



Multi-objective Particle Swarm Optimization with Integrated Fireworks Algorithm and Size Double Archiving

Yansong Zhang¹ · Yanmin Liu² · Xiaoyan Zhang¹ · Qian Song³ · Aijia Ouyang² · Jie Yang²

Received: 4 September 2024 / Accepted: 5 December 2024
© The Author(s) 2025

Abstract

The multi-objective particle swarm optimization (MOPSO) is an optimization technique that mimics the foraging behavior of birds to solve difficult optimization problems. MOPSO is well known for its strong global search capability, which efficiently locates solutions that are close to the global optimum across a wide search domain. However, similar to many other optimization algorithms, the fast convergence property of MOPSO can occasionally lead to the population entering the local optimum too soon, obstructing researchers from investigating more efficient solutions. To address this challenge, the study proposes a novel framework that integrates the fireworks algorithm (FA) into MOPSO and establishes a size-double archiving mechanism to maintain population diversity. By preventing population homogenization, this mechanism promotes the retention of better solutions. Additionally, by fusing evolutionary data analysis with particle information, the study offers new individual optimal choices and adaptive parameter tuning to improve the algorithm's robustness and adaptability and better manage the complexity of multi-objective optimization problems (MOPs). The suggested algorithm is compared with several existing MOPSOs and multi-objective evolutionary algorithms (MOEAs) in simulation experiments. Standard test problems like ZDT, UF, and DTLZ are used in the experiments. The new algorithm performs exceptionally well in terms of improving convergence and population diversity, as well as demonstrating significant competitiveness for solving MOPs.

Keywords Multi-objective particle swarm optimization · Fireworks algorithm · Double archiving · Adaptive parameters

1 Introduction

MOPs represent a distinct category of optimization problems [1–3], characterized by the simultaneous consideration of multiple objective functions during the optimization process. These functions often exhibit conflicting relationships, whereby the pursuit of one objective may result in a trade-off with others [4–6]. The focus of multi-objective optimization is not on identifying a single best solution; its real challenge lies in identifying a set of equivalent solutions, which are referred to as a pareto-optimal solution set. They achieve a state of equilibrium between multiple objectives such that no

single objective can be further optimized without sacrificing the others.

Because traditional mathematical planning techniques are best suited for tackling single-objective problems (SOP) and are difficult to apply directly to scenarios with several conflicting objectives, they have faced significant difficulties when dealing with MOPs. On the other hand, evolutionary algorithms that rely on population and meta-heuristic properties, like particle swarm optimization (PSO) [7], genetic algorithms [8], ant colony algorithms [9], and differential evolution [10], have proven to be highly effective in resolving MOPs. Other optimization problems like graphical character creation [11], beneficiation operation metrics optimization [12], evolutionary game optimal outcomes [13], inventory control [14], and optimal control of wastewater treatment processes [15] have all been solved by meta-inspired optimization algorithms. Among them, the PSO has shown excellent performance in SOP due to its simplicity, fast convergence, and small number of parameters [16]. In view of this, some scholars have begun to explore the extension of the PSO algorithm to the field of multi-objective

✉ Yanmin Liu
yanmin7813@163.com

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

² Zunyi Normal College, Zunyi 563002, China

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

optimization, with a view to utilizing its efficient characteristics to cope with more complex MOPs.

Eberhart and Kennedy proposed PSO in 1995 [17]. It is a population-based random search evolutionary method that is extensively utilized in SOP. The PSO method cannot be utilized directly for MOPs optimization because it lacks the mechanisms to preserve the Pareto solution set and the strategy to choose individual optimal solutions (pbest) and global optimal solutions (gbest) from non-dominant alternatives. Two obstacles must be overcome by researchers to apply PSO to MOPs. Finding a balance between preserving Pareto Frontiers' diversity and convergence is the first task [PF]. This depends on the characteristics and quality of the external elite archive according to the MOPSO algorithm [18]. The diversity of the solution set and the algorithm's performance might be impacted by either too quick or too slow convergence in the archive. As a result, it is critical to keep MOPSO's archive diversity and convergence in check. Creating appropriate ways to choose the pbest and gbest for every particle is the second challenge. One of the forerunners in the field of multi-objective optimization is Coello Coello, who first used PSO to solve MOPs [19], paving the way for further investigation. In an effort to meet the demands of various scenarios and enhance performance when working with complex MOPs, researchers have proposed several versions of MOPSO algorithms as the field of MOEAs continues to grow and deepen.

The elite mechanism is important in MOPSO. The algorithm is able to approach the ideal PF by gradually improving the convergence performance by preserving the non-dominated solutions in an external archive. However, during the optimization process, the number of non-dominated solutions may increase quickly [20], which presents a problem for archive maintenance. The scholars have suggested a number of density-based strategies for updating the archive to efficiently manage the non-dominated solutions in it. Among these, the crowding distance method [21], which estimates the degree of crowding among a given solution's closest neighbors in the target space, directs the updating of archives. To preserve the diversity of solutions, non-dominated solutions with high crowding distance values are preferentially kept during the archive truncation process in related studies. An additional tactic is the adaptive grid [22–24], which simplifies the archive by segmenting the target space into several grid cells and eliminating solutions found in areas with a higher concentration of particles according to the grid system. A different approach was taken by Cui et al. [25], who divided the archive into two sections: convergence and diversity. These sections are used to preserve solutions with various attributes. Significant results have been obtained and this strategy partially addresses the issue that a single archive would not be adequate to continuously preserve solutions with convergence and variety. A

technique for maintaining archives that was created recently is the reference vector method. To maintain diversified archives, a set of predefined vectors is employed as a reference. The technique is extensively employed in IDMOPSO [27] and MaOPSO-vPF [26]. This method's disadvantage is that the maintenance effect is dependent on the MOP's PF's geometry [28]. Metrics-based approaches are frequently used in MOPSO archives maintenance. Higher contribution values are given priority to be kept in the archive, and all non-dominated solutions in the archive are graded based on how much they contribute to the metrics. This method has been used by MOPSO_{hv} [30] and MaOPSO-CA [29] to manage their archives. But the success of this strategy largely depends on the measures selected; if the assessment metrics are not sound, the algorithm's performance could suffer greatly.

In terms of leader selection, Li et al. [31] constructed a virtual PF (vPF) based on the geometry of the archive and then estimated the vGD contribution value of the solution to the vPF to rank the solutions in the archive for leader selection. The solutions in the top 10% after sorting are thought to be potential candidates for gbest. This method guarantees that the chosen gbest can precisely assess each solution's contribution to the constituent PF, thereby guiding the algorithm to converge to the PF. However, Wu and Han [20, 32] et al. presented an evolutionary state estimation mechanism in adaptive MOPSO that chooses the diversity global best (d-gbest) and convergence global best (c-gbest) dynamically. This strategy aims to balance the exploration and exploitation capabilities of the algorithm, i.e., to increase the diversity of solutions in the exploration state and promote the convergence of solutions in the exploitation state. However, the pbest selection strategy is often underemphasized in most MOPSO algorithms. For example, in algorithms such as MOEA-DCOPSO [33] and NMPSO [34], pbest selection usually relies solely on the dominance relationship between new solutions and historical pbest. This simplified treatment may neglect the potential value of pbest in guiding the particle search process, thus affecting the search efficiency of the algorithm and the quality of the final solution.

As well, there are several key points to keep in mind when designing an effective MOPSO. First, there is a need to balance global and local search capabilities, i.e., to switch appropriately between exploration and exploitation states [35]. Second, since the parameters of traditional MOPSO are state-independent [36–38], the traditional parameters cannot satisfy the requirement of MOPSO to evolve states. In addition, due to the fast convergence speed of MOPSO, it is easy to fall into local optimality and premature convergence [39]. During the development process, if the number of gbest is small, it may cause the particle population to fall into the local optimal region instead of continuing to explore other potentially more

optimal regions. Therefore, rational selection of gbest in the archive in the context of MOPSO is another difficulty in MOPSO design [40, 41].

The researchers' deep thinking and inventive spirit in MOPSO research are evident in these research advances. We think that MOPSO will perform better when handling MOPs with ongoing research and development. This paper proposes an improved MOPSO, called FAMOPSO, to further improve the convergence and diversity of the MOPSO algorithm in the multi-objective optimization process. To overcome the limitations that the traditional MOPSO may encounter when handling complex problems, this study employs a number of novel strategies.

When the number of gbest is small, FA [42] is introduced to generate the best leader (fbest) to guide the population update together with gbest. The algorithm has an explosion mechanism and a diverse search strategy, which can improve diversity and effectively avoid local optimization.

This paper presents an innovative double-size archiving mechanism. Larger archives are designed to store more non-control solutions and effectively maintain archive diversity by diversifying dense areas of particles into scarce areas. While smaller archives focus on preserving elite solutions to improve convergence efficiency. To further optimize archive management, we have introduced adaptive grid techniques, shift-based density estimation (SDE) [43], and crowded distance (CD) [44] strategies that aim to increase archiving diversity and convergence speed.

Finally, this paper deeply analyzes the limitations of the traditional pbest method in the evolutionary process and innovatively introduces a particle-Active mechanism based on information fusion to realize information sharing and communication among particles. Using evolutionary information to dynamically adjust the updating parameters, we improve the adaptability and robustness of the algorithm.

In this study, we provide the reader with a comprehensive understanding of the FAMOPSO algorithm. The article is structured as follows: Sect. 2 introduces the background knowledge of PSO and related work to lay the foundation for the subsequent discussion. Section 3 describes the FAMOPSO algorithm in detail, including the fireworks algorithm to guide the population update, the dual archiving mechanism maintenance strategy, and the role of Active particles. Section 4 verifies the performance of FAMOPSO through experimental results, demonstrating its effectiveness and advantages in MOPs. Finally, Sect. 5 discusses the missing problems of FAMOPSO in terms of applications, limitations of FAMOPSO at this stage, research contributions, and future research directions. By arranging the content in this way, we hope to present the research work on the FAMOPSO algorithm in a systematic way.

2 Related Work

2.1 Introduction to MOPs

In practical application, many optimization problems involve many contradictory objectives that need to be considered and optimized at the same time. This kind of problem is called MOPs. Generally, contradictory MOPs can be mathematically expressed as:

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

$$\text{s.t. } g_i(x) \geq 0, i = 1, 2, \dots, q \quad (2)$$

$$h_j(x) = 0, j = 1, 2, \dots, r$$

where $x = (x_1, x_2, \dots, x_d)$ is a d-dimensional decision vector in the decision space R^d , denoted as: $x \in R^d$. $F(x)$ represents an m-dimensional objective vector, where $f_i(x)$ represents the objective function in the i th dimension, $g_i(x) \geq 0$ is an inequality constraint, and $h_j(x) = 0$ is an equality constraint.

A simultaneous optimal solution cannot be found in MOPs due to the conflicting aims. Finding a set of Pareto optimum sets, or non-dominated solutions, that form PF in the target space and illustrate the trade-offs between various goals is therefore crucial. These solutions give decision-makers a range of options from which to select the best plan based on their tastes and circumstances. To guarantee convergence to the true PF, discover and preserve the variety of non-dominated solutions and offer guidance for the decision-making process, MOEAs are designed.

2.2 Reasons Why PSO is Suitable for MOPs and Its Advantages

PSO is considered more suitable in solving MOPs, mainly due to the following advantages:

1. **Global search capability:** PSO is able to perform global search effectively by simulating the foraging behavior of groups of organisms in nature. Each particle adjusts its moving direction and speed according to the individual historical optimal position and the global optimal position, thus gradually approaching the global optimal solution. This method based on group intelligence makes PSO have stronger exploration ability and wider search space when dealing with MOPs.
2. **Diversity maintenance mechanism:** to maintain the diversity of the population, PSO employs a variety of strategies, such as maintaining the Pareto front and selecting multiple optimal particles. These strategies

help to avoid the algorithm from falling into local optima and ensure that a widely distributed set of Pareto-optimal solutions are found in multi-objective optimization problems.

3. Convergence improvement means: PSO also employs a variety of convergence improvement means, such as adaptive lattice variants, mutation operators, and so on. These means help to enhance the exploration ability of the algorithm, carry out sufficient exploration in the early stage, and gradually converge to the vicinity of the optimal solution in the later stage.
4. Wide range of applications: PSO algorithms have been widely used in a variety of real-world problems, including engineering design, function optimization, neural network training, path planning, and so on. In multi-objective optimization problems, PSO also shows strong application potential and wide applicability.

In addition to PSO compared with other algorithms such as DE and ACO, the specific advantages of PSO in multi-objective optimization problems are mainly reflected in the following aspects:

1. Simple and easy to implement: the principle of PSO is simple, easy to implement, and easy to understand. In contrast, although DE also has strong global search capability, its parameter setting and variation strategy are relatively complex, while ACO focuses more on simulating the positive feedback mechanism in ant foraging behavior and is relatively weak in the direct handling of multi-objective problems.
2. Efficient global search: PSO is able to quickly perform global search in the solution space through information sharing and collaboration among particles. In contrast, although DE also has some global search capability, it may be less efficient in dealing with complex multi-objective problems; ACO focuses more on local search and path optimization.
3. Flexible fitness calculation: PSO can flexibly define the fitness function according to the specific problem and deal with the trade-off between multiple objectives through the Pareto domination mechanism. In contrast, DE and ACO algorithms may be relatively fixed or not flexible enough in terms of fitness calculation and objective trade-offs.

In summary, the PSO algorithm has the advantages of strong global search capability, effective Pareto domination mechanism, perfect diversity preservation mechanism, various means of convergence improvement, and wide application areas in solving multi-objective problems. Compared with other algorithms such as DE and ACO, PSO shows higher efficiency and wider applicability in MOPs.

2.3 Introduction and Basic Definition of PSO

PSO is a traditional swarm intelligence algorithm that mimics natural social behavior to solve optimization problems. It was inspired by the foraging habits of birds. It is well-known for its quick convergence and straightforward implementation. In the particle swarm algorithm, every particle must communicate with pbest and gbest. By working together and continuously adjusting their own position and speed, the particles are able to discover new solutions within the search space. The traditional PSO is expanded to handle multiple conflicting targets against the backdrop of MOPs. Thus, rather than searching for a single worldwide optimal solution, our goal is to find a collection of non-dominated solutions. Therefore, MOPSO needs a special mechanism to maintain the diversity of solutions and ensure that the algorithm can effectively explore and converge to the real Pareto frontier. The $(t + 1)$ generation particle velocity and position update formula is:

$$x_i(t) = [x_{i,1}(t), x_{i,2}(t), \dots, x_{i,d}(t)] \quad (3)$$

$$v_i(t) = [v_{i,1}(t), v_{i,2}(t), \dots, v_{i,d}(t)] \quad (4)$$

$$V_i(t + 1) = \omega V_i(t) + c_1 r_1 (pbest_i(t) - X_i(t)) + c_2 r_2 (gbest_i(t) - X_i(t)) \quad (5)$$

