Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057168

ARTICLE

A Multi-Objective Particle Swarm Optimization Algorithm Based on Decomposition and Multi-Selection Strategy

Li Ma[1], Cai Dai[1,*], Xingsi Xue[2] and Cheng Peng[3]

[1]School of Computer Science, Shaanxi Normal University, Xi'an, 710119, China

[2]Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, 350118, China

[3]Information Construction and Management Center and Institute of Artificial Intelligence and Educational New Productivity, Ningxia Normal University, Guyuan, 756099, China

*Corresponding Author: Cai Dai. Email: cdai0320@snnu.edu.cn

Received: 09 August 2024 Accepted: 25 October 2024 Published: 03 January 2025

ABSTRACT

The multi-objective particle swarm optimization algorithm (MOPSO) is widely used to solve multi-objective optimization problems. In the article, a multi-objective particle swarm optimization algorithm based on decomposition and multi-selection strategy is proposed to improve the search efficiency. First, two update strategies based on decomposition are used to update the evolving population and external archive, respectively. Second, a multi-selection strategy is designed. The first strategy is for the subspace without a non-dominated solution. Among the neighbor particles, the particle with the smallest penalty-based boundary intersection value is selected as the global optimal solution and the particle far away from the search particle and the global optimal solution is selected as the personal optimal solution to enhance global search. The second strategy is for the subspace with a non-dominated solution. In the neighbor particles, two particles are randomly selected, one as the global optimal solution and the other as the personal optimal solution, to enhance local search. The third strategy is for Pareto optimal front (PF) discontinuity, which is identified by the cumulative number of iterations of the subspace without non-dominated solutions. In the subsequent iteration, a new probability distribution is used to select from the remaining subspaces to search. Third, an adaptive inertia weight update strategy based on the dominated degree is designed to further improve the search efficiency. Finally, the proposed algorithm is compared with five multi-objective particle swarm optimization algorithms and five multi-objective evolutionary algorithms on 22 test problems. The results show that the proposed algorithm has better performance.

KEYWORDS

Multi-objective optimization; multi-objective particle swarm optimization; decomposition; multi-selection strategy

1 Introduction

Since the year 2000, a total of 46,976 scholarly articles on multi-objective optimization problems (MOPs) [1] have been indexed in the Science Citation Index. MOPs are widely used in practical applications such as path planning [2,3], task scheduling [4] and coal production [5,6], among

others [7]. Unlike single-objective optimization problems, MOPs are inherently more complex due to conflicting objectives [8]. Therefore, it is impossible to optimize all objectives simultaneously; instead, one must prioritize them to find a set of solutions that achieve a relatively optimal balance.

Utilizing evolutionary algorithms to address Multi-Objective Problems (MOPs) remains a prominent area of research. Over recent decades, numerous multi-objective evolutionary algorithms have been developed, including the multi-objective genetic algorithm (MOGA) [9,10], multi-objective differential evolution algorithm (MODE) [11,12], multi-objective particle swarm optimization algorithm (MOPSO) [13,14] and others. The Particle Swarm Optimization Algorithm (PSO) [15], introduced by Eberhart and Kennedy in 1995, is a population-based stochastic search evolutionary algorithm that represents potential solutions through particles. Owing to its simplicity and rapid convergence, MOPSO has been extensively employed for solving MOPs [16,17]. Furthermore, MOPSO finds applications in various research domains. For instance, integrating MOPSO with data mining aids in resolving intricate data mining challenges and uncovering valuable insights [18]. In the past two decades, 5755 articles on MOPSO have been published in the Science Citation Index. As MOPs grow increasingly complex, achieving a better balance between diversity and convergence and devising an effective selection strategy to enhance search efficiency becomes crucial.

In certain traditional MOPSOs, the population tends to cluster around the leader particle, which represents the global optimal solution (gbest), leading to poor diversity [19]. To enhance the diversity of the evolutionary process, researchers have developed the clustering multiple-swarm multi-objective particle swarm optimization algorithm (CMOPSO) [20] and the coevolutionary multiple-swarm multi-objective particle swarm optimization algorithm (CMMOPSO) [21]. In 2011, the Coello team introduced a decomposition-based multi-objective particle swarm optimization algorithm (dMOPSO) [22] for addressing continuous unconstrained MOPs. A key feature of this algorithm is its use of memory reinitialization, which helps maintain population diversity. In 2014, Moubayed et al. [23] integrated dominance and decomposition to propose a multi-objective particle swarm optimization algorithm based on decomposition and dominance (D2MOPSO). Decomposition simplifies the problem by converting the multi-objective problem into a set of aggregation problems, while dominance constructs the leader archive. In 2015, Dai et al. [24] presented a novel decomposition-based multi-objective particle swarm optimization algorithm (MPSO/D). This algorithm maintains diversity by uniformly dividing the objective space and assigning a solution to each subspace. Generally, these decomposition-based MOPSOs exhibit good diversity and convergence when solving MOPs. However, they do not consider the distinct evolution of each subproblem. Additionally, for complex problems, such as those involving discontinuous Pareto-optimal fronts, new strategies and mechanisms are required.

Selecting the appropriate leader particles to guide the population's evolution is crucial for MOPSOs. Leong et al. [25] introduced a dynamic population multiple-swarm multi-objective adaptive particle swarm optimization algorithm (DMOPSO), which enhanced performance through dynamic population size adjustments and adaptive local archiving, selecting leaders based on different clusters. Zheng et al. [26] proposed a novel MOPSO that employs comprehensive learning strategies to effectively maintain population diversity. Hu et al. [27] developed a new parallel cell coordinate system, from which a group of leaders is chosen from the global archive and stored in the leader group. These leaders are selected based on environmental information and entropy to expedite convergence. Nonetheless, these algorithms only improve either diversity or convergence but fail to strike a balance between both.

Inspired by the above algorithms, it is an effective strategy to improve the diversity by using decomposition method. Meanwhile, under the decomposition framework, different selection strategies

are designed according to the different evolution of each subspace. Therefore, in order to improve the search efficiency of MOPSO, better balance diversity and convergence, and reduce the influence of (Pareto optimal front) PF discontinuity on solution set, this article integrates the decomposition method with the multi-selection strategy. The main contributions are as follows:

(1) For PF with various shapes, an advanced update strategy has been developed to maintain diversity and improve the quality of solutions. Initially, the objective space is partitioned using a set of uniformly distributed direction vectors, which categorizes the population accordingly. The diversity of the algorithm is maintained by a decomposition-based update strategy such that each subspace has a solution. Historical optimal solutions are stored in an external archive, which is updated using a strategy based on Pareto dominance and decomposition to improve solution quality. Upon completion of the iterations, the set of solutions with the lowest IGD value from both the evolving population and the external archive is chosen as the optimal solution set.
(2) According to the evolution of each subspace, a multi-selection strategy is designed. The first strategy is for the selected solution, the solution with the smallest penalty-based boundary intersection value is selected as its global optimal solution and the solution which is away from this selected solution and its global optimal solution is selected as its personal optimal solution. This strategy is to enhance global search. The second strategy is for non-dominated solutions which are directly classified by direction vectors and aggregate functions, their neighbor solutions are randomly selected as their global optimal solution and personal optimal solution. This strategy is to enhance local search. The third strategy is for the problems with discontinuity PF, which is identified by the cumulative number of iterations of the subspace without non-dominated solutions.
(3) An adaptive inertia weight update strategy dynamically adjusts the inertia weight based on the extent to which each particle is dominated by the reference point, balancing exploration and exploitation to enhance search efficiency.

The rest of this article is organized as follows: Section 2 provides an introduction to the main aspects of multi-objective optimization problems and multi-objective particle swarm optimization. Section 3 offers a detailed explanation of the proposed algorithm. Section 4 presents the experimental results along with relevant analysis. Lastly, Section 5 concludes the discussion.

2 Preliminaries

2.1 Multi-Objective Optimization Problem

In general, a MOP can be defined as [28]:

$$\begin{cases} \min & F(x) = \left(f_1(x), f_2(x), \ldots, f_M(x)\right) \\ \text{s.t.} & x \in \Omega \end{cases} \tag{1}$$

where Ω is an n-dimensional decision space; $x = (x_1, x_2, \ldots, x_n)$ is an n-dimensional decision vector; $f_i(x)$ $(i = 1, 2, \ldots, M)$ is the ith objective function; M is the number of objective functions. There are two solutions $\alpha, \beta \in \Omega$, if and only if $\forall i: f_i(\alpha) \leq f_i(\beta) \wedge \exists j: f_j(\alpha) < f_j(\beta)$ $(i, j = 1, 2, \ldots, M)$, then it can be said that α dominates β, and the dominance relationship between α and β is denoted as $\alpha \prec \beta$. In decision space, a solution is called the Pareto optimal solution when it is not dominated by any other solution. The set of all Pareto optimal solutions is called Pareto-optimal solution set (PS), and the objective set corresponding to these solutions is Pareto-optimal front (PF).

2.2 Multi-Objective Particle Swarm Optimization

PSO has been widely used in engineering practice due to its straightforward concept and minimal parameter requirements. It simulates bird foraging and uses inter-group member cooperation and information sharing to find the optimal solution [29]. Each particle represents a potential solution with two properties: velocity and position. In MOPSO, the particles seek the optimal solution according to their own and other particles' experience [30], so that the population continues to evolve and gradually approach PF. The position vector of the particle represents a candidate solution, and the velocity vector of the particle represents the velocity and direction of the particle moving in the next iteration. The information of particle i can be represented by a D-dimensional vector, the position is represented by $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and the velocity is represented by $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The particle's velocity and position are updated as follows [31]:

$$v_{id}^{t+1} = wv_{id}^{t} + c_1 r_1 \left(pbest_{id}^{t} - x_{id}^{t}\right) + c_2 r_2 \left(gbest_{id}^{t} - x_{id}^{t}\right) \tag{2}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t} \tag{3}$$

where v_{id}^{t} represents the dth dimension velocity of particle i in the tth iteration, x_{id}^{t} represents the dth dimension position of particle i in the tth iteration, $pbest_{id}^{t}$ is the individual optimal solution of particle i in the tth iteration, $gbest_{id}^{t}$ is the global optimal solution of the whole population in the tth iteration. w is the inertial weight, which controls the moving speed of particle in the search space, and a larger inertia weight can be used in earlier iterations, and then it can be gradually reduced. c_1 and c_2 are individual learning factors and social learning factors, respectively, which are used to adjust the velocity and position of particle. When c_1 is larger and c_2 is smaller, the particle will focus more on its own search experience and reduce the dependence on the group experience. The particle's exploration ability is enhanced and it is more inclined to explore new solution space. On the contrary, the exploitation ability of particles is enhanced. r_1 and r_2 are random numbers within [0, 1].

3 Proposed Algorithm

3.1 Decomposition Strategy

Firstly, a set of particles is randomly generated, initialized, and their corresponding objective values are calculated. Then, generate a set of direction vectors $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_N\}$ is generated to divide the objective space Y, as follows:

$$Y = \{Y_1, Y_2, \ldots, Y_N\} \tag{4}$$

where Y_N is the subspace corresponding to the vector γ_N, and N is the number of subspaces. For the subspace Y_i, the assigned solution set S_i is as follows:

$$\begin{cases} S_i = \left\{ x | x \in POP, \Delta\left(F\left(x\right), \gamma_i\right) = \max\limits_{1 \le i \le N} \left\{ \Delta\left(F\left(x\right), \gamma_i\right) \right\} \right\} \\ \Delta\left(F\left(x\right), \gamma_i\right) = \dfrac{\gamma_i * \left(F\left(x\right) - Z\right)^T}{\gamma_i * \left(F\left(x\right) - Z\right)}, \quad i = 1, 2, \ldots, N \end{cases} \tag{5}$$

where $Z = (Z_1, \ldots, Z_M)$ is a reference point, $Z_i = \min\{f_i(x) | x \in \Omega\} (i = 1, 2, \ldots, M)$, $\Delta\left(F\left(x\right), \gamma_i\right)$ is the cosine of the angle between γ_i and $F\left(x\right) - Z$. S_i is the solution set assigned in the ith subspace. In Fig. 1, solutions A, B and C correspond to subspace Y_1, solution D corresponds to subspace Y_2, solutions E, F and G correspond to subspace Y_3, and subspace Y_4 has no solution.

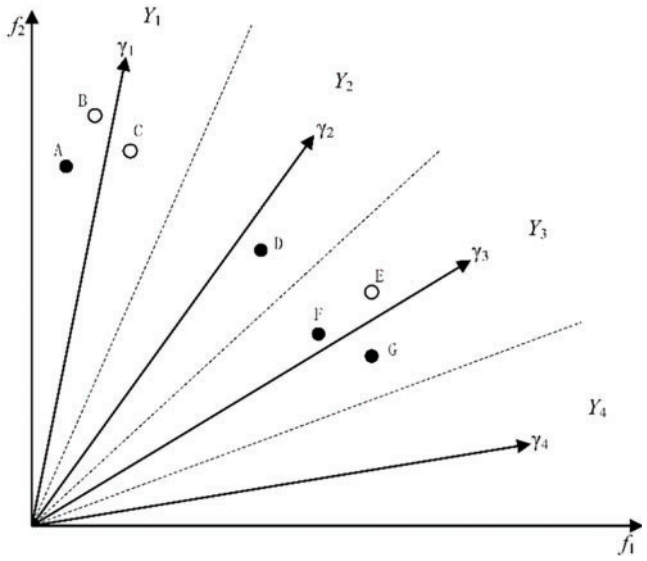

Figure 1: Decomposition strategy

3.2 Update Strategy

To maintain population diversity and enhance solution quality, this article employs enhanced update strategies for updating the offspring population POP and the external archive E, respectively.

