



OPEN Multi-objective particle swarm optimization algorithm for task allocation and archived guided mutation strategies

Jianjie Chen¹, Yanmin Liu^{2✉}, Yi Luo³, Aijia Ouyang², Jie Yang² & Wuer Bai¹

In this paper, we propose a novel multi-objective particle swarm optimization algorithm with a task allocation and archive-guided mutation strategy (TAMOPSO), which effectively solves the problem of inefficient search in traditional algorithms by assigning different evolutionary tasks to particles with different characteristics. First, TAMOPSO divides multiple subpopulations according to the particle distribution status of each iteration of the population and designs a new task allocation mechanism to improve the evolutionary search efficiency. Second, TAMOPSO adopts an adaptive Lévy flight strategy according to the population growth rate, automatically increasing the global variation probability to expand the search range when the population converges and enhancing the local variation to conduct fine search when the population disperses to realize the dynamics of global and local variations. Finally, TAMOPSO measures the contribution of particles to the population optimization through the particle evolution contribution rate index and filters out valuable historical solutions for subsequent reuse to accelerate the convergence speed; in addition, TAMOPSO improves the individual optimal particle selection mechanism, changes the bias of the traditional algorithm, ensures that each particle has an equal opportunity, and enhances the fairness of the selection process. The fairness of the selection process is enhanced at the same time. The performance of TAMOPSO is compared with ten existing algorithms on 22 standard test problems, and the experimental results show that TAMOPSO outperforms the other algorithms in several standard test problems and has better performance in solving multi-objective problems.

Keywords Multi-objective particle swarm optimization, Subpopulation partitioning, Task allocation, Lévy flight strategy, Individual optimal selection

Multi-objective optimization problems (MOPs)^{1–3} are widely used in the fields of machine learning, recommender systems, financial cloud computing, and software engineering, etc., where real-world problems usually involve multiple optimization objectives. Finding an optimal solution is extremely difficult because there are often conflicts between objective functions^{4,5}. To find a compromise solution between different objectives, a Pareto optimal solution set (POS)⁶ is introduced, and these solutions constitute a Pareto optimal frontier (PF) in the objective space⁷. Many new algorithms have been proposed and optimized to enhance the performance of MOPs⁸, such as Particle Swarm Optimization (PSO)⁹, Genetic Algorithm (GA)¹⁰, Ant Colony Optimization (ACO)¹¹, African Vulture Optimization Algorithm (AVOA)¹², and Multi-Objective Adaptive Guided Differential Evolutionary Algorithm (MOAGDE)¹³.

Among many improved algorithms, DTD—EAMO¹⁴ utilizes fast non-dominated sorting to divide the population into three sub-populations and then introduces a two-stage multi-population adaptive mutation strategy with a dual external archive mechanism, which maintains convergence and diversity of the sub-populations and archives high-quality solutions to help the populations get rid of the local optimums, promote information exchange, and finally select high-quality solutions to reorganize the new population¹⁵. Another study¹⁶ proposes a convergence contribution evaluator that does not rely on additional parameters or predefined reference points, and its design is inspired by the concept of ideal points of Pareto frontiers and partitioning strategies. MSC-PSO¹⁷ enhances the global search capability through population partitioning, and TSO-DF¹⁸

¹School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, China. ²School of Mathematics, Zunyi Normal College, Zunyi, China. ³School of Mathematics and Statistics, Guizhou University, Guiyang, China. ✉email: yanmin7813@163.com

evaluates the elite solutions for guiding the search by fusing the convergence and diversity information without sacrificing the diversity of the solutions. The method screens the optimal solutions by merging convergence and diversity metrics as a single ranking criterion. MOPSO, which is derived from the PSO extension, is widely used in solving multi-objective optimization problems due to its simple structure and fast convergence rate¹⁸. In addition, the MOFS-REPLS algorithm¹⁹, LDANet²⁰, and EAL-PDM²¹ also show their advantages in different fields, respectively. MOFS-REPLS algorithm improves the performance of multi-objective optimization in various ways, such as unique coding format, optimized initialization strategy, and double archiving mechanism; LDANet improves the performance of lung substance segmentation through innovative structure; the pair-based early active learning algorithm proposed by EAL-PDM solves the data labeling problem in the early active learning of pedestrian re-identification and improves the model performance.

Although MOPSO has significant advantages in optimization performance, there is still room for improvement when facing complex problems. There are many limitations in traditional MOPSO: The population is handled in a single way, which is difficult to cope with the complex search space and dynamic changes, and it may be too concentrated in the optimization of certain objectives, leading to slow convergence or premature convergence; the variation operation is based on simple random perturbations, which is insufficiently diversified, and the setup of the mutation probability is inflexible, which makes it difficult to balance the exploration and exploitation; the maintenance of the external archive relies on the basic criterion, which is insufficiently reflecting the diversity of the non-dominated solution. External archive maintenance relies on basic guidelines, which do not reflect the diversity of nondominated solutions and is prone to the loss of important solutions; the traditional individual optimal updating suffers from dimensional pressure and selection unfairness. In this paper, we address the potential problems of MOPSO by combining multiple mechanisms and propose TAMOPSO.

For one thing, the traditional unified treatment of the population tends to concentrate on the particle population in a local region prematurely, which reduces the diversity of the solution set and weakens the global optimization ability. To solve this problem, numerous studies have attempted to divide the population. For example, MOCPSO²² uses shift density estimation (SDE)²³ to calculate the population fitness score and randomly divide the population; ecemAMOPSO²⁴ divides the subpopulation according to the order of particle convergence; Ran Cheng and Yao Chu Jin²⁵ propose an angular penalization distance (APD) method to balance convergence and diversity; MOMTPSO²⁶ combines objective space partitioning and adaptive migration distance with the adaptive migration distance method. Combining target space partitioning and adaptive migration; the Football Team Training Algorithm (FTTA) of²⁷ and the two-population evolutionary algorithm of dealing with the population partitioning problem from different perspectives, respectively, to improve the search efficiency and algorithm performance. However, the current population partitioning methods still have the problems of a single partitioning standard and lack of dynamic adaptability. Therefore, this paper proposes to use population uniformity as a weighting factor for population division.

Second, the variation operation is crucial in MOPSO. It increases population diversity, avoids premature convergence, and enhances the global search capability while fine-tuning the particles when they are close to the optimal solution, realizing the balance between global and local search, and improving the quality of the solution. Moazen²⁸ proposes the exponential mutation operator;²⁹ proposes a particle swarm algorithm that combines elite learning; the algorithm of³⁰ adopts an adaptive response strategy, and the FDDE algorithm of³¹ uses a differential evolution strategy and integrates individual fitness and diversity contributions, and the Flower Pollination Algorithm (FPA)³² has also been applied in optimization problems. However, the existing variation operations have problems such as difficulty balancing global and local searches, high parameter sensitivity, poor stability, and low effective variation rates.

The Lévy flight strategy brings new opportunities to improve the performance of MOPSO population variability. Many previous researches have achieved remarkable results: in 2010, Xin-She Yang and Suash Deb's team first introduced it into the firefly algorithm to avoid the algorithm falling into the local optimum through long-distance jumps in order to improve the optimization ability; in 2012, Kenneth V. Price and other scholars integrated it into the differential evolution algorithm to enhance the diversity of populations and improve the convergence speed and accuracy; in 2015, Carlos A. Coello Coello's team applied Lévy flights in a multi-objective optimization algorithm to make the Pareto solution set distribution better. These studies confirmed the effectiveness of the Lévy flight strategy in different dimensions and provided theoretical and practical foundations for applying the strategy in the variational operation of TAMOPSO. However, when applying Lévy flight to MOPSO, problems such as population degradation due to too large step size, underdevelopment of localization when relying solely on it, and parameters needing to be adjusted manually may occur. The adaptive control The Lévy flight variation strategy proposed in this paper adaptively adjusts the variation method by detecting the archived growth rate, which reduces the parameter sensitivity and decreases the difficulty of adjustment while effectively improving the effective variation rate.

Third, in practice, the solutions in the archive may be overly concentrated, especially in the multidimensional objective space. Most algorithms lead to this problem by relying only on nondominated ordering to select archived solutions and not sufficiently introducing a diversity maintenance strategy. García³³ proposes a method based on nondominated ordering of metrics, Luo et al.³⁴ guide the selection of particles by establishing reference vectors and uniformly distributed vectors, and PRMOPSO³⁵ maintains the archive by using projected points and reference points. However, these methods have the limitation of being difficult to accurately measure particle diversity and mixing in low diversity contributing particles, which affects the overall diversity of the archive. To this end, this paper adopts a novel method to calculate the local uniformity of archived particles as a contribution influence factor and collaborates with adaptive grids to measure and remove particles with low contributions to improve the archive maintenance effect.

Fourth, the traditional multi-objective particle swarm optimization algorithm has limitations in individual optimal updating. On the one hand, the dimension pressure problem is highlighted with the increase of

dimension, and it is difficult to process the information effectively in the traditional way, leading to the increase of computational complexity and the decline of algorithm performance. On the other hand, the selection unfairness problem makes the evaluation criteria unreasonable and biased towards certain particles, affecting the overall search effect and convergence speed. The self-weight robust LDA (SWRLDA) method proposed in³⁶ effectively avoids problems such as optimal mean computation. The L2 paradigm selection of individual optimum has two major advantages: relieving dimensionality pressure and improving selection fairness.

To summarize, TAMOPSO proposed in this paper innovates and improves the traditional MOPSO through population partitioning, task allocation, adaptive mutation strategy, archive maintenance, and individual optimal selection method. Compared with existing MOPSO algorithms, the innovativeness of TAMOPSO can be summarized as follows:

1. Adaptive Particle Comprehensive Ranking: The algorithm combines the convergence, diversity, and evolutionary state of the particles to propose a comprehensive score for the particles and adaptively divides the population by ranking the comprehensive score of each particle; finally, the algorithm assigns a specific task for evolutionary search according to the characteristics of each population of particles to improve the performance of the algorithm.
2. Adaptive Lévy flight variation: In this paper, we use the archived information as a feedback factor for the population variation rate, and the algorithm realizes the adaptive switching of the dual-mode variation strategy (global Lévy flight and local normal perturbation) according to the feedback factor. This not only reduces the sensitivity to parameters and the difficulty of parameter tuning but also effectively enhances the effective mutation rate through the targeted mutation approach.
3. Archiving maintenance based on local uniformity: By combining the global and local densities of particles in the archive, a new metric is proposed to maintain the archive. The index takes into account both the global particle distribution and the individual particle distribution, which improves the accuracy of archive maintenance.
4. Improved Individual Optimal Update: The traditional individual optimal update faces the problems of dimensional pressure and unfair selection. For this reason, in this paper, non-dominated particles are selected as individual optimal particles by making a Pareto comparison between the current generation particles and the previous generation particles. For the mutually non-dominated particles, the distance between the particle paradigm (paradigm in the n -dimensional case) and the super-ideal point is calculated, and the particle with the closest super-ideal point is selected as the individual optimum.

Subsequently, Section [Related work](#) introduces the basics of MOPs and PSOs and related work; Section [Proposing the TAMOPSO algorithm](#) elaborates the proposed algorithm and process in detail, running through the basic ideas and algorithmic implementations; Section [Algorithm comparison and result analysis](#) describes the parameter design and performance metrics and demonstrates the effectiveness of TAMOPSO through algorithmic comparative test evaluations; and, finally, Section [Conclusion](#) concludes the research work on TAMOPSO.

Related work

Multi-objective optimization problem

Multi-objective optimization problems contain multiple objective functions $f_1(x), f_2(x), \dots, f_m(x)$, which are usually conflicting. It is not possible to optimize the values of all objective functions at the same time, even under an optimal solution, so the relationship between the objectives needs to be weighed. A multi-objective optimization problem is generally represented as (in terms of the minimum problem):

$$\min (f_1(x), f_2(x), \dots, f_m(x)) \quad (1)$$

where $x = (x_1, x_2, \dots, x_n)$ is the vector of decision variables, $x \in R^n$, n is the dimension of the decision variables. $f_1(x), f_2(x), \dots, f_m(x)$ is the objective function, each $f_i(x)$ corresponds to an objective to be optimized and m is the number of objectives.

The constraints usually include equal and unequal constraints, denoted as:

$$g_j(x) \leq 0 \quad \text{for } j = 1, 2, \dots, p \quad (2)$$

$$h_k(x) = 0 \quad \text{for } k = 1, 2, \dots, q \quad (3)$$

where $g_j(x)$ is the inequality constraint, $h_k(x)$ is the equality constraint, p and q are the number of these constraints, respectively.

Particle swarm optimization

PSO was proposed by Kennedy and Eberhart³⁷ in 1995 and belongs to the category of group intelligence. PSO simulates the movement of a group of particles in the solution space. Each particle represents a potential solution, and the particle searches for the optimal solution by continuously adjusting its speed and position:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (pBest_i - x_i(t)) + c_2 r_2 (gBest - x_i(t)) \quad (4)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

where $x_i(t)$ and $x_i(t+1)$ denote the positions of the particle relations generating t and $t+1$, respectively. $v_i(t)$ and $v_i(t+1)$ denote the velocities of the t and $t+1$ generations, respectively. The inertia weight ω controls the inertia of the particle along the direction of the original velocity. c_1 and c_2 are the learning factors for personal and global experience, respectively. r_1 and r_2 are randomly generated values on the interval $[0,1]$. $pBest_i$ is the optimal position experienced by particle i during the search process, and $gBest_i$ denotes the particle's personal optimal solution (i.e., self-perception). $gBest$ is the best position experienced by all particles in the whole population and denotes the best solution for all particles (i.e., population cognition).

Pareto ranking

Definition 1 Pareto dominance relationship.

Let x and y be two solution vectors in a multi-objective optimization problem, both of which have multiple objective function values. A solution x is said to Pareto dominate y , denoted $x < y$ if for all objective functions f_i ($i = 1, 2, \dots, m$ is the number of objective functions), there is $f_i(x) < f_i(y)$, and there exists at least one objective function f_j such that $f_j(x) < f_j(y)$.

Intuitively, if x is not worse than y on all objectives and is better than y on at least one objective, then x dominates y . If two solution vectors, x and y , have neither $x < y$ nor $y < x$, then the summation is said to be mutually non-dominating.

Definition 2 Pareto ranking.

1. Initialization: For each particle in the particle swarm, set its Pareto rank to 1 and initialize the set of dominating particles and the set of dominated particles to empty.
2. Calculate the dominance relationship: Iterate through each pair of particles x and y in the particle swarm and judge the dominance relationship between them according to the definition of the Pareto dominance relationship. If x dominates y , add y to the dominating particle set of x , and at the same time add x to the dominated particle set of y . If y dominates x , do the opposite; if x and y do not dominate each other, do not perform any operation.
3. Determine the Pareto front (first layer): Iterate over all particles and find those particles whose set of dominated particles is empty, which constitutes the Pareto front, i.e., the first layer of the Pareto optimal solution, marking their Pareto rank as 1.
4. Recursively determine the other layers: for the particles in each layer of the Pareto optimal solution that has been determined, iterate through the particles in their set of dominated particles. If all the dominated particles of a particle have been assigned a Pareto rank, then set the Pareto rank of that particle to the maximum of the Pareto ranks of all its dominated particles plus 1. Repeat this process until all particles have been assigned a Pareto rank.

Proposing the TAMOPSO algorithm

In this section, the specific procedure of the proposed TAMOPSO algorithm is described in detail. Among them, Section [Core framework of TAMOPSO](#) describes the overall framework of the TAMOPSO algorithm, Section [Population segmentation and tasking strategies](#) elaborates on the proposed population partitioning and task allocation strategy, Section [Adaptive control of the Lévy flight variation strategy](#) demonstrates the Lévy flight variation operation with adaptive control, and regarding the maintenance method of the archive and the individual optimal updating strategy, they are elaborated in Section [Uniform contribution margin metrics guide archive maintenance strategies](#) and Section [Individual optimal update strategy](#), respectively. Moreover, the motivation for introducing the corresponding new features in the algorithm is clearly explained at the end of each subsection.

Core framework of TAMOPSO

This section outlines the core framework of TAMOPSO. As shown in Fig. 1, TAMOPSO proposes the establishment of reference vectors to divide the population, task-allocation search strategy, individual optimal selection, archive maintenance, and adaptive mutation mechanism, aiming to improve the search efficiency and optimization effect. Its basic process is as follows:

First, the algorithm randomly initializes the population to provide a starting point for the subsequent search process. The population is divided into three subpopulations by creating reference vectors. This classification process is based on the relationship between the reference vectors and the individuals, with the aim of classifying the particles in the population into different subpopulations by integrating their fitness and spatial characteristics, thus making the subsequent optimization process more targeted. For the different subpopulations, the algorithm assigns different search strategies, respectively. This differentiation of strategies permits a suitable search approach for individuals with different characteristics in the population. For example, while some subpopulations focus on local search, others may focus on global search, thus improving the search efficiency of the algorithm.

When performing a population search, the algorithm not only focuses on the current optimal particle but also keeps track of the global optimal particle by maintaining an archive. The selection of individual optimal particles is based on the dominance relationship between the particles of the previous and current generations and the paradigm to ensure that high-quality particles are selected as much as possible. For non-dominated particles, if the archive is not full, they are added directly; if the archive is full, some particles are eliminated, and the particles in the archive are updated according to the uniform contribution index.

In addition, the algorithm introduces a mutation operation to guide the population to explore efficiently in the search space through an adaptive mutation strategy. This operation employs a mutation model based on Lévy flights, which can perform jumping search in the global range to avoid falling into local optima. The

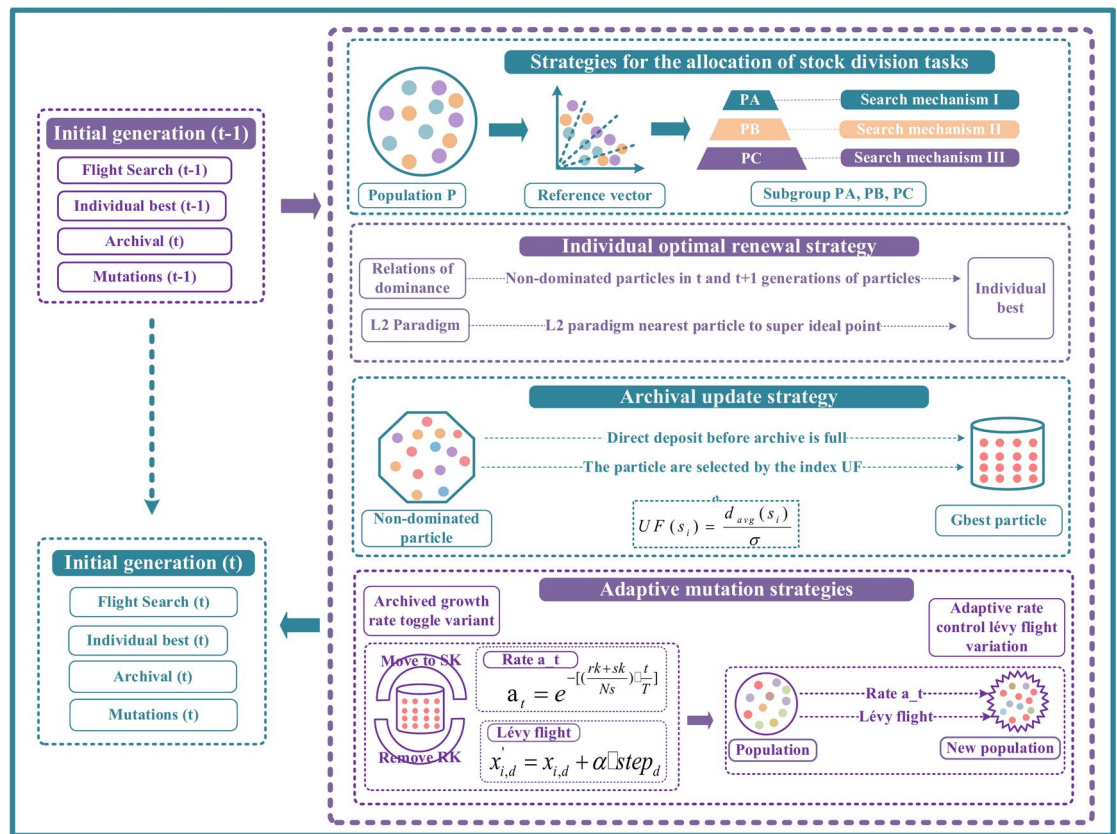


Fig. 1. Core framework of TAMOPSO.

adaptive mechanism allows the mutation operation to dynamically adjust to the current population growth rate, switching between global and local mutation to improve the mutation efficiency.

