the flatter the box is, indicating that the experimental data is more concentrated and the stability is better. The symbol “+” in the box plots represents the abnormal value in the data. As you can see from these figures, Figure 4 shows a comparison consistent with Table 3. Compared with other four comparison algorithms, there are fewer abnormal values in the experimental data of RMMOPSO, and the boxes of RMMOPSO in most of the test problems are flatter and the IGD values are smaller, which indicates that the performance of RMMOPSO is more remarkable than other four algorithms in terms of both the solution results and the stability of the algorithm.

In order to compare the convergence and distribution of the algorithms more intuitively and observe whether they really converge to the approximate PF. The Figures 5 and 6 show the distribution of the non-dominated solution sets of RMMOPSO and NMPSO, MPSOD, MMOPSO, and SMPSO on the two-objective test problem ZDT2 and the three-objective test problem UF9, respectively. As can be seen from Figure 5, on the two-objective test problem ZDT2, only NMPSO, MMOPSO, and RMMOPSO can fully converge to the true PF, but NMPSO and MMOPSO distribution is not very uniform, while other two algorithms have poor performance in convergence and distribution. It can be observed from Figure 6 that on the three-objective test problem UF9, several comparison algorithms have poor performance in convergence and distribution, and most of the non-dominated solutions obtained by RMMOPSO can fully converge to true PF. In conclusion, RMMOPSO has better performance in convergence and distribution than other algorithms.

Finally, the convergence speed is also an important indicator to compare the performance of algorithms. Figure 7 shows the IGD convergence trajectories obtained by RMMOPSO and the four MOPSOs on the two-objective test problems ZDT2, UF7, and the three-objective test problem UF9 evaluated for 10000 times. As can be seen from the figures that RMMOPSO has the fastest convergence speed and is significantly better than other four comparison algorithms.

4.5.2. Comparisons with four existing MOEAs

Figure 8 shows the statistical box plots of the IGD value distributions obtained by RMMOPSO and the other four MOEAs running independently for 30 times on some test problems, which is used to visually compare the stability of each algorithm on these test problems, where 1, 2, 3, 4, and 5 on the abscissa of the box plots are successively represented as DGEA, SPEAR, NSGAIII, MOEAD, and RMMOPSO, and the ordinate represents the IGD value of each algorithm. As shown in Figure 8, the comparison results are consistent with Table 5. As can be seen from Figure 8, although the stability of RMMOPSO on UF4, UF5, UF8, and DTLZ2 are not as good as that of other algorithms,

the IGD values of RMMOPSO are smaller than them. Furthermore, in most test problems, the experimental data of RMMOPSO has fewer abnormal values, its boxes are relatively flat, and the IGD values are smaller. It shows that the performance of RMMOPSO has obvious advantages over the other four MOEAs in both the solution results and the stability of the algorithm.