A decomposition-based update strategy is employed to enhance population diversity. After the offspring population POP is classified according to the decomposition strategy, for each subspace, if the solution set S_i corresponding to subspace Y_i has no solution, the non-dominated solution whose objective vector has the smallest angle to the direction vector γ_i corresponding to this subspace is selected as the representative solution of the subspace and retained. If the solution set S_i corresponding to subspace Y_i has only one solution, the solution is directly selected as the representative solution and retained. If the solution set S_i corresponding to subspace Y_i has multiple solutions, the dominance relationship of these solutions is determined, the non-dominated solution of these solutions whose objective vector has the smallest angle to the direction vector γ_i corresponding to this subspace is selected as the representative solution and retained. Algorithm 1 shows the update strategy based on decomposition. Therefore, for solutions in Fig. 1, solutions A, D, F and G will be selected and retained.

Algorithm 1: Update strategy for evolving population

Input:

POP (offspring population)
N (size of population)
$(\gamma_1, \gamma_2, \ldots, \gamma_N)$ (N uniformly distributed direction vectors)

Output:

POP (updated population)

1: Classify POP by Eq. (5) and get $(S_1, S_2, \ldots, S_N)$ corresponding to $(\lambda_1, \lambda_2, \ldots, \lambda_N)$
2: **For** $i = 1 : N$ **do**
3: **If** $S_i = \varnothing$ **then**
4: Select and retain the non-dominated solution with the smallest angle between the objective vector and the direction vector γ_i from POP

(Continued)

Algorithm 1 (continued)

5: **If** S_i has only one solution **then**
6: Select and retain the solution
7: **If** S_i has multiple solutions **then**
8: Select and retain the non-dominated solution with the smallest angle between the objective vector and the direction vector γ_i from S_i
9: **End**
10: **End For**

Although the above update strategy can maintain diversity well, it proves less efficient for problems with degenerated PF and more complex PF shapes. To address this, this article employs an external archive to store elite solutions obtained during iterations, updating them through a strategy based on Pareto dominance and decomposition. Firstly, the solutions in the external archive E are merged with the offspring solutions to find the non-dominated solutions. If the count of non-dominated solutions does not surpass the archive's capacity, they are included in the archive. Otherwise, they are categorized according to Eq. (5), and select the one with the smallest angle between the objective vector and the direction vector in each class of solutions into the external archive. If the number of solutions in the external archive is less than its size at this time, the solutions farthest from the solutions in the external archive are selected from the remaining non-dominated solutions to join until the number of solutions in the external archive reaches its size. Fig. 2 shows the update strategy. The size of the external archive is 4, and there are six non-dominated solutions A, B, C, D, E and F. Through the update strategy, select solutions A, C, E and F to be placed in the external archive. The pseudo-codes of the update strategy for the external archive are described in Algorithm 2.

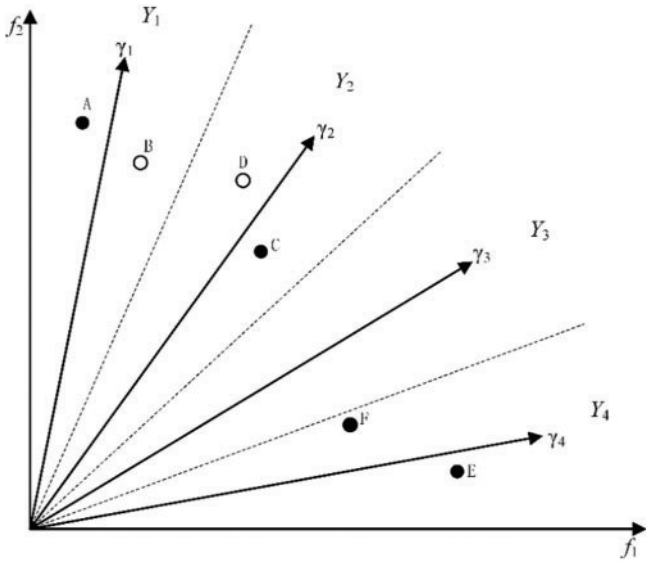

Figure 2: Update strategy for external archive

Algorithm 2: Update strategy for external archive

Input:

POP (offspring population)

(Continued)

Algorithm 2 (continued)

E (external archive)
N (size of external archive)
$(\gamma_1, \gamma_2, \ldots, \gamma_N)$ (N uniformly distributed direction vectors)

Output:

E (updated external archive)

$E' = E \cup POP$, $E = \varnothing$
Find the non-dominated solutions in E', denoted as P
If $size(P) <= N$ **then**
$E = P$
Else
Classify P by Eq. (5) and get $(S_1, S_2, \ldots, S_N)$ corresponding to $(\lambda_1, \lambda_2, \ldots, \lambda_N)$
For $i = 1: N$ **do**
Select the solution with the smallest angle between the objective vector and the direction vector γ_i from S_i and put it into E
End
If $size(E) < N$ **then**
Find the solution farthest from E from $P - \mathrm{E}$ and add it to E until $size(E) = N$
End
End

Both the evolving population and the external archive are candidate optimal solution sets. After the iteration is completed, the inverted generational distance index (IGD) [32] values of the two are calculated, respectively. This performance indicator can comprehensively reflect the convergence and diversity of the solution set, and select a set of solutions with the smaller IGD value for output. The IGD value is calculated as follows:

$$IGD\,(S, S^*) = \frac{\sum_{x^* \in S^*} d(x^*, S)}{|S^*|} \tag{6}$$

where S is the solution set obtained by the algorithm, S^* is the set of points uniformly distributed on the real PF, $d(x^*, S)$ represents the minimum Euclidean distance from the solution $x^* \in S^*$ to the solution in S, $|S^*|$ represents the number of solutions in S^*. The smaller the IGD value, the closer the non-dominated solution set is to the true PF, the more uniform the distribution, and the better the convergence and diversity.

3.3 Multi-Selection Strategy

To better balance diversity and convergence and mitigate the potential negative impact of relying on a single gbest, which can cause diversity loss and PF discontinuity, this article introduces a multi-selection strategy. This strategy primarily involves selecting suitable gbest and pbest values for representative particles based on the evolution of different subspaces.

Firstly, find the T nearest direction vectors of the direction vector γ_i of subspace Y_i, i.e., $B_i = \{i_1, i_2, \ldots, i_T\}, i = (1, 2, \ldots, N)$. The representative solutions of the subspaces corresponding to the T direction vectors are the neighbor solutions BB_i of the representative solution x_i of the subspace Y_i.

Next, the multi-selection strategy is as follows:

(1) When the subspace Y_i has no non-dominated solution, its convergence is poor, but its representative solution is crucial for maintaining diversity, so it is needed to search for non-dominated solutions with good convergence as soon as possible. Since using neighborhood information can speed up the search of the algorithm, the particle with the smallest PBI value is selected as $gbest_i$ among T neighbor particles, and the particle far away from the representative particle i and $gbest_i$ among T neighbor the particle (that is, the neighbor particle with the largest angle formed with the representative particle i and $gbest_i$) is selected as $pbest_i$. The definition formula of PBI is as follows [33]:

$$\min g^{pbi}(x|\lambda, Z) = d_1 + \theta d_2 \tag{7}$$

$$d_1 = \frac{\left\|(Z - F(x))^T \lambda\right\|}{\|\lambda\|} \tag{8}$$

$$d_2 = \|F(x) - (Z - d_1 \lambda)\| \tag{9}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_M)^T$ is a set of weight vectors, $\theta > 0$ is a preset penalty parameter, which is usually set to 5. d_1 is the distance between the projection of $F(x)$ in the direction λ and the reference point Z, and d_2 is the distance between $F(x)$ and the weight vector λ.

(2) When the subspace Y_i has non-dominated solutions, its convergence is good and the local search needs to be strengthened. Therefore, among the T neighbor particles, one particle is randomly selected as $gbest_i$, and another different neighbor particle is selected as $pbest_i$ to save storage space.

(3) When the subspace Y_i has no non-dominated solution for several consecutive generations, PF is considered to be discontinuous. To reduce the influence of PF discontinuity on the solution set, the representative particle update of the subspace should be avoided in the subsequent iteration. A *Count* attribute is set for each subspace, which records the number of iterations without non-dominated solutions in each subspace, and a threshold *TCount* is set. When the $Count_i$ value of subspace Y_i continuously increases to the threshold *TCount*, it means that PF is discontinuous. In subsequent iterations, the search for this subspace is avoided and a probability distribution strategy is used to select the subspace for search in the remaining subspace. Probability P_i is defined as follows:

$$P_i = \frac{1 - NDP_i}{\sum_j^N (1 - NDP_j)}, i = 1, 2, \ldots N \tag{10}$$

where NDP_i represents the non-dominated proportion (NDP) of subspace Y_i. The calculation of NDP_i as follows:

$$NDP_i = \frac{n_i}{\sum_{j=1}^N n_j} \tag{11}$$

where n_i represents the number of non-dominated solutions in subspace Y_i. If $Count_i$ of the subspace Y_i reaches the threshold *Tcount*, then NDP_i is set to 1, and probability P_i is 0. Algorithm 3 shows the calculation process of N subspaces selected probability.

Roulette wheel selection is employed to choose subspaces based on their probability distribution. Subspaces without non-dominated solutions have a higher likelihood of being chosen, although this does not preclude the selection of subspaces containing non-dominated solutions. The corresponding

gbest and pbest values are then determined based on whether the selected subspace includes or excludes non-dominated solutions.

The pseudo-codes of the above selection strategies are described in Algorithm 4.

Algorithm 3: Probability distribution calculation

Input:

$TCount$ (threshold of $Count$)

$(Count_1, Count_2, \ldots, Count_N)$ (the cumulative number of iterations of N subspaces without non-dominated solutions)

Output:

$(P_1, P_2, \ldots, P_N)$ (the selected probability distribution of N subspaces)

For $i = 1 : N$ **do**
Calculate NDP_i by Eq. (11)
If $Count_i >= TCount$ **then**
$NDP_i = 1$
End
Calculate P_i by Eq. (10)
End For

Algorithm 4: Selection strategy

Input:

$(x_1, x_2, \ldots, x_N)$ (the position of N particles)

$(v_1, v_2, \ldots, v_N)$ (the velocity of N particles)

$TCount$ (threshold of $Count$)

$(Count_1, Count_2, \ldots, Count_N)$ (the cumulative number of iterations of N subspaces without non-dominated solutions)

$(P_1, P_2, \ldots, P_N)$ (the selected probability distribution of N subspaces)

Output:

$(x_1, x_2, \ldots, x_N)$ (the position of N particles)

$(v_1, v_2, \ldots, v_N)$ (the velocity of N particles)

For $i = 1 : N$ **do**
If the subspace Y_i has no non-dominated solution **then**
$gbest_i$ ←Select the particle with the smallest PBI value from BB_i by Eqs. (7)–(9)
$pbest_i$ ←Select the particle farthest from the representative particle i and $gbest_i$ from BB_i
Else
$gbest_i$ ←Randomly select a particle from BB_i
$pbest_i$ ←Randomly select a particle different from $gbest_i$ from BB_i
End
If $Count_i < TCount$ **then**
Update v_i and x_i by Eqs. (2) and (3)
Else
Subspace Y_j (j $= 1, 2, \ldots, N$) is selected by roulette wheel selection according to probability $(P_1, P_2, \ldots, P_N)$
Update v_j and x_j by Eqs. (2) and (3)

(Continued)

Algorithm 4 (continued)

14: **End**
15: **End For**

3.4 Adaptive Inertia Weight Update Strategy

Inertia weight regulates the speed of particle movement and influences the balance between exploration and exploitation. Higher inertia weights enhance particles' exploration capabilities, while lower weights boost their exploitation abilities. An adaptive inertia weight update strategy is implemented as follows:

$$w_i^t = w_{max} - \frac{(w_{max} - w_{min})\, t}{t_{max}} \exp(-R_i) \tag{12}$$

$$R_i = \frac{1}{M} \sum_{j=1}^{M} \frac{f_j(x_i) - Z_j}{Z_j^* - Z_j} \tag{13}$$

where w_{max} and w_{min} are the upper and lower bounds of inertia weight, respectively, and t_{max} is the maximum number of iterations. R_i denotes the degree to which the representative solution x_i is dominated by the reference point Z. $Z_j^* = \max\{f_j(x) | x \in \Omega\} (j = 1, 2, \ldots, M)$ denotes the maximum value of all representative solutions on the jth objective function.

The inertia weight update strategy takes into account different iteration stages and the degree of each representative particle dominated by Z. For example, in the early iteration, particles are encouraged to explore areas that have not been explored, so even if a representative particle is dominated by Z to a small extent, its inertia weight will not decrease much immediately. In the same iteration, the inertia weight of the representative particle that is dominated by Z to a greater extent is larger, and *vice versa* is smaller, so as to strengthen exploration or exploitation.

3.5 Crossover and Mutation

The crossover operation of PSO can update the position and velocity of particles, which is described in Section 2.2.