$$X_i(t + 1) = X_i(t) + V_i(t + 1) \quad (6)$$

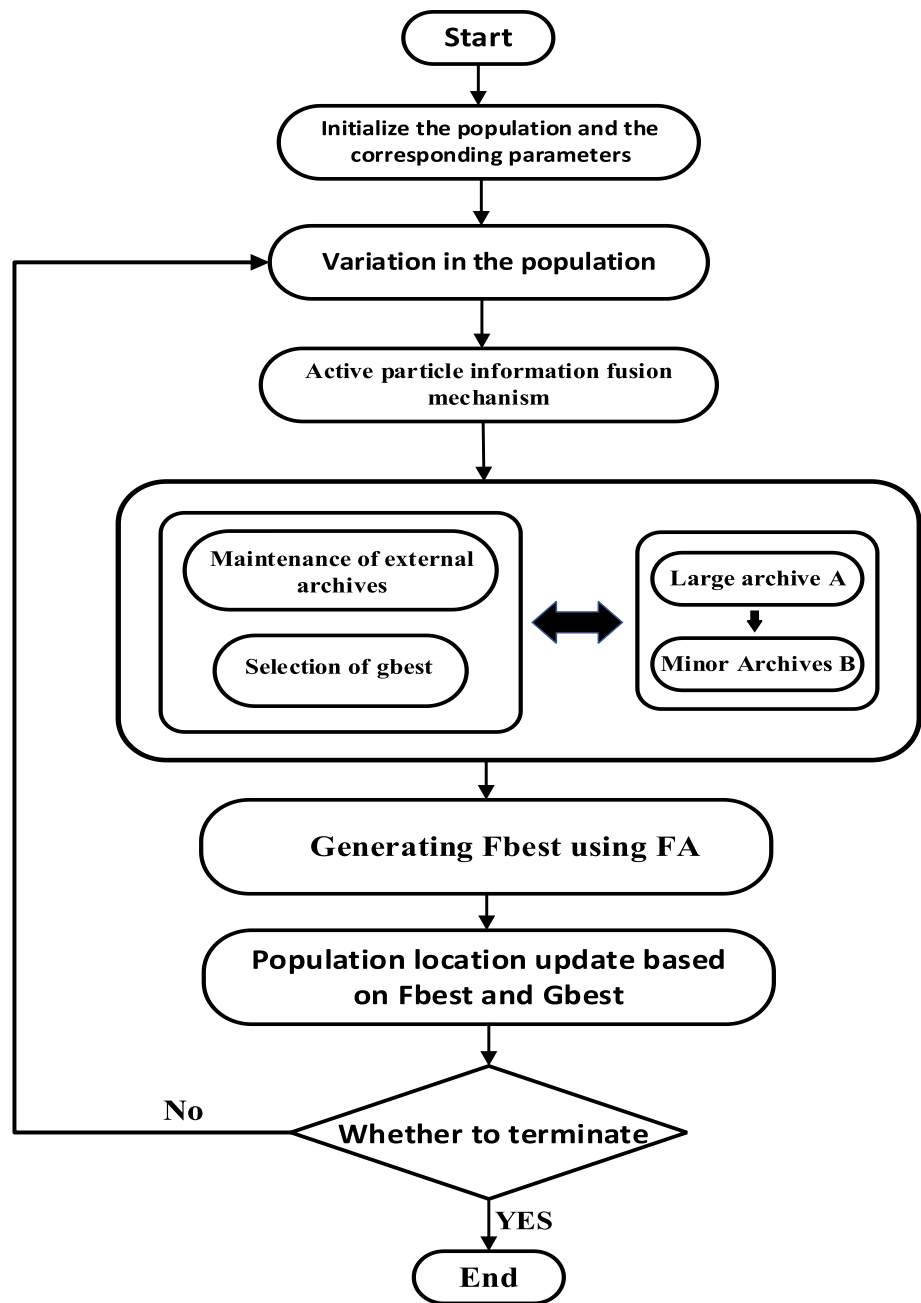
where t denotes the number of iterations, d denotes the dimension of the decision space. $x_i(t)$ and $v_i(t)$ denote the position and velocity of the i th particle, respectively. $i = 1, 2, \dots, Pmax$, $Pmax$ denotes the number of population particles, ω is the inertia weight, c_1 and c_2 are the learning factors, and r_1 and r_2 are two random vectors generated uniformly in $[0, 1]$.

3 Algorithm Design of the Proposed FAMOPSO

This study proposes an improved algorithm called FAMOPSO in an effort to further improve the performance of MOPSO. The fundamental architecture of the FAMOPSO algorithm is displayed in Fig. 1. FAMOPSO has several important improvements and innovations over the standard MOPSO.

First, we introduce FA to generate a new fbest and make it work together with the original gbest to guide population renewal. This novel updating mechanism helps the particles get rid of the local optimum and encourages them to

Fig. 1 Framework diagram of the proposed FAMOPSO



explore a wider search area, thus effectively avoiding the algorithm from falling into the local optimum prematurely.

Second, this paper proposes a double-archiving strategy to balance the algorithm's convergence speed and population diversity. The larger archive is in charge of keeping a lot of non-dominant solutions and preserving population diversity by mutating particles from dense to sparse areas. On the other hand, smaller archives concentrate on keeping the best solution that evolves well to speed up the algorithm's convergence. To facilitate the efficient management of these two archives, we present adaptive grid technology along with SDE and CD selection strategies. The diversity of archiving can be increased using adaptive grid technology, which can

dynamically modify the grid's size and shape to accommodate the distribution of solutions. Simultaneously, SDE and CD strategies determine the density of the solution based on its distribution and make appropriate decisions to expedite the algorithm's convergence.

Finally, in the aspect of particle renewal, based on the idea of multi-particle information fusion, we redesigned the individual optimal selection mechanism and realized the sharing and exchange of information between particles. In this way, particles can understand the search space more comprehensively and make more reasonable updating decisions. At the same time, we use the evolution information of particles to customize the update parameters of each particle,

and these parameters will be dynamically adjusted according to the performance of particles, which enhances the exploration ability of the algorithm.

3.1 Fireworks Algorithm Produces New Leader (fbest)

In traditional MOPSO, the particle updating process is usually guided by the gbest generated by the population. However, only a small number of gbest are usually generated in the early stages of the algorithm's operation. This phenomenon may lead the population to over-rely on these limited resources during the updating process, which increases the risk of falling into local optimal solutions. In addition, our observation also reveals that when dealing with some specific test problems, the population consistently produces only a small number of gbest from the early stage of iteration to the end of the algorithm. This situation further limits the population's ability to explore the solution space, thus making it difficult to localize to the real PF.

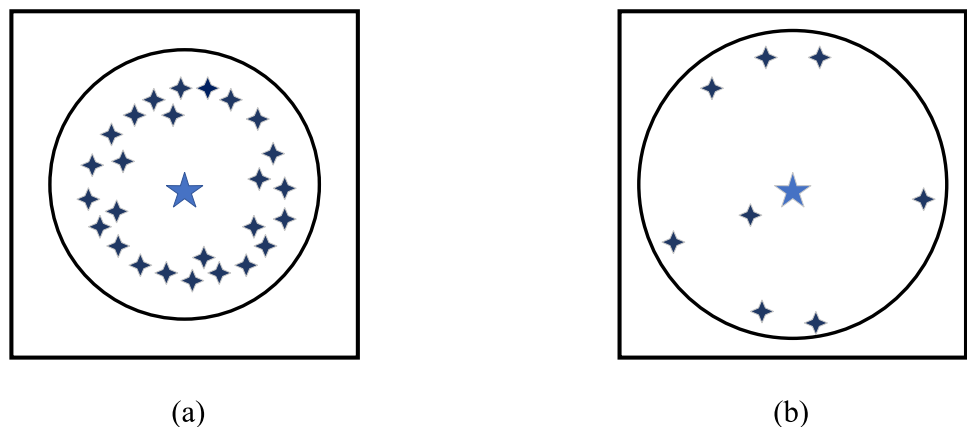
After thorough research and analysis, we found that FA has a significant advantage in generating new leaders. The core idea of this algorithm originates from the firework explosion phenomenon in nature and achieves global search and optimization of the problem by simulating the process of a firework explosion. Collectively, FA regards each feasible solution as a fireworks particle and evaluates their performance according to fitness value in each iteration. Then, the algorithm chooses the best individual as the leader of the current iteration and explodes it to generate new fireworks particles. This process helps the algorithm approach the optimal solution to the problem step by step while maintaining population diversity. It is worth noting that the fireworks algorithm will consider the balance between population diversity and convergence speed when generating new leaders and achieve this goal by adjusting parameters such as explosion radius and spark number.

In this study, we rely on the core idea of FA, which is to use FA to generate fbest and thus facilitate population renewal. Compared with the traditional leader renewal mechanism, the introduction of fbest can not only help the population get rid of the local optimal state but also significantly improve its ability to accurately locate the real PF. This improvement is expected to improve the efficiency of the algorithm in exploring and developing solutions when dealing with MOPs. The specific implementation steps are as follows: during the operation of the algorithm, we pay close attention to the number of non-dominated solutions. When the quantity is insufficient, we use FA to “explode” the existing non-dominated solutions and generate new sparks. Then evaluate these sparks and select outstanding individuals. Together with gbest, these individuals guide population renewal and explore the more optimal solution area. This FA-based mechanism increases population diversity and reduces local optimal risk. Excellent spark and a global optimal solution work together to accelerate convergence and improve solution efficiency. This innovative method provides new ideas for MOPSO, achieves a better balance between exploration and development, and obtains accurate and efficient solutions.

3.2 Specific Steps for fbest Generation

Under close scrutiny of the fireworks display, we are able to recognize two distinct explosion patterns, as shown in Fig. 2. Those that are well crafted release a starburst of sparks that are tightly focused at the center of the explosion, creating a breathtaking spectacle. On the other hand, lesser-quality fireworks display sparks that are not only few in number but also scattered. This is illustrated in the figure. In the algorithm's perspective, these high-performing fireworks can be viewed as outstanding particles in the population, which lend themselves to deeper exploration in localized areas through more abundant sparks. In contrast, the poor-quality fireworks symbolize the dominated

Fig. 2 Fireworks explosion diagram



particles, which need to expand their search to find a more optimal solution. In the framework of FA, high-quality fireworks present the properties of a large number of sparks and a limited range of explosion waves. However, in this study, we chose to represent the non-dominated solutions as “fireworks,” which means that all the selected fireworks are of high quality.

In the FA, the explosion radius and the number of sparks produced by the explosion of each firework are calculated based on its adaptation value relative to other fireworks in the firework population. For fireworks x_i , the explosion radius A_i and the number of sparks S_i are calculated as follows, respectively:

$$A_i = A \cdot \frac{F(Ns_i) - F_{\min} + \xi}{\sum_{i=1}^n (F(Ns_i) - F_{\min}) + \xi} \quad (7)$$

$$S_i = m \cdot \frac{F_{\max} - F(Ns_i) + \xi}{\sum_{i=1}^n (F_{\max} - F(Ns_i)) + \xi} \quad (8)$$

where $F_{\min} = \min \sum f_j(x_i)$, $F_{\max} = \max \sum f_j(x_i)$ are the minimum and maximum values of the sum of non-dominated particle fitness values of the current population, respectively. A is a constant used to adjust the size of the blast radius, m is also a constant used to adjust the number of sparks produced by the explosion, and ξ represents the smallest

positive constant in the computer system, which serves to avoid zero error when performing division operations.

To ensure that the number of sparks S_i is reasonable and to avoid it being too much or too little, we define the boundary of S_i as shown in the following Eq:

$$S_i = \begin{cases} \text{round}(a \cdot m) & \text{if } S_i < am \\ \text{round}(b \cdot m) & \text{if } S_i > bm \\ \text{round}(a \cdot m) & \text{otherwise} \end{cases} \quad (9)$$

where a and b are parameters that limit the number of sparks to prevent too many or too few and satisfy $a < b < 1$, and $\text{round}()$ is a rounding function based on the principle of rounding.

During a firework explosion, sparks may be affected by the explosion in randomly selected Z dimensions. We randomly determine the number of dimensions affected by the following Eq. (10):

$$z = \text{round}(d \cdot \text{rand}(0,1)) \quad (10)$$

where Z is the number of dimensions of the particle explosion, d is the positional dimension of Ns_i , and $\text{rand}(0,1)$ is a random number on $[0,1]$.

Algorithm 1: FA to generate fbest

Input:(Ns ,N) % Ns is the non-dominated particle and N is the archive capacity.

Output:(fbest) % fbest is the optimal particle generated by FA.

```

1: if size(NS)>N/2
2:   Populations were updated using Eqs (19) and (20).
3: else size(NS)<N/2
4:   Determine the number of sparks using Eqs (7) and (8).
5:   Using Eqs (9) and (10) to determine the explosion range and the
      number of dimensions of the explosion.
6:   Generate explosive sparks using Eqs (11) and (12).
7:   fbest = optimal explosion spark.
8: end if

```

In the above process, we have determined the number of firework explosions, the explosion radius, and the dimension of the explosion, and then the particles generate a large number of sparks through Eqs. (11) and (12), and then we select the batch of particles with the best performance, which is referred to as the fbest in this paper, among these sparks. Then the population is guided by the fbest and the gbest together, and the population achieves the information sharing with the population. This process greatly enriches the exploration of the search space and improves the algorithm's ability to discover Pareto optimal solutions. Its pseudocode is shown in Algorithm 1.

$$h_i = A_i \cdot \text{rand}(-1, 1) \quad (11)$$

$$\Delta N s_i^j = N s_i^j + h_i \quad (12)$$

where A_i is the explosion radius of the i th particle in Eq. (7), h_i is the magnitude of the change in particle position in Eq. (12), $\text{rand}(-1, 1)$ is a random number between $[-1, 1]$, and $N s_i^j$ denotes the j th position dimension of the i th non-dominated solution.

3.3 Large and Small Double Archiving Mechanism

In real life, we often observe the phenomenon that population size has a significant impact on the production of good talent. A larger population base implies a larger potential talent pool, which increases the likelihood of producing high-quality talent. Drawing on this idea, we design a size-dual archiving mechanism to maintain the archive of the optimization process. The initial step is to construct an archive, designated as A, with a substantial storage capacity. This archive has a maximum capacity three times the number of population particles, allowing it to hold more non-dominated particles. If A can no longer hold as many particles as possible, then an adaptive grid technique is used to keep the archive at a manageable size and guarantee that resources are used wisely. To produce particles of higher quality, secondly, mutation operations are carried out on particles with a higher density in A. Then, to correspond with the population size, archive B is created with a lower capacity. B serves as a storage space for the premium particles that were chosen from A. The population renewal process will thereafter be guided by these particles.

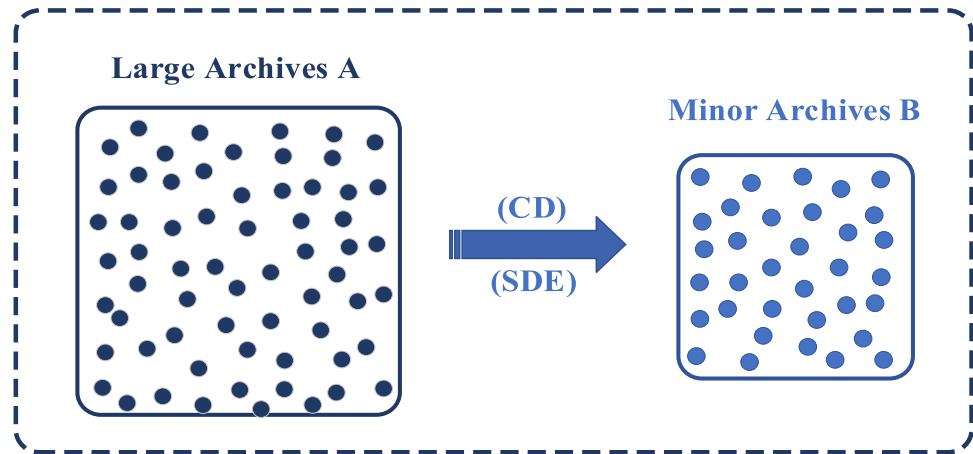
The details are as follows: if the number of particles in archive A does not reach the population size, all particles in archive A are transferred to archive B. When the number of particles in archive A exceeds the population size, a mutation operation is performed, i.e., the crowding distance of each particle is calculated, and some particles with a smaller crowding distance are selected for mutation so as to promote the migration of particles from dense areas to sparse areas and further increase the population diversity. In this process, according to Eq. 13, the particles with a large crowding distance are selected to guide the mutation. After the mutation is completed, the mutated particles with excellent performance are screened and added to archive A. It is confirmed that to balance the population stability and mutation effect, the number of population variations should be controlled between $[\text{size}(A) - N, N/2]$ (when $\text{size}(A) - N > N/2$, the number of variation is $N/2$), where $\text{size}(A)$ refers to the number of particles in archive A and N is the population size. Finally, SDE and CD are used to determine which solutions should be transferred to archive B. This method comprehensively considers the distribution of solution sets and individual convergence information, which not only maintains the diversity of solution sets but also ensures good convergence of solution sets. See Algorithm 2 for the relevant algorithms. Using the SDE and CD indicators to select high-quality, non-dominated solutions in Archive A, we can ensure that Archive B is full of potential candidate solutions until Archive B reaches its maximum capacity. Then, the excellent particles in Archive B will lead to population renewal and push the search process to the optimal frontier.

$$\text{mut}(NS_i) = \text{rand}(1) \cdot (NS_k - NS_i) + (1 - \text{rand}(1)) \cdot (NS_j - NS_i) \quad (13)$$

where NS_i is the particle with smaller CD value, NS_k and NS_j are the particles with larger CD value ($i \neq k \neq j$), and $\text{rand}(1)$ is a randomly generated random number in $[0, 1]$, the formula is oriented to the particles in the sparse region and aims to direct the particles in the dense region towards the coefficient region.