Finally, the algorithm optimizes the quality of the particles by continuously and iteratively updating the population and eventually storing the globally optimal particles in the archive. The individual strategies in the process work closely together to drive the algorithm towards a more optimal solution.

Population segmentation and tasking strategies

Population segmentation strategy

The traditional multi-objective particle swarm optimization algorithm adopts a single population evolution mode, and the global unified treatment is prone to premature convergence and Pareto frontier distribution diversity defects. To address this problem, TAMOPSO proposes a dynamic swarm optimization framework based on adaptive reference vectors: firstly, we construct a set of directional reference vectors based on the geometric features of the target space and then divide the population into multiple co-evolving sub-populations through the distribution of the particles in the reference vectors to reflect the convergence and diversity of the particles; each sub-population focuses on searching a specific region of the Pareto front so as to achieve autonomous trade-offs and diversity maintenance among conflicting targets. First, the number S of reference vectors is calculated by Eq. (6), which depends on the dimension (M) and resolution (H) of the target space.

$$S = \frac{(M+H-1)!}{H!(M-1)!} \quad (6)$$

where the resolution (H) denotes the number of segmentation segments on each target dimension. It determines the distribution density of the reference vectors in the target space. Specifically, the larger the value of H , the more finely the target space is divided and the greater the number of reference vectors generated. This helps to steer the population distribution more finely, avoid premature convergence, and maintain diversity.

The generation of reference vectors depends on the dimensionality of the target space, and the coordinates of each reference vector are determined by the values of angles uniformly distributed on the unit circle (2-dimensional) or hypersphere (high-dimensional). Assuming that the target space is 2-dimensional, the reference vectors are uniformly distributed on the unit circle in the first quadrant, with angles ranging from 0 to uniformly distributed $\pi/2$. For each reference vector r_j , its coordinates can be computed using the following equation:

$$r_j = [\cos\theta_j, \sin\theta_j] \quad (7)$$

where: θ_j is the angle of the reference vector, uniformly distributed over the range of $[0, \pi/2]$.

In n -dimensional space, point V in n -dimensional space on the sphere can be generated by the following equation:

$$V = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \quad (8)$$

where each component v_i can be calculated by:

$$\begin{aligned} v_1 &= r \cos(\theta_1) \\ v_2 &= r \sin(\theta_1) \cos(\theta_2) \\ v_3 &= r \sin(\theta_1) \sin(\theta_2) \cos(\theta_3) \\ &\vdots \\ v_{n-1} &= r \sin(\theta_1) \sin(\theta_2) \cdots \sin(\theta_{n-2}) \cos(\theta_{n-1}) \\ v_n &= r \sin(\theta_1) \sin(\theta_2) \cdots \sin(\theta_{n-2}) \sin(\theta_{n-1}) \end{aligned} \quad (9)$$

where r is the radius of the ball (for the unit sphere $r = 1$) and $\theta_1, \theta_2, \dots, \theta_{n-1}$ is an $n - 1$ angular parameter, usually uniformly distributed in the interval $[0, \pi]$ or $[0, 2\pi]$.

For particle i , the angle θ_{ij} is calculated as in Eq. 10; the angle between the particle and each reference vector can be obtained, which in turn leads to the relative position of each particle in the target space. Next, the fitness values are Pareto-ranked, and then the angle and fitness value rankings of each particle with the reference vector are normalized separately. The normalization method for both the angle and fitness value rankings is the minimum–maximum normalization method (Eq. 12), which linearly maps the angle and fitness value rankings data into the $[0, 1]$ interval to ensure that they are weighted and summed at a uniform scale. The angle score As and fitness ranking score Fs of each particle are obtained after ranking the normalized particle angle and fitness values. Where As reflects the proximity of the particle to the reference vector and Fs reflects the superiority of the particle in multi-objective optimization. These two scores are weighed and summed by adjusting their relative weights using the population homogeneity factor φ (Eq. 12), thus combining As and Fs (Eq. 13) to calculate the combined particle priority score Ps . The effect of the uniformity factor φ is that it can adaptively change the focus of the population division according to the distribution of the population during each iteration to avoid falling into a local optimum.

$$\theta_{ij} = \arccos \left(\frac{f_i \cdot r_j}{\|f_i\| \cdot \|r_j\|} \right) \quad (10)$$

in Eq. 10, f_i is the fitness value vector of the particle, r_j is the reference vector, and $\|f_i\| \cdot \|r_j\|$ is the dot product of the fitness value vector and the mode of the reference vector.

$$X_N = \frac{X_i - \min(X)}{\max(X) - \min(X)} \quad (11)$$

in Eq. 11, X_N is the normalized particle angle or fitness value ranking, X_i is the angle that needs to be normalized, and $\min(X)$ and $\max(X)$ are the minimum and maximum values of the angle and fitness value ranking in the particle.

$$\varphi = \frac{\frac{1}{\binom{N}{2}} \sum_{1 \leq i < j \leq N} \sqrt{\sum_{k=1}^M (x_{ik} - x_{jk})^2}}{\|F_{max} - F_{min}\|} \quad (12)$$

where: N is the size of the population; $\binom{N}{2} = \frac{N(N-1)}{2}$ is the number of solution pairs in the population, denoting the number of combinations of all the different solution pairs; x_i and x_j are the vectors of the objective values of the two solutions; M is the dimensionality of the objective space; and F_{max} and F_{min} are the maximum and minimum values of each objective dimension in the population, respectively. Finally, the particle composite priority score Ps is obtained:

$$Ps = \varphi \cdot As + (1 - \varphi) \cdot Fs \quad (13)$$

The priority Ps scores of all the particles are sorted and based on the sorting result, the particles are divided into three classes: subpopulation P_A (highest priority), subpopulation P_B (medium priority), and subpopulation P_C (lowest priority) in the ratio of 1:3:1. The pseudo-code for the division of the population is given in Algorithm 1 of.

The Ps score is a key measure of particle quality and is well constructed. It uses the evolutionary state of the population as the weight φ and integrates the As and Fs scores. As reflects the distribution of particles in the reference vector, and the angle between the particles and the reference vector can reflect the diversity of particles, the angle is discrete, the diversity is good, and the distribution of particles is uniform and extensive, while the angle is concentrated, the diversity is poor. Fs score is based on the ranking of the particle fitness values; the particles with smaller fitness values are ranked higher, and they get a higher Fs score, which is the visualization

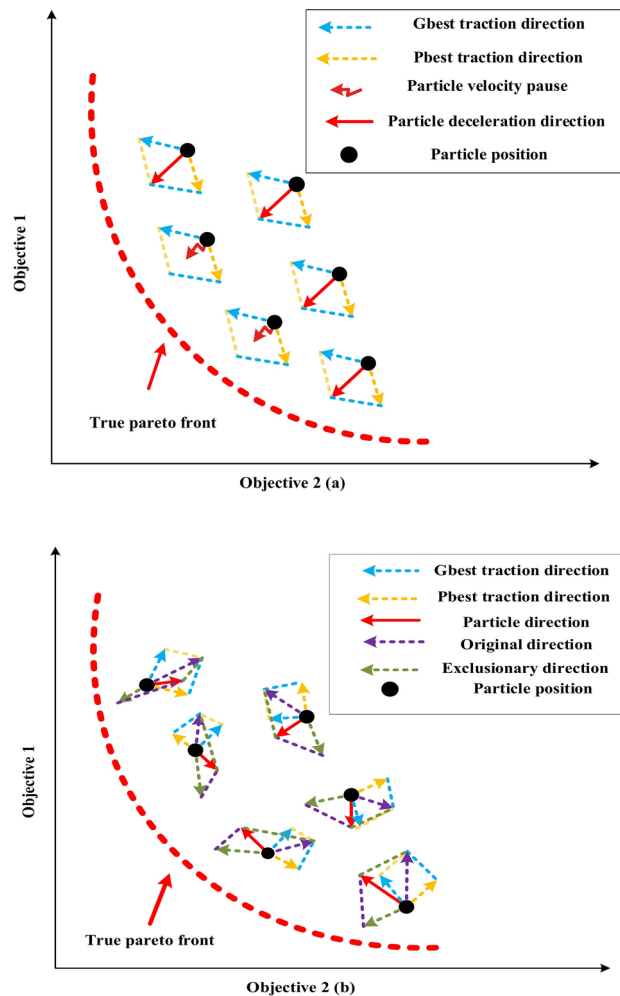


Fig. 2. Search strategy for subgroups P_A and P_B .

of the convergence of particles. The higher the score, the better the convergence is. The higher the score, the better the convergence.

To reasonably combine As and Fs , the number of high-quality particles in the archive is examined to reflect the evolutionary state of the particles, which is set as the adaptive weight. In the early stage of evolution, the diversity of the population is important, and the adaptive weights account for a large proportion in the calculation of the Ps score; As evolution advances and the particles tend to converge, the adaptive weights are adjusted to make the fs play a greater role so as to comprehensively synthesize the strengths and weaknesses of the different states of the particles.

Based on this, we can accurately classify the population in different evolutionary states according to the Ps score. For the classified particles, tasks are assigned according to their characteristics, such as allowing particles with good diversity but insufficient convergence to explore new spaces and particles with good convergence to perform local search optimization, improving the optimization effect and pushing the particles to approach the optimal solution.

In order to improve the optimization refinement, the population particles are divided into three parts according to 1:3:1. The first part is used to explore the initial search direction; the middle three parts balance the exploration and convergence in different evolution stages to optimize the quality of the solution; the last part deeply polishes the better solution region to ensure the high precision of the solution. Through this division and synergy, the particles play the maximum performance in their respective areas to enhance the performance of the optimization system.

Input: F, P

Output: P_A, P_B, P_C

% F is fitness values of particles

% P is particle swarm

1. Initialize particle-related properties

$n = \text{size}(P, 1)$; % Number of particles

for $i = 1:n$

$p(i).p_r = 1$; % Initialize the Pareto level to 1

$p(i).d_s = []$; % Initialize the set of particles dominated by this particle to

empty

$p(i).d_b_s = []$; % Initialize the set of particles dominated by this particle to be

empty

end

2. Get P and M and resolution ratio of the objective space by P and F .

3. Calculate the number of reference vectors using formula (6).

4. Calculate the uniformity factor φ using formula (12).

5. By using Equation (7), a reference vector is established.

6. Compute the angles between particles and reference vectors.

7. Calculate θ_i and r_j using cosine similarity. Store the angle as $As(i, j)$

8. Pareto sorting process

$ft = \{\}$; % Array of cells for storing different Pareto fronts

$cft = []$; % cft denotes the current Pareto frontier

for $i = 1:n$

 for $j = i + 1:n$

$dom_i = \text{all}(F(i, :) \leq F(j, :))$.

$dom_j = \text{all}(F(j, :) \leq F(i, :))$.

 if dom_i ; $p(i).d_s = [p(i).d_s; j]$; $p(j).d_b_s = [p(j).d_b_s; i]$; end

 if dom_j ; $p(j).d_s = [p(j).d_s; i]$; $p(i).d_b_s = [p(i).d_b_s; j]$; end

 end

 if isempty($p(i).d_b_s$); $cft = [cft; i]$.

end

$ft\{1\} = cft$.

level = 2.

while ~isempty(cft)

$nft = []$.

 for $idx = 1:\text{length}(cft)$

 for $dp = p(cft(idx)).d_s$

$p(dp).d_b_s(p(dp).d_b_s == cft(idx)) = []$.

 if isempty($p(dp).d_b_s$); $p(dp).p_r = \text{level}$; $nft = [nft; dp]$; end

 end

 end

$cft = nft$.

 if ~isempty(cft); $ft\{\text{level}\} = cft$; level = level + 1; end

end

9. Calculate the overall priority score Ps using formula (13).

10. Sort the P by their overall priority score Ps .

11. Divide P into subgroups in a 1:3:1 ratio:

P_A = top 1/5 of P from sorted list (Level 1)

P_B = middle 3/5 of P from sorted list (Level 2)

P_C = bottom 1/5 of P from sorted list (Level 3)

12. Output subgroups P_A, P_B, P_C .

Algorithm 1. Population segmentation.

Tasking strategy

A single search strategy is often difficult to effectively balance the conflicts between different goals. To alleviate the pressure of localized search and, at the same time, improve the search efficiency of each subpopulation, a new speed-controlled search mechanism is assigned based on the subpopulation characteristics.

The subpopulation P_A contains the best particles in the population and is therefore small. They are in the high-quality region of the Pareto front and solution space, and their task is to focus on further optimizing the existing excellent solutions. Therefore, their main task is to further improve the accuracy and quality of the

solutions through local optimization. Inspired by³⁸ a search mechanism with velocity pause (Fig. 2a) for the characteristics of P_A subpopulation, a velocity decay factor r_t Eq. (14) and a velocity pause threshold γ Eq. (15) are added to this search mechanism, where r_t gradually decays with the population evolution during iterations to improve the search accuracy, where the velocity pause threshold R is based on the particles in the archive per iteration exchange rate changes adaptively.

$$r_t = e^{-\frac{\beta \cdot t}{T}} \quad (14)$$

$$\gamma = e^{-\left(\frac{rk+sk}{NS}\right) \cdot \frac{t}{T}} \quad (15)$$

where $\beta = 0.8$ controls a constant for the decay rate; rk and sk are the particles removed from the archive and introduced into the archive in each iteration, respectively; NS is the number of particles in the archive; t is the number of current iterations; and T is the maximum number of iterations.

During the search process when the archive growth rate is large it will allow a small portion of the particles in the subpopulation P_A to move with a small probability of velocity pause i.e. the same velocity as in the previous iteration while the remaining portion performs the decelerated search mechanism. The subpopulation P_A particle velocity position is updated in the following way:

$$V_{Ai}(t+1) = \begin{cases} V_{Ai}(t); & \text{if } r < \gamma \\ r_t \cdot V_{Ai}(t) + c_1 \cdot r_1 \cdot (pbest_{Ai} - P_{Ai}) + c_1 \cdot (gbest_i - P_{Ai}); & \text{otherwise} \end{cases} \quad (16)$$

$$P_{Ai}(t+1) = P_{Ai}(t) + V_{Ai}(t+1) \quad (17)$$

where the setup parameters take the values $c_1 = 0.5$; r_1 and r are uniform random numbers between $[0,1]$.

Subpopulation P_B is the most numerous subpopulations and is mainly responsible for the global search. Since these particles are still some distance away from the optimal solution, they have a greater potential to explore other regions of the solution space. By increasing the number of particles, the diversity of the population can be enhanced, effectively avoiding premature convergence to the local optimal solution, thus ensuring a more comprehensive and extensive search process. The particles of the subpopulation P_B take on the tasks of global search and local optimization. Inspired by the search mechanism in²², in this paper, by introducing the repulsion mechanism RD and adaptive deceleration r_t , this repulsion mechanism helps the particles to disperse (Fig. 2b), enhances the coverage of the solution space, and prevents over-concentration in certain regions. It enables them to effectively explore multiple regions of the solution space to avoid premature convergence, while gradually improving the quality of the solution. Overall, the subpopulation P_B maintains the population diversity and pushes the algorithm to converge towards the global optimal solution through a strong global search capability and certain local optimization strategies. The subpopulation P_B particle velocity positions are updated as follows:

$$\begin{cases} V_{Bi}(t+1) = r_t \cdot V_{Bi}(t) + c_2 \cdot r_1 \cdot (pbest_{Bi} - P_{Bi}) + c_2 \cdot (gbest_i - P_{Bi}) + r_t \cdot RD \\ RD = r_1 \cdot c_3 \cdot (pbest_{Bi} - gbest_i) \\ P_{Bi}(t+1) = P_{Bi}(t) + V_{Bi}(t+1) \end{cases} \quad (18)$$

where the setup parameter takes the value: $c_2 = 0.8$; $c_3 = 0.1$;

The subpopulation P_C contains the worst 40 particles, which are located at the edge of the solution space or in regions far from the ideal solution, are less well adapted and have a higher Pareto ranking, meaning that they dominate fewer solutions. The angle to the reference vector is larger, showing that their target values are more directionally different from the ideal solution. The main task of the particles of the subgroup P_C is to accelerate the convergence to the global optimum (Fig. 3), which is enhanced by the global search. The velocity position update formula for subgroup P_C is as follows:

$$\begin{cases} V_{ci}(t+1) = c_4 \cdot V_{ci}(t) + c_4 \cdot (gbest_i - P_{ci}(t)) \\ P_{ci}(t+1) = P_{ci}(t) + V_{ci}(t+1) \end{cases} \quad (19)$$

where the setup parameter takes the value of $c_4 = 1.5$;

In the TAMOPSO algorithm, global search and local optimization problems in multi-objective optimization problems are solved by establishing reference vectors to simultaneously improve the efficiency and convergence accuracy of the algorithm. By dividing the population into different subpopulations, each subpopulation can focus on a specific optimization task, resulting in a broader and finer exploration in the solution space. Three different search tasks are assigned for the three subpopulations; the task of subpopulation P_A is to focus on further optimizing the existing excellent solutions, the particles of subpopulation P_B take on the tasks of global search and local optimization, and the main task of the particles of subpopulation P_C is to accelerate the convergence to the global optimum. Such a division of labor allows the algorithm to search for potential high-quality solutions in a larger range, and at the same time to converge quickly when approaching the optimal solution, and ultimately to obtain a high-quality solution. Pseudo-code for task allocation is given in Algorithm 2.

Input: lb, ub, P, V

Output: P_{Anew} , P_{Bnew} , P_{Cnew} and V_{Anew} , V_{Bnew} , V_{Cnew} .

% lb is lower bounds of the problem space

% ub is upper bounds of the problem space

% P is current particle positions

% V is previous generation particle velocity

1. Use Algorithm 2 to divide particles into three subgroups: P_A , P_B , P_C .
2. Use Formula (16) and (17) to update positions and velocities of particles in subgroup P_A .
3. Use Formula (18) to update positions and velocities of particles in subgroup P_B .
4. Use Formula (19) to update positions and velocities of particles in subgroup P_C .
5. Check for Particle Position Out of Bounds.

for each particle i and each dimension j in each subgroup.

```

    if position  $P_{new}(i, j) < P_{lb}(j)$ 
    |    $P_{new}(i, j) = P_{lb}(j)$ 
    else position  $P_{new}(i, j) > P_{ub}(j)$ 
    |    $P_{new}(i, j) = P_{ub}(j)$ 
    end

```

end

6. Check for Velocity Out of Bounds.

for each particle i and each dimension j in each subgroup.

```

    if velocity  $V_{new}(i, j) < V_{lb}(j)$ 
    |    $V_{new}(i, j) = V_{lb}(j)$ 
    Otherwise, Velocity  $V_{new}(i, j) > V_{ub}(j)$ 
    |    $V_{new}(i, j) = V_{ub}(j)$ 
    end

```

end

7. Output the updated positions P_{Anew} , P_{Bnew} , P_{Cnew} and velocities V_{Anew} , V_{Bnew} , V_{Cnew} .

Algorithm 2. Task Assignment.