The mutation operation uses polynomial mutation [34], and the calculation process is as follows:

$$P_k' = P_k + \delta(u_k - l_k) \tag{14}$$

$$\delta = \begin{cases} [2u + (1-2u)(1-\delta_1)^{\eta m+1}]^{\frac{1}{\eta m+1}} - 1, & if\ u \le 0.5 \\ 1 - [2(1-u) + 2(u-0.5)(1-\delta_2)^{\eta m+1}]^{\frac{1}{\eta m+1}}, & if\ u > 0.5 \end{cases} \tag{15}$$

where P_k denotes the population of the previous generation, P_k' denotes the newly generated population after polynomial mutation, u_k denotes the upper bound of the variable, l_k denotes the lower bound of the variable, u is a random number in [0, 1], η_m is the distribution index selected by the user, and the calculation of δ_1 and δ_2 is as follows:

$$\delta_1 = \frac{P_k - l_k}{u_k - l_k} \tag{16}$$

$$\delta_2 = \frac{u_k - P_k}{u_k - l_k} \tag{17}$$

3.6 Framework of Proposed Algorithm

Based on the above content, a multi-objective particle swarm optimization algorithm based on decomposition and multi-selection strategy (MOPSO/DMS) is proposed to improve the search efficiency and balance diversity and convergence. The overall framework of MOPSO/DMS is given in Algorithm 5. Firstly, a population POP is generated randomly, the velocity $(v_1, v_2, \ldots, v_N)$ and position $(x_1, x_2, \ldots, x_N)$ of particles are initialized, and set external archive $E = \varnothing$. A set of uniformly distributed direction vectors is used to divide the objective space and allocate POP. Then, according to the evolution of each subspace, the selection strategy of Section 3.3 is used to select the appropriate gbest and pbest for the search particles, and the position $(x_1, x_2, \ldots, x_N)$ and velocity $(v_1, v_2, \ldots, v_N)$ of the particles are updated through the crossover and mutation operations with the adaptive inertial weight update strategy of Section 3.4. Next, set $POP = POP \cup \{x_1, x_2, \ldots, x_N\}$, $E = E \cup POP$, and then update POP and E using the update strategy of Section 3.2. The iteration stops until the stopping condition is met. Finally, the one with the smaller IGD value of POP and E is the optimal solution set.

Algorithm 5: General framework of MOPSO/DMS

Input:

MOP
N (population size and external archive size)
$(\gamma_1, \gamma_2, \ldots, \gamma_N)$ (N uniformly distributed direction vectors)
T (neighbor size)
$TCount$ (threshold of $Count$)
stopping criterion

Output:

$EPOP$

1: Randomly generate an initial population POP, initialize the velocity $(v_1, v_2, \ldots, v_N)$ and position $(x_1, x_2, \ldots, x_N)$, determine the reference point $Z = (Z_1, \ldots, Z_M)$and $Z^* = (Z_1^*, \ldots, Z_M^*)$, determine $B_i = \{i_1, i_2, \ldots, i_T\}, i = (1, 2, \ldots, N)$ and BB_i, $Count_i = 0, i = (1, 2, \ldots, N)$, $EPOP = \varnothing$, $E = \varnothing$
2: **While** the stopping criterion is not met **do**
3: Divide solutions of POP into N classes by Eq. (5)
4: Use the selection strategy of Section 3.3, the adaptive inertial weight update strategy of Section 3.4 and the crossover and mutation operations of Section 3.5 to update $\{v_1, v_2, \ldots, v_N\}$ and $\{x_1, x_2, \ldots, x_N\}$, $POP = POP \cup \{x_1, x_2, \ldots, x_N\}$, $E = E \cup POP$
5: For each $j = 1, \ldots, M$, if $Z_j > \min\{f_j(x) | x \in E\}$, then update $Z_j = \min\{f_j(x) | x \in E\}$, and if $Z_j^* < \max\{f_j(x) | x \in E\}$, then update $Z_j^* = \max\{f_j(x) | x \in E\}$
6: Use update strategy of Section 3.2 to update POP and E, respectively
7: **End While**
8: **If** IGD (POP) < IGD (E) **then**
9: $EPOP = POP$
10: **Else**
11: $EPOP = E$

4 Experimental Study

4.1 Comparison Algorithms and Test Problems

To evaluate the performance and effectiveness of MOPSO/DMS, this study initially compares it with five prevalent MOPSO algorithms: competitive mechanism-based multi-objective particle swarm optimizer (CMOPSO) [35], dMOPSO [22], multi-objective particle swarm optimization with multiple search strategies (MMOPSO) [36], MPSO/D [24], speed-constrained multi-objective particle swarm optimization algorithm (SMPSO) [37], in which MPSO/D [24] and dMOPSO [22] are introduced in Section 1. CMOPSO [35] makes the particles update based on pairwise competition in each generation. MMOPSO [36] designs two search strategies for particle update, which are conducive to speeding up convergence and maintaining diversity, respectively. SMPSO [37] uses a strategy of limiting the velocity of the particles and incorporates polynomial mutation as a turbulence factor. Then MOPSO/DMS is compared with five current mainstream MOEAs including multi-objective evolutionary algorithm based on decomposition (MOEA/D) [33], MOEA/D based on differential evolution (MOEA/D-DE) [38], multi-objective evolutionary algorithm based on decision variable analyses (MOEA/DVA) [39], non-dominated neighbor immune algorithm (NNIA) [40] and clustering-based adaptive multi-objective evolutionary algorithm (CA-MOEA) [41].

All algorithms are compared on ZDT [35], DTLZ [42] and UF [43] test problems. The ZDT [35] test problem set is a widely used multi-objective test problem set, including six different forms of ZDT1-6 test problems. Each test problem has two objective functions and has different shapes of PF. Since the variables in the ZDT5 test problem are encoded in binary, ZDT1-4 and ZDT6 are selected for testing in this article. The DTLZ [42] test problem set is a kind of standard test problem set, but the number of objective functions is self-defined, usually greater than or equal to two, and has different shapes of PF. UF1-10 [43] are multi-objective test problems proposed in CEC2009, where UF1-7 are the bi-objective test functions and UF8-10 are the three-objective test functions, and has different PFs.

4.2 Parameter Settings

The number of objectives M, the number of decision variables n and the shape of PF for all test problems are shown in Table 1. Each algorithm runs independently 30 times on each test problem. For the bi-objective ZDT1-4, ZDT6 and UF1-7 test problems, the population size N is 100 and the maximum number of function evaluations is 30,000. Meanwhile, for the three-objective DTLZ1-7 and UF8-10 test problems, $N = 150$, the maximum number of function evaluations is 100,000. The initial value of inertia weight w is set to 0.9, c_1 and c_1 are 2. The crossover probability is 1, the mutation probability is $1/n$, and the distribution index for mutation η_m is 20. The Wilcoxon rank sum test with a significance level of 0.05 is used to analyze the experimental results of all algorithms. The other algorithm parameter settings are based on the pertinent settings found in the original article to ensure experiment fairness. The size of elite particle set γ is set to 10. The parameter Ta (age threshold) in dMOPSO is set to 2. In MOEA/D-DE, the parameter δ (probability of selecting the parent solution from the neighborhood) is set to 0.9 and the parameter n_r (maximal number of solutions replaced by each child solution) is set to 2. In MOEA/DVA, the parameter NCA (number of sampling solutions that recognize the control property of decision variable) is set to 20 and the parameter NIA (maximum number of tries required to judge the interaction between two variables) is set to 6. In NNIA, the parameter n_A (maximum size of active population) is set to 20 and the parameter n_C (size of clone population) is set to 100.

Table 1: Parameter setting of test problems

Problems	M	n	PF shapes
ZDT1	2	30	Convex
ZDT2	2	30	Concave
ZDT3	2	30	Convex, disconnected
ZDT4	2	10	Convex
ZDT6	2	10	Concave, nonuniform distribution
DTLZ1	3	7, $K = 5$	Linear
DTLZ2	3	12, $K = 10$	Concave
DTLZ3	3	12, $K = 10$	Concave
DTLZ4	3	12, $K = 10$	Concave
DTLZ5	3	12, $K = 10$	Degenerated
DTLZ6	3	12, $K = 10$	Degenerated
DTLZ7	3	22, $K = 20$	Mixed, irregular, disconnected
UF1	2	30	Convex
UF2	2	30	Convex
UF3	2	30	Convex
UF4	2	30	Concave
UF5	2	30	Disconnected
UF6	2	30	Disconnected
UF7	2	30	Linear
UF8	3	30	Concave
UF9	3	30	Linear, disconnected
UF10	3	30	Concave

4.3 Performance Indicator

In the experiment, hypervolume (HV) [44] and IGD [32] are used to evaluate the performance of the algorithms. HV measures the volume of the objective space between the non-dominated solution set obtained by the algorithm and the reference point, which is strictly monotone in the Pareto dominance relation. HV can be used for problems with unknown real PF or reference set without the existence of known real PF or reference set, and the diversity and convergence can be evaluated simultaneously. The larger the value of HV, the better the performance of the algorithm. The calculation is as follows:

$$HV = \delta\left(\bigcup_{i=1}^{|S|} v_i\right) \tag{18}$$

where δ represents the Lebesgue measure, which is used to measure the volume. $|S|$ denotes the number of non-dominated solutions, and v_i denotes the hypercube of the reference point and the ith solution in the solution set. In the comparative experiment of this article, the reference points are set as (1.1,1.1) and (1.1,1.1,1.1).

IGD measures the average Euclidean distance between all solutions in the actual Pareto front and the non-dominated solutions generated by the algorithm. It not only reflects the convergence of the solution set but also indicates the uniformity and diversity of the distribution. A lower IGD value signifies superior overall algorithm performance. The detailed calculation is given in Eq. (6).

4.4 Experimental Results and Analysis

This section presents the average and standard deviation of HV and IGD values obtained from running all algorithms independently 30 times across all test problems. The best results for each test question are marked in black bold. In addition, "+" means that the comparison algorithm is better than MOPSO/DMS, "−" means that the comparison algorithm is worse than MOPSO/DMS, and "=" means that the performance of the comparison algorithm is comparable to that of MOPSO/DMS. Running the algorithms 30 times independently helps calculate the average and standard deviation, minimizing chance effects and reflecting the central tendency of the experimental results.

4.4.1 Comparison with Other MOPSOs and MOEAs

Table 2 presents the HV averages and standard deviations for the five MOPSOs after 30 independent runs on all test problems. In contrast, Table 3 displays the HV averages and standard deviations for MOPSO/DMS and five MOEAs. Across 22 test problems, MOPSO/DMS attains the highest HV values on 20 problems compared to the other five MOPSOs and surpasses the five comparison MOEAs on 19 problems.

Table 2: HV average and standard deviation of MOPSO/DMS and five MOPSOs

Problems	*M*	CMOPSO	dMOPSO	MMOPSO	MPSO/D	SMPSO	MOPSO/DMS
ZDT1	2	7.1852e−1	6.8547e−1	7.1917e−1	1.0615e−1	7.0750e−1	**7.1962e−1**
		(6.29e−4) =	(1.37e−2) −	(2.86e−4) =	(2.07e−1) −	(4.21e−2) −	(**7.08e−5**)
ZDT2	2	4.4417e−1	2.1017e−1	3.9000e−1	4.3671e−2	4.4403e−1	**4.4464e−1**
		(5.95e−4) =	(1.60e−1) −	(1.23e−1) −	(1.06e−1) −	(3.13e−4) =	(**5.32e−5**)
ZDT3	2	5.9751e−1	5.2657e−1	5.7172e−1	7.3468e−2	5.1025e−1	**9.9071e−1**
		(8.57e−2) −	(1.03e−1) −	(5.28e−2) −	(1.60e−1) −	(1.52e−1) −	(**6.80e−4**)
ZDT4	2	0.0000e+0	3.5053e−1	1.3620e−2	0.0000e+0	0.0000e+0	**6.8905e−1**
		(0.00e+0) −	(1.44e−1) −	(3.39e−2) −	(0.00e+0) −	(0.00e+0) −	(**8.66e−2**)
ZDT6	2	**3.8866e−1**	3.8862e−1	3.8774e−1	3.8699e−1	3.8817e−1	3.8813e−1
		(**1.67e−4**) +	(4.88e−4) +	(1.84e−4) −	(1.23e−3) −	(1.81e−4) =	(1.27e−6)
DTLZ1	3	0.0000e+0	1.0835e−2	0.0000e+0	0.0000e+0	1.1218e−1	**8.1540e−1**
		(0.00e+0) −	(5.93e−2) −	(0.00e+0) −	(0.00e+0) −	(1.76e−1) −	(**1.01e−3**)
DTLZ2	3	5.5285e−1	4.0936e−1	5.3553e−1	5.5127e−1	4.9494e−1	**5.7070e−1**
		(1.73e−3) −	(1.55e−2) −	(3.63e−3) −	(2.94e−3) −	(1.20e−2) −	(**3.06e−5**)
DTLZ3	3	0.0000e+0	1.5921e−3	0.0000e+0	0.0000e+0	1.8329e−2	**5.7081e−1**
		(0.00e+0) −	(8.72e−3) −	(0.00e+0) −	(0.00e+0) −	(4.89e−2) −	(**2.33e−5**)
DTLZ4	3	5.4817e−1	2.9939e−1	5.4017e−1	5.0025e−1	3.5304e−1	**5.7066e−1**
		(3.10e−3) −	(5.21e−2) −	(3.20e−3) −	(4.03e−2) −	(8.38e−2) −	(**3.14e−5**)
DTLZ5	3	1.9831e−1	1.6232e−1	**1.9961e−1**	1.4214e−1	1.9950e−1	1.9940e−1
		(4.09e−4) −	(1.07e−2) −	(**2.84e−4**) =	(8.91e−3) −	(3.57e−4) =	(2.05e−4)

(Continued)

Table 2 (continued)