SDE is a transfer-based density estimation strategy that is an improved density estimation method that evaluates the density of individuals by considering their convergence and distributional information. This method not only focuses on the distribution of individuals in the population but also considers the convergence of individuals. SDE is described as follows:

Fig. 3 Schematic diagram of a large and small double archive



$$\overset{\Delta}{f}_m(X_j) = \begin{cases} f_m(X_j), & \text{if } f_m(X_i) < f_m(X_j) \\ f_m(X_i), & \text{if } f_m(X_i) \geq f_m(X_j) \end{cases} \quad (14)$$

where X_i is the particle in the large archive A, ($i, j = 1, 2, 3, \dots, N_A$, $i \neq j$) N_A is the number of particles in archive A. $m = 1, 2, \dots, M$, M is denotes the total number of targets. $f_m(X_i)$ is denotes the m th target value of the i th particle. $\overset{\Delta}{f}_m(X_j)$ denotes the target value after shifting the m th target value of the solution X_j .

CD is used to assess and maintain population diversity. It is based on the principle that crowding of individuals is measured by calculating the average distance between each individual and its neighbors, which helps the selection operation to select the better adapted individuals while maintaining the population diversity. CD is described as follows:

$$CD(X_{i,j}) = \sum_{j=1}^M \frac{f(X_{i+1,j}) - f(X_{i-1,j})}{FMax_j - FMin_j} \quad (15)$$

where $FMin_j = \min(f_j(x))$ and $FMax_j = \max(f_j(x))$ denote the minimum and maximum values of particle fitness values on the j th dimensional target, respectively, and $f(X_{i+1,j})$ denotes the j th target value for the $i + 1$ th particle.

In this study, a large-scale archive is built to simulate the influence of population size on talent output in the real world. When using evaluation indices such as SDE and CD to select outstanding individuals, increasing the number of non-dominant solutions can expand the selection range of SDE and CD and then improve the probability of selecting high-quality individuals (see Fig. 3). This mechanism is consistent with the theory that the probability of producing high-quality talents is higher when the population is large

and has been successfully applied to the process of solving optimization problems.

3.4 Active Information Fusion Strategy and Adaptive Parameters for Particle Update Detection

Under the MOPSO framework, the population particles present three unique scenarios when sharing information with pbest, revealing the different states of the particles during information exchange. First, we observe that the particle $X_i(t)$ of the current generation dominates the particle $X_i(t-1)$ of the previous generation, i.e., $X_i(t) < X_i(t-1)$ (as shown in Fig. 4, a), which clearly indicates that the particle is in an evolutionary state as it is moving towards a more optimal solution region. At this point, the pbest of the particle is updated to $X_i(t)$. However, this change results in zero results for the individual cognition part of the update formula, rendering pbest useless in exchanging information with particles.

Second, the current generation particle $X_i(t)$ is dominated by the previous generation particle $X_i(t-1)$, i.e., $X_i(t) > X_i(t-1)$ (as shown in Fig. 4, b), and we recognize that the particle is in a degenerate state. In this case, the pbest of the particle is maintained as $X_i(t-1)$, and since the particle has already shown a degenerative trend, continuing to use $X_i(t-1)$ as the leader will, with high probability, lead to the population being trapped in a locally optimal solution, which hinders the possibility of exploring a wider solution space.

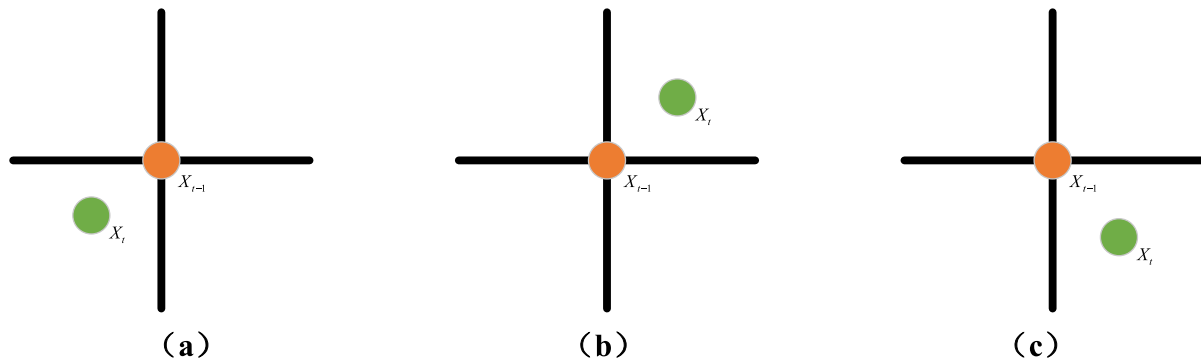


Fig. 4 Schematic of particle evolution

Algorithm 2: Maintenance of size double archive

Input:(NS, N) % NS is the non-dominated particle and N is the archive capacity.
Output:(CA, CB) % CA is the large archive and CB is the small archive.

```

1: if size(NS) > 3*N
2:   Maintenance of archived CA using adaptive grids
3: else size(NS) ≤ 3*N
4:   Deposit NSs into CA
5:   if size(CA) < N
6:     Transfer NS from CA to CB.
7:   else size(CA) > N
8:     Calculate the CD value for each particle.
9:     Num = size(CA) - N
10:    Determine the number of mutations (M)
11:    M should be not less than Num/2 and not more than N/2.
12:    Find the M particles with the smallest CD value and mutate them using Eq. (13).
13:    Remove these M particles from A.
14:    After mutation, add the optimal mutated particles to A.
15:    Calculate the SDE and CD values for each particle in A.
16:    Add the two index values of the corresponding particles and write them as SCD.
17:    Rank the SCD from largest to smallest.
18:    Place the N highest ranked NS in CB.
19:   end if
20: end if

```

Finally, examining the case where the current generation of particles $X_i(t)$ and the previous generation of particles $X_i(t-1)$ do not dominate each other, i.e., $X_i(t) \sim X_i(t-1)$ (as shown in Fig. 4c), we find that the particles have neither significantly evolved nor degraded. According to the traditional MOPSO update strategy, the pbest of the particle at this point can be $X_i(t)$ or $X_i(t-1)$. However, choosing $X_i(t)$ as the pbest will also result in zero contribution from the individual cognitive component, while choosing $X_i(t-1)$ may cause the population to stagnate in the local optimum and neither choice can effectively promote the overall evolution of the population.

The update of particles in a traditional MOPSO is dependent on tracking their pbest. But as was already mentioned, there are clear flaws in this mechanism for handling MOPs, particularly when it comes to preserving population

diversity and averting premature convergence. To address these issues, we developed a novel strategy for selecting leaders—the so-called Active particle. This type of particle is based on the fusion of multi-particle information, which seeks to direct the particle updating process by utilizing both the individual particle information and the global information of the entire population instead of depending solely on the particle's or its neighbors' best experience.

Because of the way that Active particles are designed, every particle can achieve a dynamic and varied information exchange. This facilitates information sharing and exchange between particles and improves the algorithm's capacity for exploration. To create a new leader, the Active particle actually chooses at random which member of the population is the best and combines it with the particle itself (see Eq. (16) for more information). In this way, Active particles serve as effective guides for other particles to explore by representing distinct regions in the solution space in addition to high-quality solutions. The algorithm's performance is greatly enhanced by the addition of the Active mechanism when compared to the conventional pbest method. It not only quickens the algorithm's rate of convergence but also significantly raises the caliber of the solution that is found. More importantly, Active offers fresh concepts and approaches for resolving challenging MOPs while drastically lowering the likelihood that the algorithm will enter a local optimum by preserving population diversity.

$$\text{Active}(t) = \frac{X_i(t) + \text{Active}(t-1)}{2} \cdot \text{rand}(1) + (1 - \text{rand}(1)) \cdot \text{gbest} \quad (16)$$

where $\text{Active}(t)$ denotes the activated particle of the t th iteration, $X_i(t)$ denotes the i th particle of the t th iteration, $\text{rand}(1)$ is a randomly generated vector in $[0,1]$ used to increase the randomness of the search, gbest is the global optimal particle. The sharing of information between the particle itself and the global optimal particle particles is considered in

Eq. (16), rather than a single dependence on the particle's own information.

3.5 Adaptive Parameter Tuning Mechanism

In FAMOPSO, the complexity and diversity of the information sharing mechanism of particles puts higher demands on the algorithm, which suggests that the conventional MOPSO method—which uses fixed parameters w , $c1$, and $c2$ —is insufficient in certain ways. The many requirements and difficulties that particles may encounter at various phases are ignored by this preset parameter setting. This research proposes a novel adaptive adjustment approach in recognition of this. The “dynamic adjustment”—the adaptable modification of important parameters w , $c1$, and $c2$ in accordance with the particle state in real time—is the fundamental component of this technique. This implies that the flight path of particles is now shaped by the interaction between particle performance in real time and environmental feedback, rather than by unchangeable rules. By doing this, we can more precisely satisfy the requirements of particles in various update scenarios, enhancing the overall algorithm's effectiveness and efficiency.

The specific adjustment strategies are as follows:

(1) Local optimality escape strategy: in response to the potential degradation phenomenon of particles during the iterative process or the indication of the lack of dominance relationship between particles of two consecutive generations, this paper keenly identifies that the particles may have been caught in the dilemma of local optimality. To effectively address this challenge, the strategy is to adopt a refined parameter tuning scheme, i.e., moderately reducing the values of the inertia weight w and the individual learning factor $c1$, while significantly boosting the role of the social learning factor $c2$ (the details of the tuning follow the established adaptive Eq. 17). The aim is to weaken the particles' reliance on historical trajectories, reduce the risk of premature convergence caused by individual overconfidence, and enhance the information exchange and sharing among particles, encouraging the particle community to explore the solution space with more potentials so as to help the particles break through the current limitations, move towards a broader search frontier, and discover and locate the global optimal solution.

$$w_i = W - \text{rand}(1) \cdot \left(\left| F(X_i^t) - F(X_i^{t-1}) \right| / 2^{|F(X_i^t) - F(X_i^{t-1})|} \right)$$

$$c1_i = C1 - \text{rand}(1) \cdot \left(\left| F(X_i^t) - F(X_i^{t-1}) \right| / 2^{|F(X_i^t) - F(X_i^{t-1})|} \right) \quad (17)$$

$$c2_i = C2 + \text{rand}(1) \cdot \left(\left| F(X_i^t) - F(X_i^{t-1}) \right| / 2^{|F(X_i^t) - F(X_i^{t-1})|} \right)$$

(2) Excellent solution neighborhood exploration strategy: in the journey of multi-objective optimization, when the particles show significant evolutionary momentum, i.e., their trajectories are steadily moving towards the better solution neighborhood, the strategy of this paper turns to focus on deepening the local exploration and fine search. At this point, the parameter configuration is strategically adjusted, which is manifested by increasing the values of the inertia weights w and the individual learning factor $c1$, while moderately reducing the influence of the social learning factor $c2$ (the details of the adjustment follow the well-designed adaptive Eq. 18). The purpose of this adjustment is to strengthen the ability of the particles to deeply explore the neighborhood of the current excellent solution and encourage the particles to carry out more detailed mining and iteration on the basis of the known high-quality solutions, with a view to further refining and revealing the better solutions hidden in the neighborhood, thus promoting the continuous refinement and optimization of the algorithm in the solution space.

$$w_i = W + \text{rand}(1) \cdot \left(\left| F(X_i^t) - F(X_i^{t-1}) \right| / 2^{|F(X_i^t) - F(X_i^{t-1})|} \right)$$

$$c1_i = C1 + \text{rand}(1) \cdot \left(\left| F(X_i^t) - F(X_i^{t-1}) \right| / 2^{|F(X_i^t) - F(X_i^{t-1})|} \right) \quad (18)$$

$$c2_i = C2 - \text{rand}(1) \cdot \left(\left| F(X_i^t) - F(X_i^{t-1}) \right| / 2^{|F(X_i^t) - F(X_i^{t-1})|} \right)$$

where w_i is the inertia weight of the i th particle, $c1_i$, $c2_i$ are the learning factors of the i th particle. W , $C1$, $C2$ the base parameters in Eqs. 17–18, $\text{rand}(1)$ is a random number between $[0,1]$, $F(X_i^t) = \sum f(x_{i,t})$ is the sum of the fitness values of particle i at the t th iteration, and $||$ denotes the absolute value sign. The common purpose of these three sets of formulas is to dynamically adjust the algorithm parameters according to the current status of the population, including diversity, convergence speed, and other relevant factors, to optimize the algorithm's performance and adaptability. This adaptive adjustment mechanism is an important component in the design of modern MOEAs and is of great significance in improving the solution quality and efficiency of the algorithm.

$$V_i(t+1) = w_i V_i(t) + c1_i r_1 (\text{Active}(t) - X_i(t)) + c2_i r_2 (\text{gbest}_i(t) - X_i(t)) \quad (19)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (20)$$

By utilizing the adaptive adjustment approach mentioned above, every particle can determine the optimal parameter configuration for the present state by evaluating its performance in real-time throughout the search process. The

flexibility, adaptability, and overall performance of the FAMOPSO algorithm in handling complex MOPs are much enhanced by this method, which also successfully strikes a balance between the demands of local and global exploration. In addition, this dynamic adjustment mechanism also provides more opportunities for particles to explore new regions in the solution space and dig deeper when promising regions are found, thus accelerating the optimization process and improving the quality of the understanding. Since the new leader-Active particles and adaptive parameters are used, the mathematical expressions for particle updating in this paper are shown in Eqs. (19) and (20).

3.6 In-Depth Analysis—FAMOPSO

In traditional MOPSO, the population tends to lose diversity because it relies on only a few gbest to share information with the population at the beginning of the iteration, leading to the problem of decreasing population diversity and increasing the risk of local optimality. To solve this problem, we borrow the ‘explosion mechanism’ that is unique to FA. This mechanism allows existing gbest to be ‘exploded’ into multiple new individuals, called fbest, which not only inherit some of the good traits of the original gbest, but also carry some new traits after mutation, thus providing more diverse choices and richer genetic information for the whole population. This ensures that even at the beginning of the iteration, when the number of gbest is relatively small, there are enough leaders to guide the population in a better direction. In addition, since fbest are obtained by expanding existing gbest, there is a degree of variability among them that helps to maintain or even improve the overall diversity level of the population.

The experimental results show that, compared to the traditional MOPSO algorithm without any improvements, the FAMOPSO method proposed in this paper is able to significantly reduce the probability of falling into a local optimum while maintaining a high search efficiency. In particular, its advantage is more obvious when dealing with complex multi-peak functions or objective spaces with highly non-linear characteristics. This shows that by introducing the ‘explosion mechanism’ in FA and applying it to the MOPSO framework, the drawbacks of relying on a small number of gbests can be effectively mitigated, and the non-dominated solution front obtained in the end is closer to the real Pareto front.

Second, in the traditional MOPSO archive management strategy, the single archive mechanism dominates, and its core mechanisms all revolve around the particle deletion operation, which is triggered immediately when the archive capacity is exceeded, with the aim of retaining the particles with superior performance as potential candidates for gbest. This process is essentially a one-time screening of the

archive content, which directly determines the gbest candidate set.

The innovative double-size archive strategy proposed in this paper is inspired by the principle of a positive correlation between population size and the rate of output of elite talent in human societies, i.e., the larger the base of high-quality individuals, the higher the probability of breeding better individuals. Based on this, we design an oversized primary grand archive, three times the size of the population, with the aim of broadly accommodating non-dominated solutions and ensuring maximum preservation of diversity and convergence. When the grand archive is saturated, it is fine-tuned using an adaptive grid technique, which effectively increases the overall diversity of the population through spatial partitioning with intelligent culling of particles in high-density regions, constituting the first round of selection of non-dominated solutions. We then introduce a secondary mini-archive that matches the population size and use two indicators, SDE and CD, as a scoring system to evaluate the particles in the primary archive in depth. Only the top particles that pass the assessment of these two strict metrics can be promoted to the secondary archive until it reaches its maximum capacity limit. This link is the second round of fine screening for non-dominated solutions. Compared to the traditional approach, our strategy adds an additional layer of filtering mechanism that significantly improves the quality of non-dominated particles as gbest candidates, thus promoting more efficient convergence while increasing the diversity of the algorithm.

In addition, to further enrich the genetic diversity of the large archive and to prevent premature convergence of the algorithm, we embed a mutation mechanism. An adaptive grid strategy is used to locate the densest regions of particle distribution and to target some particles for mutation. This local fine-tuning strategy has little impact on the overall performance of the algorithm due to the limited number of particles, but on the other hand, it can induce particles to migrate to the low-density region, which effectively expands the search space and strengthens the algorithm’s ability to explore the unexplored areas, demonstrating a significant advantage in maintaining the diversity of the population and avoiding the trap of local optimality.