Adaptive control of the Lévy flight variation strategy

The variation operation in common MOPSO usually relies on a small range of random perturbations, which tends to cause the particles to oscillate in a local region of the search space and fall into a local optimal solution, which prevents effective global search. Too large a variation amplitude may lead to population degradation. And when the magnitude of variation is too small, the particle population may oscillate around the known solution, slowing down the convergence speed and making it difficult to further improve the quality of the solution. In addition, a fixed or limited variation amplitude may cause the particle population to fall into a local region with local oscillations, which in turn leads to premature convergence and a lack of global search capability, especially in multi-peak optimization problems, where the variation frequency and amplitude fail to be dynamically adjusted, resulting in inefficient search.

In the TAMOPSO variational design, Lévy flights are used as a way of global variability and combined with local variability to balance global and local fine variability. The step size L_i (Eq. 20) generated by the Lévy distribution has a strong jumping property, which enables the particles to make a wide range of jumps and jump out of the local optimal region, avoiding the local oscillations and search limitations of the traditional variational methods.

$$L_i = \frac{u_i}{|v_i|^{\frac{1}{\beta}}} \quad (20)$$

where u_i and v_i are random variables sampled from the normal and Lévy distributions, respectively. β is a parameter of the Lévy distribution that takes the value $1 < \beta \leq 2$ over the range and determines the heavy-tailed nature of the distribution.

Global and local variation in the variation operation is switched according to the variation probability γ (Eq. 15). The mutation probability γ gradually decays with iterations, thus allowing the particle swarm to perform extensive global exploration at the beginning of the search and refine the search through local perturbations at a later stage. In addition, the inclusion of an archival growth rate factor in the mutation probability ensures that the particles do not continue to make excessive jumps as they approach the optimal solution, which avoids population degradation and improves the mutation efficiency. This adaptive mutation strategy improves the convergence speed of the particle population and ensures that the particles can find the global optimal solution more accurately in the final stage. The global and local particle mutation formulas are as follows:

$$P_{new}(i, j) = \begin{cases} P(i, j) + \varepsilon \cdot N(0, 1) \cdot (P(j, j) - P(k, j)); & \text{if } r > \gamma \\ P(i, j) + L_i(j); & \text{otherwise} \end{cases} \quad (21)$$

where $P(j, j)$ and $P(k, j)$ are two randomly selected particles; r is a random number; ε is the magnitude of the perturbation controlling the local variation; In this paper, we set the size $\varepsilon = 0.1$; a smaller perturbation allows the particles to be fine-tuned around their current position, which helps the algorithm to exploit the promising regions that have been found; $u_i \sim N(0, \sigma^2)$ is a normally distributed random variable controlling the direction of the jumps; σ is a scale factor for the Lévy flight step:

$$\sigma = \left(\frac{\Gamma(1+\beta) \cdot \sin\left(\frac{\pi \cdot \beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}} \quad (22)$$

where Γ is the Gamma function, which is a generalization of the factorial; $v_i \sim N(0, 1)$ is a standard normally distributed random variable that controls the scale of the jumps; and β is a parameter of the Lévy distribution, which usually takes values between $1 < \beta \leq 2$, and controls the shape of the distribution of steps.

Input: $P, \gamma, P_{lb}, P_{ub}$

Output: P_{new}

% P is particle positions.

% γ can be calculated using the formula (15).

% P_{lb} are lower bounds of the problem space.

% P_{ub} are upper bounds of the problem space.

% P_{new} are new particle positions.

1. for each particle i generate a random number r to decide the mutation type.

```

    if  $r > \gamma$  (Condition for global mutation)
    |   Perform global mutation using Formula (21).
    else local mutation
    |   Perform local mutation using Formula (21).
    end

```

end

2. Ensure new particle positions are within bounds.

for each particle i and each dimension j .

```

    if  $P_{new}(i, j) < P_{lb}(j)$ 
    |    $P_{new}(i, j) = P_{lb}(j)$ 
    else  $P_{new}(i, j) > P_{ub}(j)$ 
    |    $P_{new}(i, j) = P_{ub}(j)$ 
    end

```

end

3. Return the updated particle positions P_{new} .

Algorithm 3. Adaptive Control of Lévy Flight Variation.

Compared with the traditional Lévy flight implementation, the adaptive control mechanism of TAMOPSO focuses on global exploration (large jumps in Lévy flights) at the early stage of the search through the dynamically decaying variance probability γ and gradually shifts to local perturbations at the later stage for refined exploitation, which solves the problem of ineffective jumps or local oscillations at the later stage due to the fixed probability of the traditional method; meanwhile, TAMOPSO suppresses large jumps at the later

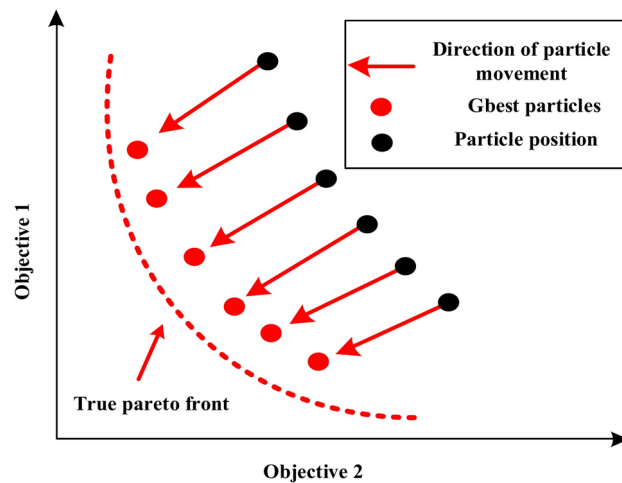


Fig. 3. Search strategy of P_C .

stage by combining a dual-mode variance strategy (global Lévy flights and local normal perturbations) with a dynamically decaying At the same time, TAMOPSO suppresses large jumps in the late stage by dynamically decaying the mutation probability γ and the archival growth rate factor to avoid population degradation and combines the dual-mode mutation strategy (global Lévy flight and local normal perturbation) to achieve adaptive switching of the stage, complemented by the dynamically adjusted Lévy parameter β and the scale factor σ , to synergistically optimize the distribution of the step lengths and the search strengths to reduce the dependence on manual parameter tuning. These improvements allow the algorithm to balance global exploration and local exploitation in multi-peak optimization, significantly improving convergence speed, solution quality, and the ability to avoid premature stagnation. The pseudo-code of Algorithm 3 gives the pseudo-code of the Lévy flight variant with adaptive control.

Uniform contribution margin metrics guide archive maintenance strategies

There are several major problems with common archive maintenance strategies. First, the dominance relationship judgment is inefficient, the dominance between particles needs to be compared frequently, and the computational complexity will rise rapidly when the number of particles increases. Second, if the archive is not limited, the number of particles may keep increasing, leading to increased storage and computational pressure. In addition, archiving usually suffers from insufficient diversity of solutions and easy loss of boundary solutions. Meanwhile, the range of the target space expands or contracts with particle changes, and the fixed grid division cannot be dynamically adjusted, which affects the adaptability of the dynamic target space. Finally, in high-dimensional spaces, the distribution of solutions tends to be biased towards certain target dimensions, leading to insufficient coverage of other dimensions, which affects the equilibrium of solutions. These problems significantly affect the performance of the algorithm, especially in the case of high dimensionality and diversity requirements, which poses a higher challenge to the archive design.

To address the above challenges this paper proposes a new metric-guided archive maintenance strategy. When the archive does not reach the threshold, the algorithm only compares the non-dominated particles in the population after iteration with the particles in the archive in a Pareto comparison, if the new particles dominate the particles in the archive, the dominated particles are deleted from the archive; if the new particles are dominated by the particles in the archive, the new particles are ignored directly, which reduces the size of the archive and the computational redundancy, and improves the efficiency. When the archive capacity exceeds the limitation threshold of 200, particles are filtered by grid division and new indicators. An adaptive grid (hypercube grid in high dimensions) is established based on the fitness values of the particles in the archive and the candidate new particles. In the process of generating the adaptive grid, firstly, for each objective function dimension, the basic range is determined according to the maximum and minimum values of the particle adaptation value, and the boundary is extended outward by 5% (if all the values are the same, the extension of the unit range is mandatory); then the extended range is equally divided into 49 intervals, and 50 equally spaced grid splitting points are generated; finally, by judging the location of the particle's adaptation value in each dimension in each interval, the corresponding multidimensional grid index is determined. This process ensures that the distribution of particles in the target space is uniformly quantified by dynamically adjusting the grid boundaries and interval divisions, providing the basis for subsequent density calculation and diversity maintenance. After establishing the adaptive mesh, each particle is assigned to the grid $g(x_j)$ in the target space:

$$\begin{cases} g(x_j) = [g_1(x_j), g_2(x_j), \dots, g_m(x_j)] \\ g_i(x_j) = \underset{k}{\operatorname{argmin}} |f_i(x_j) - f_{grid,i}[k]| \end{cases} \quad (23)$$

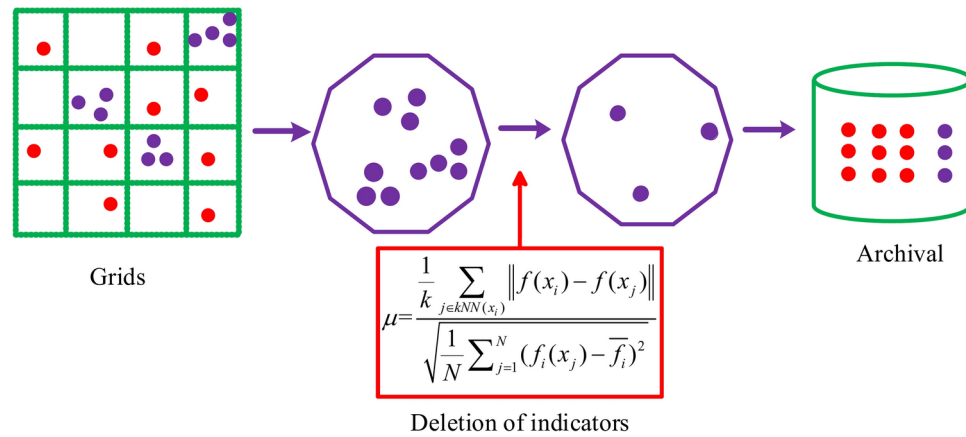


Fig. 4. Uniform contribution factor guides archival maintenance.

For each mesh g_i , the particle density is defined as $d(g) = \sum_{j=1}^N \mathcal{J}(x_i \in g)$ where $\mathcal{J}(\cdot)$ (Eq. 24) is an indicator function of whether the particle belongs to the mesh g_i . The value is 1 when the condition inside the parentheses is true; otherwise, the value is 0. For each particle i , check whether its grid indexes in all dimensions are equal to that of the i th particle, and if they are equal, then the counts are added by 1. Finally, the counts of all the particles are summed up, and the density of the grid where the i th particle is located is obtained.

$$J(x_i \in g) = \begin{cases} 1, & \text{The particle } x_i \text{ belongs to the lattice } g \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

After obtaining the grids and grid densities, all the particles in the densest grids are selected, and then the local uniformity contribution μ (Eq. 25) metric is calculated for these particles; the particles in the high-density grids are measured by the local uniformity contribution metric, and some of the particles that have the smallest contributing to the archive uniformity by the number of exceeding the threshold (The threshold is a predefined size for the archive, which is set to 200, the population size, and setting the threshold to the archive size ensures that the high-quality particles in the archive are maintained within the predefined value for archive maintenance) are deleted (Fig. 4). In high-dimensional target spaces, the distribution of certain target dimensions may dominate the archive update, leading to uneven coverage. The distribution balance between local neighboring particles is measured by the new metric uniform contribution index, which prioritizes the retention of particles that contribute more to the uniformity.

$$\mu = \frac{\frac{1}{k} \sum_{j \in kNN(x_i)} \|f(x_i) - f(x_j)\|}{\sqrt{\frac{1}{N} \sum_{j=1}^N (f_i(x_j) - \bar{f}_i)^2}} \quad (25)$$

where $\bar{f}_i = \frac{1}{N} \sum_{j=1}^N f_i(x_j)$ is the mean value of the i th objective; $kNN(x_i)$ is the set of nearest neighbor solutions, and the algorithm selects two particles as a nearest neighbor; and $f(x_i)$ denotes the value of the i th objective.

It is important to note that this mesh is built adaptively and will dynamically adjust (Eq. 27) the mesh boundaries according to the range of target values of the current particles to ensure the adaptability of the target space. Adapting to the dynamic changes in the target space improves the robustness of the archive update.

$$f_{grid,i} = (\min_i^{ext}, \max_i^{ext}, N_g) \quad (26)$$

In addition, extending the mesh boundary through (Eq. 27) ensures that the boundary solution is preserved. Increasing the retention probability of the boundary solution ensures the integrity of the Pareto front.

$$\min_i^{ext} = \min(f_i) - \epsilon; \max_i^{ext} = \max(f_i) - \epsilon \quad (27)$$

TAMOPSO provides a fine-grained design treatment for archive maintenance, which solves common archive updating and maintenance problems. First, the diversity of the archive is effectively maintained through dynamic grid partitioning and density-driven particle screening to avoid excessive concentration of particles in the target space. Specifically, the algorithm can dynamically adjust the grid division according to the fitness values of the particles and preferentially retains the particles in low-density regions, thus ensuring the uniform distribution of the optimized solutions in the target space. In addition, the archive capacity is effectively controlled, and the archive size is maintained by eliminating particles in high-density regions when the capacity is exceeded, and avoiding the loss of critical solutions (e.g., boundary solutions). This is achieved through adaptive extension of the boundary and the computation of the uniformity contribution factor, allowing the archive to not only

reflect the convergence of the Pareto front, but also to fully cover the target space. Redundant computations are also reduced through efficient dominance relation checking, which improves the speed of updating the archive and can significantly enhance the efficiency of the algorithm, especially when the number of particles is large. Finally, the ability to adjust the mesh and archive content according to the particle distribution ensures that a good solution distribution is always maintained in the dynamically changing target space. The algorithmic flow of the new index-guided archive maintenance strategy is given in Algorithm 4.

Input: t , $Pbest$, $Fbest$, rk , sk

Output: $gbest_{new}$ and $Sfit_{new}$.

% t is current iteration number

% $Pbest$ is the best particle in the first rank of the population

% $Fbest$ is the best particle in the first rank of fitness value

% rk is the particles that need to be added in the archive

% sk is the particles that need to be removed in the archive

1. Find the best particle j from the first rank of the population based on fitness value.

if $t == 0$:

$gbest = Pbest$

$Sfit_{new} = Fbest$

 Return the updated $gbest_{new}$ and $Sfit_{new}$.

else

 Check if the current particle dominates any particle in the archive;

if $sk == 0$ and $rk == 0$

 Do not update the archive.

else if $sk > 0$:

 Remove the dominated solutions from the archive.

else if $rk > 0$:

 Add the dominating solutions to the temporary archive.

end

 Check Archive Size.

if archive size exceeds resize

 Compute the grid partition of all particles using Formula (24).

 Select the particles in low-density grids (grid particle count ≤ 2).

 Use Formula (25) to remove particles from high-density grids until the archive size satisfies the limit.

end if

 Return the updated $gbest_{new}$ and $Sfit_{new}$.

end if

Algorithm 4. New indicators guide archive maintenance policies.

Individual optimal update strategy

In traditional optimization algorithms, the update of the individual optimal solution usually relies on the comparison of single-objective fitness. Specifically, particles will choose the solution with a smaller fitness value to update their individual optimal solution. Although this approach performs well in single-objective optimization problems, it often faces problems such as dimensionality pressure and unfair selection in multi-objective optimization problems or more complex optimization problems.

Input: P_t, t, P_{t-1}
Output: P_{best} .

% P_t is current generation particle
 % t is current iteration number
 % P_{t-1} is previous generation individual best particle
 % P_{best} is best individual positions

if $t == 0$:
 | $P_{best} = P_t$
else
 | Set the ideal point to zero.
 | Loop Over Each Particle to Update Individual Best Positions.
 | **for** each particle i
 | | **if** the current generation particle i dominates the previous generation
 | | particle i .
 | | | Update the individual best to the current particle position.
 | | **else if** the current generation particle is dominated by the previous
 | | | generation particle.
 | | | Update the individual best to the previous generation particle position.
 | | **else if** neither particle dominates the other:
 | | | Calculate the L_2 norm between the current solution and the ideal
 | | | point:
 | | | | **if the** L_2 norm of the current particle is smaller than the previous
 | | | | | P_{best} : Update $P_{best}[i]$ with the current particle.
 | | | | **end**
 | | **end**
 | **end**
 | Update the overall P_{best} .
end if

Algorithm 5. P_{best} update strategy.

To solve these problems, this paper proposes an individual optimal solution updating method based on the Pareto dominance criterion and L_2 paradigm (L_n paradigm in the n -dimensional case). Specifically, when particles update individual optimal solutions, the Pareto dominance criterion is first utilized to compare the fitness of the solutions. The Pareto dominance criterion can consider the relative relationship between multiple objectives simultaneously, rather than relying solely on the fitness value of one objective. When the fitness vectors between particles do not dominate each other, traditional methods are often unable to fairly choose which solution is superior. In this case, relying solely on the fitness of one objective to update the individual optimal solution may lead to unreasonable results, and it may even appear that some solutions are over-optimized while others are ignored. To solve this problem, when the solutions between particles are not dominated by each other, the algorithm selects the solution that is closer to the super-ideal point by calculating the L_2 paradigm distance

Problems	N	M	D	FES
ZDT1~ZDT3	200	2	30	10,000
ZDT4-ZDT6	200	2	10	10,000
UF1~UF7	200	2	30	10,000
UF8~UF10	200	3	30	10,000
ZDTL1	200	3	7	10,000
DTLZ2~DTLZ6	200	3	12	10,000
DTLZ7	200	3	22	10,000

Table 1. Relevant parameter settings.

between the particle fitness vector and the super-ideal point (where the super-ideal point is chosen to be the zero point). In this way, the algorithm can select a solution more fairly in a multi-objective optimization problem, avoiding over-optimization of one objective while considering the optimization of other objectives, ensuring the balance and effectiveness of the global optimization. Algorithm 5 demonstrates the individual optimal update strategy.