Problems	M	CMOPSO	dMOPSO	MMOPSO	MPSO/D	SMPSO	MOPSO/DMS
DTLZ6	3	1.7172e−1	1.8682e−1	2.0039e−1	4.7693e−2	2.6733e−2	**2.0086e−1**
		(6.95e−2) −	(2.73e−4) −	(9.60e−5) =	(5.02e−2) −	(6.93e−2) −	(**5.00e−5**)
DTLZ7	3	2.6635e−1	2.4483e−1	2.6012e−1	1.0710e−1	1.1586e−1	**2.8201e−1**
		(9.61e−3) −	(5.15e−3) −	(2.11e−2) −	(4.14e−2) −	(8.02e−2) −	(**5.77e−4**)
UF1	2	5.7681e−1	5.8660e−2	5.7148e−1	5.0103e−1	2.3873e−1	**6.9553e−1**
		(2.34e−2) −	(5.66e−2) −	(3.40e−2) −	(2.59e−2) −	(7.32e−2) −	(**8.72e−3**)
UF2	2	6.4863e−1	6.0805e−1	6.4274e−1	6.0598e−1	6.0200e−1	**7.1738e−1**
		(6.69e−3) −	(6.62e−3) −	(5.03e−3) −	(1.03e−2) −	(6.16e−3) −	(**3.92e−4**)
UF3	2	3.3733e−1	2.9085e−1	3.0739e−1	2.1338e−1	1.8236e−1	**7.0484e−1**
		(4.64e−2) −	(1.16e−2) −	(4.65e−2) −	(1.92e−2) −	(2.35e−2) −	(**8.80e−3**)
UF4	2	2.9055e−1	2.5940e−1	3.7164e−1	3.2826e−1	2.9185e−1	**4.4469e−1**
		(1.41e−2) −	(7.68e−3) −	(3.21e−3) −	(7.71e−3) −	(1.24e−2) −	(**1.88e−5**)
UF5	2	1.0517e−2	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	**5.8386e−2**
		(2.17e−2) −	(0.00e+0) −	(0.00e+0) −	(0.00e+0) −	(0.00e+0) −	(**3.66e−2**)
UF6	2	2.3540e−1	0.0000e+0	1.0704e−1	3.1426e−3	4.5334e−3	**3.2282e−1**
		(5.74e−2) −	(0.00e+0) −	(8.05e−2) −	(9.50e−3) −	(1.20e−2) −	(**5.85e−2**)
UF7	2	4.2868e−1	1.5356e−1	4.3445e−1	3.9816e−1	1.5294e−1	**5.5374e−1**
		(1.06e−1) −	(8.08e−2) −	(1.01e−1) −	(5.71e−2) −	(8.24e−2) −	(**7.76e−3**)
UF8	3	1.8404e−2	2.7455e−1	2.8679e−1	5.8485e−2	1.2380e−1	**5.6301e−1**
		(1.80e−2) −	(2.51e−2) −	(2.10e−2) −	(2.30e−2) −	(5.50e−2) −	(**1.10e−3**)
UF9	3	1.8064e−2	2.1140e−1	3.3659e−1	1.2352e−1	1.5956e−1	**7.8987e−1**
		(2.19e−2) −	(2.11e−2) −	(4.72e−2) −	(3.02e−2) −	(5.14e−2) −	(**5.20e−3**)
UF10	3	0.0000e+0	9.0919e−2	2.9531e−3	0.0000e+0	0.0000e+0	**2.8184e−1**
		(0.00e+0) −	(5.18e−5) −	(8.72e−3) −	(0.00e+0) −	(0.00e+0) −	(**8.28e−2**)
+/−/=		1/19/2	1/21/0	0/19/3	0/22/0	0/19/3	——

Table 3: HV average and standard deviation of MOPSO/DMS and five MOEAs

Problems	M	MOEA/D	MOEA/ D-DE	MOEA/ DVA	NNIA	CA-MOEA	MOPSO /DMS
ZDT1	2	**8.7157e−1**	9.7309e−3	1.1919e−1	6.1733e−1	5.3288e−1	7.1962e−1
		(**3.26e−6**) +	(3.12e−2) −	(2.91e−2) −	(5.84e−2) −	(7.17e−2) −	(7.08e−5)
ZDT2	2	**5.3489e−1**	4.5953e−4	1.1950e−1	9.5326e−2	2.6467e−2	4.4464e−1
		(**6.70e−8**) +	(2.52e−3) −	(3.00e−2) −	(9.11e−2) −	(4.49e−2) −	(5.32e−5)
ZDT3	2	4.5598e−1	4.1375e−3	0.0000e+0	6.2580e−1	5.7712e−1	**9.9071e−1**
		(1.46e−1) −	(1.58e−2) −	(0.00e+0) −	(8.87e−2) −	(9.64e−2) −	(**6.80e−4**)
ZDT4	2	2.2009e−1	0.0000e+0	0.0000e+0	6.6377e−1	3.1564e−1	**6.8905e−1**
		(1.44e−1) −	(0.00e+0) −	(0.00e+0) −	(5.04e−2) =	(1.74e−1) −	(**8.66e−2**)
ZDT6	2	2.7981e−1	3.1656e−1	1.2378e−1	3.6599e−1	1.1730e−1	**3.8813e−1**
		(2.44e−2) −	(9.89e−2) −	(7.11e−2) −	(1.17e−2) =	(4.53e−2) −	(**1.27e−6**)

(Continued)

Table 3 (continued)

Problems	M	MOEA/D	MOEA/ D-DE	MOEA/ DVA	NNIA	CA-MOEA	MOPSO /DMS
DTLZ1	3	5.2753e−1	3.0008e−1	0.0000e+0	4.5913e−1	9.9563e−2	**8.1540e−1**
		(3.39e−1) −	(3.43e−1) −	(0.00e+0) −	(3.59e−1) −	(1.62e−1) −	(**1.01e−3**)
DTLZ2	3	5.6009e−1	5.3244e−1	4.6710e−1	5.4838e−1	5.5557e−1	**5.7070e−1**
		(1.84e−3) −	(2.77e−3) −	(5.62e−3) −	(2.72e−3) −	(1.96e−3) −	(**3.06e−5**)
DTLZ3	3	0.0000e+0	3.3448e−2	0.0000e+0	0.0000e+0	0.0000e+0	**5.7081e−1**
		(0.00e+0) −	(1.15e−1) −	(0.00e+0) −	(0.00e+0) −	(0.00e+0) −	(**2.33e−5**)
DTLZ4	3	4.2938e−1	5.1146e−1	2.0877e−1	5.5051e−1	5.4994e−1	**5.7066e−1**
		(1.41e−1) −	(2.69e−2) −	(3.77e−2) −	(2.34e−3) −	(3.84e−2) −	(**3.14e−5**)
DTLZ5	3	1.8560e−1	1.9571e−1	1.6854e−1	**2.0016e−1**	1.9824e−1	1.9940e−1
		(1.49e−3) −	(2.89e−4) −	(5.43e−3) −	(**1.22e−4**) +	(4.55e−4) −	(2.05e−4)
DTLZ6	3	1.5720e−1	1.9549e−1	8.3751e−3	2.0045e−1	1.9524e−1	**2.0086e−1**
		(5.64e−2) −	(7.37e−3) −	(2.10e−2) −	(9.07e−4) =	(7.02e−3) −	(**5.00e−5**)
DTLZ7	3	2.2818e−1	5.4292e−2	1.7283e−5	2.6068e−1	2.4510e−1	**2.8201e−1**
		(1.51e−2) −	(4.07e−2) −	(9.47e−5) −	(8.76e−3) −	(4.80e−3) −	(**5.77e−4**)
UF1	2	4.1015e−1	5.0457e−1	2.9494e−1	5.0129e−1	5.6980e−1	**6.9553e−1**
		(5.75e−2) −	(4.26e−2) −	(3.96e−2) −	(4.75e−2) −	(2.88e−2) −	(**8.72e−3**)
UF2	2	5.6885e−1	6.4001e−1	4.7023e−1	6.4132e−1	6.4116e−1	**7.1738e−1**
		(3.29e−2) −	(1.57e−2) −	(1.27e−2) −	(1.15e−2) −	(5.27e−3) −	(**3.92e−4**)
UF3	2	3.1600e−1	3.4678e−1	1.0313e−1	2.9621e−1	2.4883e−1	**7.0484e−1**
		(4.91e−2) −	(2.15e−2) −	(1.71e−2) −	(4.45e−2) −	(3.37e−2) −	(**8.80e−3**)
UF4	2	2.7068e−1	3.0814e−1	3.0643e−1	3.3444e−1	3.4148e−1	**4.4469e−1**
		(6.38e−3) −	(8.50e−3) −	(2.62e−3) −	(5.33e−3) −	(2.91e−3) −	(**1.88e−5**)
UF5	2	0.0000e+0	0.0000e+0	0.0000e+0	1.3419e−2	5.0292e−2	**5.8386e−2**
		(0.00e+0) −	(0.00e+0) −	(0.00e+0) −	(2.59e−2) −	(5.51e−2) −	(**3.66e−2**)
UF6	2	1.0233e−1	4.8943e−2	0.0000e+0	1.4644e−1	8.2584e−2	**3.2282e−1**
		(7.68e−2) −	(3.72e−2) −	(0.00e+0) −	(9.37e−2) −	(6.70e−2) −	(**5.85e−2**)
UF7	2	2.0744e−1	3.3866e−1	1.3939e−1	3.1586e−1	4.1111e−1	**5.5374e−1**
		(7.34e−2) −	(1.09e−1) −	(2.93e−2) −	(9.38e−2) −	(7.62e−2) −	(**7.76e−3**)
UF8	3	1.4933e−1	2.5461e−1	3.9425e−4	1.2091e−1	2.9444e−1	**5.6301e−1**
		(6.40e−2) −	(2.68e−2) −	(1.38e−3) −	(1.12e−1) −	(2.69e−2) −	(**1.10e−3**)
UF9	3	2.7904e−1	3.6017e−1	7.1285e−3	7.4661e−2	3.0574e−1	**7.8987e−1**
		(5.54e−2) −	(5.72e−2) −	(1.22e−2) −	(8.14e−2) −	(5.08e−2) −	(**5.20e−3**)
UF10	3	4.1457e−2	0.0000e+0	0.0000e+0	0.0000e+0	2.5762e−5	**2.8184e−1**
		(2.85e−2) −	(0.00e+0) −	(0.00e+0) −	(0.00e+0) −	(1.41e−4) −	(**8.28e−2**)
+/−/=		2/20/0	0/22/0	0/22/0	1/18/3	0/22/0	——

In Table 2, CMOPSO, dMOPSO, MMOPSO, MOPSO/D and SMPSO are better than MOPSO/DMS on 1, 1, 0, 0 and 0 test problems, respectively. MOPSO/DMS is better than CMOPSO, dMOPSO, MMOPSO, MOPSO/D and SMPSO on 19, 21, 19, 22 and 19 test problems, respectively.

The HV values obtained by MOPSO/DMS are not significantly different from those of CMOPSO, dMOPSO, MMOPSO, MOPSO/D and SMPSO on 2, 0, 3, 0 and 3 test problems, respectively.

In Table 3, MOEA/D, MOEA/D-DE, MOEA/DVA, NNIA and CA-MOEA perform better than MOPSO/DMS on 2, 0, 0, 1 and 0 test problems, respectively. MOPSO/DMS performs better than MOEA/D, MOEA/D-DE, MOEA/DVA, NNIA and CA-MOEA on 20, 22, 22, 18 and 22 test problems, respectively. The HV values of MOPSO/DMS were not significantly different from those of NNIA on ZDT4, ZDT6 and DTLZ6.

In Tables 2 and 3, for 8 test problems with concave PF (including ZDT2, ZDT6, DTLZ2-4, UF4, UF8 and UF10), the HV values of MOPSO/DMS on 6, 7, 8, 8 and 6 test problems are greater than those of CMOPSO, dMOPSO, MMOPSO, MOPSO/D and SMPSO, respectively. And the HV values on 7, 8, 8, 7 and 8 test problems are greater than MOEA/D, MOEA/D-DE, MOEA/DVA, NNIA and CA-MOEA, respectively. For 6 test problems with convex PF (including ZDT1, ZDT3, ZDT4 and UF1-3), MOPSO/DMS performs better than CMOPSO, dMOPSO, MMOPSO, MOPSO/D and SMPSO on 5, 6, 5, 6 and 6 test problems, respectively. And it performs better than MOEA/D, MOEA/D-DE, MOEA/DVA, NNIA and CA-MOEA on 5, 6, 6, 5 and 6 test problems, respectively. For DTLZ5 with degenerated PF, NNIA performs better than MOPSO/DMS, MMOPSO and SMPSO perform similar to MOPSO/DMS, and the other seven comparison algorithms is worse than MOPSO/DMS. On DTLZ6 with degenerated PF, only MMOPSO and NNIA perform similarly to MOPSO/DMS, and the remaining comparison algorithms perform worse. MOPSO/DMS achieves larger HV values on ZDT3, UF5-6 and UF9 with disconnected PF. In addition, MOPSO/DMS has the largest HV value on DTLZ1 and UF7 with linear PF, while its HV value on DTLZ7 with complex PF is also larger than other ten comparison algorithms.

Table 4 shows the IGD average and standard deviation of five MOPSOs obtained by running 30 times independently on all test problems. Table 5 shows the IGD average and standard deviation of MOPSO/DMS and five MOEAs.