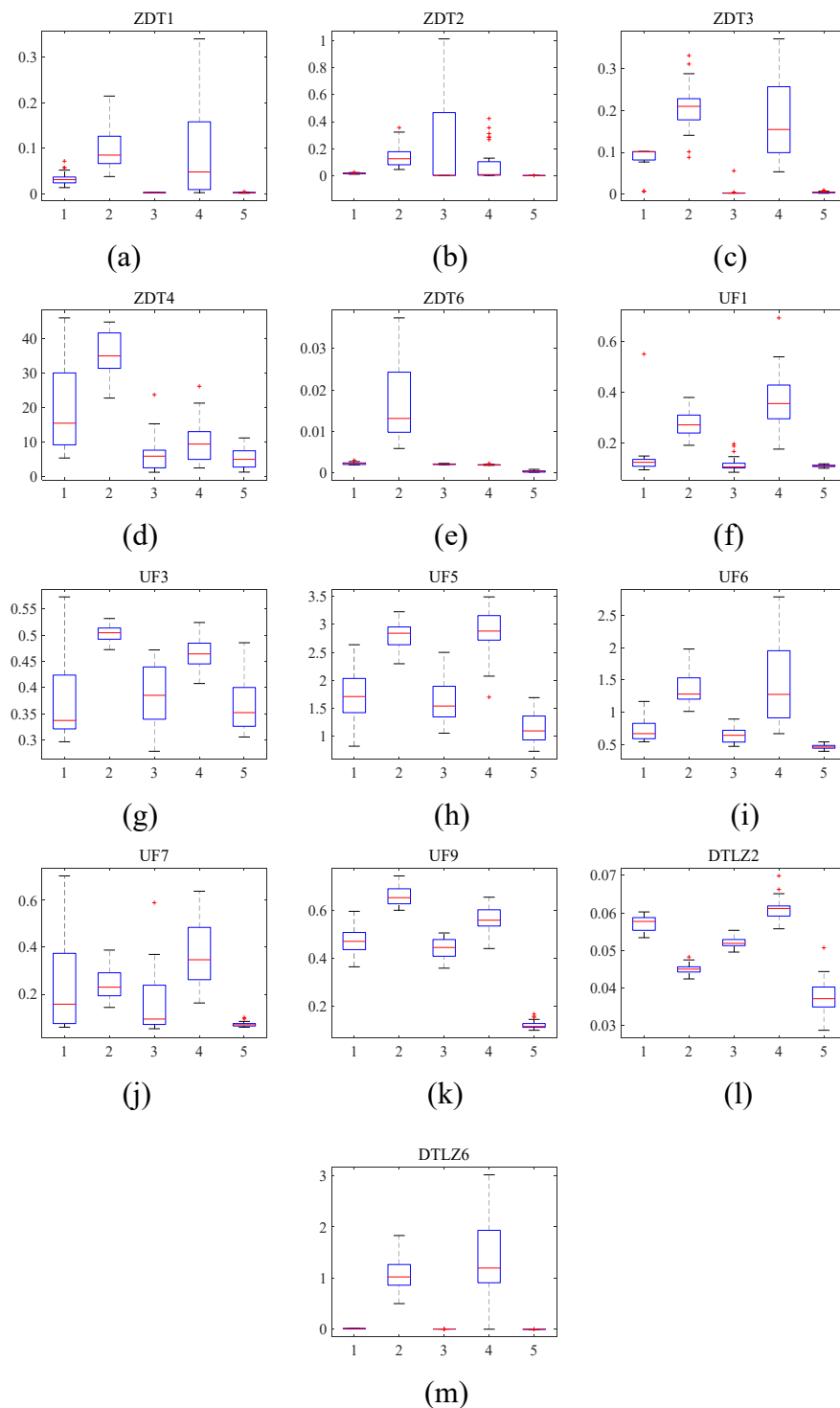


Figure 4. Box statistical plots of IGD values of five algorithms on different test problems.

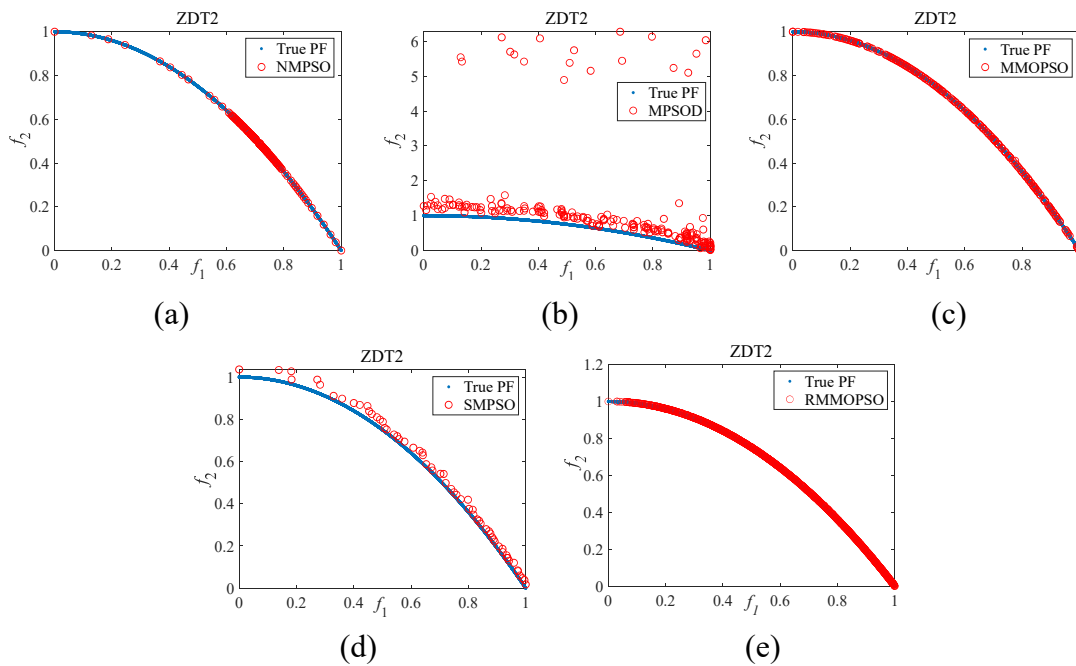


Figure 5. Approximate PF of the five algorithms on the ZDT2 test problem.

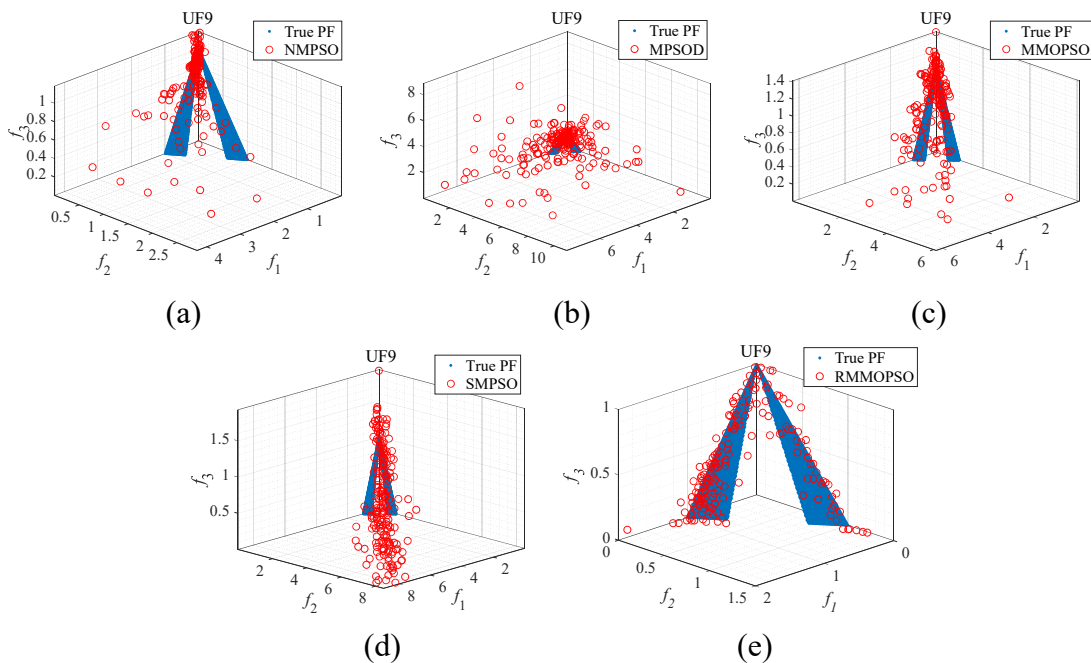


Figure 6. Approximate PF of the five algorithms on UF9 test problem.