It is worth noting that for the maintenance of the large archive, we use the adaptive grid technique, a strategy that enhances population diversity by dynamically adjusting the grid division in the search space to adapt to changes in the distribution of populations. Specifically, the adaptive grid automatically adjusts the size and position of the grid according to the distribution of individuals in the target space. This method can cover the solution space more effectively, especially for high-dimensional problems, and helps to maintain population diversity and avoid falling into local optima. The effectiveness of this method has been verified

in the literature [22–24], showing that it can significantly improve the diversity exploration capability of the algorithm. In the small archive pairwise particle screening process, we use the SDE metric, which is an improved density estimation method that enhances population diversity by changing the transfer probability of individuals. Unlike traditional density estimation methods, SDE considers not only the distribution of individuals in the population but also their convergence. Specifically, SDE shifts individuals with poor convergence to crowded regions and assigns them a higher density value, making them more likely to be removed during the selection process. The effectiveness of this strategy is also supported in the literature [43], highlighting its role in improving the overall performance of the algorithm. Overall, both adaptive grids and SDEs are effective strategies for increasing the diversity of MOEAs. Adaptive grids dynamically adjust the partitioning of the search space, while SDE alters the transfer probability of individuals, both of which help to maintain the diversity of the population and prevent the algorithm from prematurely converging to a local optimal solution.

Finally, we analyze the three scenarios of particle updating and find that the traditional pbest is no longer suitable for guiding particle updating. Therefore, in this paper, we innovatively propose the concept of Active particles to replace the traditional pbest, which not only takes into account the information of individuals but also fully integrates the information of the global optimum, thus effectively improving the performance of the algorithm. In addition, we incorporate an adaptive parameter tuning strategy, aiming to break through the bottleneck of the existing algorithms. These initiatives aim to enhance the algorithm's ability to explore and develop complex multi-objective optimization problems and inject new research vitality and practical paths into the whole field. Through these improvements, we expect to push the multi-objective optimization algorithms in the direction of being more efficient and smarter.

In summary, this paper integrates the FA strategy for generating fbest, the size double archiving strategy, and the design of Active particles and adaptive parameters. It not only effectively solves the problem of insufficient diversity of the MOPSO algorithm at the initial stage but also significantly improves the solving efficiency and accuracy of the algorithm for complex MOPs through the refined management and screening mechanism.

3.7 Complexity Analysis of the Proposed Algorithm

To deeply analyze the computational complexity of the FAMOPSO algorithm, this paper focuses on its core components, covering the fireworks algorithm to generate the fbest, the maintenance mechanism of the small and large double archives, the generation strategy of Active particles, and the dynamic update process of the adaptive parameters.

The population size is set to be N , the maximum capacity of the small archive and the large archive are N and $3N$, respectively, the target dimension is M , the dimension of the decision space is D , and the number of non-dominated solutions is denoted as G , where the number of non-dominated solutions exceeding the archive capacity is defined as Z (i.e., $G - N = Z$).

In the maintenance process of the size-double archive, two key aspects are involved: the maintenance operation of the archive and the mutation processing of the particles. The computational complexity of archive maintenance is divided into three scenarios based on different situations:

1. If the sum of the newly generated non-dominated particles of the population and the total number of existing archived particles does not exceed N , they are added directly, and then the computational complexity is $O(I)$.
2. When the sum is between N and $3N$, two evaluation indexes, SDE and CD, need to be introduced, both of which have a computational complexity of $O(GM)$, so the overall complexity in this case is $O(G + 2GM)$.
3. If the sum exceeds $3N$, the adaptive grid strategy is firstly used to eliminate redundant particles, and the complexity of this step is $O(GZ)$, so the overall complexity is raised to $O(G + GZ + 2GM)$.

The computational complexity of the particle mutation is stable at $O(N/2 + GM)$. Taken together, the maximum computational complexity of the size double archive maintenance part can be expressed as $O(G + GZ + 3GM + N/2)$. Further, the computational complexity involved in the process of generating fbest under the action of FA is $O((N + 4) \times (N/4))$, while the computational complexity of Active particle generation and adaptive parameter adjustment are $O(N^2)$ and $O(N)$, respectively. In particular, it is noted that the computational complexity of Active particle generation is consistent with the process of selecting pbest in the original MOPSO algorithm.

In conclusion, the overall computational complexity of the FAMOPSO algorithm is jointly determined by the key steps of the maintenance of the size double archive, the generation of the fbest, the generation of Active particles, and the timely tuning of the adaptive parameters, and the combined effect of these factors shapes the efficiency and performance characteristics of the algorithm in solving complex multi-objective optimization problems.

4 Experimental Research

This section uses three well-known benchmark test sets, ZDT [45], UF [46], and DTLZ [47], to thoroughly assess the performance of the suggested FAMOPSO method. These

test sets encompass a range of attributes and challenges in MOPs, enabling a thorough assessment of the algorithm's efficacy.

We carefully designed a series of experiments and conducted a thorough comparison with other popular MOPSO algorithms to thoroughly examine the performance of the FAMOPSO algorithm. These algorithms, which give a strong basis for our comparison, include MMOPSO [48], MOPSOCD [49], NMPSO [34], SMPSO [50], and dMOPSO [51]. They reflect several research directions and technological advancements in the field of multi-objective optimization. Additionally, we compare this algorithm with a number of competitive MOEAs to assess FAMOPSO's efficiency even more. Among these MOEAs are MOEAD [52], NSGAIII [8], SPER [53], CMOAD [54], and IMMOEA [55], which are recognized as highly proficient examples in the optimization field and exhibit exceptional performance while handling MOPs.

To guarantee the impartiality of any comparative assessment of algorithmic performance, it is essential to ensure that all pertinent parameters established by the comparative algorithm are aligned with the original references. Every test question undergoes 30 separate runs of each algorithm. Using the MATLAB R2020b platform, the experimental results of every algorithm were acquired on an Intel (R) Core (TM) i7-7700 CPU @ 3.60 GHz Windows 10 machine. The transparency and repeatability of the experiment are guaranteed by the source code of the comparison algorithm, which is made available by PlatEMO [56]. The purpose of the experiment is to assess FAMOPSO's performance using a number of quantitative and qualitative metrics. These consist of stability, computational efficiency, diversity, and convergence. These indications allow for a thorough evaluation of FAMOPSO's performance in managing MOPs from a variety of angles.

The experimental results will provide a comprehensive analysis of the performance differences between FAMOPSO and the comparison algorithm on each benchmark set. The capacity of FAMOPSO to explore and utilize the solution space will be examined, as will its efficacy in maintaining solution diversity and avoiding premature convergence. Furthermore, the performance of FAMOPSO will be examined across a range of MOPs types and difficulties, with a view to establishing its applicability and robustness. This chapter is intended to provide readers with a comprehensive and transparent evaluation of the performance of the FAMOPSO algorithm, based on careful planning of the experimental process and an in-depth discussion of the results.

4.1 Test Questions

To deeply analyze the performance of the FAMOPSO algorithm, we carefully selected and used three different

Table 1 Parameter Settings for 22 test questions

Problems	N	M	D	Fes
ZDT1–ZDT3	200	2	30	10,000
ZDT4 and ZDT6	200	2	10	10,000
UF1–UF7	200	2	30	10,000
UF8–UF10	200	3	30	10,000
DTLZ1	200	3	7	10,000
DTLZ2–DTLZ6	200	3	12	10,000
DTLZ7	200	3	22	10,000

benchmark suites for experimental evaluation. These kits include the diversification problems and PF shapes in the optimization field, which ensures the comprehensive evaluation of the algorithm in various situations. For example, we choose five bi-objective problems in the ZDT suite, which are widely studied in the industry and can effectively test the ability of the algorithm to deal with bi-objective problems with different characteristics. At the same time, we also use seven bi-objective problems, three tri-objective problems of the UF suite, and seven tri-objective problems of the DTLZ suite, each of which considers different PF shapes and dimensions, so as to comprehensively evaluate the performance of the algorithm in MOPs.

It should be pointed out that, due to the particularity of the ZDT5 problem (discrete optimization), we did not take it into consideration. Similarly, DTLZ8 and DTLZ9 are not included because they involve constrained optimization, which is beyond the scope of this study.

Table 1 lists the parameter settings of each test question in detail, including the overall size of the particle swarm (“ N ”), the number of optimization targets (“ M ”), the dimension of the decision space (“ D ”), and the maximum number of evaluations (“Fes”). These parameters are carefully designed according to the uniqueness of each test question to provide a challenging search space for the algorithm and evaluate its performance more accurately. Through these comprehensive benchmark tests, we can fully understand the performance of FAMOPSO on MOPS with different difficulties and types, which is of great significance for deeply understanding the efficiency of the algorithm and guiding its future application and improvement.

4.2 Comprehensive Performance Evaluation Metrics

In FAMOPSO, the key metrics for evaluating the performance of the algorithms include inverted generational distance (IGD) [57] and hypervolume (HV) [58]. These metrics are critical for measuring the performance of the algorithms in solving MOPs.

Mathematical definitions:

$$\text{IGD} = \frac{1}{|P^*|} \sum_{p \in P^*} \min_{P} \|P - p\| \quad (21)$$

where P^* is a set of uniformly distributed reference points representing the ideal Pareto front; P is the solution set on the approximate Pareto front generated by the algorithm; $\|\cdot\|$ denotes the Euclidean distance; and $|P^*|$ is the number of reference points. The smaller the value of IGD, the closer the solution set generated by the algorithm is to the ideal Pareto front, i.e., the better the convergence and diversity.

$$\text{HV} = \text{Volume} \left(\bigcup_{i=1}^{|P|} v_i \right) \quad (22)$$

where v_i is the volume of the goal space occupied by each solution in the solution set. A larger value of HV indicates that the non-dominated solutions found cover a wider range of the goal space, thus reflecting the exploratory capability and diversity of the algorithm.

Reasons for selection: IGD considers both convergence and diversity. It not only focuses on the distance of the solution from the true Pareto front (i.e., convergence), but also indirectly promotes diversity in the solution set by minimizing the distance from all reference points to the nearest solution. HV directly quantifies the volume of the target space occupied by the set of approximate solutions and is an important measure of diversity. A high HV value means that the algorithm is able to find more good-quality solutions spread across different regions, which is particularly important for multi-objective optimization.

Discussion of limitations: although IGD and HV are powerful tools for evaluating the performance of MOEAs, they have some limitations:

1. Computational complexity: especially for high-dimensional problems, computing IGD and HV can be very time-consuming as they require the evaluation of a large number of reference points or distances between solutions.
2. Dependence on the choice of reference points: the accuracy of IGD is highly dependent on the choice of reference points. Inappropriate reference points may lead to misinterpretation of the algorithm's performance.
3. Does not fully reflect preferences: both metrics focus on the overall quality of the solution set and do not directly reflect the preferences of a particular decision maker. In some applications, more specific metrics may be needed to assess the quality of solutions that satisfy specific needs.

In summary, we use IGD and HV as key metrics for evaluating the performance of the FAMOPSO algorithm. IGD enables a comprehensive assessment of the convergence and

diversity of the algorithm, while HV provides a method of evaluation that does not require reference to the Pareto frontier. Although these metrics have their own strengths and limitations, together they provide us with a comprehensive view that helps us better understand the FAMOPSO.

4.3 Experimental Parameterization

In this paper, we provide an in-depth look at the performance of FAMOPSO on MOPs and compare it to several popular algorithms for MOPSOs, as well as five representative MOEAs. To ensure the fairness of the comparison, we carefully configured the relevant parameters of all the algorithms involved in the comparison to ensure that they are consistent with the settings reported in the respective original references. Such a rigorous approach aims to eliminate any performance bias due to misconfigured parameters, thus ensuring the fairness and reliability of the experimental results. In terms of specific implementation, the FA-related parameters in FAMOPSO were carefully tuned to $m = 100$, $A = 40$, $a = 0.3$, and $b = 0.8$. Meanwhile, in the adaptive parameter update mechanism, the parameters were set to $W = 0.4$ and $C1 = C2 = 2$. The careful selection and tuning of these parameters are intended to enable the FAMOPSO algorithm to maintain a balance between exploration and exploitation, resulting in superior performance on MOPs.

4.4 Experimental Results

Comparison with five other MOPSOs: Tables 2 and 3 provide a detailed account of the performance of the six MOPSOs on the ZDT, UF, and DTLZ test problems. The data set includes the standard deviation and mean of the IGD and HV metrics, thus providing a comprehensive quantitative assessment of the algorithms' performance. In Tables 2 and 3, the terms "Mean" and "Std" refer to the mean and standard deviation, respectively. The best IGD and HV values are highlighted in bold font to facilitate the rapid identification of the algorithms' strengths on different test problems.

Through an examination of the outcomes condensed in Table 2's final row, we can see that FAMOPSO performed admirably across the 22 benchmark test cases. In particular, FAMOPSO completes 12 test problems with the best results, a feat that fully validates its exceptional performance. MMOPSO, which showed the best performance on five test instances, trailed it closely. While SMPSO performed well on the ZDT6 test problem and MMOPSO performed admirably on the DTLZ test problem, the other MOPSO algorithms in the comparison did not outperform each other on the majority of the test problems. However, their performance on the UF test suite was not as strong. Specifically, SMPSO performs noticeably worse on the DTLZ6 test problem. The aforementioned observations indicate that, while

Table 2 IGD values for the six MOPSOs on the three sets of test problems

Problem	M	D		MMOPSO	MOPSOCD	NMPSO	SMPSO	dMOPSO	FAMOPSO
ZDT1	2	30	Mean	2.6903e−3	2.8044e−3	2.9713e−2	7.6673e−2	5.0208e−2	2.8246e−3
			Std	1.06e−5	8.52e−4	1.01e−2	8.59e−2	1.43e−2	9.23e−5
ZDT2	2	30	Mean	2.1051e−1	1.0202e−1	2.0089e−2	7.4416e−2	4.4934e−2	3.5633e−3
			Std	2.57e−1	1.79e−1	1.03e−2	1.12e−1	2.08e−2	4.83e−4
ZDT3	2	30	Mean	5.2883e−3	2.8715e−2	9.1301e−2	1.6430e−1	4.1424e−2	2.8368e−3
			Std	1.34e−2	3.99e−2	1.89e−2	8.54e−2	8.57e−2	4.43e−4
ZDT4	2	10	Mean	8.2536e+0	1.8067e+1	1.4735e+1	9.3244e+0	3.3939e+0	1.7915e+1
			Std	4.78e+0	8.15e+0	7.79e+0	4.03e+0	4.75e+0	5.27e+0
ZDT6	2	10	Mean	2.0754e−3	2.9631e−3	2.2838e−3	1.9518e−3	6.2746e−3	2.3143e−3
			Std	9.33e−5	1.33e−3	2.39e−4	9.94e−5	1.07e−2	3.90e−4
UF1	2	30	Mean	1.2868e−1	6.2357e−1	1.2262e−1	3.6849e−1	6.9534e−1	1.1676e−1
			Std	5.23e−2	1.60e−1	1.31e−2	8.76e−2	8.22e−2	3.55e−3
UF2	2	30	Mean	7.1399e−2	1.4311e−1	8.4490e−2	9.9507e−2	9.5654e−2	8.5598e−2
			Std	7.99e−3	1.65e−2	6.24e−3	9.97e−3	7.42e−3	4.40e−3
UF3	2	30	Mean	3.8726e−1	4.1694e−1	3.4596e−1	4.5627e−1	3.3170e−1	3.0490e−1
			Std	5.31e−2	9.99e−2	5.37e−2	3.57e−2	7.98e−3	5.65e−3
UF4	2	30	Mean	5.8074e−2	7.8401e−2	6.2087e−2	1.1174e−1	1.3797e−1	5.5194e−2
			Std	2.98−3	8.73e−3	8.79e−3	1.03e−2	4.95e−3	1.10e−3
UF5	2	30	Mean	1.6462e+0	3.9357e+0	1.5447e+0	2.9202e+0	3.2144e+0	1.0688e+0
			Std	3.66e−1	4.16e−1	4.54e−1	4.91e−1	3.37e−1	1.77e−1
UF6	2	30	Mean	6.0221e−1	2.9698e+0	6.8682e−1	1.3941e+0	2.3115e+0	4.8692e−1
			Std	1.33e−1	5.68e−1	1.76e−1	4.20e−1	5.22e−1	1.93e−2
UF7	2	30	Mean	1.7088e−1	6.0732e−1	1.7469e−1	3.7920e−1	3.3824e−1	8.8484e−2
			Std	1.22e−1	1.15e−1	1.24e−1	1.20e−1	6.53e−2	8.13e−3
UF8	3	30	Mean	2.7928e−1	7.5427e−1	4.9245e−1	4.0503e−1	3.4859e−1	2.5018e−1
			Std	1.01e−2	1.81e−1	8.12e−2	6.29e−2	3.86e−2	2.71e−2
UF9	3	30	Mean	4.3548e−1	8.9333e−1	4.6979e−1	5.6094e−1	5.8263e−1	1.3962e−1
			Std	4.12e−2	1.68e−1	5.75e−2	4.60e−2	4.65e−2	1.05e−2
UF10	3	30	Mean	1.2203e+0	5.1165e+0	1.6161e+0	2.7528e+0	9.4059e−1	2.6067e+0
			Std	3.29e−1	5.49e−1	3.72e−1	3.42e−1	1.92e−2	2.53e−1
DTLZ1	3	7	Mean	3.0510e+0	1.9600e+1	5.8179e+0	3.9067e+0	1.1684e+1	2.0491e+1
			Std	2.52e+0	5.26e+0	2.83e+0	3.56e+0	7.93e+0	2.40e+0
DTLZ2	3	12	Mean	5.5185e−2	9.2590e−2	5.7068e−2	6.4321e−2	1.0524e−1	5.2582e−2
			Std	1.80e−3	1.05e−2	1.59e−3	4.28e−3	5.24e−3	1.75e−3
DTLZ3	3	12	Mean	9.6447e+1	1.1974e+2	1.1605e+2	4.6451e+1	9.5917e+1	1.6858e+2
			Std	2.60e+1	3.77e+1	2.05e+1	4.17e+1	6.18e+1	3.05e+1
DTLZ4	3	12	Mean	5.1636e−2	3.1887e−1	9.1581e−2	3.9089e−1	3.2648e−1	1.7403e−1
			Std	1.34e−3	3.79e−2	1.22e−1	1.35e−1	3.47e−2	2.73e−2
DTLZ5	3	12	Mean	3.6868e−3	3.5864e−2	7.0440e−3	3.6616e−3	2.5766e−2	5.3804e−3
			Std	2.87e−4	1.03e−2	7.61e−4	4.17e−4	3.90e−3	5.73e−4
DTLZ6	3	12	Mean	3.3295e−3	1.2781e−1	1.2465e−2	1.2067e+0	2.0379e−2	4.4021e−3
			Std	5.09e−4	2.87e−1	2.41e−3	1.00e+0	4.47e−4	1.03e−3
DTLZ7	3	22	Mean	1.5873e−1	7.5281e−2	6.6849e−2	7.6062e−1	1.2158e−1	4.3578e−2
			Std	2.33e−1	1.78e−2	3.59e−2	2.98e−1	3.80e−2	1.15e−3
				5/22	0/22	0/22	3/22	2/22	12/22