Table 4: IGD average and standard deviation of MOPSO/DMS and five MOPSOs

Problems	M	CMOPSO	dMOPSO	MMOPSO	MPSO/D	SMPSO	MOPSO/DMS
ZDT1	2	4.4430e−3	2.9824e−2	5.1781e−3	3.4449e+0	1.7694e−2	**3.9273e−3**
		(2.77e−4) −	(1.14e−2) −	(3.07e−4) −	(2.80e+0) −	(5.36e−2) −	(**6.64e−5**)
ZDT2	2	4.0546e−3	3.9152e−1	8.1678e−2	6.9355e+0	4.9142e−3	**3.7904e−3**
		(1.68e−4) =	(2.88e−1) −	(1.77e−1) −	(6.21e+0) −	(2.47e−4) −	(**2.67e−5**)
ZDT3	2	8.1201e−2	1.7565e−1	4.0770e−2	3.2475e+0	**5.3549e−3**	5.4210e−3
		(1.52e−1) −	(2.36e−1) −	(1.31e−1) −	(2.84e+0) −	(**1.14e−4**) +	(1.06e−4)
ZDT4	2	8.4590e+0	5.2656e−1	3.7597e+0	3.5226e+1	8.0935e+0	**1.0824e−2**
		(5.53e+0) −	(2.06e−1) −	(4.67e+0) −	(6.51e+0) −	(4.44e+0) −	(**1.82e−2**)
ZDT6	2	3.2684e−3	3.3535e−3	4.2664e−3	4.5000e−3	3.8603e−3	**2.7368e−3**
		(1.47e−4) =	(4.16e−4) −	(1.90e−4) −	(1.35e−3) −	(2.19e−4) −	(**1.72e−6**)
DTLZ1	3	1.3980e+1	1.0028e+1	2.5071e+0	8.6788e+0	3.3508e+0	**1.5389e−2**
		(3.80e+0) −	(6.12e+0) −	(1.84e+0) −	(2.12e+0) −	(3.45e+0) −	(**4.40e−5**)
DTLZ2	3	4.7341e−2	1.2056e−1	6.0370e−2	4.8525e−2	7.0634e−2	**3.9856e−2**
		(6.66e−4) −	(7.66e−3) −	(1.69e−3) −	(9.78e−4) −	(5.09e−3) −	(**2.95e−5**)

(Continued)

Table 4 (continued)

Problems	M	CMOPSO	dMOPSO	MMOPSO	MPSO/D	SMPSO	MOPSO/DMS
DTLZ3	3	1.6173e+2	5.3008e+1	9.1678e+1	1.3622e+2	4.6183e+1	**3.9836e−2**
		(4.07e+1) −	(5.08e+1) −	(2.81e+1) −	(1.58e+1) −	(4.16e+1) −	(**1.95e−5**)
DTLZ4	3	4.9268e−2	3.2361e−1	5.9265e−2	8.4652e−2	3.8013e−1	**3.9851e−2**
		(1.25e−3) −	(3.39e−2) −	(1.34e−3) −	(3.25e−2) −	(1.54e−1) −	(**3.42e−5**)
DTLZ5	3	5.4745e−3	3.2630e−2	4.6178e−3	5.4551e−2	**4.2280e−3**	4.3076e−3
		(4.01e−4) −	(4.95e−3) −	(3.77e−4) =	(6.30e−3) −	(**4.41e−4**) +	(2.34e−4)
DTLZ6	3	1.2221e−1	2.5636e−2	4.6292e−3	6.0955e−1	1.6213e+0	**2.8620e−3**
		(3.01e−1) −	(3.05e−4) −	(5.26e−4) −	(3.79e−1) −	(9.03e−1) −	(**4.13e−5**)
DTLZ7	3	8.5869e−2	1.1940e−1	1.8947e−1	4.1292e−1	6.2094e−1	**4.6439e−2**
		(8.74e−2) −	(1.24e−2) −	(2.06e−1) −	(1.00e−1) −	(3.57e−1) −	(**9.77e−4**)
UF1	2	1.0939e−1	6.6361e−1	1.1040e−1	1.5281e−1	4.1190e−1	**1.8035e−2**
		(3.51e−2) −	(1.10e−1) −	(5.44e−2) −	(2.15e−2) −	(8.92e−2) −	(**5.02e−3**)
UF2	2	6.1032e−2	1.0456e−1	6.6932e−2	9.1767e−2	1.0502e−1	**4.7487e−3**
		(6.95e−3) −	(9.89e−3) −	(5.86e−3) −	(8.12e−3) −	(6.05e−3) −	(**1.99e−4**)
UF3	2	2.9883e−1	3.4140e−1	3.3061e−1	4.5448e−1	4.8454e−1	**1.1283e−2**
		(5.86e−2) −	(8.78e−3) −	(4.80e−2) −	(1.89e−2) −	(2.73e−2) −	(**7.27e−3**)
UF4	2	1.1039e−1	1.3187e−1	5.4448e−2	8.3135e−2	1.1222e−1	**3.7859e−3**
		(1.03e−2) −	(6.10e−3) −	(2.45e−3) −	(5.72e−3) −	(9.69e−3) −	(**6.82e−7**)
UF5	2	9.2388e−1	2.8395e+0	1.5105e+0	2.2254e+0	2.8742e+0	**4.1212e−1**
		(2.47e−1) −	(4.12e−1) −	(4.45e−1) −	(2.39e−1) −	(5.98e−1) −	(**8.84e−2**)
UF6	2	3.0365e−1	1.9224e+0	4.8660e−1	8.6764e−1	1.2888e+0	**1.4366e−1**
		(5.29e−2) −	(6.30e−1) −	(1.50e−1) −	(1.26e−1) −	(4.78e−1) −	(**4.72e−2**)
UF7	2	1.5795e−1	3.9035e−1	1.3966e−1	1.3019e−1	4.2069e−1	**2.0655e−2**
		(1.57e−1) −	(1.03e−1) −	(1.30e−1) −	(4.56e−2) −	(1.13e−1) −	(**4.82e−3**)
UF8	3	5.9347e−1	3.6244e−1	2.7626e−1	5.2963e−1	4.3901e−1	**4.4330e−2**
		(1.06e−1) −	(4.51e−2) −	(1.25e−2) −	(4.78e−2) −	(7.66e−2) −	(**4.93e−4**)
UF9	3	8.9394e−1	6.1939e−1	4.2563e−1	6.4446e−1	6.1520e−1	**3.4952e−2**
		(1.23e−1) −	(4.22e−2) −	(4.86e−2) −	(3.92e−2) −	(6.34e−2) −	(**2.30e−3**)
UF10	3	4.2617e+0	9.4519e−1	1.0936e+0	4.1326e+0	2.7117e+0	**1.9339e−1**
		(5.40e−1) −	(3.99e−3) −	(3.72e−1) −	(3.90e−1) −	(5.33e−1) −	(**3.81e−2**)
+/−/=		0/20/2	0/22/0	0/21/1	0/22/0	2/20/0	——

Table 5: IGD average and standard deviation of MOPSO/DMS and five MOEAs

Problems	M	MOEA/D	MOEA/ D-DE	MOEA/ DVA	NNIA	CA-MOEA	MOPSO/ DMS
ZDT1	2	**3.8873e−3**	4.2613e+0	5.7612e−1	1.2045e−1	1.7117e−1	3.9273e−3
		(**7.83e−7**) =	(2.60e+0) −	(6.41e−2) −	(9.71e−2) −	(9.67e−2) −	(6.64e−5)
ZDT2	2	3.8070e−3	5.2652e+0	7.7893e−1	4.8582e−1	6.0061e−1	**3.7904e−3**
		(1.30e−8) =	(2.51e+0) −	(1.51e−1) −	(2.23e−1) −	(1.39e−1) −	(**2.67e−5**)

(Continued)

Table 5 (continued)

Problems	M	MOEA/D	MOEA/ D-DE	MOEA/ DVA	NNIA	CA-MOEA	MOPSO/ DMS
ZDT3	2	2.3645e−1	3.5584e+0	7.4289e−1	1.7123e−1	1.7640e−1	**5.4210e−3**
		(1.39e−1) −	(2.25e+0) −	(1.76e−1) −	(1.25e−1) −	(1.04e−1) −	(**1.06e−4**)
ZDT4	2	**1.0096e−2**	3.6987e+0	5.1530e+0	4.8437e−2	3.9687e−1	1.0824e−2
		(**3.95e−3**) +	(1.39e+0) −	(1.55e+0) −	(4.33e−2) −	(2.10e−1) −	(1.82e−2)
ZDT6	2	3.1034e−3	9.7434e−2	8.8206e−1	1.8897e−2	2.8150e−1	**2.7368e−3**
		(2.56e−7) −	(1.47e−1) −	(9.72e−1) −	(8.65e−3) −	(8.07e−2) −	(**1.72e−6**)
DTLZ1	3	2.8457e−2	8.2419e−1	3.8690e−1	2.2093e−1	4.9653e−1	**1.5389e−2**
		(8.38e−5) −	(1.08e+0) −	(1.13e−1) −	(2.37e−1) −	(2.98e−1) −	(**4.40e−5**)
DTLZ2	3	4.5115e−2	6.1656e−2	9.3347e−2	5.7550e−2	4.7676e−2	**3.9856e−2**
		(5.66e−4) −	(8.62e−4) −	(3.34e−3) −	(2.25e−3) −	(8.77e−4) −	(**2.95e−5**)
DTLZ3	3	6.9463e−2	2.7696e+1	6.1863e+1	8.1004e+0	1.6394e+1	**3.9836e−2**
		(2.68e−4) −	(2.89e+1) −	(9.93e+0) −	(4.52e+0) −	(4.13e+0) −	(**1.95e−5**)
DTLZ4	3	3.3679e−1	1.6745e−1	4.4644e−1	5.6065e−2	6.4017e−2	**3.9851e−2**
		(2.95e−1) −	(7.43e−2) −	(2.56e−2) −	(1.30e−3) −	(9.01e−2) −	(**3.42e−5**)
DTLZ5	3	2.5445e−2	9.7954e−3	3.2644e−2	**4.0240e−3**	5.2090e−3	4.3076e−3
		(7.73e−4) −	(2.69e−4) −	(2.76e−3) −	(**2.35e−4**) +	(5.32e−4)	(2.34e−4)
DTLZ6	3	1.1977e−1	2.2561e−2	1.0632e+0	4.0207e−3	8.4756e−3	**2.8620e−3**
		(2.69e−1) −	(4.25e−2) −	(3.67e−1) −	(1.52e−3) −	(7.31e−3) −	(**4.13e−5**)
DTLZ7	3	1.2272e−1	7.1935e−1	1.0404e−1	9.8637e−2	8.9325e−2	**4.6439e−2**
		(1.68e−2) −	(2.68e−1) −	(1.85e−3) −	(8.59e−2) −	(8.44e−3) −	(**9.77e−4**)
UF1	2	3.3028e−1	1.5363e−1	3.1722e−1	1.6423e−1	1.1758e−1	**1.8035e−2**
		(1.09e−1) −	(4.34e−2) −	(3.85e−2) −	(4.26e−2) −	(2.59e−2) −	(**5.02e−3**)
UF2	2	1.9581e−1	7.8256e−2	1.7342e−1	7.1367e−2	6.8456e−2	**4.7487e−3**
		(7.50e−2) −	(2.70e−2) −	(9.78e−3) −	(2.22e−2) −	(7.93e−3) −	(**1.99e−4**)
UF3	2	3.3607e−1	2.6907e−1	5.8351e−1	3.2431e−1	3.9024e−1	**1.1283e−2**
		(2.71e−2) −	(1.87e−2) −	(3.03e−2) −	(5.02e−2) −	(4.42e−2) −	(**7.27e−3**)
UF4	2	1.2555e−1	1.0057e−1	9.6702e−2	8.0855e−2	7.6173e−2	**3.7859e−3**
		(5.38e−3) −	(7.17e−3) −	(2.35e−3) −	(3.59e−3) −	(2.75e−3) −	(**6.82e−7**)
UF5	2	1.4872e+0	1.7218e+0	2.1310e+0	9.3587e−1	5.7706e−1	**4.1212e−1**
		(4.11e−1) −	(3.59e−1) −	(1.90e−1) −	(3.52e−1) −	(1.61e−1) −	(**8.84e−2**)
UF6	2	5.3473e−1	5.8121e−1	1.5396e+0	4.2449e−1	4.5471e−1	**1.4366e−1**
		(1.75e−1) −	(8.13e−2) −	(1.97e−1) −	(1.63e−1) −	(8.33e−2) −	(**4.72e−2**)
UF7	2	5.0192e−1	2.2686e−1	4.0358e−1	3.0255e−1	1.6172e−1	**2.0655e−2**
		(1.31e−1) −	(1.61e−1) −	(3.37e−2) −	(1.52e−1) −	(1.15e−1) −	(**4.82e−3**)
UF8	3	6.0369e−1	3.1007e−1	1.0043e+0	5.5536e−1	3.4557e−1	**4.4330e−2**
		(2.61e−1) −	(2.41e−2) −	(1.04e−1) −	(3.64e−1) −	(4.07e−2) −	(**4.93e−4**)
UF9	3	5.3960e−1	4.0216e−1	1.0300e+0	8.0921e−1	4.5379e−1	**3.4952e−2**
		(9.60e−2) −	(5.45e−2) −	(1.06e−1) −	(2.65e−1) −	(6.09e−2) −	(**2.30e−3**)

(Continued)

Table 5 (continued)

Problems	*M*	MOEA/D	MOEA/ D-DE	MOEA/ DVA	NNIA	CA-MOEA	MOPSO/ DMS
UF10	3	7.3454e−1 (9.30e−2) −	2.0185e+0 (3.40e−1) −	5.9044e+0 (6.38e−1) −	3.3342e+0 (1.16e+0) −	1.0350e+0 (3.26e−1) −	**1.9339e−1 (3.81e−2)**
+/−/=		1/19/2	0/22/0	0/22/0	1/21/0	0/22/0	——

In Table 4, for 22 test problems, only SMPSO achieves smaller IGD values than MOPSO/DMS on ZDT3 and DTLZ5, while MOPSO/DMS achieves smaller IGD values than CMOPSO, dMOPSO, MMOPSO, MOPSO/D and SMPSO on 20, 22, 21, 22 and 20 test problems, respectively. The IGD values that CMOPSO and MOPSO/DMS achieved on ZDT2 and ZDT6 are quite close. Meanwhile, MMOPSO and MOPSO/DMS have very close IGD values on DTLZ5.