Problem	M	D		CMOPSO	MOPSO	MOPSODC	MPSOD	NMPSO	TAMOPSO
ZDT1	2	30	Mean	3.1089e-3	1.5425e+0	2.8044e-3	8.8460e-2	3.2785e-2	2.8629e-03
			Std	(3.79e-4)-	(5.67e-2) -	(8.52e-4) ≈	(2.98e-2)-	(1.52e-2)-	(1.16e-04)
ZDT2	2	30	Mean	2.9354e-3	3.0281e+0	1.2560e-1	1.4717e-1	2.7148e-2	2.8956e-03
			Std	(3.93e-4) ≈	(1.75e-1)-	(1.60e-1)-	(1.05e-1)-	(3.98e-2)-	(1.36e-04)
ZDT3	2	30	Mean	3.8697e-3	1.1887e+0	4.2593e-2	2.0128e-1	9.7415e-2	3.3762e-03
			Std	(8.99e-4)-	(1.24e-1)-	(5.38e-2)-	(5.69e-2)-	(7.77e-3)-	(2.16e-04)
ZDT4	2	10	Mean	1.8359e+1	9.2757e+0	1.9527e+1	3.6605e+1	1.4874e+1	1.3611e+00
			Std	(6.38e+0)-	(3.73e+0)-	(7.14e+0)-	(6.02e+0)-	(5.91e+0)-	(5.51e-01)
ZDT6	2	10	Mean	1.5804e-3	5.4694e+0	3.7702e-3	1.8335e-2	2.3036e-3	2.1733e-03
			Std	(3.30e-5) +	(4.00e-1)-	(1.75e-3)-	(1.40e-2)-	(2.54e-4)-	(1.20e-04)
UF1	2	30	Mean	9.8341e-2	5.0532e-1	6.1797e-1	2.6626e-1	1.2835e-1	1.5294e-01
			Std	(1.88e-2) +	(6.61e-2)-	(1.46e-1)-	(4.71e-2)-	(2.88e-2) +	(1.12e-02)
UF2	2	30	Mean	6.3474e-2	1.0123e-1	1.3692e-1	1.1442e-1	8.3768e-2	9.0843e-02
			Std	(7.13e-3) +	(6.93e-3) +	(1.51e-2)-	(9.45e-3)-	(6.84e-3) +	(4.64e-03)
UF3	2	30	Mean	3.9185e-1	4.5927e-1	3.7261e-1	4.9642e-1	3.6293e-1	3.0269e-01
			Std	(3.57e-2)-	(2.95e-2)-	(6.00e-2)-	(1.53e-2)-	(5.83e-2)-	(7.37e-03)
UF4	2	30	Mean	1.1075e-1	1.8241e-1	7.9462e-2	9.8798e-2	6.2638e-2	5.7955e-02
			Std	(9.39e-3)-	(3.86e-3)-	(8.54e-3)-	(5.38e-3)-	(6.99e-3)-	(2.60e-03)
UF5	2	30	Mean	8.3862e-1	2.6031e+0	4.0151e+0	2.6669e+0	1.6637e+0	1.8086e+00
			Std	(2.64e-1) +	(3.27e-1)-	(3.48e-1)-	(2.14e-1)-	(3.95e-1) +	(1.42e-01)
UF6	2	30	Mean	3.9394e-1	2.1800e+0	2.6883e+0	1.3505e+0	6.5427e-1	7.4356e-01
			Std	(5.71e-2) +	(3.08e-1)-	(7.61e-1)-	(2.83e-1)-	(1.41e-1)-	(8.77e-02)
UF7	2	30	Mean	1.4658e-1	5.7117e-1	5.8562e-1	2.3903e-1	2.2935e-1	1.3887e-01
			Std	(1.40e-1) ≈	(6.51e-2)-	(1.46e-1)-	(5.84e-2)-	(1.93e-1)-	(1.20e-02)
UF8	3	30	Mean	5.9317e-1	4.2545e-1	7.6196e-1	5.5388e-1	4.9773e-1	2.6714e-01
			Std	(9.31e-2)-	(3.02e-2)-	(1.67e-1)-	(4.56e-2)-	(1.34e-1)-	(4.11e-02)
UF9	3	30	Mean	8.5089e-1	5.4095e-1	8.3789e-1	6.7275e-1	4.6825e-1	1.4466e-01
			Std	(1.26e-1)-	(3.44e-2)-	(1.53e-1)-	(4.57e-2)-	(5.89e-2)-	(1.10e-02)
UF10	3	30	Mean	4.6280e+0	1.4478e+0	5.0918e+0	4.1226e+0	1.5282e+0	7.2324e-01
			Std	(6.37e-1)-	(2.64e-1)-	(9.66e-1)-	(3.31e-1)-	(3.97e-1)-	(9.85e-02)
DTLZ1	3	7	Mean	1.5466e+1	6.3265e+0	1.9884e+1	1.1179e+1	5.7439e+0	1.7377e+01
			Std	(3.86e+0) ≈	(2.97e+0) +	(4.20e+0)-	(2.40e+0) +	(3.27e+0) +	(4.52e+00)
DTLZ2	3	12	Mean	4.1141e-2	4.9427e-2	8.9362e-2	4.5134e-2	5.6008e-2	7.4888e-02
			Std	(6.16e-4) +	(2.03e-3)-	(7.11e-3)-	(1.23e-3)-	(1.96e-3)-	(2.99e-03)
DTLZ3	3	12	Mean	1.8936e+2	1.7146e+2	1.0683e+2	1.4872e+2	1.1931e+2	1.3561e+02
			Std	(3.51e+1)-	(5.90e+1)-	(3.92e+1) +	(1.49e+1) ≈	(2.91e+1) ≈	(4.79e+01)
DTLZ4	3	12	Mean	4.5427e-2	1.9778e-1	3.3380e-1	1.3705e-1	7.4063e-2	2.9281e-01
			Std	(6.61e-3) +	(1.12e-1) +	(4.73e-2)-	(4.26e-2) +	(8.83e-2) +	(2.88e-02)
DTLZ5	3	12	Mean	4.5395e-3	5.0605e-3	3.4049e-2	5.3891e-2	7.0950e-3	2.2463e-02
			Std	(3.66e-4) +	(6.49e-4) +	(8.36e-3)-	(6.00e-3)-	(8.37e-4) +	(1.99e-03)
DTLZ6	3	12	Mean	8.2014e-2	7.3512e+0	1.1154e-1	1.0759e+0	1.3617e-2	2.8026e-03
			Std	(2.37e-1) ≈	(1.76e-1)-	(2.73e-1)-	(5.11e-1)-	(3.28e-3)-	(1.51e-04)
DTLZ7	3	22	Mean	4.8487e-2	6.3261e+0	7.1281e-2	5.5927e-1	4.7756e-2	7.2873e-02
			Std	(3.59e-3) +	(5.28e-1)-	(2.05e-2) ≈	(1.43e-1)-	(7.70e-3) +	(3.54e-02)
Best/All +/-/≈				8/22 9/9/4	0/22 4/18/0	1/22 1/19/2	0/22 2/19/1	2/22 7/14/1	11/22 -

Table 2. IGD values five MOPSOs and TAMOPSOs on three sets of test problems.

Problem	M	D		CMOPSO	MOPSO	MOPSOCD	MPSOD	NMPSO	TAMOPSO
ZDT1	2	30	Mean	7.1997e-1	0.0000e+0	7.2098e-1	5.9247e-1	6.8722e-1	7.2122e-01
			Std	(5.99e-4)-	(0.00e+0)-	(1.21e-3) ≈	(4.19e-2)-	(1.53e-2)-	(2.10e-04)
ZDT2	2	30	Mean	4.4466e-1	0.0000e+0	3.2725e-1	2.7506e-1	4.2701e-1	4.4569e-01
			Std	(6.75e-4)-	(0.00e+0)-	(1.39e-1)-	(9.42e-2)-	(3.79e-2)-	(3.42e-04)
ZDT3	2	30	Mean	6.0022e-1	1.9031e-3	5.7804e-1	4.5734e-1	5.6886e-1	6.0024e-01
			Std	(1.54e-3) ≈	(3.78e-3)-	(4.02e-2)-	(5.41e-2)-	(4.15e-3)-	(1.06e-03)
ZDT4	2	10	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	3.5731e-03
			Std	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(1.41e-02)
ZDT6	2	10	Mean	3.9034e-1	0.0000e+0	3.8828e-1	3.7438e-1	3.8976e-1	3.8927e-01
			Std	(3.79e-5)+	(0.00e+0)-	(1.66e-3)-	(1.32e-2)-	(2.29e-4)+	(4.40e-04)
UF1	2	30	Mean	5.7477e-1	1.6147e-1	8.6441e-2	3.6480e-1	5.2284e-1	4.8126e-01
			Std	(1.92e-2)+	(4.24e-2)-	(7.34e-2)-	(4.86e-2)-	(4.51e-2)+	(1.63e-02)
UF2	2	30	Mean	6.4484e-1	6.1041e-1	5.4861e-1	5.7726e-1	6.1994e-1	6.1459e-01
			Std	(5.97e-3)+	(6.86e-3)-	(1.88e-2)-	(1.09e-2)-	(7.37e-3)+	(5.48e-03)
UF3	2	30	Mean	2.6340e-1	1.8157e-1	2.5821e-1	1.7522e-1	2.7408e-1	3.4207e-01
			Std	(2.44e-2)-	(2.13e-2)-	(4.86e-2)-	(1.52e-2)-	(5.50e-2)-	(7.31e-03)
UF4	2	30	Mean	2.8985e-1	2.1166e-1	3.3288e-1	3.0795e-1	3.6063e-1	3.6096e-01
			Std	(1.21e-2)-	(2.77e-3)-	(1.12e-2)-	(6.69e-3)-	(9.80e-3) ≈	(3.63e-03)
UF5	2	30	Mean	2.3976e-2	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+00
			Std	(3.53e-2)+	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+00)
UF6	2	30	Mean	1.4004e-1	0.0000e+0	4.3298e-4	0.0000e+0	2.9290e-2	2.1788e-03
			Std	(7.30e-2)+	(0.00e+0)-	(2.37e-3)+	(0.00e+0)-	(3.14e-2)+	(3.83e-03)
UF7	2	30	Mean	4.2454e-1	5.8889e-2	4.8192e-2	2.6922e-1	3.5697e-1	3.7550e-01
			Std	(9.66e-2)+	(3.47e-2)-	(5.17e-2)-	(5.80e-2)-	(1.28e-1) ≈	(1.65e-02)
UF8	3	30	Mean	1.5244e-2	2.6058e-1	8.4371e-3	5.0200e-2	2.7115e-1	3.3479e-01
			Std	(1.67e-2)-	(3.93e-2)-	(1.10e-2)-	(2.13e-2)-	(7.52e-2)-	(1.35e-02)
UF9	3	30	Mean	2.9380e-2	2.5611e-1	4.6082e-2	1.0255e-1	3.1604e-1	6.1236e-01
			Std	(3.05e-2)-	(2.56e-2)-	(4.39e-2)-	(2.98e-2)-	(5.49e-2)-	(1.39e-02)
UF10	3	30	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	7.1769e-02
			Std	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(2.50e-02)
DTLZ1	3	7	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+00
			Std	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+00)
DTLZ2	3	12	Mean	5.5978e-1	5.4053e-1	4.8488e-1	5.5045e-1	5.6953e-1	4.9988e-01
			Std	(1.30e-3)+	(5.42e-3)+	(1.20e-2)-	(2.80e-3)+	(9.20e-4)+	(4.20e-03)
DTLZ3	3	12	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+00
			Std	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+0) ≈	(0.00e+00)
DTLZ4	3	12	Mean	5.5436e-1	4.6188e-1	2.6705e-1	4.2385e-1	5.6147e-1	3.3446e-01
			Std	(4.00e-3)+	(5.46e-2)+	(6.25e-2)-	(5.85e-2)+	(4.08e-2)+	(2.12e-02)
DTLZ5	3	12	Mean	1.9894e-1	1.9782e-1	1.7352e-1	1.4148e-1	1.9800e-1	1.7683e-01
			Std	(3.31e-4)+	(1.81e-3)+	(5.82e-3) ≈	(7.63e-3)-	(6.08e-4)+	(4.14e-03)
DTLZ6	3	12	Mean	1.8197e-1	0.0000e+0	1.7487e-1	1.5735e-2	1.9783e-1	2.0033e-01
			Std	(5.35e-2)-	(0.00e+0)-	(6.03e-2)-	(3.62e-2)-	(7.97e-4) ≈	(9.41e-04)
DTLZ7	3	22	Mean	2.7364e-1	0.0000e+0	2.6958e-1	6.1776e-2	2.8137e-1	1.5256e-01
			Std	(3.02e-3)+	(0.00e+0)-	(8.46e-3)+	(3.88e-2)-	(1.14e-3)+	(9.50e-03)
Best/All +/-/≈				7/22 10/9/3	0/22 3/18/3	0/22 2/15/5	0/22 2/17/3	3/22 8/8/6	10/22 —

Table 3. HV values for the five MOPSOs and TAMOPSOs on the three sets of test problems.

Algorithm comparison and result analysis

Parameter settings and performance indicators

TAMOPSO employs 22 test problems from three benchmark suites, ZDT³⁹, UF⁴⁰ and DTLZ⁴¹, to evaluate the algorithm performance. These test problems cover different difficulty levels; problems like ZDT1-ZDT3 are relatively basic, while some of the problems in the UF and DTLZ series are more complex. Complex problems may have multimodal, discontinuous, or irregular PF shapes, which make it difficult for the algorithms to find the optimal solution and place higher demands on the algorithms searching ability and their ability to balance convergence and diversity. In terms of dimensionality, the number of decision variables is different; in the two-objective test problem, the number of decision variables is set to 30 for ZDT1-ZDT3 and UF1-UF7 and 10 for

ZDT4 and ZDT6, while in the three-objective test problem, the number of decision variables is 30 for UF8-UF10, 7 for DTLZ1, 12 for DTLZ2-DTLZ6, and 22 for DTLZ7, and the different dimensions mean that the algorithm needs to deal with different sizes of the solution space, and the increase in dimensionality will increase the difficulty of the algorithm in searching for the optimal solution. In terms of objective space characteristics, these problems have different PF shapes, such as concave, convex, multimodal, discontinuous, and irregular, etc. The PFs of the ZDT series may have specific geometrical shapes, while the UF and DTLZ series of problems cover more complex shapes, and these different characteristics can be used to check whether the algorithms can make the solutions as close to the real PFs and good on them as possible when they deal with different types of optimization problems. to the real PF and well-distributed over it.

The relevant parameter settings are presented in Table 1, where FEs denote the maximum number of evaluations. With five existing MOPSO algorithms (CMOPSO⁴², NMPISO⁴³, MPSOD⁴⁴, MOPSOCD⁴⁵, MOPSO) and five MOEA algorithms (DGEA⁴⁶, SPEAR⁴⁷, MOEADD⁴⁸, NSGAIII⁴⁹, IDBEA⁵⁰) Comparison experiments were conducted. To ensure fairness, the parameter settings of all comparison algorithms were consistent with the original text. Each algorithm was run independently 30 times on each test problem. The mean (mean) and standard deviation (Std) of the IGD and HV values were recorded, and the best results for IGD and HV are shown in bold. All experiments were implemented on an AMD Ryzen 7 4800U with Radeon Graphics 1.80 GHz, Windows 11, via MATLAB R2022b. source code for the comparison algorithms is provided by PlatEMO⁵¹.

Performance evaluation indicators

In this paper, the inverse generation distance (IGD)⁵² and hypervolume (HV)⁵³ are used as evaluation metrics for algorithm performance. Although both metrics can synthesize convergence and diversity, a single metric cannot fully evaluate the experimental algorithm. Therefore, these two metrics will be used to evaluate the performance of TAMOPSO with the selected MOPSO and MOEA algorithms.

The IGD (Eq. 28) provides important information about the accuracy of the algorithm and the quality of the solution set by measuring the distance between the approximate solution set and the true Pareto frontier. A smaller value of IGD indicates that the algorithm is closer to the true Pareto frontier in terms of solution quality and distribution.

$$IGD(P^*, S) = \frac{1}{|P^*|} \sum_{x^* \in P^*} dist(x^*, S) \quad (28)$$

where P^* denotes the set of uniformly distributed solutions along PF in the objective space; S is the Pareto optimal set obtained by the algorithm; and $dist(x^*, S)$ is the minimum Euclidean distance between solution x^* and solution S in P^* ;

HV (Eq. 29) is used to measure the volume covered by the solution set in the objective space, which can effectively assess the overall quality of the solution set, especially for problems with high dimensionality of the objective space. HV reflects the dominance and diversity of the optimized solution set, and it is one of the key metrics for assessing the performance of multi-objective optimization algorithms.

$$HV(S, Z^*) = \delta(\cup [f_1(x), Z_1^*] \times \cdots \times [f_m(x), Z_m^*]) \quad (29)$$

where $Z^* = Z_1^*, Z_1^*, \dots, Z_m^*$ is the reference point in the target space where all Pareto optimal solutions dominate; δ is the Lebesgue measure.

Comparison of experimental results

Comparison of IGD and HV with five existing MOPSOs

Tables 2 and 3 show the mean and standard deviation of IGD and HV obtained by the six MOPSO algorithms (CMOPSO, MOPSO, MOPSOCD, MPSOD, NMPISO, and TAMOPSO) after 30 independent runs on 22 test problems. According to the test results in the table, TAMOPSO performed the best among the 22 test problems, obtaining 11 best IGD values and 10 best HV values, making it the algorithm with the highest number of best results among all the algorithms. In addition, to ensure that the conclusions are statistically reliable, a normality test was carried out on all HV and IGD values in the algorithm. The test results show that some of the data do not conform to normal distribution. In view of this, we introduced a non-parametric statistical hypothesis testing approach to the experimental results. The Wilcoxon rank sum test⁵⁴ was applied to assess whether there was a significant difference between the different algorithms at a significance level of $\alpha = 0.05$. In each of the corresponding tables, the last row of the data is denoted by the symbols “+”, “-”, and “≈”, respectively: The results of the other algorithms are significantly better than those of TAMOPSO, significantly lower than TAMOPSO, and like TAMOPSO at the statistical level.

Table 2 demonstrates the significant advantages over the other five MOPSOs. Specifically, in the ZDT series, TAMOPSO outperforms the other five algorithms in four test problems, and only slightly underperforms the algorithm CMOPSO in the ZDT6 problem. However, TAMOPSO performs particularly well in the UF series, outperforming the other five algorithms in all six test problems, showing its strong optimization ability in high-dimensional problems. Although in the DTLZ series, TAMOPSO's advantage is more limited, outperforming the other algorithms in only one test problem, when considering the 22 test functions of the three series, ZDT, UF, and DTLZ, TAMOPSO occupies a dominant position in 11 test problems, outperforming all the other algorithms in terms of number. This result indicates that TAMOPSO has a more stable and comprehensive advantage in solving complex multi-peak and multi-objective optimization problems.