Table 3 HV values of six MOPSOs on three sets of test problems

Problem	M	D		MMOPSO	MOPSOCD	NMPSO	SMPSO	dMOPSO	FAMOPSO
ZDT1	2	30	Mean	7.2152e-1	7.2098e-1	6.8912e-1	6.2638e-1	6.6027e-1	7.2163e-1
			Std	2.13e-4	1.21e-3	1.27e-2	1.04e-1	1.67e-2	2.48e-4
ZDT2	2	30	Mean	3.0239e-1	3.5891e-1	4.3373e-1	3.6652e-1	3.7744e-1	4.4559e-1
			Std	1.72e-1	1.42e-1	1.34e-2	1.10e-1	3.16e-2	3.55e-4
ZDT3	2	30	Mean	5.9827e-1	5.8591e-1	5.7102e-1	5.4996e-1	5.9443e-1	5.9905e-1
			Std	9.69e-3	2.77e-2	7.16e-3	6.42e-2	1.86e-2	4.23e-4
ZDT4	2	10	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	5.0389e-2	0.0000e+0
			Std	0.00e+0	0.00e+0	0.00e+0	0.00e+0	1.05e-1	0.00e+0
ZDT6	2	10	Mean	3.8987e-1	3.8905e-1	3.8978e-1	3.9001e-1	3.8526e-1	3.8953e-1
			Std	8.88e-5	1.22e-3	2.06e-4	8.82e-5	1.30e-2	1.21e-4
UF1	2	30	Mean	5.3676e-1	9.2488e-2	5.3366e-1	2.7269e-1	4.6742e-2	5.4495e-1
			Std	5.29e-2	9.05e-2	2.34e-2	7.64e-2	4.14e-2	6.06e-3
UF2	2	30	Mean	6.3955e-1	5.3943e-1	6.1819e-1	6.0895e-1	6.1614e-1	6.1713e-1
			Std	6.69e-3	2.07e-2	6.27e-3	8.85e-3	5.84e-3	4.82e-3
UF3	2	30	Mean	2.5904e-1	2.2645e-1	2.8723e-1	2.0610e-1	3.0710e-1	3.3648e-1
			Std	4.26e-2	7.63e-2	4.79e-2	2.95e-2	9.14e-3	4.81e-3
UF4	2	30	Mean	3.6671e-1	3.3391e-1	3.6166e-1	2.9204e-1	2.5291e-1	3.6790e-1
			Std	3.92e-3	1.08e-2	1.28e-2	1.19e-2	6.09e-3	1.82e-3
UF5	2	30	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	1.0807e-3
			Std	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	4.18e-3
UF6	2	30	Mean	5.5921e-2	0.0000e+0	2.2070e-2	8.5820e-4	0.0000e+0	4.3732e-2
			Std	5.88e-2	0.00e+0	3.27e-2	3.69e-3	0.00e+0	7.63e-3
UF7	2	30	Mean	3.9930e-1	2.6253e-2	3.9036e-1	1.8057e-1	2.0807e-1	4.4982e-1
			Std	8.56e-2	3.79e-2	9.47e-2	9.27e-2	4.97e-2	1.20e-2
UF8	3	30	Mean	2.8012e-1	1.0132e-2	2.7633e-1	1.6277e-1	2.6800e-1	2.8351e-1
			Std	1.90e-2	1.28e-2	6.28e-2	5.88e-2	1.87e-2	1.31e-2
UF9	3	30	Mean	3.3282e-1	3.0344e-2	3.1197e-1	2.0677e-1	2.2680e-1	6.0437e-1
			Std	4.02e-2	3.39e-2	5.18e-2	4.81e-2	2.27e-2	1.48e-2
UF10	3	30	Mean	4.0344e-4	0.0000e+0	0.0000e+0	0.0000e+0	9.1107e-2	0.0000e+0
			Std	1.61e-3	0.00e+0	0.00e+0	0.00e+0	9.11e-4	0.00e+0
DTLZ1	3	7	Mean	4.6282e-3	0.0000e+0	0.0000e+0	8.7218e-2	7.5171e-3	0.0000e+0
			Std	1.82e-2	0.00e+0	0.00e+0	1.79e-1	4.12e-2	0.00e+0
DTLZ2	3	12	Mean	5.3858e-1	4.7859e-1	5.6875e-1	5.0633e-1	4.3708e-1	5.5118e-1
			Std	3.44e-3	1.87e-2	9.49e-4	9.32e-3	8.68e-3	1.87e-3
DTLZ3	3	12	Mean	0.0000e+0	0.0000e+0	0.0000e+0	1.8692e-2	0.0000e+0	0.0000e+0
			Std	0.00e+0	0.00e+0	0.00e+0	4.60e-2	0.00e+0	0.00e+0
DTLZ4	3	12	Mean	5.4716e-1	2.8241e-1	5.5358e-1	3.3133e-1	2.8513e-1	4.8668e-1
			Std	3.17e-3	5.10e-2	5.63e-2	8.75e-2	3.75e-2	1.87e-2
DTLZ5	3	12	Mean	2.0007e-1	1.7164e-1	1.9818e-1	1.9978e-1	1.7367e-1	1.9861e-1
			Std	1.81e-4	9.56e-3	3.83e-4	2.92e-4	5.55e-3	3.95e-4
DTLZ6	3	12	Mean	2.0105e-1	1.7311e-1	1.9804e-1	5.6456e-2	1.9007e-1	2.0078e-1
			Std	6.55e-5	5.43e-2	6.40e-4	8.08e-2	4.46e-4	2.95e-4
DTLZ7	3	22	Mean	2.6611e-1	2.6843e-1	2.7842e-1	9.3087e-2	2.4349e-1	2.8109e-1
			Std	2.39e-2	7.06e-3	5.81e-3	7.43e-2	5.43e-3	1.27e-4
				4/22	0/22	2/22	3/22	2/22	11/22

Table 4 IGD values for six MOEAs on three sets of test problems

Problem	M	D		MOEADD	NSGAIII	SPEAR	CMOEAD	IMMOEA	FAMOPSO
ZDT1	2	30	Mean	8.6326e−2	8.2582e−2	1.9690e−1	7.6815e−2	1.7796e−1	2.8246e−3
			Std	1.72e−2	1.60e−2	3.18e−2	2.03e−2	2.15e−2	9.23e−5
ZDT2	2	30	Mean	1.0432e−1	1.7325e−1	3.9734e−1	1.0194e−1	1.9158e−1	3.5633e−3
			Std	1.99e−2	9.95e−2	1.03e−1	3.24e−2	4.09e−2	4.83e−4
ZDT3	2	30	Mean	1.5886e−1	6.9247e−2	1.4876e−1	7.6479e−2	2.1767e−1	2.8368e−3
			Std	2.30e−2	1.32e−2	1.64e−2	2.25e−2	2.92e−2	4.43e−4
ZDT4	2	10	Mean	6.1773e−1	2.1725e+0	2.0714e+0	5.1103e−1	3.2341e−2	1.7915e+1
			Std	1.84e−1	8.03e−1	8.69e−1	1.20e−1	4.87e−2	5.27e+0
ZDT6	2	10	Mean	3.8204e−1	1.2808e+0	1.1861e+0	3.3455e−1	1.6430e+0	2.3143e−3
			Std	1.12e−1	2.01e−1	2.32e−1	7.25e−2	8.65e−2	3.90e−4
UF1	2	30	Mean	1.3344e−1	1.1385e−1	1.3509e−1	1.8121e−1	2.7483e−1	1.1676e−1
			Std	2.89e−2	2.25e−2	1.98e−2	5.78e−2	5.28e−2	3.55e−3
UF2	2	30	Mean	6.2640e−2	7.6538e−2	7.0205e−2	1.2438e−1	7.0939e−2	8.5598e−2
			Std	6.39e−3	9.35e−3	6.79e−3	4.92e−2	8.41e−3	4.40e−3
UF3	2	30	Mean	4.5366e−1	4.6436e−1	4.3317e−1	3.8479e−1	3.7964e−1	3.0490e−1
			Std	1.28e−2	1.05e−2	1.29e−2	2.90e−2	3.22e−2	5.65e−3
UF4	2	30	Mean	8.6800e−2	8.9096e−2	8.4889e−2	9.6218e−2	1.2938e−1	5.5194e−2
			Std	3.03e−3	2.29e−3	1.86e−3	3.90e−3	2.87e−3	1.10e−3
UF5	2	30	Mean	1.0421e+0	1.2058e+0	1.1569e+0	1.0071e+0	1.9081e+0	1.0688e+0
			Std	1.65e−1	2.47e−1	2.29e−1	1.95e−1	2.72e−1	1.77e−1
UF6	2	30	Mean	6.6902e−1	6.0557e−1	6.7217e−1	6.0059e−1	1.2405e+0	4.8692e−1
			Std	8.28e−2	1.02e−1	7.76e−2	8.56e−2	1.87e−1	1.93e−2
UF7	2	30	Mean	1.4216e−1	1.5980e−1	1.8194e−1	2.8119e−1	3.5026e−1	8.8484e−2
			Std	6.26e−2	9.22e−2	6.40e−2	1.27e−1	4.76e−2	8.13e−3
UF8	3	30	Mean	3.0516e−1	3.7205e−1	3.1975e−1	3.3829e−1	2.7639e−1	2.5018e−1
			Std	1.76e−2	3.83e−2	1.70e−2	2.63e−2	9.64e−3	2.71e−2
UF9	3	30	Mean	4.8055e−1	4.7151e−1	4.8668e−1	5.5770e−1	5.1450e−1	1.3962e−1
			Std	7.82e−2	5.39e−2	6.49e−2	9.26e−2	3.78e−2	1.05e−2
UF10	3	30	Mean	2.9712e+0	1.5924e+0	1.9042e+0	1.0999e+0	1.2679e+0	2.6067e+0
			Std	6.43e−1	3.36e−1	3.18e−1	2.06e−1	3.23e−1	2.53e−1
DTLZ1	3	7	Mean	1.1353e+0	8.1668e−1	9.8204e−1	2.3438e−1	8.2161e+0	2.0491e+1
			Std	2.85e−1	3.02e−1	3.72e−1	1.71e−1	2.23e+0	2.40e+0
DTLZ2	3	12	Mean	3.9015e−2	3.9584e−2	4.3404e−2	4.0342e−2	7.7760e−2	5.2582e−2
			Std	4.71e−4	6.64e−4	1.64e−3	8.78e−4	3.37e−3	1.75e−3
DTLZ3	3	12	Mean	3.8962e+1	2.6228e+1	3.4797e+1	1.4918e+1	1.1483e+2	1.6858e+2
			Std	1.06e+1	6.17e+0	1.05e+1	6.21e+0	1.80e+1	3.05e+1
DTLZ4	3	12	Mean	3.9236e−2	4.0305e−2	4.4172e−2	7.5572e−2	1.5872e−1	1.7403e−1
			Std	5.70e−4	6.90e−4	1.69e−3	1.27e−1	3.48e−2	2.73e−2
DTLZ5	3	12	Mean	2.4304e−2	7.0339e−3	2.3258e−2	1.9334e−2	1.6755e−2	5.3804e−3
			Std	1.96e−3	6.50e−4	2.24e−3	6.82e−4	2.14e−3	5.73e−4
DTLZ6	3	12	Mean	1.3592e−1	2.1161e−2	1.4970e−1	4.8575e−2	5.9566e+0	4.4021e−3
			Std	1.34e−1	1.15e−2	1.26e−1	1.56e−1	2.17e−1	1.03e−3
DTLZ7	3	22	Mean	3.4179e−1	1.2798e−1	2.0136e−1	1.2450e−1	2.4539e−1	4.3578e−2
			Std	2.71e−1	1.97e−2	4.60e−2	3.83e−2	2.71e−2	1.15e−3
				4/22	2/22	0/22	3/22	1/22	13/22

these algorithms exhibit certain strengths, only FAMOPSO is capable of demonstrating consistent efficacy in addressing the majority of test problems, thereby substantiating the soundness and practicality of its design.