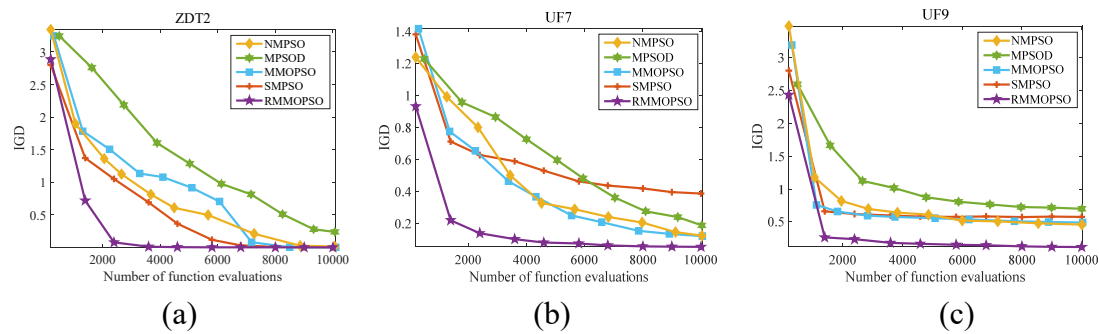


Figure 7. IGD convergence trajectories of RMMOPSO and four MOPSOs on ZDT2, UF7 and UF9.

In the same way, in order to visually compare the convergence and diversity of each algorithm, Figures 9 and 10 show the approximate PFs of RMMOPSO, DGEA, SPEAR, NSGAIII, and MOEAD on the two-objective test problem ZDT2 and the three-objective test problem UF9, respectively. As can be seen from Figure 9, on two-objective test problem ZDT2, only RMMOPSO can fully converge to the true PF, while the other four algorithms performed poorly in both convergence and distribution. It can be observed from Figure 10 that on the three-objective test problem UF9, although some non-dominated solutions of the four comparison algorithms converge to the true PF, most of them fail to converge and have poor distribution; only RMMOPSO has a relatively good convergence and distribution. To sum up, RMMOPSO has advantages over other algorithms in terms of convergence and distribution.

Next, we compare the convergence speed between the proposed RMMOPSO and the other four MOEAs on the two-objective test problems ZDT2, UF7, and the three-objective test problem UF9. Figure 11 shows the IGD convergence trajectories obtained by evaluating each algorithm on ZDT2, UF7, and UF9 for 10000 times. As can be seen from the figures, RMMOPSO has the fastest convergence speed, which is significantly better than the other four comparison algorithms.

In summary, from the box plot, the PF diagram, and the convergence trajectory diagram, the proposed RMMOPSO had a better comprehensive performance compared with the selected competitive algorithms, especially on the benchmark suites ZDT and UF. Therefore, it has superior advantages than the other comparison algorithms in solving the most complex multi-objective optimization problems.

4.6. Overall performance of RMMOPSO

As shown in Tables 7 and 8, in order to further compare the overall performance of RMMOPSO with all comparison algorithms, we also rank the average IGD and HV values of RMMOPSO and all comparison algorithms on the 22 test problems. Generally speaking, the smaller and more frequent ranking indicates the better overall performance of the algorithm. For instance, in Table 7, the rank 1 value of RMMOPSO is 11, which indicates that when RMMOPSO and all comparison algorithms are compared on 22 test problems, RMMOPSO achieves 11 best IGD values. In addition, we also use the Friedman test [44] to calculate the average ranking of all algorithms. The numbers with the most rank 1 and the best average ranking in the tables are shown in bold. It can be seen from Table 7 and Table 8 that the final ranking of the proposed RMMOPSO is the first in both the average IGD value and the HV value, and the number of rank 1 is also the most. This indicates that RMMOPSO can

obtain a better convergence and diversity solutions when solving the most MOPs. Therefore, it has a better advantage than other algorithms in solving MOPs.