As can be seen from the HV-based performance evaluation results in Table 3, TAMOPSO still obtains the largest number of the best HV results. In the ZDT4 test problem, TAMOPSO shows a significant advantage even though the five MOPSO algorithms have an HV value of 0, indicating that MOPSO is unable to converge at the

Problem	M	D		DGEA	IDBEA	MOEADD	NSGAIII	SPEAR	TAMOPSO
ZDT1	2	30	Mean	1.1374e+0	5.2658e-1	8.7398e-2	7.8574e-2	1.8199e-1	2.8629e-03
			Std	(2.39e-1)-	(7.04e-2)-	(1.79e-2)-	(1.18e-2)-	(2.64e-2)-	(1.16e-04)
ZDT2	2	30	Mean	9.6323e-1	1.9155e+0	1.0371e-1	1.5173e-1	3.9370e-1	2.8956e-03
			Std	(3.96e-1)-	(2.02e-1)-	(1.91e-2)-	(3.17e-2)-	(1.18e-1)-	(1.36e-04)
ZDT3	2	30	Mean	9.0312e-1	3.1473e-1	1.6224e-1	6.9395e-2	1.4980e-1	3.3762e-03
			Std	(2.16e-1)-	(4.08e-2)-	(1.70e-2)-	(1.02e-2)-	(1.49e-2)-	(2.16e-04)
ZDT4	2	10	Mean	6.6065e+0	2.5208e+1	6.8717e-1	2.0896e+0	1.8424e+0	1.3611e+00
			Std	(3.82e+0)-	(6.33e+0)-	(2.35e-1)+	(6.88e-1)-	(6.54e-1)-	(5.51e-01)
ZDT6	2	10	Mean	1.0150e-2	4.5030e+0	3.7309e-1	1.2639e+0	1.2020e+0	2.1733e-03
			Std	(4.33e-2)+	(3.39e-1)-	(1.15e-1)-	(2.28e-1)-	(2.19e-1)-	(1.20e-04)
UF1	2	30	Mean	6.7604e-1	4.0411e-1	1.2437e-1	1.1049e-1	1.4101e-1	1.5294e-01
			Std	(1.42e-1)-	(7.25e-2)-	(1.96e-2)+	(1.17e-2)+	(1.89e-2)+	(1.12e-02)
UF2	2	30	Mean	1.6866e-1	1.5814e-1	6.4422e-2	7.4636e-2	7.2375e-2	9.0843e-02
			(Std)	(2.04e-2)-	(1.90e-2)-	(5.59e-3)+	(4.51e-3)+	(7.97e-3)+	(4.64e-03)
UF3	2	30	Mean	5.6405e-1	6.0056e-1	4.5899e-1	4.6563e-1	4.3503e-1	3.0269e-01
			Std	(4.66e-2)-	(2.50e-2)-	(8.41e-3)-	(1.17e-2)-	(1.37e-2)-	(7.37e-03)
UF4	2	30	Mean	1.2094e-1	1.0740e-1	8.5987e-2	8.8882e-2	8.5695e-2	5.7955e-02
			Std	(9.81e-3)-	(3.76e-3)-	(2.79e-3)-	(2.95e-3)-	(1.81e-3)-	(2.60e-03)
UF5	2	30	Mean	2.8977e+0	3.0335e+0	1.2249e+0	1.1543e+0	1.0835e+0	1.8086e+00
			Std	(6.20e-1)-	(2.96e-1)-	(2.33e-1)+	(3.10e-1)+	(1.85e-1)+	(1.42e-01)
UF6	2	30	Mean	2.5936e+0	1.8714e+0	6.8185e-1	6.1161e-1	6.4877e-1	7.4356e-01
			Std	(5.81e-1)-	(2.68e-1)-	(1.09e-1)+	(9.38e-2)+	(7.89e-2)+	(8.77e-02)
UF7	2	30	Mean	6.7997e-1	4.6326e-1	1.4727e-1	1.5408e-1	2.0297e-1	1.3887e-01
			Std	(1.66e-1)-	(7.49e-2)-	(6.69e-2)≈	(9.38e-2)≈	(6.61e-2)-	(1.20e-02)
UF8	3	30	Mean	7.3590e-1	3.6217e-1	3.0571e-1	3.8533e-1	3.1945e-1	2.6714e-01
			Std	(1.19e-1)-	(3.41e-2)-	(2.30e-2)-	(4.06e-2)-	(1.91e-2)-	(4.11e-02)
UF9	3	30	Mean	7.7057e-1	5.4342e-1	4.8945e-1	4.8484e-1	4.7647e-1	1.4466e-01
			Std	(9.02e-2)-	(5.18e-2)-	(4.75e-2)-	(6.15e-2)-	(6.95e-2)-	(1.10e-02)
UF10	3	30	Mean	4.8196e+0	2.8855e+0	2.8032e+0	1.6533e+0	1.9346e+0	7.2324e-01
			Std	(7.90e-1)-	(4.91e-1)-	(6.52e-1)-	(3.29e-1)-	(3.66e-1)-	(9.85e-02)
DTLZ1	3	7	Mean	1.3562e+1	3.0893e+0	1.1202e+0	7.6821e-1	8.4065e-1	1.7377e+01
			Std	(6.63e+0)+	(1.38e+0)+	(4.20e-1)+	(2.73e-1)+	(3.32e-1)+	(4.52e+00)
DTLZ2	3	12	Mean	1.2052e-1	4.3938e-2	3.8912e-2	3.9689e-2	4.3004e-2	7.4888e-02
			Std	(1.76e-2)-	(1.66e-3)+	(3.66e-4)+	(5.34e-4)+	(1.63e-3)+	(2.99e-03)
DTLZ3	3	12	Mean	1.0677e+2	7.4428e+1	4.0488e+1	2.7961e+1	3.3862e+1	1.3561e+02
			Std	(6.64e+1)≈	(1.26e+1)+	(7.60e+0)+	(6.83e+0)+	(9.49e+0)+	(4.79e+01)
DTLZ4	3	12	Mean	4.7887e-1	6.1805e-2	3.9192e-2	5.6964e-2	4.4235e-2	2.9281e-01
			Std	(1.43e-1)+	(9.07e-2)+	(5.67e-4)+	(9.15e-2)+	(2.02e-3)+	(2.88e-02)
DTLZ5	3	12	Mean	6.4028e-2	1.2453e-2	2.4311e-2	6.5353e-3	2.3208e-2	2.2463e-02
			Std	(3.08e-2)-	(2.48e-3)+	(2.02e-3)-	(5.23e-4)+	(2.46e-3)≈	(1.99e-03)
DTLZ6	3	12	Mean	2.4057e+0	7.0771e-1	1.1156e-1	2.1687e-2	1.3570e-1	2.8026e-03
			Std	(1.46e+0)-	(3.66e-1)-	(8.98e-2)-	(1.07e-2)-	(7.21e-2)-	(1.51e-04)
DTLZ7	3	22	Mean	4.3175e+0	9.5009e-1	3.3870e-1	1.2819e-1	2.0026e-1	7.2873e-02
			Std	(1.42e+0)-	(8.34e-1)-	(2.79e-1)-	(2.19e-2)-	(3.66e-2)-	(3.54e-02)
Best/All +/-/≈				0/22 3/18/1	0/22 5/17/0	4/11 9/12/1	5/22 9/12/1	1/22 8/13/1	12/22 —

Table 4. IGD values for the five MOEAs and TAMOPSO on the three sets of test problems.

final iteration, whereas TAMOPSO can achieve better convergence at the final iteration. This further validates the superior performance of TAMOPSO in terms of convergence and diversity.

Combining the predominance of IGD and HV in Tables 2 and 3, it can be concluded that TAMOPSO optimizes the particle swarm search process by introducing an advanced mechanism, which improves the algorithm's global search ability and local accuracy. In the tests of ZDT series and UF series, the performance of TAMOPSO is even better, reflecting that the algorithm has strong adaptability and performance in dealing with high dimensional and complex objective optimization problems. In the DTLZ series, although the performance is not as good as the former two, it still shows better stability and reliability. Overall, TAMOPSO performs well on several standard test functions, especially in the ZDT series and UF series, and shows obvious advantages in

Problem	M	D		DGEA	IDBEA	MOEADD	NSGAIII	SPEAR	TAMOPSO
ZDT1	2	30	Mean	7.1468e-3	1.5543e-1	6.0235e-1	6.1773e-1	4.9011e-1	7.2122e-01
			Std	(2.30e-2)-	(4.15e-2)-	(2.23e-2)-	(1.52e-2)-	(2.74e-2)-	(2.10e-04)
ZDT2	2	30	Mean	1.7430e-2	0.0000e+0	3.0151e-1	2.5517e-1	7.7049e-2	4.4569e-01
			Std	(7.39e-2)-	(0.00e+0)-	(2.29e-2)-	(3.00e-2)-	(5.56e-2)-	(3.42e-04)
ZDT3	2	30	Mean	5.5081e-2	3.7668e-1	4.9544e-1	5.5584e-1	4.9962e-1	6.0024e-01
			Std	(7.49e-2)-	(2.97e-2)-	(2.11e-2)-	(6.48e-3)-	(1.88e-2)-	(1.06e-03)
ZDT4	2	10	Mean	0.0000e+0	0.0000e+0	9.2977e-2	0.0000e+0	1.3890e-3	3.5731e-03
			Std	(0.00e+0)-	(0.00e+0)-	(8.58e-2)+	(0.00e+0)-	(5.44e-3)≈	(1.41e-02)
ZDT6	2	10	Mean	3.8548e-1	0.0000e+0	7.3412e-2	0.0000e+0	0.0000e+0	3.8927e-01
			Std	(2.36e-2)≈	(0.00e+0)-	(4.60e-2)-	(0.00e+0)-	(0.00e+0)-	(4.40e-04)
UF1	2	30	Mean	8.0010e-2	2.2724e-1	5.3487e-1	5.4919e-1	5.1049e-1	4.8126e-01
			Std	(6.88e-2)-	(6.64e-2)-	(2.41e-2)-	(2.24e-2)+	(2.46e-2)-	(1.63e-02)
UF2	2	30	Mean	5.1203e-1	5.0648e-1	6.3496e-1	6.2692e-1	6.2624e-1	6.1459e-01
			Std	(3.02e-2)-	(2.53e-2)-	(6.78e-3)+	(5.12e-3)≈	(9.54e-3)+	(5.48e-03)
UF3	2	30	Mean	1.2226e-1	1.0205e-1	2.0779e-1	1.9329e-1	2.2383e-1	3.4207e-01
			Std	(2.54e-2)-	(1.40e-2)-	(6.97e-3)-	(9.97e-3)-	(1.02e-2)-	(7.31e-03)
UF4	2	30	Mean	2.7957e-1	2.9540e-1	3.2549e-1	3.2248e-1	3.2415e-1	3.6096e-01
			Std	(1.11e-2)-	(4.86e-3)-	(2.59e-3)-	(3.95e-3)-	(2.34e-3)-	(3.63e-03)
UF5	2	30	Mean	0.0000e+0	0.0000e+0	0.0000e+0	4.7737e-5	4.4937e-4	0.0000e+00
			Std	(0.00e+0)≈	(0.00e+0)≈	(0.00e+0)≈	(0.00e+0)≈	(1.66e-3)+	(0.00e+00)
UF6	2	30	Mean	0.0000e+0	0.0000e+0	1.4218e-2	2.5932e-2	1.6387e-2	2.1788e-03
			Std	(0.00e+0)-	(0.00e+0)-	(1.82e-2)+	(2.19e-2)+	(1.65e-2)+	(3.83e-03)
UF7	2	30	Mean	4.1057e-2	9.5430e-2	3.8884e-1	3.9073e-1	3.3385e-1	3.7550e-01
			Std	(5.15e-2)-	(4.46e-2)-	(6.46e-2)≈	(7.74e-2)≈	(4.67e-2)-	(1.65e-02)
UF8	3	30	Mean	2.0375e-2	1.6709e-1	1.9610e-1	2.5441e-1	1.8780e-1	3.3479e-01
			Std	(2.26e-2)-	(4.45e-2)-	(3.51e-2)-	(4.40e-2)-	(3.96e-2)-	(1.35e-02)
UF9	3	30	Mean	6.4458e-2	2.0722e-1	2.5404e-1	2.7794e-1	2.6440e-1	6.1236e-01
			Std	(4.29e-2)-	(4.34e-2)-	(4.06e-2)-	(4.74e-2)-	(5.61e-2)-	1.39e-02
UF10	3	30	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	7.1769e-02
			Std	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(0.00e+0)-	(2.50e-02)
DTLZ1	3	7	Mean	0.0000e+0	0.0000e+0	0.0000e+0	9.8199e-3	7.8766e-3	0.0000e+00
			Std	(0.00e+0)≈	(0.00e+0)≈	(0.00e+0)≈	(3.63e-2)+	(3.75e-2)-	(0.00e+00)
DTLZ2	3	12	Mean	4.1913e-1	5.5308e-1	5.6395e-1	5.6222e-1	5.5640e-1	4.9988e-01
			Std	(2.79e-2)-	(3.09e-3)+	(1.45e-3)+	(1.06e-3)+	(2.45e-3)+	(4.20e-03)
DTLZ3	3	12	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+00
			Std	(0.00e+0)≈	(0.00e+0)≈	(0.00e+0)≈	(0.00e+0)≈	(0.00e+0)≈	(0.00e+00)
DTLZ4	3	12	Mean	1.8655e-1	5.4412e-1	5.6351e-1	5.5436e-1	5.5434e-1	3.3446e-01
			Std	(7.73e-2)-	(3.87e-2)+	(1.65e-3)+	(3.96e-2)+	(3.73e-3)+	(2.12e-02)
DTLZ5	3	12	Mean	1.4377e-1	1.9209e-1	1.8318e-1	1.9699e-1	1.8536e-1	1.7683e-01
			Std	(3.08e-2)-	(1.72e-3)+	(1.84e-3)+	(5.19e-4)+	(2.11e-3)+	(4.14e-03)
DTLZ6	3	12	Mean	1.4098e-2	1.8294e-2	1.1758e-1	1.8733e-1	1.0298e-1	2.0033e-01
			Std	(4.75e-2)-	(3.24e-2)-	(4.06e-2)-	(6.86e-3)-	(3.44e-2)-	(9.41e-04)
DTLZ7	3	22	Mean	2.2175e-4	6.4058e-2	2.2002e-1	2.2649e-1	1.9950e-1	1.5256e-01
			Std	(1.21e-3)-	(3.38e-2)-	(2.09e-2)+	(9.23e-3)+	(1.24e-2)+	(9.50e-03)
Best/All +/-/≈				0/22 0/18/4	0/22 3/16/3	4/22 7/11/4	6/22 7/11/4	1/22 7/12/3	10/22 —

Table 5. HV values for the five MOEAs and TAMOPSO on the three sets of test problems.

overall optimization quality, solution efficiency and global convergence compared with the other five particle swarm algorithms.

Comparison of IGD and HV with five existing MOEAs

To further validate the performance of the TAMOPSO algorithm, this paper compares it with five common multi-objective evolutionary algorithms MOEAs (DGEA, IDBEA, MOEADD, NSGAIII, SPEAR). Tables 4 and 5 show the mean and standard deviation of these six algorithms on IGD and HV metrics, respectively, after 30 independent runs on 22 standard test problems.

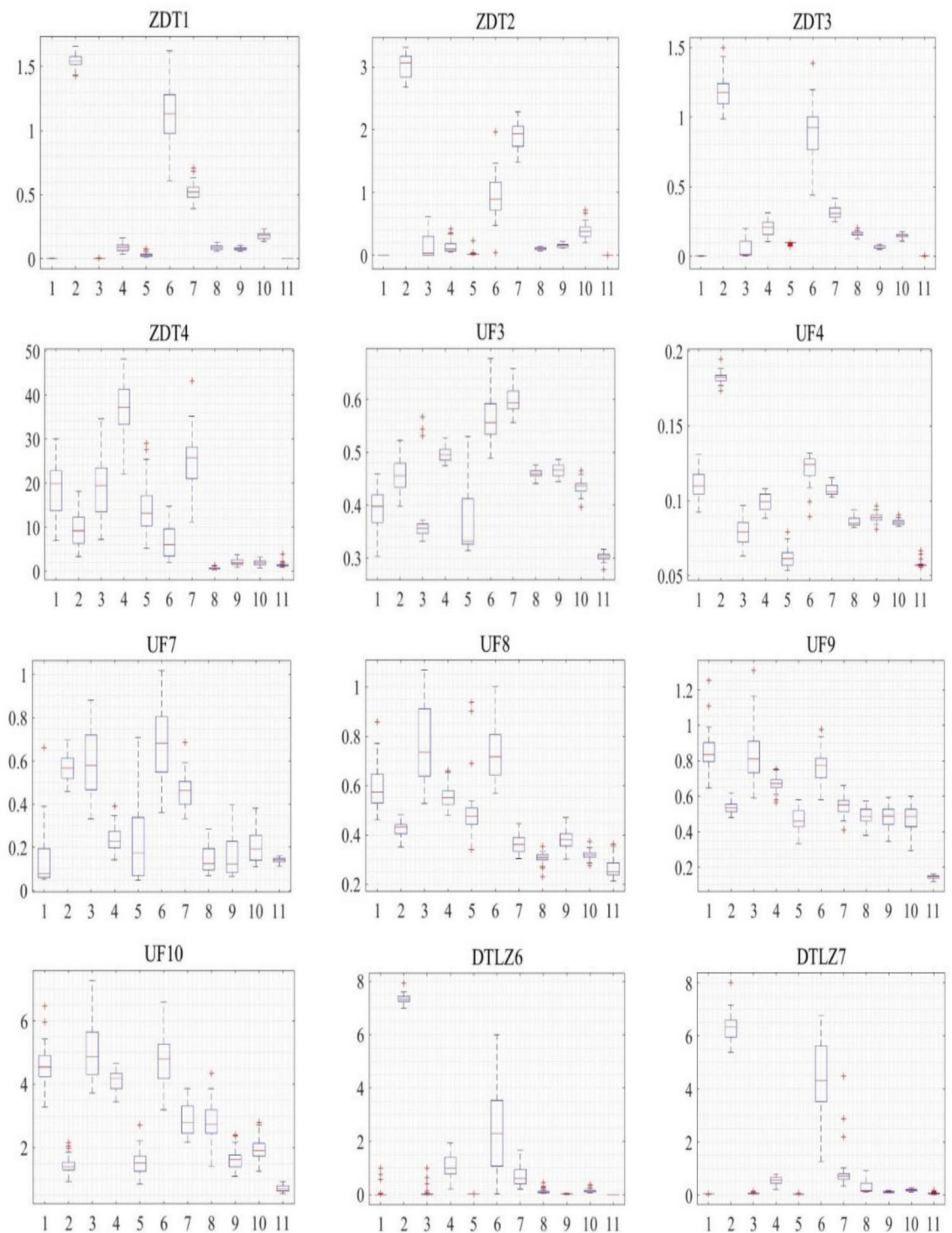


Fig. 5. Box plots of 10 comparison algorithms and TAMOPSO on ZDT3, UF9, and DTLZ6 test problems.