Table 3 provides a comprehensive illustration of the comparative outcomes between FAMOPSO and the other five MOPSOs algorithms in relation to HV metrics. As with the IGD comparison results, FAMOPSO also demonstrates

Table 5 HV values for the six MOEAs on the three sets of test problems

Problem	M	D		MOEADD	NSGAIII	SPEAR	CMOEAD	IMMOEA	FAMOPSO
ZDT1	2	30	Mean	6.0381e-1	6.1264e-1	4.7437e-1	6.2433e-1	5.0439e-1	7.2163e-1
			Std	2.24e-2	2.04e-2	3.11e-2	2.18e-2	2.33e-2	2.48e-4
ZDT2	2	30	Mean	3.0043e-1	2.4228e-1	7.4807e-2	3.0137e-1	2.0907e-1	4.4559e-1
			Std	2.44e-2	5.38e-2	4.65e-2	3.42e-2	3.70e-2	3.55e-4
ZDT3	2	30	Mean	4.8664e-1	5.5508e-1	5.0524e-1	5.4943e-1	4.5767e-1	5.9905e-1
			Std	2.25e-2	9.32e-3	2.16e-2	1.50e-2	2.46e-2	4.23e-4
ZDT4	2	10	Mean	1.0775e-1	2.4592e-3	0.0000e+0	1.4078e-1	6.7910e-1	0.0000e+0
			Std	7.54e-2	1.35e-2	0.00e+0	7.33e-2	6.19e-2	0.00e+0
ZDT6	2	10	Mean	6.9722e-2	0.0000e+0	0.0000e+0	7.3508e-2	0.0000e+0	3.8953e-1
			Std	4.00e-2	0.00e+0	0.00e+0	2.84e-2	0.00e+0	1.21e-4
UF1	2	30	Mean	5.2151e-1	5.4405e-1	5.1851e-1	5.0133e-1	3.5780e-1	5.4495e-1
			Std	3.57e-2	3.61e-2	2.75e-2	4.16e-2	5.25e-2	6.06e-3
UF2	2	30	Mean	6.3771e-1	6.2660e-1	6.2938e-1	6.0768e-1	6.3734e-1	6.1713e-1
			Std	8.12e-3	6.49e-3	7.51e-3	2.70e-2	7.60e-3	4.82e-3
UF3	2	30	Mean	2.1149e-1	1.9477e-1	2.2408e-1	2.4047e-1	2.6657e-1	3.3648e-1
			Std	1.11e-2	9.11e-3	1.49e-2	2.79e-2	2.65e-2	4.81e-3
UF4	2	30	Mean	3.2392e-1	3.2206e-1	3.2525e-1	3.0891e-1	2.6745e-1	3.6790e-1
			Std	3.81e-3	3.54e-3	2.02e-3	4.48e-3	3.65e-3	1.82e-3
UF5	2	30	Mean	0.0000e+0	1.4802e-4	9.4576e-5	4.5574e-3	0.0000e+0	1.0807e-3
			Std	0.00e+0	8.11e-4	5.18e-4	1.19e-2	0.00e+0	4.18e-3
UF6	2	30	Mean	1.7339e-2	2.2184e-2	1.7606e-2	2.1557e-2	0.0000e+0	4.3732e-2
			Std	1.73e-2	2.14e-2	1.62e-2	2.48e-2	0.00e+0	7.63e-3
UF7	2	30	Mean	3.9243e-1	3.7938e-1	3.5298e-1	3.1574e-1	2.0032e-1	4.4982e-1
			Std	5.80e-2	8.03e-2	4.73e-2	7.72e-2	4.43e-2	1.20e-2
UF8	3	30	Mean	2.0983e-1	2.6146e-1	1.8391e-1	1.8959e-1	3.0431e-1	2.8351e-1
			Std	3.11e-2	3.63e-2	3.14e-2	3.97e-2	6.11e-3	1.31e-2
UF9	3	30	Mean	2.6195e-1	2.8700e-1	2.5690e-1	2.6460e-1	2.5522e-1	6.0437e-1
			Std	6.85e-2	4.33e-2	6.33e-2	5.50e-2	3.08e-2	1.48e-2
UF10	3	30	Mean	0.0000e+0	0.0000e+0	0.0000e+0	1.0739e-4	1.8886e-4	0.0000e+0
			Std	0.00e+0	0.00e+0	0.00e+0	5.88e-4	1.03e-3	0.00e+0
DTLZ1	3	7	Mean	0.0000e+0	1.0455e-3	7.9350e-3	3.4245e-1	0.0000e+0	0.0000e+0
			Std	0.00e+0	4.75e-3	3.88e-2	2.79e-1	0.00e+0	0.00e+0
DTLZ2	3	12	Mean	5.6353e-1	5.6275e-1	5.5587e-1	5.5976e-1	4.9286e-1	5.5118e-1
			Std	1.23e-3	1.64e-3	2.66e-3	1.97e-3	5.93e-3	1.87e-3
DTLZ3	3	12	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
			Std	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
DTLZ4	3	12	Mean	5.6356e-1	5.6177e-1	5.5449e-1	5.4597e-1	4.5227e-1	4.8668e-1
			Std	1.74e-3	1.80e-3	3.26e-3	5.51e-2	2.06e-2	1.87e-2
DTLZ5	3	12	Mean	1.8296e-1	1.9667e-1	1.8527e-1	1.8909e-1	1.8574e-1	1.9861e-1
			Std	2.11e-3	5.54e-4	2.07e-3	1.02e-3	2.31e-3	3.95e-4
DTLZ6	3	12	Mean	1.0062e-1	1.8650e-1	1.0788e-1	1.7759e-1	0.0000e+0	2.0078e-1
			Std	4.55e-2	9.39e-3	4.31e-2	3.44e-2	0.00e+0	2.95e-4
DTLZ7	3	22	Mean	2.1799e-1	2.2672e-1	1.9803e-1	2.2205e-1	1.8294e-1	2.8109e-1
			Std	1.95e-2	7.94e-3	1.52e-2	2.79e-2	1.06e-2	1.27e-4
				3/22	1/22	0/22	2/22	3/22	13/22

excellent performance in the majority of the 22 benchmark tests, significantly outperforming the other algorithms that were compared. The empirical results presented here demonstrate that the proposed FAMOPSO algorithm not only

exhibits superior convergence but also maintains solution diversity when solving MOPs.

It is important to remember that the FAMOPSO algorithm's advantage is a result of its special algorithmic

design, not chance. To find a better solution during the search process, FAMOPSO incorporates an update strategy for the fireworks algorithm, which successfully balances the algorithm's exploratory and exploitative capabilities. Convergence speed and algorithm stability are further enhanced by the addition of the size-double archiving mechanism, which guarantees that the algorithm can keep the best-performing population updated to inform iterations. Furthermore, FAMOPSO uses the Active leader strategy, which greatly enhances the algorithm's search performance and strengthens it when handling different MOPs. The innovative designs of the FAMOPSO demonstrate a high level of competitiveness when compared to the selected algorithms for MOPSOs.

Comparison with five existing MOEAs: this paper presents a comparative analysis of the performance of MOEADD, NSGAIII, SPEAR, CMOEAD, IMMOEA, and the FAMOPSO algorithm on a series of standardized test problems, including ZDT1-ZDT4, ZDT6, UF1-UF10, and DTLZ1-DTLZ7, with respect to the IGD and HV metrics. The specific data are detailed in Tables 4 and 5, which list the standard deviation and mean values of each algorithm in detail.

We discover that FAMOPSO exhibits notable benefits in the 22 benchmarking tasks by examining the IGD value data in Table 4. More than half of the test issues yield ideal outcomes for FAMOPSO, which completely proves its superior performance. On the majority of the test issues, however, the other MOEAs in the comparison performed poorly. For instance, while NSGAIII did quite poorly on the ZDT series of test problems, it did well on the DTLZ series. Similar to this, MOEADD performed well on the DTLZ4 test problems but much worse on the DTLZ6 test problems. The comparison between FAMOPSO and the other five MOEAs algorithms on the HV metrics is presented in detail in Table 5. Comparing FAMOPSO against the other examined algorithms, it scores better in 13 out of 22 benchmark tests, which is similar to the IGD comparison results. These empirical findings clearly show that, when solving MOPs, the suggested FAMOPSO algorithm retains solution variety in addition to having superior convergence. The experimental results suggest that while each of these algorithms has some distinct advantages, only FAMOPSO consistently performs exceptionally well on the majority of the investigated issues, demonstrating the validity and broad applicability of its design.

4.5 Comparison of FAMOPSO with Broader MOEAs

To explore the performance advantages of FAMOPSO, we have carefully selected two widely recognized algorithms in the field of MOEAs as benchmarks for comparison: MOEA/D and NSGA-III. Through a multi-dimensional and

detailed comparison, FAMOPSO demonstrates its significant unique advantages. Below are the main advantages of FAMOPSO summarized from an exhaustive analysis of the three algorithms:

Strong global search capability: FAMOPSO—using the global search capability of PSO and the fbest generated by FA, it is able to quickly find the potential optimal solution region in the solution space. This global search ability makes FAMOPSO more efficient in dealing with complex MOPs.

MOEA/D: although it also has strong search capability, its decomposition-based strategy may limit the scope of global search in some cases. NSGA-III: emphasizes population diversity through the reference point mechanism, but performance metrics computation and diversity maintenance may be more time-consuming when dealing with hyper-dimensional objective optimization problems.

Fast convergence: FAMOPSO—due to the gbest filtered by PSO's unique information sharing mechanism and size double archiving mechanism, it is guaranteed that FAMOPSO usually converges to PF faster. MOEA/D: although the computational complexity is lower, it performs well on low-dimensional problems but may face the problem of slower convergence due to decreased selection pressure on high-dimensional problems. NSGA-III uses a reference point mechanism to maintain population diversity, but this approach may increase the computational burden and affect the convergence speed when dealing with super-dimensional objective optimization problems.

High flexibility: FAMOPSO further improves the performance and adaptability of the algorithm through Active particle collaboration mechanisms, adaptive parameters, and the introduction of mutation strategies inside the archive. MOEA/D: while also supporting the use in combination with other algorithms, the decomposition-based strategy may limit the flexibility of its application to some specific problems. NSGA-III: while suitable for handling high-dimensional objective optimization problems, it may require additional strategy tuning when dealing with unconstrained and constrained problems.

In summary, FAMOPSO demonstrates obvious advantages over MOEA/D and NSGA-III in terms of global search capability, convergence speed, and flexibility. These advantages make FAMOPSO more efficient and more widely applicable when dealing with complex MOPs.

4.6 Comparison of MOPSOs Box Plots

To comprehensively evaluate the stability performance of each algorithm on 22 test questions, we construct Fig. 5, which depicts in detail the box-type statistical chart of IGD values of FAMOPSO and five other MOPSOs in 30 independent runs. In this box diagram, the horizontal axis is labeled from 1 to 6 and represents the following algorithms:

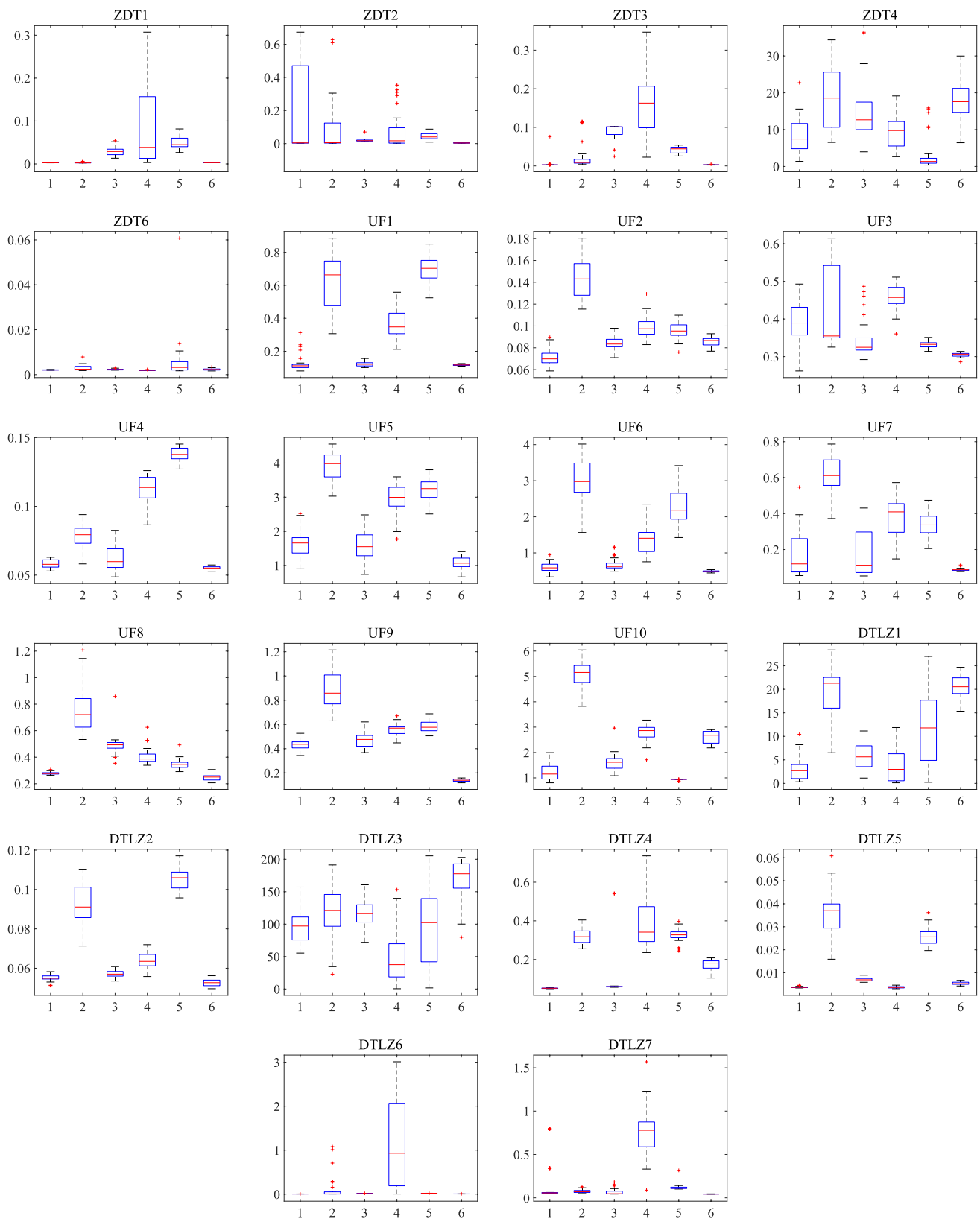


Fig. 5 Comparison of box plots of 6 MOPSOs

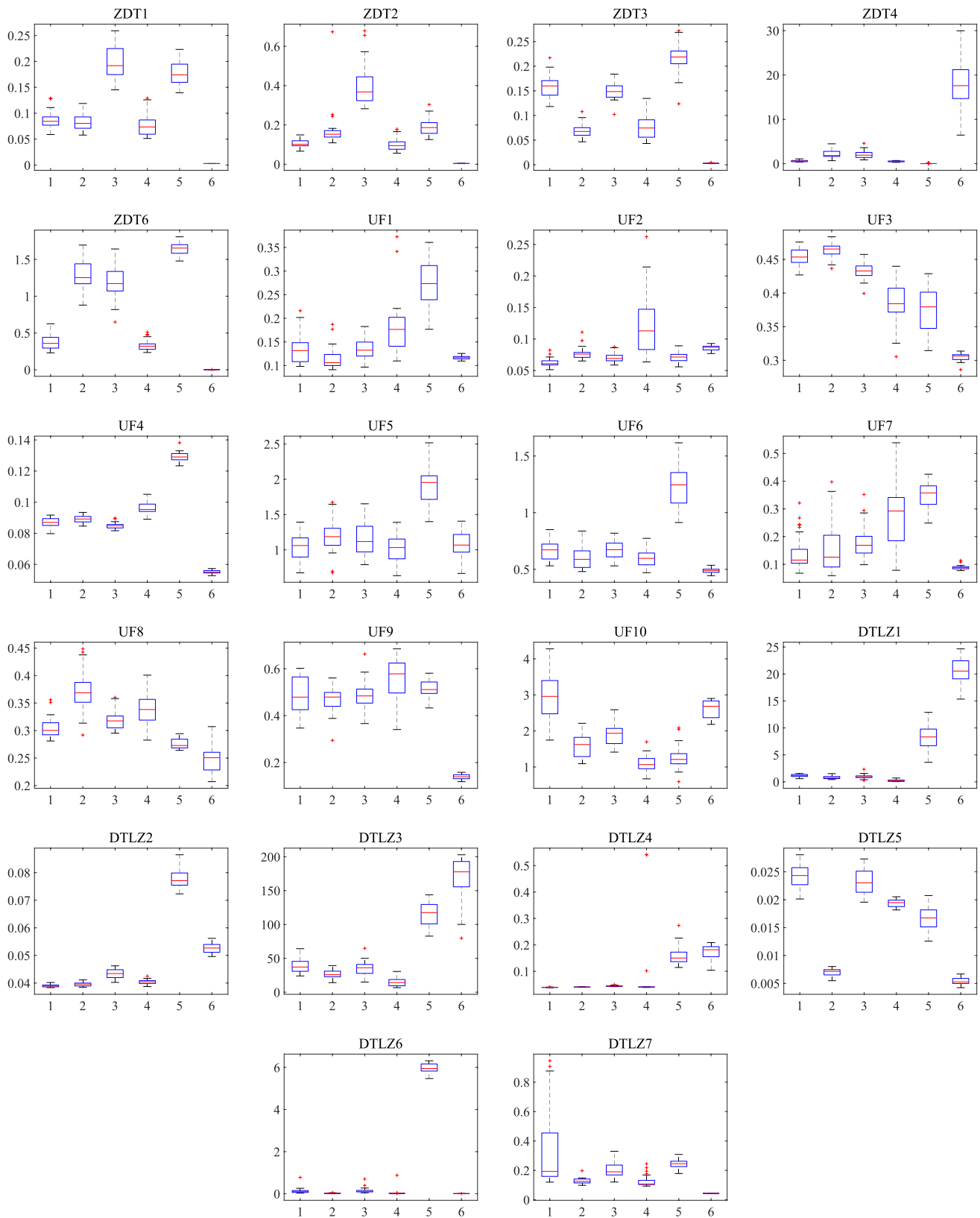


Fig. 6 Comparison of box plots of 6 MOEAs

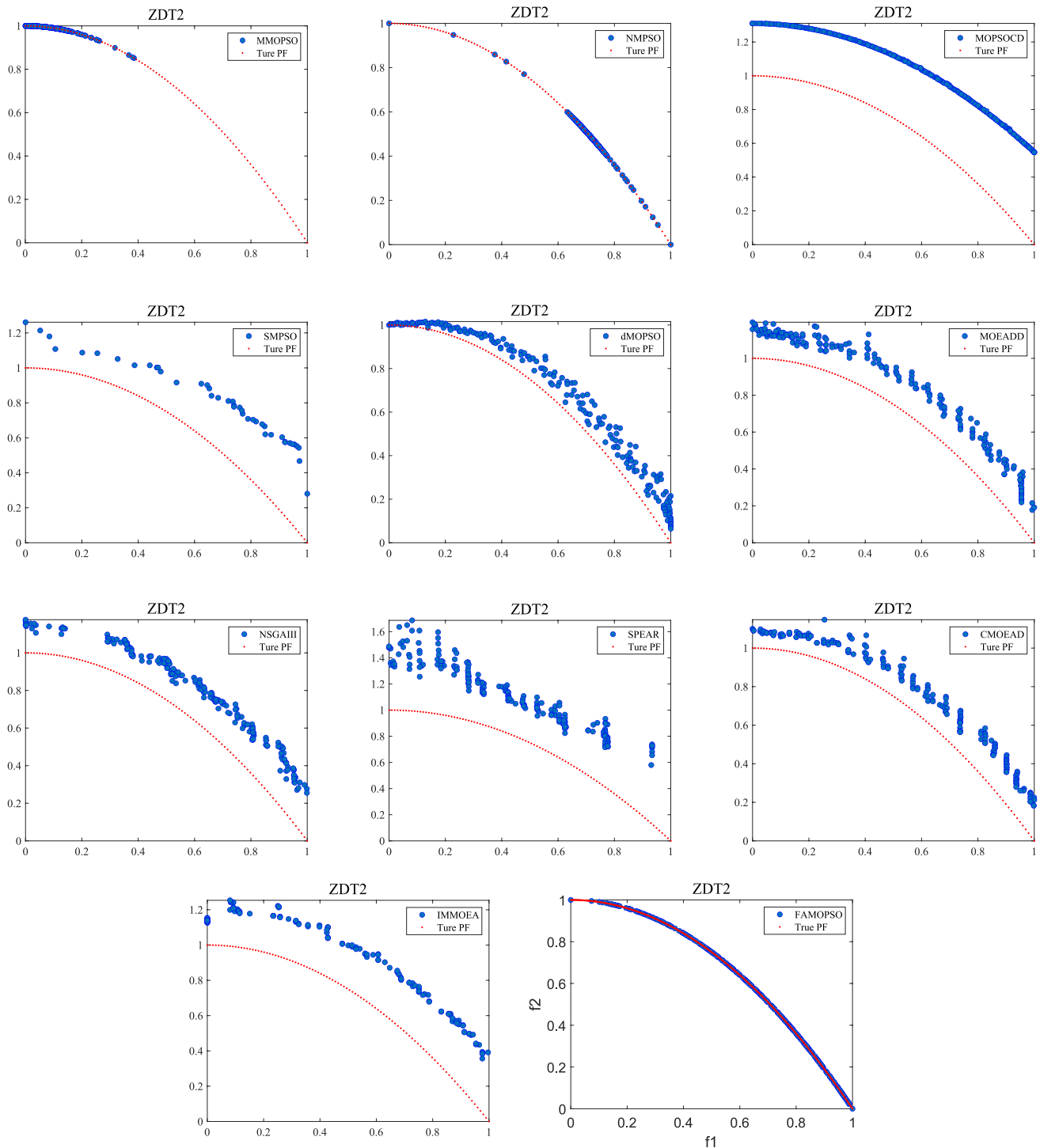


Fig. 7 Illustration of the IGD front for ZDT2

MMOPSO, MOPSOCD, NMPSO, SMPSO, dMOPSO, and FAMOPSO. The vertical axis shows the IGD values of the corresponding algorithms. As an effective method of data visualization, the box diagram is capable of clearly illustrating the distribution characteristics of the data in question. The box represents the range between the lower quartile

(Q1) and the upper quartile (Q3) of the IGD value. The plus symbol indicates an outlier in the data set. A smaller height of the box indicates a greater degree of data concentration, while a reduced number of outliers suggests enhanced algorithmic stability.

Figure 5 illustrates that the FAMOPSO box diagram is relatively flat on the majority of test problems, indicating a generally low IGD value. This suggests that FAMOPSO exhibits enhanced stability and superior performance in addressing MOPs. In comparison to the other five algorithms, FAMOPSO demonstrates a reduced number of outliers and a more stable data distribution, which serves to reinforce the conclusions presented in Table 2. Consequently, FAMOPSO is not only capable of achieving superior outcomes when solving MOPs, but it is also demonstrably more stable than the other five MOPSOs.

4.7 Comparison of MOEAs Box Plots

Figure 6 illustrates the performance of FAMOPSO and five additional MOEAs on 22 test questions. The statistical box diagram of the IGD value distribution obtained by 30 independent runs allows for an intuitive comparison of the stability of each algorithm. The abscissa of the figure represents the various algorithms, including MOEADD, NSGAIII, SPEAR, CMOEAD, IMMOEA, and FAMOPSO, while the ordinate depicts the IGD value of each algorithm.

As illustrated in Fig. 6, while the IGD values of FAMOPSO on ZDT4, DTLZ1, DTLZ3, and other test problems exhibit considerable volatility and are less stable than those of other algorithms, in the majority of test problems, the anomalous values of FAMOPSO's experimental data are scarce, the distribution is relatively flat, and the IGD values are typically minimal. This indicates that FAMOPSO exhibits superior comprehensive performance in addressing multi-objective optimization problems, particularly in terms of result quality and algorithm stability.

Further analysis demonstrates that FAMOPSO exhibits remarkable stability on the majority of test problems, primarily due to its distinctive algorithmic design. FAMOPSO incorporates the update strategy of the fireworks algorithm, which enhances the exploration capability. Concurrently, the large and small double archiving mechanisms ensure effective retention of the gbest with optimal performance and improve convergence speed and stability. Additionally, the Active leadership strategy enhances the search capability to address complex problems.

4.8 Comparison of IGD Frontier Maps for Each Algorithm

To provide a comprehensive assessment of the optimization performance of each algorithm, an exhaustive experimental analysis was conducted. In this process, a series of standard test problems, including ZDT2, ZDT3, UF9, and DTLZ7, were used to compare the performance of FAMOPSO with

that of ten state-of-the-art comparison algorithms. The distribution of the PF optimal solution sets for these problems is illustrated in Figs. 7, 8, 9, and 10. These graphs offer an intuitive representation of the convergence and diversity of the algorithms under evaluation. Examining Figs. 7, 8, 9, and 10 in detail reveals that FAMOPSO performs better on the ZDT2, ZDT3, UF9, and DTLZ7 problems than the majority of the algorithms used in the comparison. With regard to each test problem, FAMOPSO excels, especially at solving the dual-objective test problem ZDT2. Only MMOPSO and FAMOPSO can precisely locate the PF, as seen in Fig. 7. While MMOPSO can locate the PF as well, its distribution uniformity over the frontier is much less than FAMOPSO's. Similarly, FAMOPSO demonstrates its superior performance for the ZDT3 problem (Fig. 8), outperforming the other comparison algorithms in terms of localization and distribution accuracy at the PF. As demonstrated in Fig. 9, FAMOPSO fully demonstrates its advantages in convergence and diversity by exhibiting better distribution uniformity near the real PF on the UF9 problem, even in the face of the three-objective problem's complexity. The comparison's other algorithms exhibit shortcomings in both areas. Looking further at Fig. 10, we can see that FAMOPSO performs significantly better in terms of particle diversity, even though NMPSO and NSGAIII demonstrate good convergence on the more difficult DTLZ7 problem.

It is evident that FAMOPSO exhibits a more robust and pronounced performance advantage in comparison to alternative algorithms. This further substantiates the efficacy of the learning strategy proposed in this study, which not only ensures the uniform distribution of solutions but also guarantees convergence.

4.9 Comparison of IGD Indicator Convergence Rates

In addition to the aforementioned experimental analysis, another crucial index for evaluating the efficacy of the algorithm is its convergence speed. Figure 11 illustrates the convergence paths of the IGD values generated by all comparison algorithms following 10,000 evaluations by FAMOPSO on ZDT2, ZDT3, UF4, and DTLZ7. The aforementioned paths are classified into two distinct categories. The initial four graphs illustrate the IGD convergence paths of FAMOPSO and five MOPSOs, while the subsequent four graphs depict the IGD convergence paths of FAMOPSO and five MOEAs. As illustrated in the figure, FAMOPSO exhibits a markedly accelerated convergence rate, conferring a significant advantage in enhancing the algorithm's execution efficiency and calculation speed. A further examination of the final state of each algorithm in the process of convergence reveals that the IGD value of FAMOPSO is the lowest.

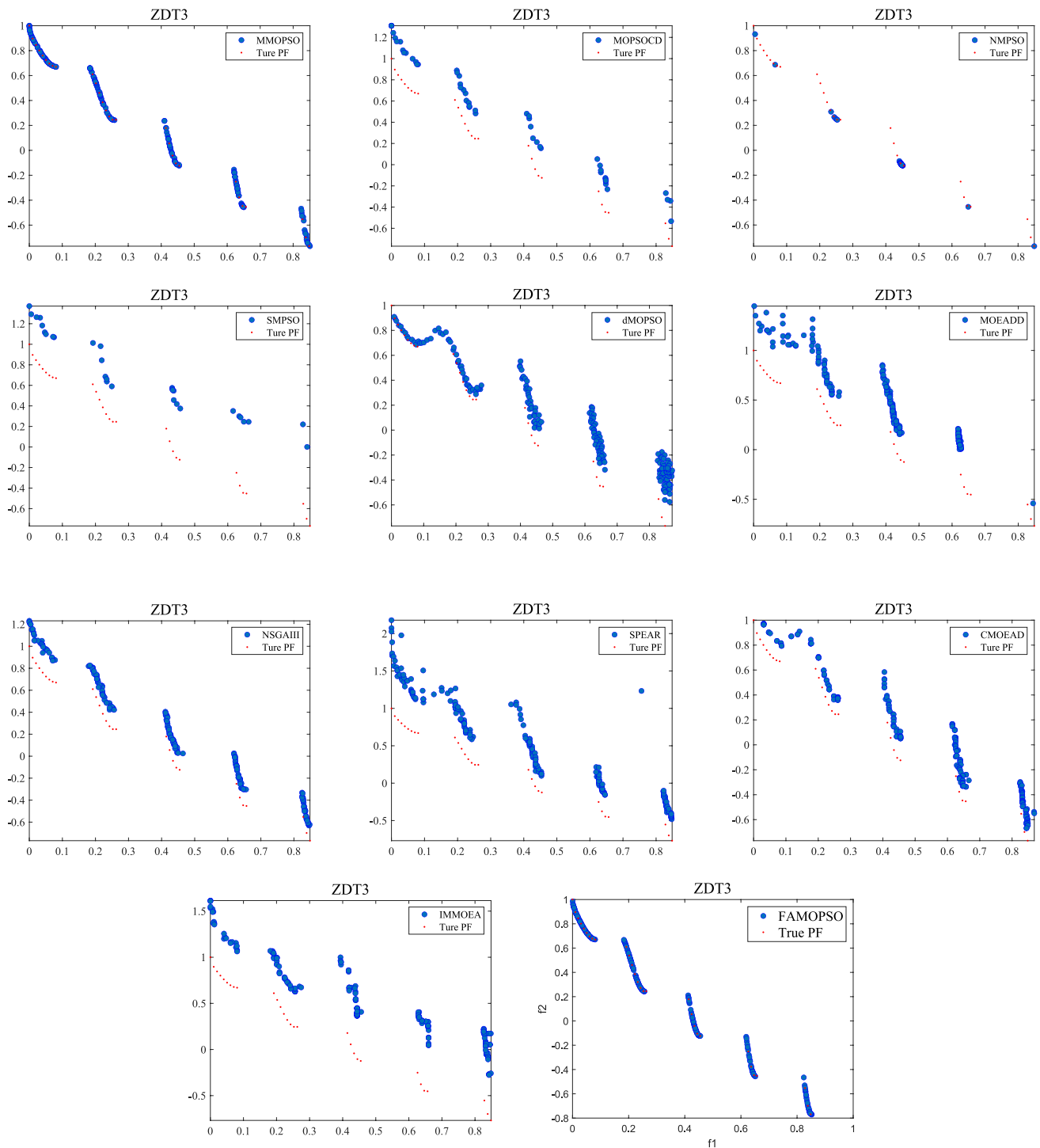


Fig. 8 Illustration of the IGD front for ZDT3

This not only underscores the notable enhancement in convergence performance of FAMOPSO but also highlights its exceptional capacity for diversity and uniform distribution. These results serve to further confirm the comprehensive

performance advantages of the learning strategy proposed by FAMOPSO in dealing with MOPs.

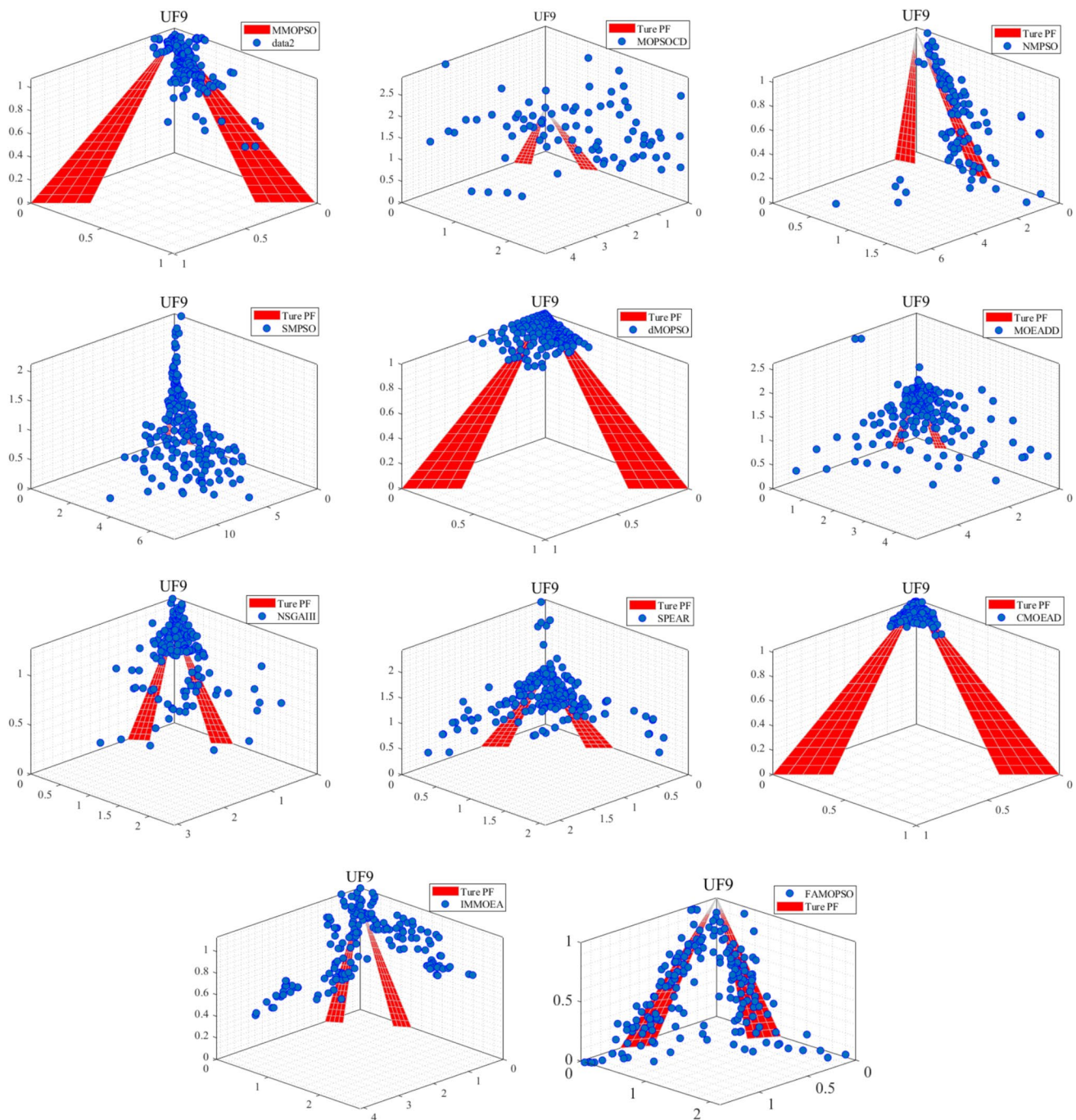


Fig. 9 Illustration of the IGD front for UF9

4.10 Analysis of the Effectiveness of the Algorithm

As mentioned earlier, FAMOPSO incorporates several strategies to enhance the performance of the original MOPSO. To rigorously validate the effectiveness of these strategies, we designed a series of ablation experiments aimed at evaluating the contribution of specific components to the

overall performance by systematically removing them. The core of the ablation experiments lies in the control variable approach, i.e., by investigating the change in the system's performance in the absence or weakening of a component, we can gain a deeper understanding of the component's role and value in the overall system.