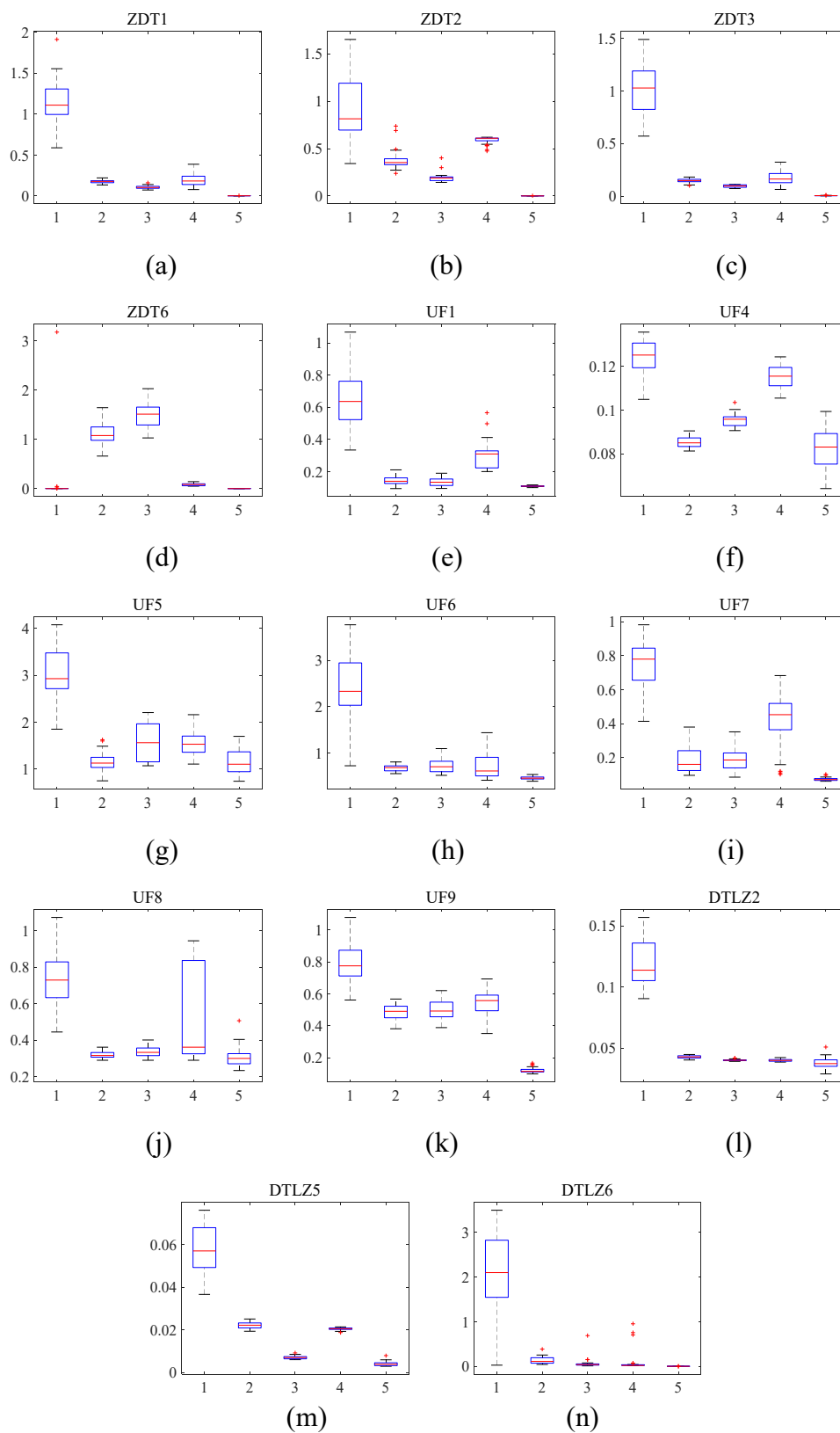


Figure 8. Box statistical plots of IGD values of five algorithms on different test problems.

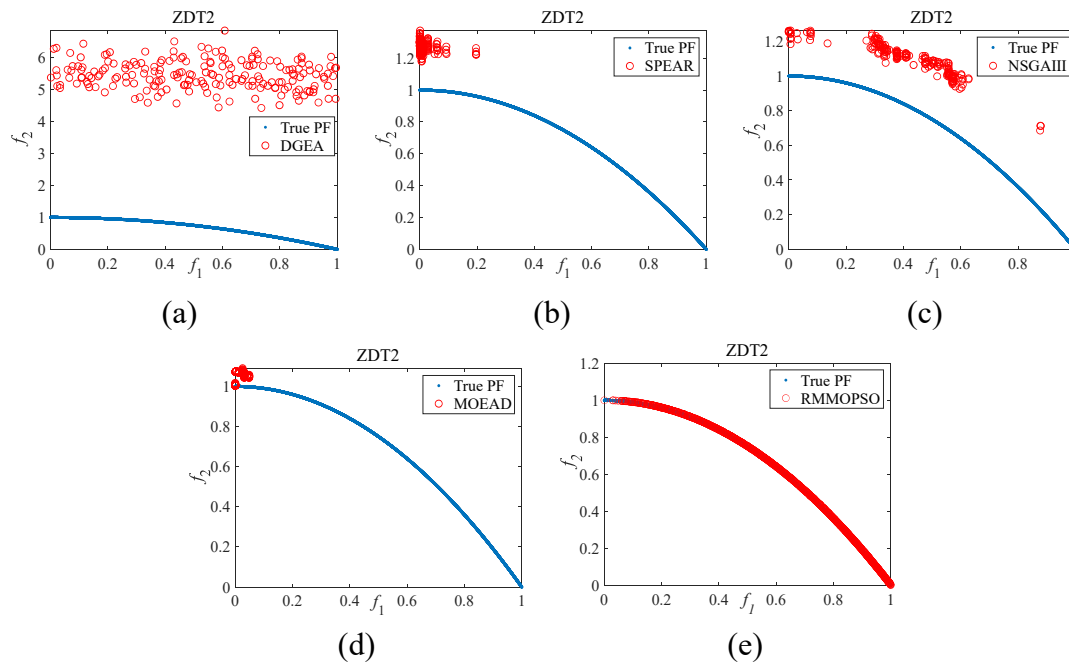


Figure 9. Approximate PF of the five algorithms on the ZDT2 test problem.