Table 5 shows that the IGD values of MOEA/D and NNIA are smaller than MOPSO/DMS on ZDT4 and DTLZ5, respectively. Meanwhile, MOPSO/DMS obtains smaller IGD values than MOEA/D, MOEA/D-DE, MOEA/DVA, NNIA and CA-MOEA on 19, 22, 22, 21 and 22 test problems of 22 test problems, respectively. In addition, MOPSO/DMS achieves similar IGD values with MOEA/D on ZDT1 and ZDT2.

In Tables 4 and 5, it can be found that for ZDT1, ZDT3-4 and UF1-3 with convex PF, except that SMPSO performs better than MOPSO/DMS on ZDT3 and MOEA/D performs similar to MOPSO/DMS on ZDT1 and better than MOPSO/DMS on ZDT4, other comparison algorithms are worse than MOPSO/DMS. For ZDT2, ZDT6, DTLZ2-4, UF4, UF8 and UF10 with concave PF, other comparison algorithms perform worse than MOPSO/DMS except that CMOPSO is similar to MOPSO/DMS on ZDT2 and ZDT6 and MOEA/D is similar to MOPSO/DMS on ZDT2. For ZDT3, UF5-6, and UF9 with disconnected PF, except that SMPSO has a smaller IGD value than MOPSO/DMS on ZDT3, MOPSO/DMS has smaller IGD values than other comparison algorithms. Moreover, for DTLZ5 with degenerated PF, SMPSO and NNIA perform better than MOPSO/DMS, and MMOPSO performs comparable to MOPSO/DMS, while MOPSO/DMS performs better than all comparison algorithms on DTLZ6 with degenerated PF. In addition, the IGD values of MOPSO/DMS are the smallest on DTLZ1 and UF7 with linear PF and DTLZ7 with complex PF.

Next, the performance of all algorithms is visually observed through Figs. 3–6. The final population of all algorithms on ZDT4 with m = 2 and convex PF is displayed in Fig. 3, which clearly illustrates the superior performance of MOPSO/DMS. Compared to the other ten algorithms, it exhibits superior diversity and convergence even if its distribution is not particularly uniform. Consequently, MOPSO/DMS performs better.

The final population distribution of all algorithms on the three-objective DTLZ3 with concave PF is presented in Fig. 4. Fig. 4 indicates that MOPSO/DMS, except for MOEA/D, has distinct advantages over the other nine algorithms in terms of diversity and convergence. Its distribution is reasonably uniform and quite close to the true PF, while the other nine methods have poor diversity and convergence. Therefore, MOPSO/DMS performs well on DTLZ3 with concave PF.

Fig. 5 presents the final population distribution of all algorithms on the three-objective DTLZ6 with degenerated PF. The final population distributions of CMOPSO, MMOPSO, SMPSO, MOEA/D-DE, NNIA, CA-MOEA and MOPSO/DMS are close to the true PF, among which

CMOPSO, CA-MOEA and MOPSO/DMS have good diversity and convergence, while the solution sets of MMOPSO, SMPSO, MOEA/D-DE and NNIA are not evenly distributed. The diversity and convergence of the remaining four algorithms are poor. Therefore, MOPSO/DMS is effective on DTLZ6 test problems with degenerated PF.

Fig. 6 shows the final population distribution of each algorithm on the three-objective UF9 with linear and disconnected PF. From Fig. 6, it is not difficult to see that compared with the other ten algorithms, the final population distribution of MOPSO/DMS is closer to the true PF, with better diversity and convergence, and wider population coverage. Therefore, MOPSO/DMS performs better on UF9 test problem with linear and disconnected PF.

The trajectory of the IGD values for all algorithms on ZDT4, DTLZ1, and UF9 is displayed in Fig. 7, with the horizontal and vertical coordinates representing the number of evaluations and IGD values, respectively. In Fig. 7, for ZDT4, MPSOD and MOEA/DVA perform the worst, for DTLZ1, CMOPSO performs the worst, while for UF9, all the algorithms except MOPSO/DMS perform worse. It is not difficult to find that MOPSO/DMS has obvious advantages on all three test problems, especially on UF9, where MOPSO/DMS achieves the smallest IGD value, and it also verifies the fast convergence speed of the PSO framework.