Based on the IGD evaluation results in Table 4, the overall performance of TAMOPSO is significantly better than the other five algorithms on the ZDT, UF and DTLZ series of problems. In the ZDT series of problems, TAMOPSO performs well in all the test problems, only ZDT4 is second only to MOEADD while the other IGD values are lower than the other algorithms, which indicates that it is able to better converge to the Pareto front and maintain the diversity of the solution set, especially in the face of the complex nonlinear Pareto front, and it shows its strong global search ability. TAMOPSO also outperforms the other algorithms in the UF series of problems, with significantly lower IGD values than the other algorithms in most of the test problems, demonstrating its search efficiency in high-dimensional spaces and its ability to deal with goal conflicts. Although TAMOPSO outperforms other algorithms in only two test problems in the DTLZ series of problems,

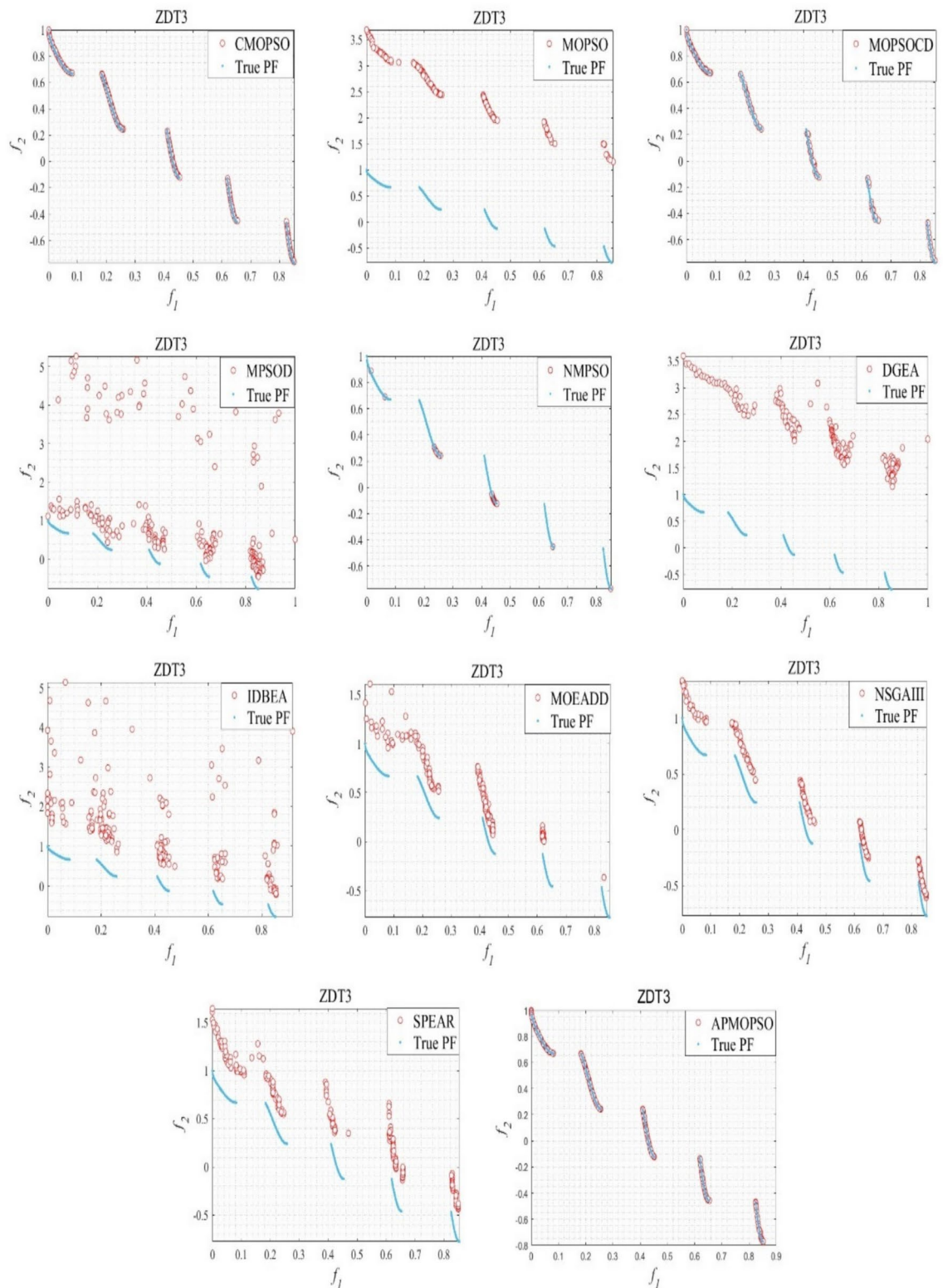


Fig. 6. Distribution of Pareto-optimal solution sets for the 10 algorithms compared and TAMOPSO on the ZDT3 test problem.

it still ranks second in most of the problems, showing strong robustness and stability, especially in dealing with complex high-dimensional problems.

In the HV evaluation results in Table 5, TAMOPSO's performance is not as impressive as in the IGD metrics but still shows a relatively clear advantage. For the ZDT series, TAMOPSO outperforms the other five algorithms in terms of HV on four problems, and only slightly lags and fails to outperform the algorithm NSGAIII on the

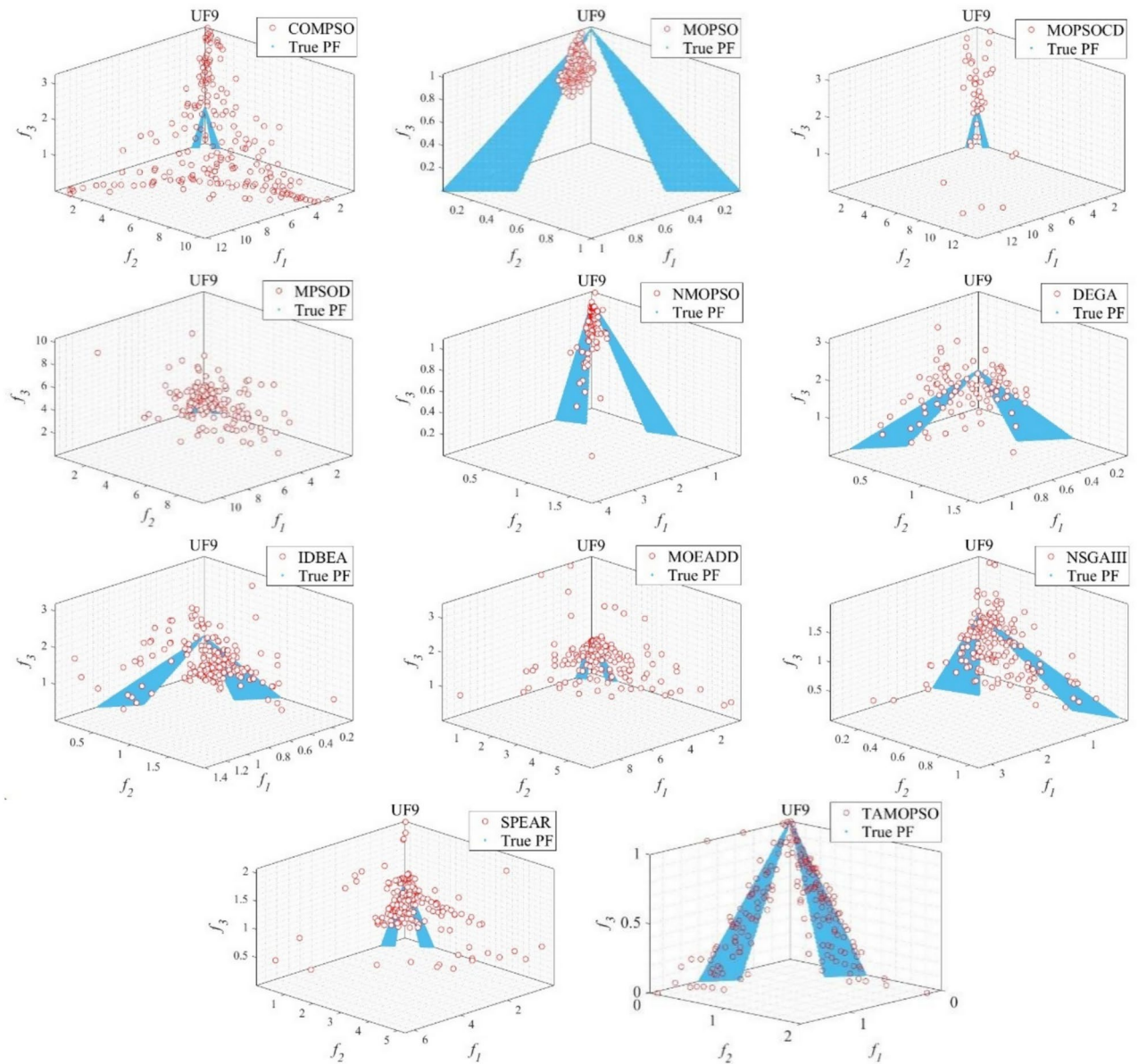


Fig. 7. Distribution of Pareto-optimal solution sets for the 10 algorithms and TAMOPSO on the UF9 test problem.

ZDT4 problem, which indicates that TAMOPSO is still capable of balancing the goal conflicts and converging the Pareto frontiers. For the UF series, TAMOPSO also leads to several test problems, and it outperforms the other algorithms in terms of HV on five test problems, showing its excellent optimization ability in high-dimensional problems. In contrast, TAMOPSO's HV value in the DTLZ series is slightly less impressive, and although it exceeds the other algorithms in only one test problem, it still performs relatively well in the other test problems, placing it at the top of the list.

Overall, TAMOPSO performs well in several test problems of the ZDT and UF series, demonstrating strong global search capabilities and adaptability to multi-objective optimization problems. In the DTLZ series of problems, TAMOPSO maintains a good optimization performance despite facing higher dimensionality and complex Pareto frontiers, proving its robustness and stability in complex multi-objective problems. Overall, the excellent performance of TAMOPSO on these test functions shows that it is an efficient and adaptable multi-objective optimization algorithm suitable for handling a wide range of complex multi-objective optimization tasks.

Visual comparison of algorithms

Box plots measure algorithm stability

Figure 5 illustrates a box plot based on the IGD value data for the MOPSO and MOEAs algorithms in Tables 2 and 4. The 1 to 11 on the horizontal axis represent different algorithms: CMOPSO, MOPSO, MOPSOCD, MPSOD,

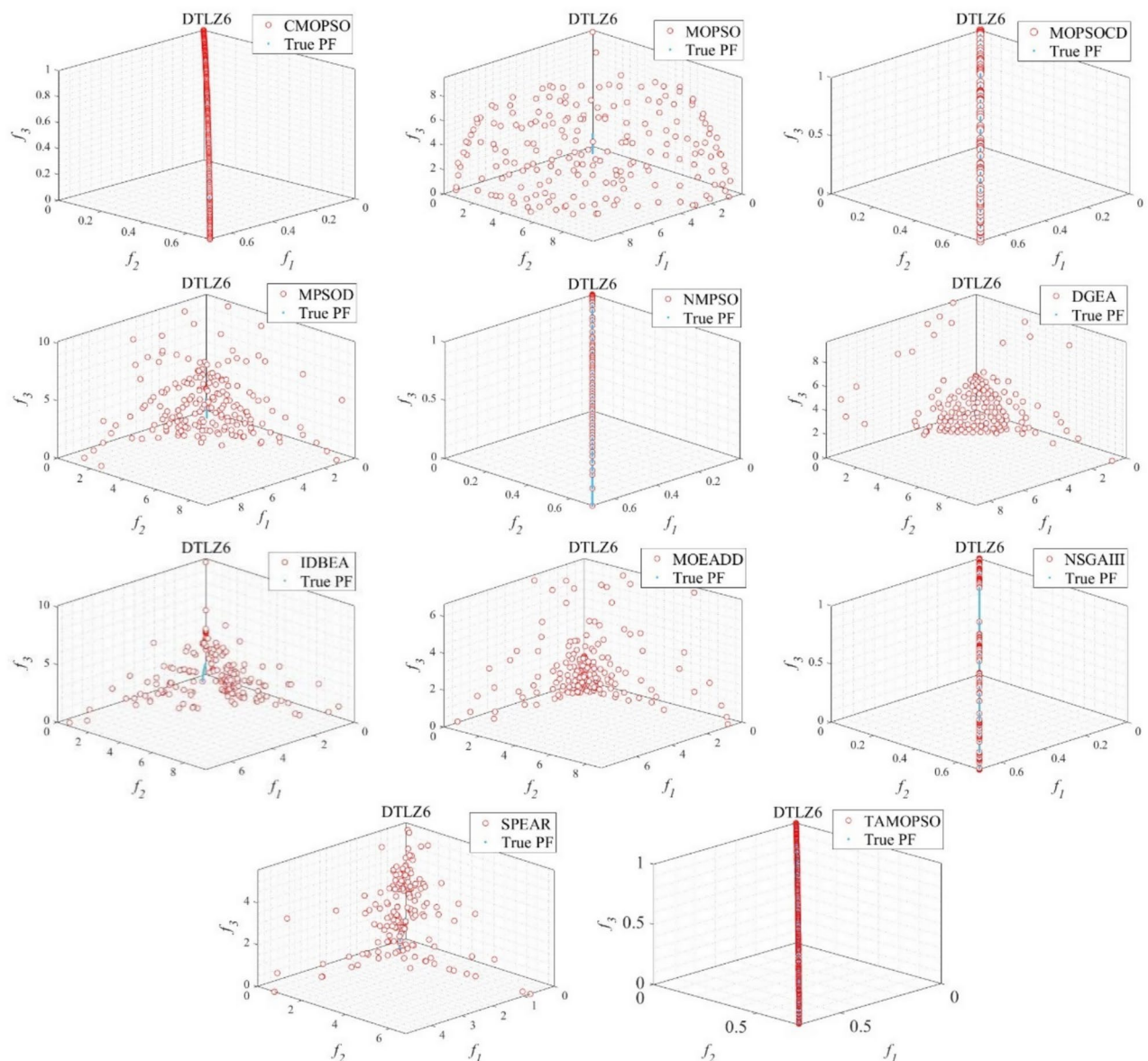


Fig. 8. Distribution of Pareto-optimal solution sets for the 10 compared algorithms and TAMOPSO on the DTLZ6 test problem.

NMPSO, DGEA, IDBEA, MOEADD, NSGAIII, SPEAR, and TAMOPSO, respectively. By scrutinizing the box plots corresponding to these algorithms, we can intuitively find that the box gap of TAMOPSO is significantly smaller, and the upper and lower quartile intervals of its box are relatively narrow. This phenomenon indicates that TAMOPSO exhibits the smallest solution set volatility among all the algorithms, suggesting that its stability is significantly better than the other algorithms in different runs. Therefore, TAMOPSO is not only able to obtain better solution quality but also has higher stability when solving multi-objective optimization problems, which makes it more reliable and advantageous in practical applications.

Comparison of Pareto frontiers

Figures 6, 7, and 8 show the distribution of Pareto-optimal solution sets for the ten comparison algorithms (CMOPSO, MOPSO, MOPSOCD, MPSOD, NMPSO, DGEA, IDBEA, MOEADD, NSGAIII, and SPEAR) versus TAMOPSO on the test problems ZDT3, UF9, and DTLZ6. By observing the distribution of the solution set in the figure, we can intuitively see that TAMOPSO outperforms other algorithms in both its convergence to the PF and its distribution on the PF.

Figure 6 For the bi-objective test problem ZDT3, the performance of TAMOPSO is particularly outstanding compared to the other comparison algorithms. First, intuitively, only CMOPSO and TAMOPSO have solution sets that are both uniformly distributed and converge to the true Pareto front. In particular, the solution set of TAMOPSO not only shows strong convergence to the frontier but also shows a more uniform distribution of the solution set. By analyzing the data in Table 2, the solution set of TAMOPSO is more significant in approaching

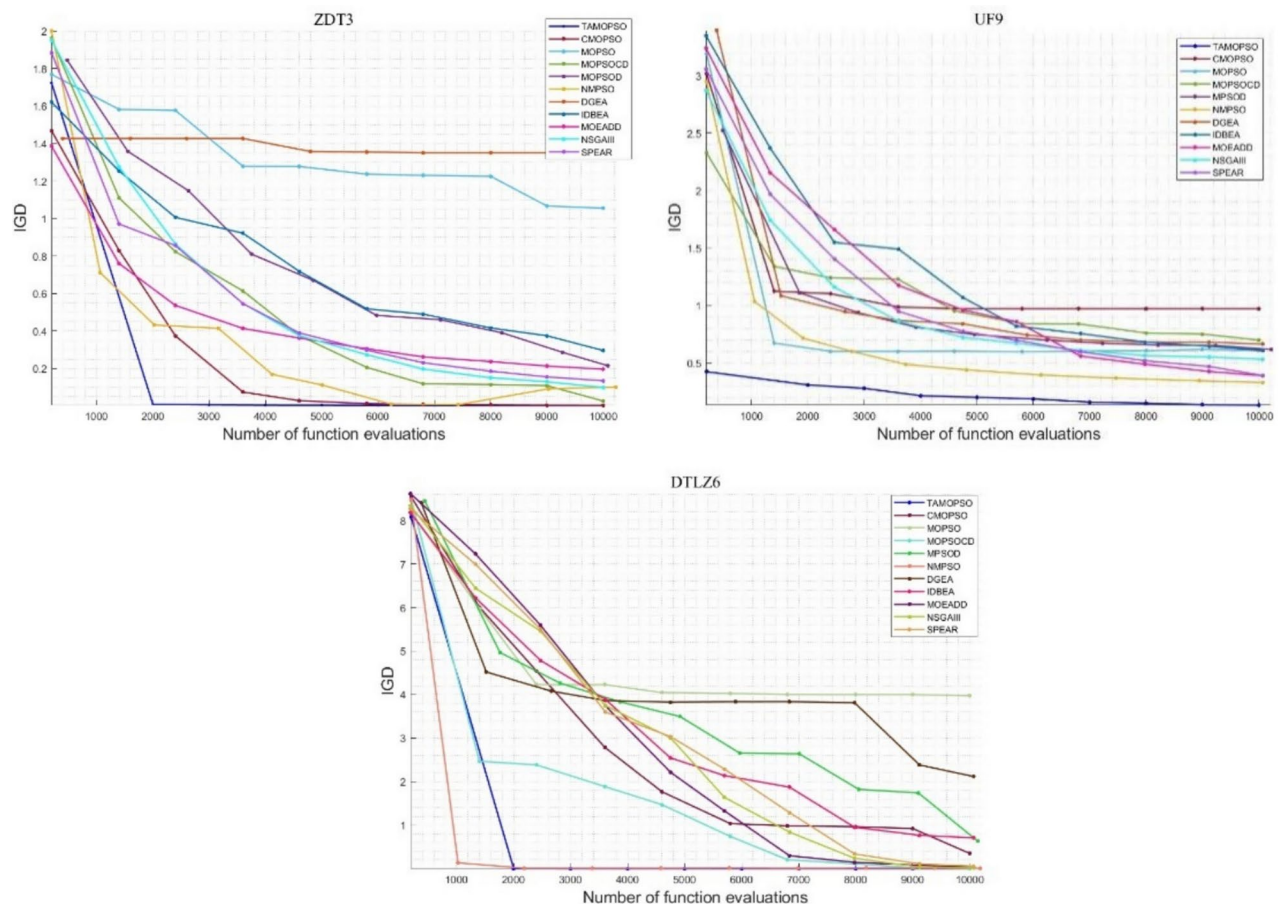


Fig. 9. 10 comparison algorithms and TAMOPSO on ZDT3, UF9, DTLZ6 test problems Convergence trajectory plot.

Algorithm	ZDT1-4,6		UF1-10		DTLZ1-7		Overall	
	Friedman test	Rank	Friedman test	Rank	Friedman test	Rank	Friedman test	Rank
CMOPSO	3.30	2	4.90	6	3.86	2	4.20	3
MOPSO	10.0	11	7.30	7	8.00	9	8.14	9
MOPSOD	4.20	3	8.90	10	7.43	8	7.36	6
MPSOD	7.60	8	7.70	8	8.14	10	7.82	8
NMPSO	4.40	4	3.60	2	4.86	5	4.18	2
DGEA	7.80	9	10.10	11	9.43	11	9.36	10
IDBEA	9.60	10	8.10	9	5.86	6	7.73	7
MOEADD	5.0	5	4.20	4	4.00	3	4.32	4
NSGAIII	5.80	6	4.30	5	3.29	1	4.32	4
SPEAR	6.60	7	3.90	3	4.14	4	4.59	5
TAMOPSO	1.70	1	3	1	7.00	7	3.98	1

Table 6. Friedman rank test of IGD values of TAMOPSO with each comparison algorithm.

the true Pareto frontier compared to CMOPSO. This indicates that TAMOPSO has a greater ability to guide the convergence of the solution set to more accurately approximate the true optimal solution. In addition, the solution sets of the other compared algorithms either fail to approach the true Pareto frontier sufficiently in terms of convergence, or there are obvious localized concentrations in the distributions, resulting in a more limited diversity of solution sets. For the three-objective test problem UF9, the advantages of TAMOPSO are even more obvious.