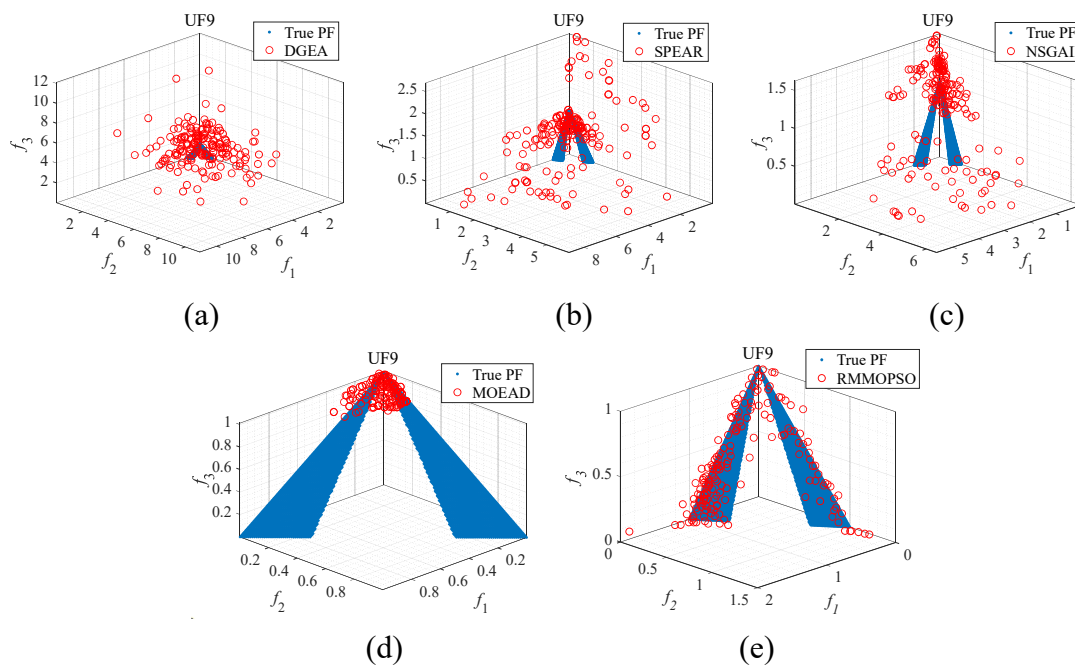


Figure 10. Approximate PF of the five algorithms on UF9 test problem.

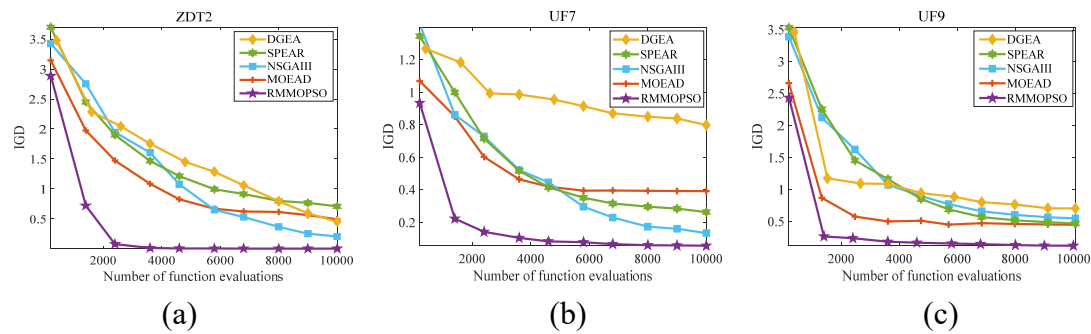


Figure 11. IGD convergence trajectories of RMMOPSO and four MOEAs on ZDT2, UF7 and UF9.

Table 7. Frequency of ranks of IGD of RMMOPSO and 8 comparison algorithms on 22 problems.

Algorithm	Rank1	Rank2	Rank3	Rank4	Rank5	Rank6	Rank7	Rank8	Rank9	Average Ranking	Final ranking
NMPSO	1	2	6	4	2	4	2	1	0	4.32	4
MPSOD	0	0	0	2	4	2	6	7	1	6.68	8
MMOPSO	2	11	2	2	3	2	0	0	0	2.95	2
SMPSO	1	1	1	1	3	2	7	6	0	6.09	7
DGEA	0	0	0	0	0	2	1	3	16	8.50	9
SPEAR	2	4	3	4	4	1	3	1	0	4.09	3
NSGAIII	0	0	7	7	4	2	1	0	1	4.41	5
MOEAD	5	2	1	0	1	5	2	4	2	5.05	6
RMMOPSO	11	2	2	2	1	2	0	0	2	2.91	1

Table 8. Frequency of ranks of HV of RMMOPSO and 8 comparison algorithms on 22 problems.