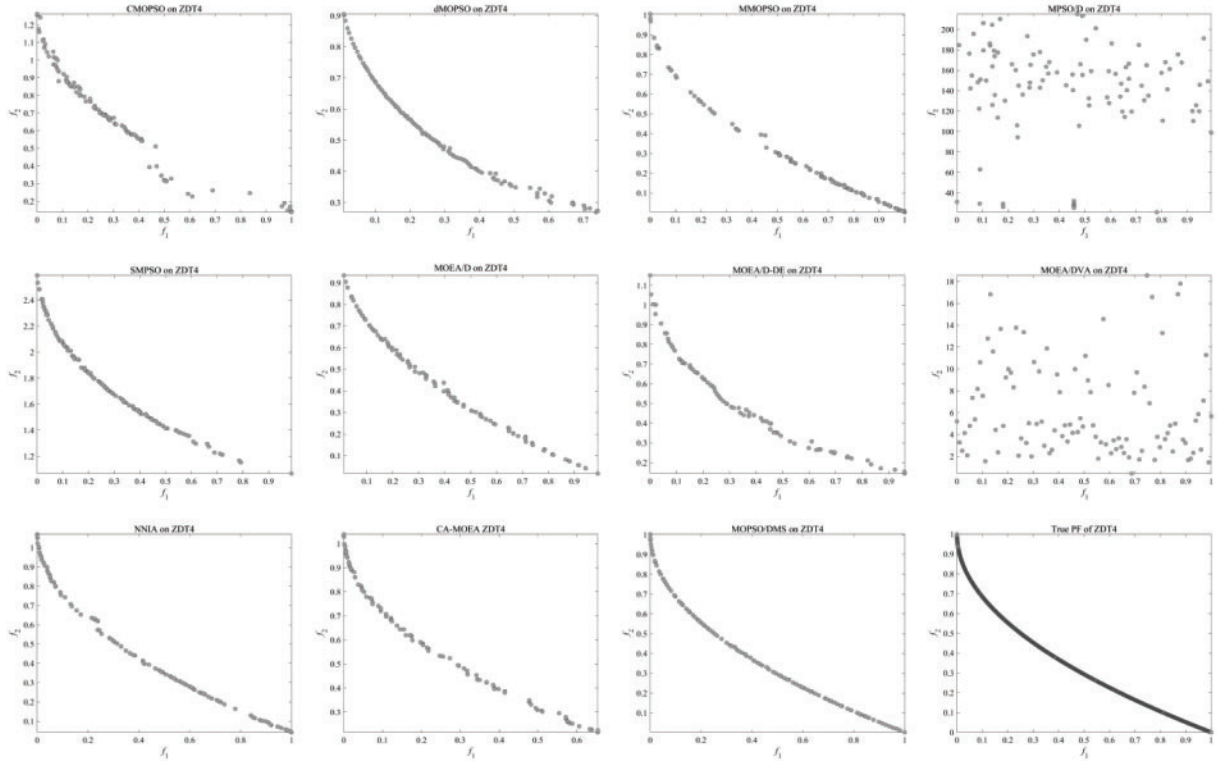

Figure 3: Final population distribution on ZDT4

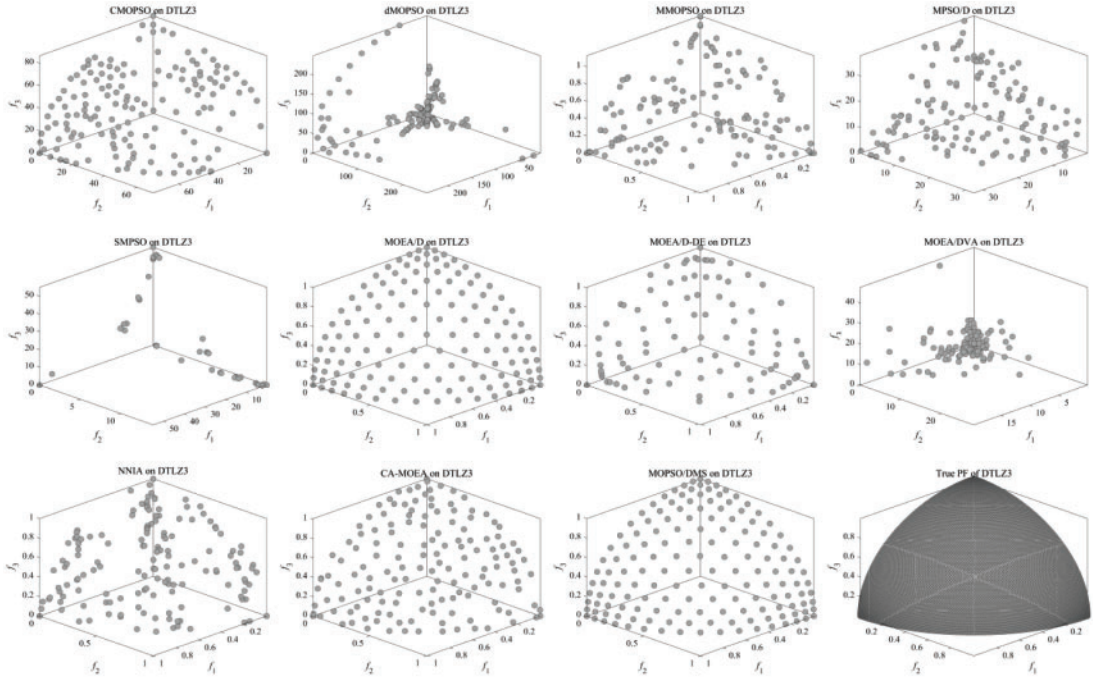

Figure 4: Final population distribution on DTLZ3

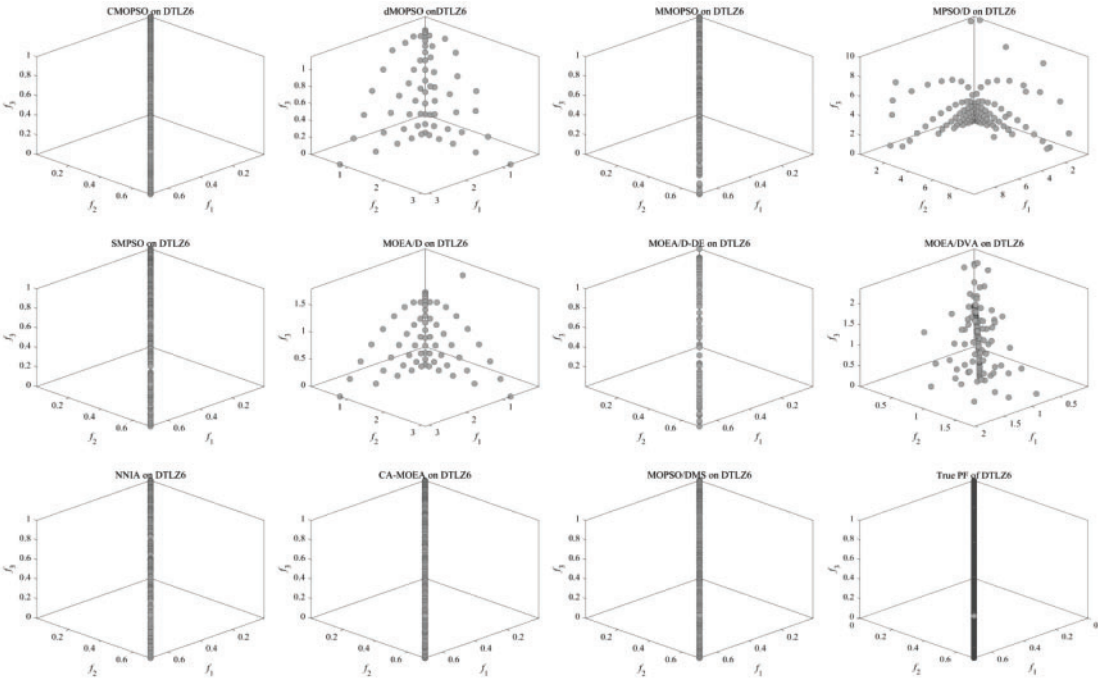

Figure 5: Final population distribution on DTLZ6

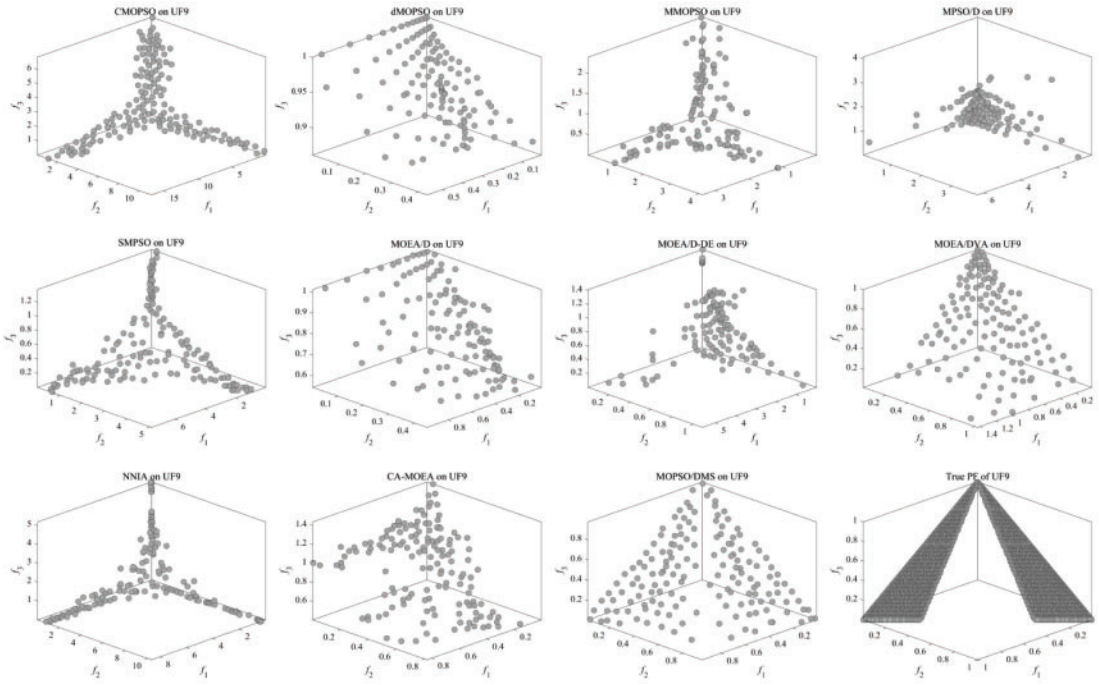

Figure 6: Final population distribution on UF9

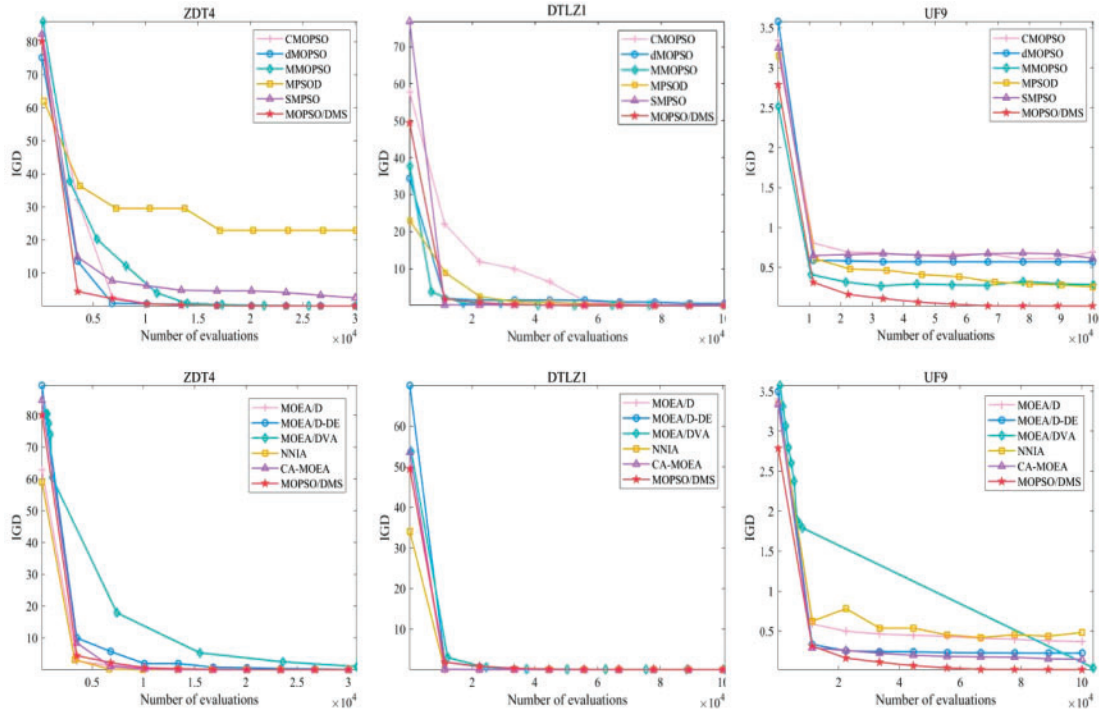

Figure 7: Change of IGD value with the number of assessments

4.4.2 Ablation Experiment

The primary innovation of MOPSO/DMS lies in its decomposition of the objective space, where it employs a multi-selection strategy to choose the appropriate gbest and pbest for each search particle based on the evolution of each subspace. This approach also minimizes the impact of PF discontinuity on the solution set. Additionally, both the evolving population and the external archive are updated, with the superior solution set being selected for output post-iteration. To enhance search efficiency further, an adaptive inertia weight update strategy is introduced. Ablation experiments are conducted to validate MOPSO/DMS's effectiveness. To assess the multi-selection strategy's efficacy, MOPSO/DMS-I uses the selection strategy from MOPSO/D, while MOPSO/DMS-II disregards PF discontinuity to test the strategy's ability to avoid searching discontinuous subspaces. Lastly, MOPSO/DMS-III employs a linear inertia weight adjustment strategy to evaluate the adaptive inertia weight update strategy's impact on improving search efficiency. Tables 6 and 7 show the comparison results of HV values and IGD values of the four algorithms on 22 test problems, respectively.

Table 6: HV average and standard deviation of four algorithms

Problems	*M*	MOPSO/ DMS-I	MOPSO /DMS-II	MOPSO/ DMS-III	MOPSO/ DMS
ZDT1	2	2.4443e−1 (2.77e−1) −	7.1961e−1 (8.61e−5) =	**7.1964e−1** (**3.77e−5**) =	7.1962e−1 (7.08e−5)
ZDT2	2	4.2454e−2 (9.63e−2) −	**4.4465e−1** (**2.34e−5**) =	4.4463e−1 (1.92e−5) =	4.4464e−1 (5.32e−5)
ZDT3	2	1.3736e−1 (2.20e−1) −	9.9046e−1 (7.25e−4) =	9.8063e−1 (7.98e−4) −	**9.9071e−1** (**6.80e−4**)
ZDT4	2	0.0000e+0 (0.00e+0) −	6.8210e−1 (7.78e−2) −	6.1855e−1 (5.59e−2) −	**6.8905e−1** (**8.66e−2**)
ZDT6	2	3.8693e−1 (1.77e−3) =	**3.8814e−1** (**1.56e−6**) =	3.8722e−1 (1.51e−6) =	3.8813e−1 (1.27e−6)
DTLZ1	3	0.0000e+0 (0.00e+0) −	8.1533e−1 (1.06e−3) =	8.1533e−1 (9.29e−4) =	**8.1540e−1** (**1.01e−3**)
DTLZ2	3	5.5056e−1 (1.72e−3) −	5.7068e−1 (3.35e−5) =	5.7062e−1 (4.29e−5) =	**5.7070e−1** (**3.06e−5**)
DTLZ3	3	0.0000e+0 (0.00e+0) −	5.7080e−1 (2.99e−5) =	5.7071e−1 (1.93e−5) =	**5.7081e−1** (**2.33e−5**)
DTLZ4	3	5.2177e−1 (1.73e−2) −	5.7065e−1 (3.64e−5) =	5.7064e−1 (2.68e−5) =	**5.7066e−1** (**3.14e−5**)
DTLZ5	3	1.4337e−1 (8.96e−3) −	1.9114e−1 (2.19e−4) −	1.9376e−1 (1.78e−4) −	**1.9940e−1** (**2.05e−4**)
DTLZ6	3	3.1014e−2 (4.74e−2) −	2.0081e−1 (3.61e−5) =	2.0080e−1 (4.64e−5) =	**2.0086e−1** (**5.00e−5**)
DTLZ7	3	1.0637e−1 (4.02e−2) −	2.7519e−1 (1.35e−3) −	2.7559e−1 (1.95e−3) −	**2.8201e−1** (**5.77e−4**)
UF1	2	4.9879e−1 (2.91e−2) −	6.9021e−1 (8.33e−3) −	6.7541e−1 (4.21e−3) −	**6.9553e−1** (**8.72e−3**)

(Continued)

Table 6 (continued)

Problems	M	MOPSO/ DMS-I	MOPSO /DMS-II	MOPSO/ DMS-III	MOPSO/ DMS
UF2	2	6.0690e−1 (1.05e−2) −	7.0714e−1 (4.08e−4) −	7.0790e−1 (2.82e−4) −	**7.1738e−1 (3.92e−4)**
UF3	2	2.1332e−1 (2.00e−2) −	7.0136e−1 (8.26e−3) =	**7.0583e−1 (3.55e−4)** =	7.0484e−1 (8.80e−3)
UF4	2	3.2930e−1 (8.79e−3) −	4.4468e−1 (1.23e−5) =	4.4468e−1 (1.84e−5) =	**4.4469e−1 (1.88e−5)**
UF5	2	0.0000e+0 (0.00e+0) −	5.5514e−2 (7.63e−2) −	5.5768e−2 (5.91e−2) −	**5.8386e−2 (3.66e−2)**
UF6	2	4.6641e−3 (1.14e−2) −	3.1678e−1 (8.34e−2) −	3.1722e−1 (7.62e−2) −	**3.2282e−1 (5.85e−2)**
UF7	2	4.0368e−1 (5.96e−2) −	5.5276e−1 (8.09e−3) =	5.5169e−1 (7.75e−3) =	**5.5374e−1 (7.76e−3)**
UF8	3	5.7600e−2 (2.43e−2) −	5.6249e−1 (1.55e−3) =	5.6259e−1 (5.43e−4) =	**5.6301e−1 (1.10e−3)**
UF9	3	1.2395e−1 (3.16e−2) −	7.8252e−1 (5.11e−3) −	7.8926e−1 (1.08e−2) =	**7.8987e−1 (5.20e−3)**
UF10	3	0.0000e+0 (0.00e+0) −	2.7100e−1 (1.18e−1) −	2.8026e−1 (8.84e−2) =	**2.8184e−1 (8.28e−2)**
+/−/=		0/21/1	0/9/13	0/8/14	——

Table 7: IGD average and standard deviation of four algorithms

Problems	M	MOPSO/ DMS-I	MOPSO/ DMS-II	MOPSO /DMS-III	MOPSO/ DMS
ZDT1	2	1.1237e−1 (1.10e−1) −	3.9479e−3 (8.96e−5) =	3.9338e−3 (5.99e−5) =	**3.9273e−3 (6.64e−5)**
ZDT2	2	8.5428e−2 (5.26e−2) −	**3.7854e−3 (9.62e−6)** =	3.7863e−3 (1.11e−5) =	3.7904e−3 (2.67e−5)
ZDT3	2	3.0465e−2 (8.64e−2) −	5.5528e−3 (1.93e−4) −	5.5251e−3 (1.90e−4) −	**5.4210e−3 (1.06e−4)**
ZDT4	2	5.3474e−1 (1.15e−1) −	1.6280e−1 (1.92e−1) −	1.6119e−2 (4.51e−2) −	**1.0824e−2 (1.82e−2)**
ZDT6	2	8.8147e−3 (2.90e−3) −	2.7369e−3 (1.04e−6) =	2.7379e−3 (6.19e−7) =	**2.7368e−3 (1.72e−6)**
DTLZ1	3	1.5459e−2 (2.60e−5) =	1.5394e−2 (3.33e−5) =	1.5396e−2 (3.65e−5) =	**1.5389e−2 (4.40e−5)**
DTLZ2	3	3.9999e−2 (4.36e−5) =	3.9862e−2 (2.64e−5) =	**3.9854e−2 (2.33e−5)** =	3.9856e−2 (2.95e−5)

(Continued)

Table 7 (continued)

Problems	M	MOPSO/ DMS-I	MOPSO/ DMS-II	MOPSO /DMS-III	MOPSO/ DMS
DTLZ3	3	3.9866e−2 (2.12e−5) =	3.9849e−2 (1.55e−5) =	3.9837e−2 (2.45e−5) =	**3.9836e−2 (1.95e−5)**
DTLZ4	3	4.0897e−2 (1.24e−3) −	3.9855e−2 (2.75e−5) =	3.9853e−2 (4.50e−5) =	**3.9851e−2 (3.42e−5)**
DTLZ5	3	2.5420e−2 (5.84e−4) −	5.3930e−3 (3.24e−4) −	4.3104e−3 (1.99e−4) =	**4.3076e−3 (2.34e−4)**
DTLZ6	3	2.5557e−2 (5.19e−4) −	2.8680e−3 (3.69e−5) =	2.8688e−3 (3.27e−5) =	**2.8620e−3 (4.13e−5)**
DTLZ7	3	5.9818e−1 (2.30e−1) −	4.7851e−2 (1.02e−3) −	4.6494e−2 (8.25e−4) =	**4.6439e−2 (9.77e−4)**
UF1	2	2.3238e−2 (9.97e−3) −	1.9511e−2 (6.88e−3) −	2.1852e−2 (8.36e−3) −	**1.8035e−2 (5.02e−3)**
UF2	2	5.6495e−3 (1.87e−3) −	4.8242e−3 (2.98e−4) −	4.7999e−3 (2.33e−4) −	**4.7487e−3 (1.99e−4)**
UF3	2	**8.3742e−3 (6.97e−4) +**	1.2709e−2 (6.12e−3) −	1.1663e−2 (6.50e−3) =	1.1283e−2 (7.27e−3)
UF4	2	1.0902e−2 (3.76e−3) −	3.7873e−3 (3.02e−6) =	3.7863e−3 (9.50e−7) =	**3.7859e−3 (6.82e−7)**
UF5	2	5.9448e−1 (1.37e−1) −	4.4952e−1 (5.88e−2) −	4.2616e−1 (8.81e−2) −	**4.1212e−1 (8.84e−2)**
UF6	2	1.9280e−1 (5.62e−2) −	1.5352e−1 (5.53e−2) −	1.8025e−1 (4.10e−2) −	**1.4366e−1 (4.72e−2)**
UF7	2	2.6774e−2 (1.88e−2) −	2.3743e−2 (3.86e−3) −	2.1837e−2 (5.61e−3) −	**2.0655e−2 (4.82e−3)**
UF8	3	4.5984e−2 (8.71e−4) −	4.4649e−2 (7.85e−4) =	4.4444e−2 (5.41e−4) =	**4.4330e−2 (4.93e−4)**
UF9	3	3.7494e−2 (3.73e−3) −	3.6106e−2 (3.91e−3) −	3.5256e−2 (2.45e−3) =	**3.4952e−2 (2.30e−3)**
UF10	3	2.0375e−1 (4.45e−2) −	2.1537e−1 (6.81e−2) −	1.9625e−1 (6.74e−2) =	**1.9339e−1 (3.81e−2)**
+/−/=		1/18/3	0/12/10	0/7/15	——

In Table 6, for 22 test problems, MOPSO/DMS achieves the highest HV value on 18 test problems, and MOPSO/DMS performs better than MOPSO/DMS-I, MOPSO/DMS-II and MOPSO/DMS-III on 21, 9 and 8 test problems, respectively. Meanwhile, in Table 7, among 22 test problems, MOPSO/DMS performs best on 19 test problems, and is superior to MOPSO/DMS-I, MOPSO/DMS-II and MOPSO/DMS-III on 18, 12 and 7 test problems, respectively. In general, MOPSO/DMS performs best.