Figure 7 shows the distribution of the solution sets of the 11 algorithms on this problem, and TAMOPSO can satisfy two key requirements at the same time: one is convergence, and the other is diversity of solution sets. As can be seen from the figure, the solution set of TAMOPSO not only converges perfectly to the true Pareto frontier

Algorithm	ZDT1-4,6		UF1-10		DTLZ1-7		Overall	
	Friedman test	rank	Friedman test	rank	Friedman test	rank	Friedman test	rank
CMOPSO	3.10	2	4.65	5	3.64	2	3.98	2
MOPSO	9.90	10	7.65	8	7.57	8	8.14	10
MOPSOCD	4.10	3	6.90	7	7.29	7	6.39	7
MPSOD	7.30	8	7.85	9	8.14	9	7.82	8
NMPSO	4.30	4	4.00	2	3.14	1	3.80	1
DGEA	6.30	6	9.35	11	9.29	10	8.64	11
IDBEA	9.50	9	8.45	10	6.00	5	7.91	9
MOEADD	5.60	5	4.50	4	5.14	3	4.95	5
NSGAIII	7.00	7	4.10	3	3.64	2	4.61	4
SPEAR	7.30	8	5.10	6	5.43	4	5.70	6
TAMOPSO	1.60	1	3.45	1	6.71	6	4.07	3

Table 7. Friedman rank test for HV values of TAMOPSO with each comparison algorithm.

		ZDT1	ZDT2	UF3	UF4	DTLZ6	DTLZ7
C1	0.1	3.0631e-03	2.8005e-03	2.9482e-01	5.6004e-02	2.8102e-03	2.2968e-01
	0.5	2.8179e-03	2.9067e-03	3.0650e-01	5.7117e-02	2.8089e-03	7.7639e-02
	1	2.8518e-03	2.7447e-03	3.1253e-01	5.7116e-02	2.9459e-03	2.5861e-01
	1.5	2.9921e-03	2.9316e-03	3.1187e-01	6.1023e-02	2.8344e-03	1.1196e-01
C2	0.4	2.7168e-03	2.8168e-03	3.0659e-01	7.1090e-02	2.9372e-03	9.7009e-02
	0.8	2.8359e-03	2.6543e-03	3.0538e-01	5.5591e-02	2.6831e-03	6.2136e-02
	1.2	2.9618e-03	2.8303e-03	3.2796e-01	5.6520e-02	2.7103e-03	6.2781e-02
	1.6	2.9028e-03	2.7587e-03	3.3622e-01	5.6991e-02	2.9542e-03	6.5266e-02
C3	0.05	2.9651e-03	2.8775e-03	3.0849e-01	5.7181e-02	2.7705e-03	8.5807e-02
	0.1	2.9378e-03	2.8114e-03	2.8900e-01	5.7162e-02	2.8196e-03	5.6843e-02
	0.15	2.9472e-03	3.1533e-03	3.1254e-01	5.7236e-02	2.8602e-03	5.0399e-02
	0.2	3.1144e-03	2.9644e-03	3.0956e-01	5.7108e-02	2.7654e-03	5.7946e-02
C4	0.5	2.8953e-03	2.8522e-03	3.0942e-01	9.9525e-02	2.7533e-03	1.4017e-01
	1	2.7631e-03	3.2853e-03	3.1771e-01	5.6981e-02	2.8591e-03	4.2451e-02
	1.5	2.6377e-03	2.8046e-03	3.0628e-01	5.6507e-02	2.7197e-03	7.0547e-02
	2	2.7952e-03	2.9811e-03	3.0682e-01	5.8362e-02	2.8456e-03	5.2127e-02

Table 8. IGD value data for different parameter settings.

but also has a uniform distribution of solution sets covering the entire range of the Pareto frontier. In contrast, the solution sets of other algorithms either have obvious deficiencies in convergence or are concentrated in a certain localized region, resulting in inadequate coverage of the Pareto front and poor diversity of the solution sets. For example, some algorithms form locally concentrated solution sets in certain parts of the frontier, failing to effectively cover the entire diversity region of the Pareto frontier. Therefore, the performance of TAMOPSO in the UF9 problem is significantly better than other algorithms both in terms of diversity and convergence, indicating that it can better balance the accuracy and diversity of the solution set in high-dimensional multi-objective optimization problems.

Finally, Fig. 8 shows the performance of the 11 algorithms under the complex multi-objective test problem DTLZ6. In this problem, TAMOPSO still performs well and can achieve a good balance of both convergence and diversity. Although CMOPSO, MOPSOCD, NMPSO and TAMOPSO are all able to converge to the true Pareto front, the distribution of the solution sets shows that only CMOPSO and TAMOPSO can distribute the solution sets uniformly over the Pareto front, ensuring better diversity. In contrast, although MOPSOCD and NMPSO also converge to the true frontier, their solution sets are unevenly distributed with obvious concentrated regions, which may lead to over-convergence of locally optimal solutions in some cases, limiting the diversity of solution sets. Therefore, TAMOPSO not only excels in convergence, but also has a significant advantage in maintaining the diversity of solution sets. In summary, TAMOPSO performs well in multi-objective optimization problems with strong convergence and excellent diversity of solution sets and can provide solution sets that are uniform and close to the true Pareto frontier in different test problems, which is significantly better than the other compared algorithms.

Comparison of convergence trajectories

To further validate the performance of the algorithms, we compared the convergence speeds of ten existing algorithms with TAMOPSO on three classical test problems, ZDT3, UF9 and DTLZ6, and made a visual comparison. By observing the convergence trajectories of the algorithms in Fig. 9, it can be clearly seen that TAMOPSO is able to reach the optimal IGD value earlier compared to the other compared algorithms. This phenomenon indicates that TAMOPSO is not only able to effectively guide the convergence of the populations, but also significantly accelerates the convergence process, showing a strong convergence speed advantage. Specifically, TAMOPSO can quickly find solutions closer to the true Pareto frontier in each round of iterations, thus reducing the number of iterations in the optimization process. This accelerated convergence speed is of great significance for solving complex multi-objective optimization problems, as it can provide high-quality solution sets in a shorter period, which greatly improves the computational efficiency of the algorithm.

Friedman rank test

The Friedman rank test is an ideal tool when dealing with the comparison of multiple algorithms with multiple test problems. By statistically analyzing significant differences in algorithm performance, the Friedman rank test can help to more accurately determine the strengths and weaknesses of different algorithms without the stringent data distribution requirements of traditional methods. Its support for multiple comparisons makes it possible to more effectively evaluate the combined performance of algorithms when faced with complex multi-objective optimization problems.

From the IGD values of the TAMOPSO algorithm and the Friedman rank test results of each compared algorithm in Table 6, TAMOPSO consistently ranks first in the rank rankings of the ZDT and UF series of problems, which shows its excellent performance in these problems. Despite its relatively weak performance in the DTLZ series of problems, which leads to its lower rank in this series, overall, TAMOPSO still maintains the first place in the rank ranking, which fully proves its balanced advantage in terms of diversity and convergence. This suggests that TAMOPSO can provide more stable and better solutions when dealing with different types of multi-objective optimization problems, especially in the ZDT and UF series of optimization problems.

The analysis of the Friedman rank test results of TAMOPSO in terms of HV values versus each of the compared algorithms in Table 7 also shows that TAMOPSO once again ranked first in the rankings of the ZDT and UF series, which further validates its combined advantages in terms of convergence and diversity. Despite the dip in rank on the DTLZ problem set, resulting in its third place in overall HV, this result also reveals the potential room for improvement of the TAMOPSO algorithm on the DTLZ test problems. Thus, despite TAMOPSO's slightly inferior performance on some of the test problems, its overall superiority is still evident, especially on the ZDT and UF problems, demonstrating the algorithm's strong potential and scalability in the field of multi-objective optimization.

Sensitivity analysis of experimental parameters

This subsection develops a sensitivity analysis around the algorithm's parameters, aiming to provide insights into the following key elements: assessing how variations in parameter values affect the algorithm's performance in different test problems; determining the optimal or recommended parameter ranges for various scenarios; and demonstrating TAMOPSO's robustness under different configurations.

Six representative test problems, namely ZDT1, ZDT2, UF3, UF4, DTLZ6, and DTLZ7, are selected for the experiment, which covers multi-objective optimization scenarios of different types and difficulties and can comprehensively and efficiently reflect the effects of parameter changes on the performance of the algorithm. In the experiments, the algorithm parameters C1, C2, C3, and C4 are emphasized for testing. Because the inertia weight W in this paper is adaptively transformed by high-quality particles in the test archive, to ensure the accuracy and validity of the experimental results, the inertia weight is not tested in this parameter sensitivity analysis experiment.

The experiments were conducted with different values for each parameter: 0.1, 0.5, 1, and 1.5 for C1; 0.4, 0.8, 1.2, and 1.6 for C2; 0.05, 0.1, 0.15, and 0.2 for C3; and 0.5, 1, 1.5, and 2 for C4. The experimental data are organized as shown in Table 8 to evaluate the performance of the algorithm in different test problems by comparing the IGD values with different parameter settings. Evaluate the impact of parameter changes on the performance of the algorithm in different test problems.

From the data in the table, the different values of each parameter have different impacts on the performance of the algorithm under different test problems. For parameter C1, when $C1=0.5$, the number of preferences reaches half, while the number of preferences for other parameters does not reach half; for parameter C2, only when $C2=0.4$ is preferred in ZDT1, and most of the remaining preferences occur when $C2=0.8$; among the values of C3, the largest number of preferences is found when $C3=0.1$; the experimental parameter preferences of C4 are similar to that of C2, and five out of six test problems are preferred when $C4=1.5$, and only when $C4=1$ is preferred; $C4=1$ is preferred by the algorithms of all the test problems. The experimental parameter preference for C4 was like C2, with 5 out of 6 test problems at $C4=1.5$ and only 1 out of 6 test problems at $C4=1$ in DTLZ7.

It is thus determined that the optimal or recommended parameter ranges under these test problems is $C1=0.5$, $C2=0.8$, $C3=0.1$, and $C4=1.5$. Under this parameter combination, the algorithm has the highest number of optimal numbers in different test problems, which suggests that it has a better robustness under different configurations, which is more conducive to the algorithm to achieve a good performance in the multi-objective optimization scenario. This series of experimental results provides a strong parameter selection basis for the practical application and further optimization of the algorithm and fully demonstrates the robust characteristics of TAMOPSO under different configurations.

Problem	M	D		TAMOPSO-234	TAMOSPO-134	TAMOPSO-124	TAMOPSO-123	TAMOPSO-0
ZDT1	2	30	Mean	2.9371e-03	3.1609e-03	3.7491e-03	2.8817e-03	2.8629e-03
			Std	(1.04e-04)-	(1.32e-04)-	(2.04e-04)-	(1.34e-04) ≈	(1.16e-04)
ZDT2	2	30	Mean	2.9267e-03	2.9730e-03	3.7931e-03	2.8086e-03	2.8956e-03
			Std	(3.97e-05)-	(7.14e-05)-	(3.08e-04)-	(2.11e-04) +	(1.36e-04)
ZDT3	2	30	Mean	3.8426e-03	3.6345e-03	4.5978e-03	3.7203e-03	3.3762e-03
			Std	(6.20e-04)-	(3.90e-04)-	(4.28e-04)-	(2.17e-04)-	(2.16e-04)
ZDT4	2	10	Mean	1.6817e+00	1.1197e+00	5.3567e+00	8.4032e-01	1.3611e+00
			Std	(5.69e-01)-	(3.68e-02) +	(4.34e+00)-	(2.39e-05) +	(5.51e-0)
ZDT6	2	10	Mean	2.2419e-03	2.2630e-03	3.8294e-03	2.2468e-03	2.1733e-03
			Std	(3.43e-05)-	(9.91e-05)-	(4.31e-04)-	(6.03e-05)-	(1.20e-04)
UF1	2	30	Mean	1.6351e-01	1.5757e-01	1.6135e-01	1.5448e-01	1.5294e-01
			Std	(7.30e-03)-	(2.17e-02)-	(3.64e-03)-	(1.12e-02) ≈	(1.12e-02)
UF2	2	30	Mean	9.3067e-02	9.1226e-02	9.4648e-02	9.4693e-02	9.0843e-02
			Std	(3.09e-04)-	(1.67e-03) ≈	(5.77e-03)-	(4.41e-03)-	(4.64e-03)
UF3	2	30	Mean	3.1224e-01	3.1948e-01	3.2547e-01	3.0419e-01	3.0269e-01
			Std	(2.57e-03)-	(8.53e-03)-	(3.67e-02)-	(5.74e-03) ≈	(7.37e-03)
UF4	2	30	Mean	5.8109e-02	7.0447e-02	7.3498e-02	6.1396e-02	5.7955e-02
			Std	(8.72e-04) ≈	(1.02e-02)-	(1.13e-02)-	(3.38e-03)-	(2.60e-03)
UF5	2	30	Mean	1.8899e+00	1.8127e+00	1.8191e+00	1.9590e+00	1.8086e+00
			Std	(1.56e-01)-	(1.69e-01) ≈	(2.01e-01)≈	(1.91e-01)-	(1.42e-01)
UF6	2	30	Mean	7.5577e-01	7.6515e-01	7.9081e-01	7.5963e-01	7.4356e-01
			Std	(2.53e-02)-	(6.55e-02)-	(1.08e-01)-	(5.24e-02)-	(8.77e-02)
UF7	2	30	Mean	1.4125e-01	1.4124e-01	1.4495e-01	1.4345e-01	1.3887e-01
			Std	(9.61e-03)-	(8.52e-03)-	(5.10e-03)-	(1.06e-02)-	(1.20e-02)
UF8	3	30	Mean	2.8796e-01	2.9218e-01	2.9273e-01	2.8850e-01	2.6714e-01
			Std	(1.03e-02)-	(4.72e-02)-	(4.71-02)-	(3.20e-02)-	(4.11e-02)
UF9	3	30	Mean	1.5581e-01	1.8176e-01	1.4843e-01	1.6498e-01	1.4466e-01
			Std	(3.99e-03)-	(1.43e-02)-	(2.07e-03)-	(1.01e-03)-	(1.10e-02)
UF10	3	30	Mean	7.9701e-01	7.6215e-01	7.2674e-01	9.4586e-01	7.2324e-01
			Std	(2.31e-02)-	(6.74e-02)-	(1.96e-03) ≈	(6.07e-05)-	(9.85e-02)
DTLZ1	3	7	Mean	1.8914e+01	1.8097e+01	1.9604e+01	1.9132e+01	1.7377e+01
			Std	(1.29e+00)-	(5.93e+00)-	(1.62e+00)-	(1.96e+00)-	(4.52e+00)
DTLZ2	3	12	Mean	7.6498e-02	6.7355e-02	1.6707e-01	6.3372e-02	7.4888e-02
			Std	(2.94e-04)-	(1.98e-03) +	(1.23e-02)-	(2.42e-03) +	(2.99e-03)
DTLZ3	3	12	Mean	1.4392e+02	1.4846e+02	1.7339e+02	1.5399e+02	1.3561e+02
			Std	(7.73e+00)-	(2.42e+00)-	(8.76e-01)-	(1.28e+01)-	(4.79e+01)
DTLZ4	3	12	Mean	3.3881e-01	2.9318e-01	3.1388e-01	3.0004e-01	2.9281e-01
			Std	(1.35e-02)-	(2.28e-02) ≈	(2.92e-04)-	(8.98e-03)-	(2.88e-02)
DTLZ5	3	12	Mean	2.5106e-02	1.6618e-02	2.3458e-02	1.8531e-02	2.2463e-02
			Std	(3.32e-03)-	(2.75e-03)-	(2.91e-03) ≈	(1.00e-03) +	(1.99e-03)
DTLZ6	3	12	Mean	2.8801e-03	3.1408e-03	6.4282e-02	3.0349e-03	2.8026e-03
			Std	(6.26e-05) ≈	(1.33e-04)-	(2.05e-02)-	(1.83e-04)-	(1.51e-04)
DTLZ7	3	22	Mean	8.5309e-02	4.9453e-02	8.6371e-02	8.1188e-02	7.2873e-02
			Std	(1.69e-02)-	(5.14e-03) +	(1.56e-02)-	(5.03e-03)-	(3.54e-02)
Best/All +/-/≈				0/22 0/20/2	2/22 3/16/3	0/22 0/19/3	3/22 4/15/3	17/22 —

Table 9. Ablation experiment IGD data table.

Ablation study

In this paper, multiple strategies are innovatively combined to achieve the optimization of the MOPSO algorithm. To deeply investigate the actual effectiveness of each strategy in the algorithm, an ablation experiment is designed, which evaluates the impact of these mechanisms on the overall performance of the algorithm by removing specific mechanisms from the algorithm.