Algorithm	Rank1	Rank2	Rank3	Rank4	Rank5	Rank6	Rank7	Rank8	Rank9	Average Ranking	Final ranking
NMPSO	3	3	10	5	0	1	0	0	0	3.52	3
MPSOD	0	1	3	0	4	3	3	7	1	6.77	8
MMOPSO	4	9	4	3	1	0	1	0	0	3.07	2
SMPSO	1	3	4	1	1	3	6	3	0	5.50	7
DGEA	0	2	2	0	0	1	1	2	14	8.00	9
SPEAR	0	3	5	1	7	1	2	3	0	5.18	5
NSGAIII	0	2	4	5	4	5	1	1	0	5.18	6
MOEAD	4	2	2	3	1	2	5	3	0	4.93	4
RMMOPSO	10	4	5	0	0	2	1	0	0	2.84	1

4.7. Effectiveness of the QRBL mechanism

In order to verify the effectiveness of the proposed algorithm using the QRBL mechanism to initialize population, we set an experiment to compare RMMOPSO with its variant RMMOPSO_NQ, where RMMOPSO_NQ represents RMMOPSO without using the QRBL mechanism to initialize population. The parameter settings of this experiment are the same as the above experiments. RMMOPSO and RMMOPSO_NQ are independently run for 30 times on each of the three suits benchmark problems to obtain the statistical means and standard deviations (standard deviations in

parentheses) of the IGD indicator. The comparison results are shown in Table 9, and the best IGD value on each test problem is shown in bold. As can be seen from Table 9, RMMOPSO significantly outperforms RMMOPSO_NQ on most test problems, and obtains 17 best IGD values on a total of 22 test problems. The above experiment verifies that the QRBL mechanism plays an important role in the initialization process of RMMOPSO. This is because the quasi-reflected point of a point is more likely to be close to the optimal solutions than the point itself. Introducing the QRBL mechanism into the initial population of MOPSO not only ensures randomness of the initial population of the algorithm, but also improves operation efficiency and optimization speed of the algorithm.

Table 9. The IGD results of RMMOPSO and its variant RMMOPSO_NQ.

Problem	RMMOPSO_NQ	RMMOPSO	Problem	RMMOPSO_NQ	RMMOPSO
ZDT1	2.7118×10^{-3} (8.74×10^{-4})	2.4244×10^{-3} (7.53×10^{-4})	UF7	6.8097×10^{-2} (9.03×10^{-3})	7.1205×10^{-2} (9.59×10^{-3})
ZDT2	1.8708×10^{-3} (1.30×10^{-3})	1.3945×10^{-3} (5.22×10^{-4})	UF8	3.4222×10^{-1} (5.82×10^{-2})	3.0972×10^{-1} (5.52×10^{-2})
ZDT3	3.8673×10^{-3} (1.32×10^{-3})	4.5011×10^{-3} (1.63×10^{-3})	UF9	1.2357×10^{-1} (7.74×10^{-3})	1.2163×10^{-1} (1.68×10^{-2})
ZDT4	7.4147×10^0 (4.48×10^0)	5.1690×10^0 (2.79×10^0)	UF10	2.3529×10^0 (4.87×10^{-1})	2.2367×10^0 (5.26×10^{-1})
ZDT6	3.5416×10^{-4} (1.73×10^{-4})	3.6063×10^{-4} (1.81×10^{-4})	DTLZ1	1.9466×10^1 (4.19×10^0)	1.9967×10^1 (4.27×10^0)
UF1	1.1255×10^{-1} (4.20×10^{-3})	1.1080×10^{-1} (4.63×10^{-3})	DTLZ2	4.1152×10^{-2} (6.12×10^{-3})	3.7415×10^{-2} (4.61×10^{-3})
UF2	1.0385×10^{-1} (1.10×10^{-2})	8.2546×10^{-2} (5.08×10^{-3})	DTLZ3	1.8562×10^2 (1.13×10^1)	1.8253×10^2 (2.05×10^1)
UF3	3.8276×10^{-1} (6.32×10^{-2})	3.6552×10^{-1} (4.90×10^{-2})	DTLZ4	2.6749×10^{-1} (7.30×10^{-2})	3.2503×10^{-1} (1.47×10^{-1})
UF4	8.4892×10^{-2} (9.80×10^{-3})	8.2298×10^{-2} (9.22×10^{-3})	DTLZ5	4.3817×10^{-3} (1.07×10^{-3})	4.0912×10^{-3} (1.09×10^{-3})
UF5	1.2948×10^0 (3.23×10^{-1})	1.1471×10^0 (2.73×10^{-1})	DTLZ6	3.8972×10^{-4} (2.73×10^{-4})	3.3970×10^{-4} (1.28×10^{-4})
UF6	4.6394×10^{-1} (2.93×10^{-2})	4.6209×10^{-1} (3.43×10^{-2})	DTLZ7	2.7982×10^{-1} (2.36×10^{-1})	2.5548×10^{-1} (2.16×10^{-1})
Best/all	2/11	9/11	Best/all	3/11	8/11

5. Conclusions

In order to achieve a balance between convergence and diversity and to improve the overall performance of the algorithm, a multi-objective particle swarm optimization with reverse multi-leaders is proposed. The convergence strategy of the global ranking and the diversity strategy of the mean angular distance are put forward to update the convergence and diversity archives, respectively. Therefore, the convergence of the algorithm is improved, and the diversity of the population is maintained. At the same time, the global leaders are selected by the proposed reverse selection method, which can lead the particles to quickly fly to the true PF. In addition, the information fusion strategy is further applied to the update of *pbest* to improve the convergence of the algorithm. Finally, a new particle velocity updating method combines two global leaders to guide the flight of particles in the population. In this way, the performance of RMMOPSO is enhanced, and

the balance between the convergence and diversity is effectively achieved.

In the simulation experiments, RMMOPSO is compared with selected advanced MOPSOs and competitive MOEAs on three suites of benchmark problems. The experimental results also verify that RMMOPSO has an improved comprehensive performance and can better balance the convergence and diversity.