5 Conclusions

Multi-objective optimization problems frequently arise in various aspects of production and daily life. To enhance solution quality and optimize search efficiency, this article introduces a novel multi-objective particle swarm optimization algorithm that integrates decomposition and a multi-selection strategy. Initially, the decomposition technique maintains diversity significantly, while an external archive stores historical optimal solutions. The evolving population and the external archive are updated through distinct strategies. Concurrently, based on the evolution of each subspace with or without non-dominated solutions, the multi-selection strategy selects suitable global and personal optimal solutions for each particle's update process. Additionally, this approach minimizes the impact of PF discontinuity on the final solution set. An adaptive inertia weight update strategy is also implemented to boost algorithm performance. The proposed algorithm was compared with several mainstream MOPSOs and MOEAs across 22 test problems, demonstrating superior overall performance. Current research on this algorithm is confined to two-dimensional or three-dimensional benchmark problems. Future work should focus on applying the algorithm to real-world issues and extending its applicability to many-objective evolutionary scenarios.

Acknowledgement: The authors would like to thank all reviewers and editors for their constructive comments and feedback on our manuscript, which helped us improve the quality of our manuscript.

Funding Statement: This work was supported by National Natural Science Foundations of China (nos. 12271326, 62102304, 61806120, 61502290, 61672334, 61673251), China Postdoctoral Science Foundation (no. 2015M582606), Industrial Research Project of Science and Technology in Shaanxi Province (nos. 2015GY016, 2017JQ6063), Fundamental Research Fund for the Central Universities (no. GK202003071), Natural Science Basic Research Plan in Shaanxi Province of China (no. 2022JM-354).

Author Contributions: Study conception and design: Li Ma, Cai Dai; data collection: Li Ma, Xingsi Xue; analysis and interpretation of results: Li Ma, Cai Dai, Cheng Peng; draft manuscript preparation: Li Ma. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011. doi: 10.1016/j.swevo.2011.03.001.

[2] Y. Zhang, D. Gong, and J. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, no. 3, pp. 172–185, 2013. doi: 10.1016/j.neucom.2012.09.019.

[3] Z. Chen, H. Wu, Y. Chen, L. Cheng, and B. Zhang, "Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm," *Appl. Soft Comput.*, vol. 116, no. 19, 2022, Art. no. 10819. doi: 10.1016/j.asoc.2021.108192.

[4] A. A. Alabbadi and M. F. Abulkhair, "Multi-objective task scheduling optimization in spatial crowdsourcing," *Algorithms*, vol. 14, no. 3, 2021, Art. no. 77. doi: 10.3390/a14030077.
[5] Z. Cui *et al.*, "Hybrid many-objective particle swarm optimization algorithm for green coal production problem," *Inf. Sci.*, vol. 518, no. 7, pp. 256–271, 2020. doi: 10.1016/j.ins.2020.01.018.
[6] X. Cai, J. Zhang, Z. Ning, Z. Cui, and J. Chen, "A many-objective multistage optimization-based fuzzy decision-making model for coal production prediction," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 12, pp. 3665–3675, 2021. doi: 10.1109/TFUZZ.2021.3089230.
[7] B. Zhang, Q. Pan, L. Gao, L. L. Meng, X. Y. Li and K. K. Peng, "A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 12, pp. 4984–4999, 2019. doi: 10.1109/TSMC.2019.2916088.
[8] K. Li and R. Chen, "Batched data-driven evolutionary multiobjective optimization based on manifold interpolation," *IEEE Trans. Evol. Comput.*, vol. 27, no. 1, pp. 126–140, 2022. doi: 10.1109/TEVC.2022.3162993.
[9] C. Pizzuti, "A multi-objective genetic algorithm for community detection in networks," in *2009 21st IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2009, pp. 379–386. doi: 10.1109/ICTAI.2009.58.
[10] S. Wikaisuksakul, "A multi-objective genetic algorithm with fuzzy c-means for automatic data clustering," *Appl. Soft Comput.*, vol. 24, pp. 679–691, Nov. 2014. doi: 10.1016/j.asoc.2014.08.036.
[11] Y. Han *et al.*, "Multi-strategy multi-objective differential evolutionary algorithm with reinforcement learning," *Knowl.-Based Syst.*, vol. 277, no. 8, Oct. 2023, Art. no. 110801. doi: 10.1016/j.knosys.2023.110801.
[12] X. Yu, Z. Hu, W. Luo, and Y. Xue, "Reinforcement learning-based multi-objective differential evolution algorithm for feature selection," *Inf. Sci.*, vol. 661, Mar. 2024, Art. no. 120185. doi: 10.1016/j.ins.2024.120185.
[13] H. Han, Y. Liu, Y. Hou, and J. Qiao, "Multi-modal multi-objective particle swarm optimization with self-adjusting strategy," *Inf. Sci.*, vol. 629, no. 1, pp. 580–598, Jun. 2023. doi: 10.1016/j.ins.2023.02.019.
[14] Y. Li, Y. Zhang, and W. Hu, "Adaptive multi-objective particle swarm optimization based on virtual Pareto front," *Inf. Sci.*, vol. 625, pp. 206–236, May 2023. doi: 10.1016/j.ins.2022.12.079.
[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95-Int. Conf. Neural Netw.*, Nov. 1995, vol. 4, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.
[16] X. Xu, J. Li, M. Zhou, J. Xu, and J. Cao, "Accelerated two-stage particle swarm optimization for clustering not-well-separated data," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 11, pp. 4212–4223, 2018. doi: 10.1109/TSMC.2018.2839618.
[17] J. Zheng, Z. Zhang, J. Zou, S. Yang, J. Ou and Y. Hu, "A dynamic multi-objective particle swarm optimization algorithm based on adversarial decomposition and neighborhood evolution," *Swarm Evol. Comput.*, vol. 69, no. 9, 2022, Art. no. 100987. doi: 10.1016/j.swevo.2021.100987.
[18] S. Carstensen and J. C. W. Lin, "TKU-PSO: An efficient particle swarm optimization model for top-K high-utility itemset mining," *Int. J. Interact. Multimed. Artif. Intell.*, Jan. 2024. doi: 10.9781/ijimai.2024.01.002.
[19] H. Han, L. Zhang, A. Yinga, and J. Qiao, "Adaptive multiple selection strategy for multi-objective particle swarm optimization," *Inf. Sci.*, vol. 624, no. 2, pp. 235–251, May 2023. doi: 10.1016/j.ins.2022.12.077.
[20] G. T. Pulido and C. A. Coello Coello, "Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer," in *Genetic and Evolutionary Computation—GECCO 2004*, Berlin, Heidelberg, Berlin Heidelberg: Springer, 2004, vol. 3102, pp. 225–237.
[21] Z. H. Zhan, J. Li, J. Cao, J. Zhang, H. S. H. Chung and Y. H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, 2013. doi: 10.1109/TSMCB.2012.2209115.
[22] S. Z. Martínez and C. A. Coello Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proc. 13th Annu. Conf. Genetic Evol. Comput.*, New York, NY, USA, Association for Computing Machinery, Jul. 2011, pp. 69–76. doi: 10.1145/2001576.2001587.
[23] 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," *Evol. Comput.*, vol. 22, no. 1, pp. 47–77, 2014. doi: 10.1162/EVCO_a_00104.

[24] C. Dai, Y. Wang, and M. Ye, "A new multi-objective particle swarm optimization algorithm based on decomposition," *Inf. Sci.*, vol. 325, no. 9, pp. 541–557, 2015. doi: 10.1016/j.ins.2015.07.018.
[25] W. F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 38, no. 5, pp. 1270–1293, 2008. doi: 10.1109/TSMCB.2008.925757.
[26] Y. J. Zheng, H. F. Ling, J. Y. Xue, and S. Y. Chen, "Population classification in fire evacuation: A multiobjective particle swarm optimization approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 70–81, 2013. doi: 10.1109/TEVC.2013.2281396.
[27] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 1–18, 2013. doi: 10.1109/TEVC.2013.2296151.
[28] N. Ye, C. Dai, and X. Xue, "A two-archive many-objective optimization algorithm based on D-Domination and decomposition," *Algorithms*, vol. 15, no. 11, 2022, Art. no. 392. doi: 10.3390/a15110392.
[29] X. Shu, Y. Liu, J. Liu, M. Yang, and Q. Zhang, "Multi-objective particle swarm optimization with dynamic population size," *J. Comput. Des. Eng.*, vol. 10, no. 1, pp. 446–467, 2023. doi: 10.1093/jcde/qwac139.
[30] X. Chen, H. Tianfield, and W. Du, "Bee-foraging learning particle swarm optimization," *Appl. Soft Comput*, vol. 102, no. 11, 2021, Art. no. 107134. doi: 10.1016/j.asoc.2021.107134.
[31] T. Guan, F. Han, and H. Han, "A modified multi-objective particle swarm optimization based on levy flight and double-archive mechanism," *IEEE Access*, vol. 7, pp. 183444–183467, 2019. doi: 10.1109/ACCESS.2019.2960472.
[32] H. Cheng, L. Li, and L. You, "A weight vector adjustment method for decomposition-based multiobjective evolutionary algorithms," *IEEE Access*, vol. 11, pp. 42324–42330, 2023. doi: 10.1109/ACCESS.2023.3270806.
[33] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007. doi: 10.1109/TEVC.2007.892759.
[34] K. Liagkouras and K. Metaxiotis, "An elitist polynomial mutation operator for improved performance of MOEAs in computer networks," in *2013 22nd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2013, pp. 1–5. doi: 10.1109/ICCCN.2013.6614105.
[35] X. Zhang, X. Zheng, R. Cheng, J. Qiu, and Y. Jin, "A competitive mechanism based multi-objective particle swarm optimizer with fast convergence," *Inf. Sci.*, vol. 427, no. 2, pp. 63–76, 2018. doi: 10.1016/j.ins.2017.10.037.
[36] Q. Lin, J. Li, Z. Du, J. Chen, and Z. Ming, "A novel multi-objective particle swarm optimization with multiple search strategies," *Eur. J. Oper. Res.*, vol. 247, no. 3, pp. 732–744, 2015. doi: 10.1016/j.ejor.2015.06.071.
[37] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna and E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *2009 IEEE Symp. Comput. Intell. Multi-Criteria Decis.-Mak. (MCDM)*, IEEE, Mar. 2009, pp. 66–73. doi: 10.1109/MCDM.2009.4938830.
[38] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2008. doi: 10.1109/TEVC.2008.925798.
[39] X. Ma *et al.*, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275–298, 2015. doi: 10.1109/TEVC.2015.2455812.
[40] M. Gong, L. Jiao, H. Du, and L. Bo, "Multiobjective immune algorithm with nondominated neighbor-based selection," *Evol. Comput.*, vol. 16, no. 2, pp. 225–255, 2008. doi: 10.1162/evco.2008.16.2.225.
[41] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2758–2770, 2018. doi: 10.1109/TCYB.2018.2834466.
[42] Y. Tian *et al.*, "Evolutionary large-scale multi-objective optimization: A survey," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 174:1–174:34, Oct. 2021. doi: 10.1145/3470971.

[43] M. Abdel-Basset, R. Mohamed, S. Mirjalili, R. K. Chakrabortty, and M. Ryan, "An efficient marine predators algorithm for solving multi-objective optimization problems: Analysis and validations," *IEEE Access*, vol. 9, pp. 42817–42844, 2021. doi: 10.1109/ACCESS.2021.3066323.

[44] X. Li, X. L. Li, K. Wang, and Y. Li, "A multi-objective particle swarm optimization algorithm based on enhanced selection," *IEEE Access*, vol. 7, pp. 168091–168103, 2019. doi: 10.1109/ACCESS.2019.2954542.