Specifically, the TAMOPSO algorithm was set up to compare its performance with four variants of the algorithm: firstly, TAMOPSO with the removal of the individual optimal update strategy (labeled TAMOPSO—234); secondly, TAMOPSO with the removal of the population delimitation and task allocation strategy (labeled TAMOPSO—134); thirdly, TAMOPSO with the removal of the uniform contribution metrics

Problem	M	D		TAMOPSO-234	TAMOSPO-134	TAMOPSO-124	TAMOPSO-123	TAMOPSO-0
ZDT1	2	30	Mean	7.2120e-01	7.2087e-01	7.2005e-01	7.2103e-01	7.2122e-01
			Std	(8.81e-05) ≈	(1.93e-04)-	(1.94e-04)-	(3.23e-04)-	(2.10e-04)
ZDT2	2	30	Mean	4.4558e-01	4.4489e-01	4.4362e-01	4.4584e-01	4.4569e-01
			Std	(2.22e-05) ≈	(5.88e-04)-	(1.80e-03)-	(4.83e-04) +	(3.42e-04)
ZDT3	2	30	Mean	5.9995e-01	5.9951e-01	5.9911e-01	6.0004e-01	6.0024e-01
			Std	(7.93e-05)-	(5.64e-04)-	(7.59e-04)-	(1.40e-03) ≈	(1.06e-03)
ZDT4	2	10	Mean	0.0000e+00	0.0000e+00	0.0000e+00	9.0888e-02	3.5731e-03
			Std	(0.00e+00)-	(0.00e+00)-	(0.00e+00)-	(2.62e-05) +	(1.41e-02)
ZDT6	2	10	Mean	3.8848e-01	3.8849e-01	3.8448e-01	3.8893e-01	3.8927e-01
			Std	(1.27e-04)-	(3.57e-04)-	(1.23e-03)-	(7.83e-04) ≈	(4.40e-04)
UF1	2	30	Mean	4.6843e-01	4.7985e-01	4.7254e-01	4.8049e-01	4.8126e-01
			Std	(1.04e-02)-	(2.73e-02)-	(3.31e-03)-	(2.11e-02)-	(1.63e-02)
UF2	2	30	Mean	6.1231e-01	6.1244e-01	6.1227e-01	6.1198e-01	6.1459e-01
			Std	(1.10e-03)-	(1.85e-03)-	(6.15e-03)-	(4.25e-03)-	(5.48e-03)
UF3	2	30	Mean	3.3423e-01	3.1729e-01	3.2633e-01	3.4004e-01	3.4207e-01
			Std	(1.93e-03)-	(1.41e-02)-	(1.97e-02)-	(3.29e-03)-	(7.31e-03)
UF4	2	30	Mean	3.6009e-01	3.4683e-01	3.5016e-01	3.5553e-01	3.6096e-01
			Std	(1.12e-03)-	(1.35e-02)-	(7.50e-03)-	(4.48e-03)-	(3.63e-03)
UF5	2	30	Mean	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
			Std	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00)
UF6	2	30	Mean	0.0000e+00	0.0000e+00	3.0697e-04	4.6617e-04	2.1788e-03
			Std	(0.00e+00)-	(0.00e+00)-	(5.32e-04)-	(8.07e-04)-	(3.83e-03)
UF7	2	30	Mean	3.7286e-01	3.7511e-01	3.7181e-01	3.6943e-01	3.7550e-01
			Std	(8.85e-03)-	(9.89e-03) ≈	(9.13e-03)-	(1.58e-02)-	(1.65e-02)
UF8	3	30	Mean	3.1627e-01	3.3237e-01	3.2508e-01	3.3277e-01	3.3479e-01
			Std	(5.57e-04)-	(8.22e-03) ≈	(1.91e-02)-	(8.26e-03)-	(1.35e-02)
UF9	3	30	Mean	6.0235e-01	5.6385e-01	6.0218e-01	5.9112e-01	6.1236e-01
			Std	(5.10e-03)-	(1.21e-02)-	(4.79e-03)-	(8.48e-04)-	(1.39e-02)
UF10	3	30	Mean	7.3597e-02	9.1864e-02	4.2648e-02	9.0905e-02	7.1769e-02
			Std	(1.39e-04) +	(1.52e-03) +	(4.25e-02)-	(3.99e-06) +	(2.50e-02)
DTLZ1	3	7	Mean	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
			Std	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00)
DTLZ2	3	12	Mean	4.9738e-01	5.1525e-01	3.8528e-01	5.1870e-01	4.9988e-01
			Std	(2.38e-04)-	(3.14e-03) +	(9.89e-04)-	(6.18e-03) +	(4.20e-03)
DTLZ3	3	12	Mean	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
			Std	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00) ≈	(0.00e+00)
DTLZ4	3	12	Mean	2.8278e-01	3.5615e-01	3.2655e-01	3.4691e-01	3.3446e-01
			Std	(3.36e-02)-	(1.68e-02) +	(4.91e-03)-	(2.52e-02) +	(2.12e-02)
DTLZ5	3	12	Mean	1.7179e-01	1.6244e-01	1.7302e-01	1.8126e-01	1.7683e-01
			Std	(6.34e-03)-	(3.41e-03)-	(4.34e-03)-	(1.07e-03) +	(4.14e-03)
DTLZ6	3	12	Mean	2.0083e-01	2.0046e-01	1.5233e-01	1.9999e-01	2.0033e-01
			Std	(1.96e-05) +	(2.31e-04) +	(1.34e-02)-	(8.34e-04)-	(9.41e-04)
DTLZ7	3	22	Mean	1.5021e-01	1.5077e-01	1.4813e-01	1.4403e-01	1.5256e-01
			Std	(1.17e-03)-	(7.79e-04)-	(3.58e-03)-	(5.97e-03)-	(9.50e-03)
Best/All +/-/≈				1/22 5/13/4	2/22 8/10/4	0/22 0/19/3	4/22 8/10/4	12/22 —

Table 10. Ablation experiment HV data table.

guiding the archival maintenance strategy (labeled TAMOPSO—124); and the fourth is TAMOPSO (labeled TAMOPSO—123), which removes the Lévy flight variation strategy for adaptive control. During the experiments, the parameter settings of these four variants of the algorithm were kept identical to those of TAMOPSO to ensure a fair and accurate comparison of the algorithms' performance.

Subsequently, the four variants of the algorithms were tested on problems in the ZDT, UF and DTLZ test suites, respectively. Each test problem was run independently for 30 times, and the results of each run were recorded and statistically analyzed for the IGD and HV metrics to obtain the mean and standard deviation of these two metrics, and the comparison results are presented in Tables 9 and 10. Meanwhile, Wilcoxon rank sum

test is applied to detect the significance of the performance difference between TAMOPSO and the four variants of the algorithm.

From the experimental results, in the IGD data table in Table 9, the four variants of the algorithm are 0, 2, 0, and 3, while the original TAMOPSO algorithm has as many as 17; similarly, in Table 10, the four variants of the algorithm are 1, 2, 0, and 4, while the original TAMOPSO algorithm has 12. These data fully confirm that the organic combination of the four strategies significantly improves the comprehensive performance of TAMOPSO and plays a crucial role in the algorithm.

Conclusion

In this paper, a novel multi-objective optimization algorithm is proposed to overcome the problems of premature convergence, lack of diversity, and difficult coordination of inter-objective conflicts in traditional algorithms by introducing multiple mechanisms. First, for the uniform distribution of the population, the algorithm assigns proportionate weights to the convergence and diversity of each particle and optimizes the evolutionary process by using a sub-population strategy. Second, it is difficult to balance the target conflicts with a single search strategy. A new speed control mechanism is designed to adjust the search strategy according to the subpopulation characteristics, which improves the local search efficiency and balances the relationship between exploration and exploitation. In terms of mutation operation, an adaptive Lévy flight mutation strategy is proposed to balance the global and local search by detecting the growth rate and local mutation of the archive, which effectively reduces the difficulty of parameter adjustment and improves the mutation rate. To better approximate the Pareto frontier, this paper also innovatively introduces the concept of archive particle uniformity contribution factor to optimize the archive quality by deleting low contribution particles. Finally, to address the problems of unfair selection and dimension pressure of individual optimal updates in traditional algorithms, this paper selects optimal particles by Pareto comparison and L_2 paradigm, which significantly improves the fairness of selection and reduces the dimension pressure. These innovative mechanisms enable the algorithm to show significant advantages in terms of convergence speed, solution set diversity and global optimization capability. Through in-depth comparative analysis of experimental results, the TAMOPSO algorithm was comprehensively compared with five existing MOPSO algorithms and five MOEA algorithms from multiple dimensions such as convergence, diversity, and stability. Its performance on 22 standard benchmark problems was evaluated. The experimental results show that TAMOPSO demonstrates significant advantages in terms of convergence, diversity, and stability in these benchmark problems, outperforming the other comparison algorithms.

Data availability

The datasets used or analyzed during the current study are available, and the following data were downloaded: <https://github.com/coldwaterforever/TAMOPSO.git>.

Received: 20 January 2025; Accepted: 22 April 2025

Published online: 06 May 2025

References

1. Zhang, B. et al. An automatic multi-objective evolutionary algorithm for the hybrid flowshop scheduling problem with consistent sublots. *Knowl.-Based Syst.* **238**, 107819. <https://doi.org/10.1016/j.knosys.2021.107819> (2022).
2. Yuan, M. et al. Coordinated allocation of water resources and wastewater emission permits based on multi-objective optimization model: From the perspective of conflict between equity and economic benefits. *J. Clean. Prod.* **372**, 133733. <https://doi.org/10.1016/j.jclepro.2022.133733> (2022).
3. Rodríguez-Molina, A. et al. Adaptive controller tuning method based on online multi-objective optimization: A case study of the four-bar mechanism. *IEEE Trans. Cybern.* **51**(3), 1272–1285. <https://doi.org/10.1109/TCYB.2019.2903491> (2019).
4. Palakonda, V., Mallipeddi, R. & Suganthan, P. N. An ensemble approach with external archive for multi- and many-objective optimization with adaptive mating mechanism and two-level environmental selection. *Inf. Sci.* **555**, 164–197. <https://doi.org/10.1016/j.ins.2020.11.040> (2021).
5. Li, K. & Chen, R. Batched data-driven evolutionary multi-objective optimization based on manifold interpolation. *IEEE Trans. Evol. Comput.* **27**(1), 126–140. <https://doi.org/10.1109/TEVC.2022.3162993> (2022).
6. Ma, L. et al. An adaptive localized decision variable analysis approach to large-scale multi-objective and many-objective optimization. *IEEE Trans. Cybern.* **52**(7), 6684–6696. <https://doi.org/10.1109/TCYB.2020.3041212> (2021).
7. Yang, J. et al. A particle swarm algorithm based on the dual search strategy for dynamic multi-objective optimization. *Swarm Evol. Comput.* **83**, 101385. <https://doi.org/10.1016/j.swevo.2023.101385> (2023).
8. Zheng, J. et al. A dynamic multi-objective particle swarm optimization algorithm based on adversarial decomposition and neighborhood evolution. *Swarm Evol. Comput.* **69**, 100987. <https://doi.org/10.1016/j.swevo.2021.100987> (2022).
9. Deb, K. & Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**(4), 577–601. <https://doi.org/10.1109/TEVC.2013.2281535> (2013).
10. Wang, S. et al. Cooperative task allocation for multi-robot systems based on multi-objective ant colony system. *IEEE Access* **10**, 56375–56387. <https://doi.org/10.1109/ACCESS.2022.3165198> (2022).
11. Zheng, R. et al. A multi-strategy enhanced African vultures optimization algorithm for global optimization problems. *J. Comput. Des. Eng.* **10**(1), 329–356. <https://doi.org/10.1093/jcde/qwac135> (2023).
12. Duman, S., Akbel, M. & Kahraman, H. T. Development of the multi-objective adaptive guided differential evolution and optimization of the MO-ACOPF for wind/PV/tidal energy sources. *Appl. Soft Comput.* **112**, 107814. <https://doi.org/10.1016/j.asoc.2021.107814> (2021).
13. Song, Y. et al. A dual-time dual-population multi-objective evolutionary algorithm with application to the portfolio optimization problem. *Eng. Appl. Artif. Intell.* **133**, 108638. <https://doi.org/10.1016/j.engappai.2024.108638> (2024).
14. Zhang, Y. et al. A multi-objective particle swarm optimization based on local ideal points. *Appl. Soft Comput.* **161**, 111707. <https://doi.org/10.1016/j.asoc.2024.111707> (2024).
15. Tang, Y. et al. Multi-subswarm cooperative particle swarm optimization algorithm and its application. *Inf. Sci.* <https://doi.org/10.1016/j.ins.2024.120887> (2024).

16. Xu, Q. et al. Two-stage particle swarm optimization with dual-indicator fusion ranking for multi-objective problems. *Inf. Sci.* <https://doi.org/10.1016/j.ins.2024.121032> (2024).
17. Coello, C. A. C., & Lechuga, M. S. MOPSO: A proposal for multiple objective particle swarm optimization[C]//Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). IEEE, 2002, 2: 1051–1056. <https://doi.org/10.1109/CEC.2002.1004388>
18. Xu, X. et al. Accelerated two-stage particle swarm optimization for clustering not-well-separated data. *IEEE Trans. Syst. Man Cybern. Syst.* **50**(11), 4212–4223. <https://doi.org/10.1109/TSMC.2018.2839618> (2018).
19. Fu, Q. et al. MOFS-REPLS: A large-scale multi-objective feature selection algorithm based on real-valued encoding and preference leadership strategy. *Inf. Sci.* **667**, 120483. <https://doi.org/10.1016/j.ins.2024.120483> (2024).
20. Chen, Y. et al. LDANet: Automatic lung parenchyma segmentation from CT images. *Comput. Biol. Med.* **155**, 106659. <https://doi.org/10.1016/j.combiomed.2023.106659> (2023).
21. Liu, W. et al. Pair-based uncertainty and diversity promoting early active learning for person re-identification. *ACM Trans. Intell. Syst. Technol. (TIST)* **11**(2), 1–15. <https://doi.org/10.1145/3372121> (2020).
22. Zhang, Y. et al. MOCPSO: A multi-objective cooperative particle swarm optimization algorithm with dual search strategies. *Neurocomputing* **562**, 126892. <https://doi.org/10.1016/j.neucom.2023.126892> (2023).
23. Li, M., Yang, S. & Liu, X. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Trans. Evol. Comput.* **18**(3), 348–365. <https://doi.org/10.1109/TEVC.2013.2262178> (2013).
24. Huang, W. & Zhang, W. Adaptive multi-objective particle swarm optimization with multi-strategy based on energy conversion and explosive mutation. *Appl. Soft Comput.* **113**, 107937. <https://doi.org/10.1016/j.asoc.2021.107937> (2021).
25. Cheng, R. et al. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **20**(5), 773–791. <https://doi.org/10.1109/TEVC.2016.2519378> (2016).
26. Liang, Z. et al. Multi-objective multi-task particle swarm optimization based on objective space division and adaptive transfer. *Expert Syst. Appl.* **255**, 124618. <https://doi.org/10.1016/j.eswa.2024.124618> (2024).
27. Tian, Z. & Gai, M. Football team training algorithm: A novel sport-inspired meta-heuristic optimization algorithm for global optimization. *Expert Syst. Appl.* **245**, 123088. <https://doi.org/10.1016/j.eswa.2023.123088> (2024).
28. Wang, C. et al. A dual-population based evolutionary algorithm for multi-objective location problem under uncertainty of facilities. *IEEE Trans. Intell. Transp. Syst.* **23**(7), 7692–7707. <https://doi.org/10.1109/TITS.2021.3071786> (2021).
29. Moazen, H. et al. PSO-ELPM: PSO with elite learning, enhanced parameter updating, and exponential mutation operator. *Inf. Sci.* **628**, 70–91. <https://doi.org/10.1016/j.ins.2023.01.103> (2023).
30. Wang, M. et al. A dynamic multi-objective optimization algorithm with a dual mechanism based on prediction and archive. *Swarm Evol. Comput.* **90**, 101693. <https://doi.org/10.1016/j.swevo.2024.101693> (2024).
31. Cheng, J. et al. Differential evolution algorithm with fitness and diversity ranking-based mutation operator. *Swarm Evol. Comput.* **61**, 100816. <https://doi.org/10.1016/j.swevo.2020.100816> (2021).
32. Salgotra, R. & Singh, U. Application of mutation operators to flower pollination algorithm. *Expert Syst. Appl.* **79**, 112–129. <https://doi.org/10.1016/j.eswa.2017.02.035> (2017).
33. García, I. C., Coello, C. A. C., Arias-Montano, A. MOPSOhv: A new hypervolume-based multi-objective particle swarm optimizer[C]//2014 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2014: 266–273. <https://doi.org/10.1109/CEC.2014.6900540>
34. Luo, J. et al. A many-objective particle swarm optimizer based on indicator and direction vectors for many-objective optimization. *Inf. Sci.* **514**, 166–202. <https://doi.org/10.1016/j.ins.2019.11.047> (2020).
35. Zhang, Y. et al. Adaptive projection plane and reference point strategy for multi-objective particle swarm optimization. *Alex. Eng. J.* **105**, 381–401. <https://doi.org/10.1016/j.aej.2024.07.006> (2024).
36. Yan, C. et al. Self-weighted robust LDA for multiclass classification with edge classes. *ACM Trans. Intell. Syst. Technol. (TIST)* **12**(1), 1–19. <https://doi.org/10.1145/3418284> (2020).
37. Kennedy J, Eberhart R. Particle swarm optimization[C]//Proceedings of ICNN'95-international conference on neural networks. IEEE, 1995, 4: 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
38. Shami, T. M. et al. Velocity pausing particle swarm optimization: A novel variant for global optimization. *Neural Comput. Appl.* **35**(12), 9193–9223. <https://doi.org/10.1007/s00521-022-08179-0> (2023).
39. Zitzler, E., Deb, K. & Thiele, L. Comparison of multi-objective evolutionary algorithms: Empirical results. *Evol. Comput.* **8**(2), 173–195. <https://doi.org/10.1162/106365600568202> (2000).
40. Zhang Q, Zhou A, Zhao S, et al. Multi-objective optimization test instances for the CEC 2009 special session and competition. 2008.
41. Deb K, Thiele L, Laumanns M, et al. Scalable test problems for evolutionary multi-objective optimization[M]//Evolutionary multi-objective optimization: theoretical advances and applications. London: Springer London, 2005: 105–145.
42. Zhang, X. et al. A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf. Sci.* **427**, 63–76. <https://doi.org/10.1016/j.ins.2017.10.037> (2018).
43. Lin, Q. et al. Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **22**(1), 32–46. <https://doi.org/10.1109/TEVC.2016.2631279> (2016).
44. Dai, C., Wang, Y. & Ye, M. A new multi-objective particle swarm optimization algorithm based on decomposition. *Inf. Sci.* **325**, 541–557. <https://doi.org/10.1016/j.ins.2015.07.018> (2015).
45. Raquel C R, Naval Jr P C. An effective use of crowding distance in multi-objective particle swarm optimization[C]//Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation. 2005: 257–264. <https://doi.org/10.1145/1068009.1068047>
46. He, C., Cheng, R. & Yazdani, D. Adaptive offspring generation for evolutionary large-scale multi-objective optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **52**(2), 786–798. <https://doi.org/10.1109/TSMC.2020.3003926> (2020).
47. Tian, Y. et al. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Trans. Evol. Comput.* **23**(2), 331–345. <https://doi.org/10.1109/TEVC.2018.2866854> (2018).
48. Jiang, S. & Yang, S. A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization. *IEEE Trans. Evol. Comput.* **21**(3), 329–346. <https://doi.org/10.1109/TEVC.2016.2592479> (2017).
49. Li, K. et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput.* **19**(5), 694–716. <https://doi.org/10.1109/TEVC.2014.2373386> (2014).
50. Asafuddoula, M., Ray, T. & Sarker, R. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Trans. Evol. Comput.* **19**(3), 445–460. <https://doi.org/10.1109/TEVC.2014.2339823> (2014).
51. Tian, Y. et al. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput. Intell. Mag.* **12**(4), 73–87. <https://doi.org/10.1109/MCI.2017.2742868> (2017).
52. Zhou, A., Jin, Y., Zhang, Q. et al. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion[C]//2006 IEEE international conference on evolutionary computation. IEEE, 2006: 892–899. <https://doi.org/10.1109/CEC.2006.1688406>
53. While, L. et al. A faster algorithm for calculating hypervolume. *IEEE Trans. Evol. Comput.* **10**(1), 29–38. <https://doi.org/10.1109/TEVC.2005.851275> (2006).
54. Zhou, Y., Chen, Z., Huang, Z. & Xiang, Y. A multiobjective evolutionary algorithm based on objective-space localization selection. *IEEE Trans. Cybern.* **52**(5), 3888–3901. <https://doi.org/10.1109/TCYB.2020.3016426> (2020).

Acknowledgements

This paper was partially supported by the Key Laboratory of Evolutionary Artificial Intelligence of Guizhou Province (Qianjiaoji [2022] 059), the Key Talent Program of Digital Economy of Guizhou Province, the National Natural Science Foundation of China (NSFC 62372495), and Guizhou Provincial Key Laboratory Program – Zunyi Jiang-Flavor Chinese Spirits Molecular Research Key Laboratory (ZZSG[2024]014), the Guizhou Provincial Department of Education Colleges and Universities Science and Technology Innovation Team (QJJ[2023]084).

Author contributions

J.C.: methodology, software, investigation, validation, writing—original manuscript. Y. L.: project conceptualization, funding acquisition, methodology, project management, writing, review. Y.L.: data management. A.O.: validation. J.Y.: writing review. W.B.: editing.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025