Acknowledgments

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

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Y. Wang, W. Gao, M. Gong, H. Li, J. Xie, A new two-stage based evolutionary algorithm for solving multi-objective optimization problems, *Inf. Sci.*, **611** (2022), 649–659. <https://doi.org/10.1016/j.ins.2022.07.180>
2. Q. Zhu, Q. Lin, W. Chen, K. C. Wong, C. A. C. Coello, J. Li, et al., An external archive-guided multiobjective particle swarm optimization algorithm, *IEEE Trans. Cybern.*, **47** (2017), 2794–2808. <https://doi.org/10.1109/TCYB.2017.2710133>
3. L. Ma, M. Huang, S. Yang, R. Wang, X. Wang, An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization, *IEEE Trans. Cybern.*, **52** (2021), 6684–6696. <https://doi.org/10.1109/TCYB.2020.3041212>
4. G. Acampora, R. Schiattarella, A. Vitiello, Using quantum amplitude amplification in genetic algorithms, *Expert Syst. Appl.*, **209** (2022), 118203. <https://doi.org/10.1016/j.eswa.2022.118203>
5. H. Zhao, C. Zhang, An ant colony optimization algorithm with evolutionary experience-guided pheromone updating strategies for multi-objective optimization, *Expert Syst. Appl.*, **201** (2022), 117151. <https://doi.org/10.1016/j.eswa.2022.117151>
6. Z. Zeng, M. Zhang, H. Zhang, Z. Hong, Improved differential evolution algorithm based on the sawtooth-linear population size adaptive method, *Inf. Sci.*, **608** (2022), 1045–1071. <https://doi.org/10.1016/j.ins.2022.07.003>
7. R. Nand, B. N. Sharma, K. Chaudhary, Stepping ahead firefly algorithm and hybridization with evolution strategy for global optimization problems, *Appl. Soft. Comput.*, **109** (2021), 107517. <https://doi.org/10.1016/j.asoc.2021.107517>
8. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Icnn95-international Conference on Neural Networks*, **4** (1995), 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
9. C. A. C. Coello, M. S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in *Pro. 2002 Congr. Evol. Comput. CEC'02 (Cat. No. 02TH8600)*, *IEEE*, **2** (2002), 1051–1056. <https://doi.org/10.1109/CEC.2002.1004388>

10. Y. Cui, X. Meng, J. Qiao, A multi-objective particle swarm optimization algorithm based on two-archive mechanism, *Appl. Soft. Comput.*, **119** (2022), 108532. <https://doi.org/10.1016/j.asoc.2022.108532>
11. Y. Li, Y. Zhang, W. Hu, Adaptive multi-objective particle swarm optimization based on virtual Pareto front, *Inf. Sci.*, **625** (2023), 206–236. <https://doi.org/10.1016/j.ins.2022.12.079>
12. D. Sharma, S. Vats, S. Saurabh, Diversity preference-based many-objective particle swarm optimization using reference-lines-based framework, *Swarm Evol. Comput.*, **65** (2021), 100910. <https://doi.org/10.1016/j.swevo.2021.100910>
13. Y. Hu, Y. Zhang, D. Gong, Multiobjective particle swarm optimization for feature selection with fuzzy cost, *IEEE Trans. Cybern.*, **51** (2020), 874–888. <https://doi.org/10.1109/TCYB.2020.3015756>
14. L. Li, L. Chang, T. Gu, W. Sheng, W. Wang, On the norm of dominant difference for many-objective particle swarm optimization, *IEEE Trans. Cybern.*, **51** (2019), 2055–2067. <https://doi.org/10.1109/TCYB.2019.2922287>
15. L. Yang, X. Hu, K. Li, A vector angles-based many-objective particle swarm optimization algorithm using archive, *Appl. Soft. Comput.*, **106** (2021), 107299. <https://doi.org/10.1016/j.asoc.2021.107299>
16. B. Wu, W. Hu, J. Hu, G. G. Yen, Adaptive multiobjective particle swarm optimization based on evolutionary state estimation, *IEEE Trans. Cybern.*, **51** (2019), 3738–3751. <https://doi.org/10.1109/TCYB.2019.2949204>
17. H. Han, W. Lu, J. Qiao, An adaptive multiobjective particle swarm optimization based on multiple adaptive methods, *IEEE Trans. Cybern.*, **47** (2017), 2754–2767. <https://doi.org/10.1109/TCYB.2017.2692385>
18. W. Huang, W. Zhang, Adaptive multi-objective particle swarm optimization with multi-strategy based on energy conversion and explosive mutation, *Appl. Soft. Comput.*, **113** (2021), 107937. <https://doi.org/10.1016/j.asoc.2021.107937>
19. K. Li, R. Chen, G. Fu, X. Yao, Two-archive evolutionary algorithm for constrained multiobjective optimization, *IEEE Trans. Evol. Comput.*, **23** (2018), 303–315. <https://doi.org/10.1109/TEVC.2018.2855411>
20. J. Liu, R. Liu, X. Zhang, Recursive grouping and dynamic resource allocation method for large-scale multi-objective optimization problem, *Appl. Soft. Comput.*, **130** (2022), 109651. <https://doi.org/10.1016/j.asoc.2022.109651>
21. M. Ergezer, D. Simon, Mathematical and experimental analyses of oppositional algorithms, *IEEE Trans. Cybern.*, **44** (2014), 2178–2189. <https://doi.org/10.1109/TCYB.2014.2303117>
22. Y. Xiang, Y. Zhou, M. Li, Z. Chen, A vector angle-based evolutionary algorithm for unconstrained many-objective optimization, *IEEE Trans. Evol. Comput.*, **21** (2016), 131–152. <https://doi.org/10.1109/TEVC.2016.2587808>
23. H. Wang, L. Jiao, X. Yao, Two_Arch2: An improved two-archive algorithm for many-objective optimization, *IEEE Trans. Evol. Comput.*, **19** (2014), 524–541. <https://doi.org/10.1109/TEVC.2014.2350987>
24. M. Garza-Fabre, G. T. Pulido, C. A. C. Coello, Ranking methods for many-objective optimization, *Mex. Int. Conf. Artif. Intell.*, **5845** (2009), 633–645. https://doi.org/10.1007/978-3-642-05258-3_56

25. W. Huang, W. Zhang, Multi-objective optimization based on an adaptive competitive swarm optimizer, *Inf. Sci.*, **583** (2022), 266–287. <https://doi.org/10.1016/j.ins.2021.11.031>
26. S. Chen, X. Wang, J. Gao, W. Du, X. Gu, An adaptive switching-based evolutionary algorithm for many-objective optimization, *Knowl. Based Syst.*, **248** (2022), 108915. <https://doi.org/10.1016/j.knosys.2022.108915>
27. Y. Liu, D. Gong, J. Sun, Y. Jin, A many-objective evolutionary algorithm using a one-by-one selection strategy, *IEEE Trans. Cybern.*, **47** (2017), 2689–2702. <https://doi.org/10.1109/TCYB.2016.2638902>
28. E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, *Evol. Comput.*, **8** (2000), 173–195. <https://doi.org/10.1162/106365600568202>
29. Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, S. Tiwari, Multi-objective optimization test instances for the CEC 2009 special session and competition, *Mech. Eng. New York*, **264** (2008), 1–30.
30. K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multi-objective optimization, *Evol. Mult. Opt. London.*, (2005), 105–145. https://doi.org/10.1007/1-84628-137-7_6
31. A. M. Zhou, Y. C. Jin, Q. F. Zhang, B. Sendhoff, E. Tsang, Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion, in *2006 IEEE Int. Conf. Evol. Comput.*, (2006), 892–899. <https://doi.org/10.1109/CEC.2006.1688406>
32. L. While, P. Hingston, L. Barone, S. Huband, A faster algorithm for calculating hypervolume, *IEEE Trans. Evol. Comput.*, **10** (2006), 29–38. <https://doi.org/10.1109/TEVC.2005.851275>
33. Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, J. Chen, et al., Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems, *IEEE Trans. Evol. Comput.*, **22** (2018), 32–46. <https://doi.org/10.1109/TEVC.2016.2631279>
34. C. Dai, Y. Wang, M. Ye, A new multi-objective particle swarm optimization algorithm based on decomposition, *Inf. Sci.*, **325** (2015), 541–557. <https://doi.org/10.1016/j.ins.2015.07.018>
35. Q. Lin, J. Li, Z. Du, J. Chen, Z. Ming, A novel multiobjective particle swarm optimization with multiple search strategies, *Eur. J. Oper. Res.*, **247** (2015), 732–744. <https://doi.org/10.1016/j.ejor.2015.06.071>
36. A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, E. Alba, SMPSO: a new PSO-based metaheuristic for multi-objective optimization, in *2009 IEEE Symp. Comput. Intell. MCDM.*, (2009), 66–73. <https://doi.org/10.1109/MCDM.2009.4938830>
37. C. He, R. Cheng, D. Yazdani, Adaptive offspring generation for evolutionary large-scale multi-objective optimization, *IEEE Trans. Syst. Man Cybern. Syst.*, **52** (2020), 786–798. <https://doi.org/10.1109/TSMC.2020.3003926>
38. S. Jiang, S. Yang, A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization, *IEEE Trans. Evol. Comput.*, **21** (2017), 329–346. <https://doi.org/10.1109/TEVC.2016.2592479>
39. K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: solving problems with box constraints, *IEEE Trans. Evol. Comput.*, **18** (2013), 577–601. <https://doi.org/10.1109/TEVC.2013.2281535>

40. Q. F. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.*, **11** (2007), 712–731. <https://doi.org/10.1109/TEVC.2007.892759>
41. Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: a matlab platform for evolutionary multi-objective optimization [educational forum], *IEEE Comput. Intell. Mag.*, **12** (2017), 73–87. <https://doi.org/10.1109/MCI.2017.2742868>
42. Y. Zhou, Z. Chen, Z. Huang, Y. Xiang, A multiobjective evolutionary algorithm based on objective-space localization selection, *IEEE Trans. Cybern.*, **52** (2020), 3888–3901. <https://doi.org/10.1109/TCYB.2020.3016426>
43. M. Sheng, Z. Wang, W. Liu, X. Wang, S. Chen, X. Liu, A particle swarm optimizer with multi-level population sampling and dynamic p-learning mechanisms for large-scale optimization, *Knowl. Based Syst.*, **242** (2022), 108382. <https://doi.org/10.1016/j.knosys.2022.108382>
44. J. Lu, J. Zhang, J. Sheng, Enhanced multi-swarm cooperative particle swarm optimizer, *Swarm Evol. Comput.*, **69** (2022), 100989. <https://doi.org/10.1016/j.swevo.2021.100989>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)