Specifically, three experimental groups were set up in this study, and one key strategy was excluded from each

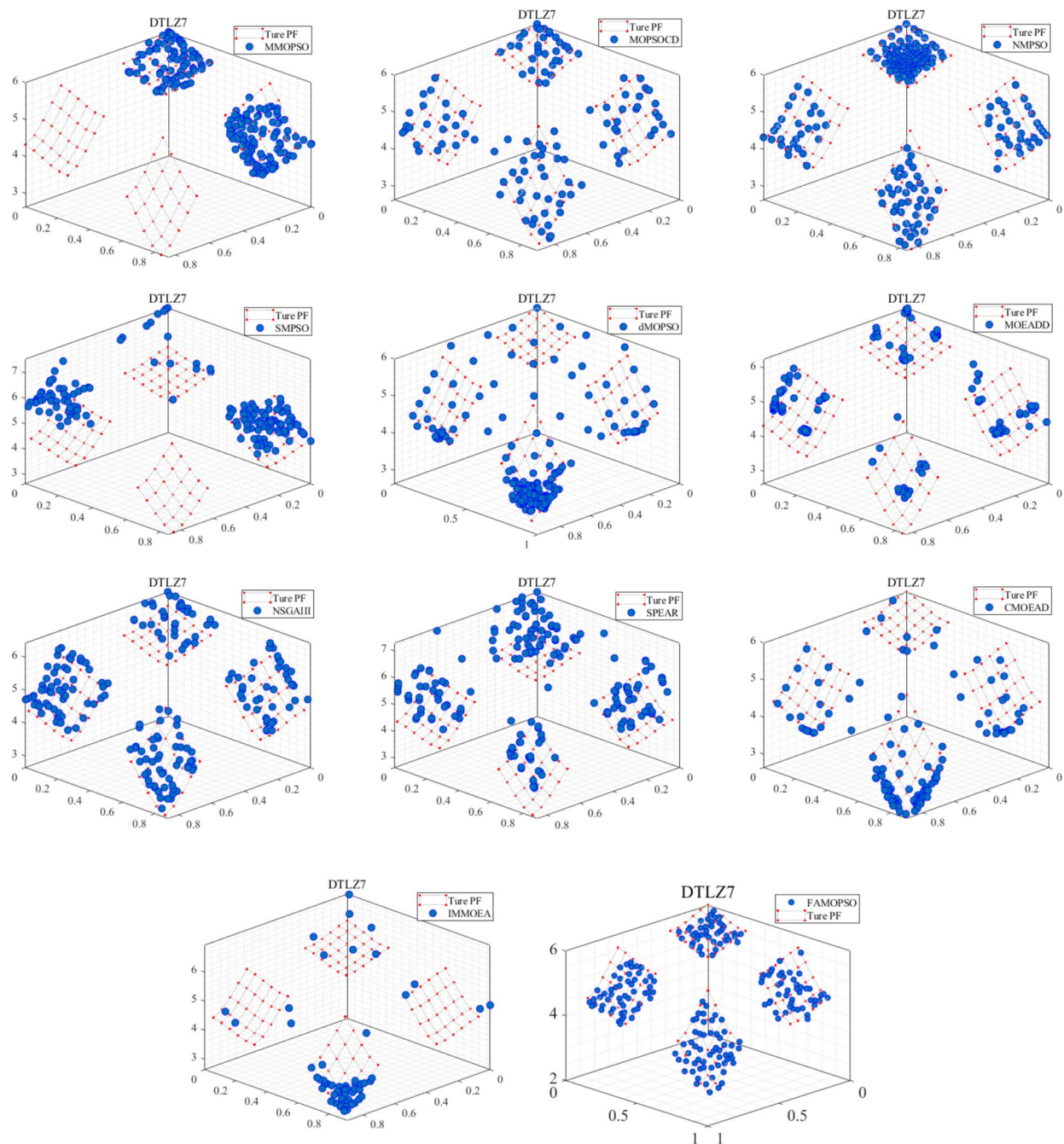


Fig. 10 Illustration of the IGD front for DTLZ7

group to highlight its importance: (1) FAMOPSO without the size-double archive strategy (FAMOPSO-23); (2) FAMOPSO without the fireworks explosion algorithm for generating fbest (FAMOPSO-13); and (3) FAMOPSO lacking adaptive parameter tuning and an information-based fusion strategy to generate Active particles for FAMOPSO (FAMOPSO-12). Meanwhile, the same parameter settings as in the above experiments were maintained, and the standard FAMOPSO was used as a control group.

All three variants of the algorithm and FAMOPSO itself were run independently on the ZDT, UF, and DTLZ test suites for 30 runs to obtain the statistical mean and standard deviation of the IGD and HV metrics. If the experimental group shows a significant performance dip compared to the control group, it is a clear indication that the excluded strategies play a key role in improving the performance of the algorithm. Detailed IGD vs. HV comparison results can

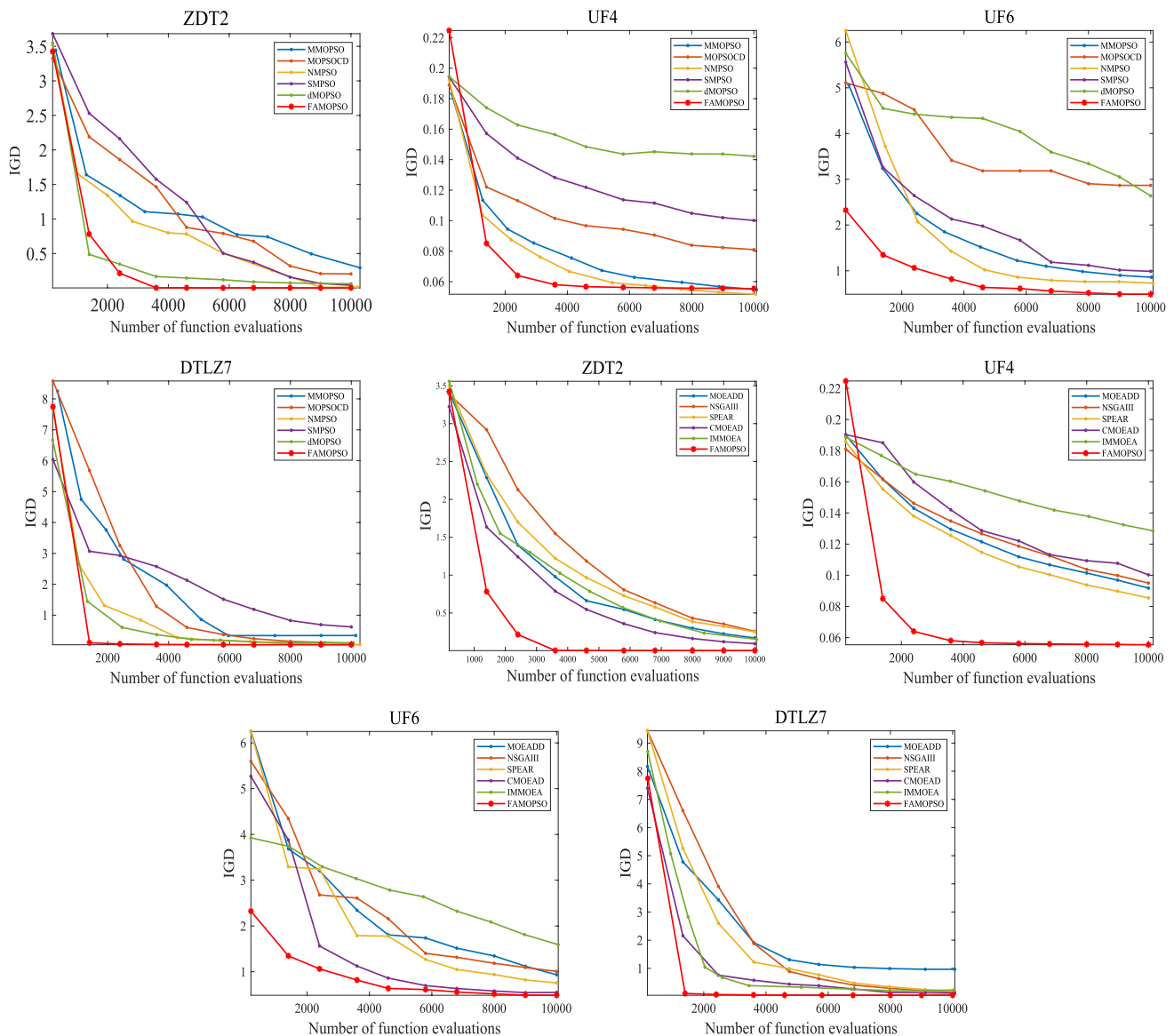


Fig. 11 IGD convergence trajectories for FAMOPSO and ten comparison algorithms on four test problems

be found in Tables 6 and 7, where the best results are highlighted in bold black.

From the data in Tables 6 and 7, it can be clearly observed that there is a significant difference in the performance of its variant algorithms compared to FAMOPSO. Specifically, FAMOPSO-23 obtained only one best IGD value on 22 test problems and failed to achieve the best HV value on any problem, while FAMOPSO-13 obtained one best IGD value and two best HV values, and FAMOPSO-12 performed slightly better, obtaining five best IGD values and one best HV value.

In contrast, FAMOPSO, which integrates multiple strategies, performs well on both metrics, obtaining a total of 15 best IGD values and 15 best HV values.

These results strongly demonstrate that the size-double archiving strategy, the firework explosion algorithm for generating fbest, adaptive parameter tuning, and the information fusion-based strategy for generating Active particles all play an integral role in enhancing the performance of the FAMOPSO algorithm. Through an exhaustive comparative analysis between the experimental and control groups, we validate the effectiveness of the proposed strategies and confirm that each of the improvement methods has a positive impact on the original MOPSO algorithm. This finding not only highlights the importance of the use of integrated strategies but also provides valuable insights and directions for the design of future MOEAs.

Table 6 Comparative results of IGD values for FAMOPSO and its variants

Problem		FAMOPSO-23 IGD	FAMOPSO-13 IGD	FAMOPSO-12 IGD	FAMOPSO IGD
ZDT1	Mean	5.8799e−3	3.0860e−3	2.7676e−3	2.8246e−3
	Std	5.43e−4	1.48e−4	1.55e−4	9.23e−5
ZDT2	Mean	5.9273e−3	4.4049e−2	3.6196e−3	3.5633e−3
	Std	5.93e−4	1.54e−1	5.19e−3	4.83e−4
ZDT3	Mean	1.0870e−2	5.4330e−3	2.6162e−3	2.8368e−3
	Std	2.25e−3	1.35e−3	3.04e−4	4.43e−4
ZDT4	Mean	2.0262e+1	8.4583e+0	2.2629e+1	1.7915e+1
	Std	5.83e+0	6.23e+0	3.77e+0	5.27e+0
ZDT6	Mean	2.6125e−3	2.7591e−3	2.6010e−3	2.3143e−3
	Std	2.65e−4	9.94e−4	6.00e−4	3.90e−4
UF1	Mean	1.1917e−1	1.1723e−1	1.5543e−1	1.1676e−1
	Std	5.90e−3	4.95e−3	1.27e−2	3.55e−3
UF2	Mean	8.6618e−2	1.1108e−1	1.0032e−1	8.5598e−2
	Std	4.76e−3	1.03e−2	8.29e−3	4.40e−3
UF3	Mean	3.0638e−1	3.4008e−1	3.1090e−1	3.0449e−1
	Std	7.40e−3	5.61e−2	7.59e−3	5.65e−3
UF4	Mean	6.5401e−2	7.1989e−2	5.4480e−2	5.5194e−2
	Std	5.85e−3	7.51e−3	1.12e−3	1.10e−3
UF5	Mean	1.2255e+0	1.3084e+0	1.6998e+0	1.0688e+0
	Std	2.34e−1	2.47e−1	1.79e−1	1.77e−1
UF6	Mean	5.0499e−1	5.1609e−1	6.4045e−1	4.8692e−1
	Std	3.31e−2	4.51e−2	6.81e−2	1.93e−2
UF7	Mean	9.1638e−2	9.0219e−2	1.3372e−1	8.848e−2
	Std	7.70e−3	8.79e−3	1.33e−2	8.13e−3
UF8	Mean	2.6384e−1	3.1670e−1	2.3857e−1	2.5018e−1
	Std	3.34e−2	2.96e−2	2.71e−2	2.71e−2
UF9	Mean	1.4677e−1	1.4792e−1	1.6174e−1	1.3962e−1
	Std	1.31e−2	1.10e−2	1.59e−2	1.05e−2
UF10	Mean	2.6319e+0	2.6078e+0	2.6860e+0	2.6067e+0
	Std	2.95e−1	3.82e−1	1.45e−1	2.53e−1
DTLZ1	Mean	1.9930e+0	2.0952e+1	2.0177e+1	2.0491e+1
	Std	3.01e+0	3.32e+0	3.25e+0	2.40e+0
DTLZ2	Mean	1.5344e−1	5.3547e−2	6.6020e−2	5.2582e−2
	Std	1.69e−2	3.32e+0	3.37e−3	1.75e−3
DTLZ3	Mean	1.8344e+2	1.7943e+2	1.5595e+2	1.6858e+2
	Std	2.67e+1	2.62e+1	2.62e+1	3.05e+1
DTLZ4	Mean	1.7817e−1	2.0170e−1	2.3016e−1	1.7403e−1
	Std	3.29e−2	3.60e−2	3.61e−2	2.73e−2
DTLZ5	Mean	2.3086e−2	5.6477e−3	1.5327e−2	5.3804e−3
	Std	7.81e−3	6.43e−4	2.37e−3	5.73e−4
DTLZ6	Mean	1.3333e−2	4.6380e−3	4.7559e−3	4.4021e−3
	Std	4.65e−3	1.17e−3	1.32e−3	1.03e−3
DTLZ7	Mean	1.6414e−1	4.9687e−2	4.4539e−2	4.3578e−2
	Std	3.74e−2	3.86e−3	1.92e−3	1.15e−3
Best/All		1	1	5	15

Table 7 Comparative results of HV values for FAMOPSO and its variants

Problem	IGD	FAMOPSO-23 HV	FAMOPSO-13 HV	FAMOPSO-12 HV	FAMOPSO HV
ZDT1	Mean	7.1556e-1	7.2060e-1	7.2149e-1	7.2163e-1
	Std	1.03e-3	2.36e-4	1.70e-4	2.48e-4
ZDT2	Mean	4.4216e-1	4.2173e-1	4.4597e-1	4.4559e-1
	Std	1.08e-1	8.99e-2	1.94e-1	3.55e-4
ZDT3	Mean	5.9389e-1	5.9726e-1	5.9875e-1	5.9905e-1
	Std	1.01e-3	1.60e-3	3.35e-4	4.23e-4
ZDT4	Mean	0.0000e+0	7.5238e-3	0.0000e+0	0.0000e+0
	Std	0.00e+0	4.12e-2	0.00e+0	0.00e+0
ZDT6	Mean	3.7939e-1	3.8950e-1	3.8955e-1	3.8953e-1
	Std	2.68e-3	2.04e-4	1.35e-4	1.21e-4
UF1	Mean	5.3636e-1	5.3838e-1	4.7943e-1	5.4495e-1
	Std	1.01e-2	1.02e-2	1.68e-2	6.06e-3
UF2	Mean	6.1597e-1	5.7409e-1	5.9678e-1	6.1713e-1
	Std	4.97e-3	1.62e-2	8.90e-3	4.82e-3
UF3	Mean	3.3505e-1	3.0160e-1	3.3030e-1	3.3648e-1
	Std	8.06e-3	4.68e-2	8.57e-3	4.81e-3
UF4	Mean	3.5438e-1	3.4424e-1	3.6760e-1	3.6790e-1
	Std	7.18e-3	1.02e-2	2.43e-3	1.82e-3
UF5	Mean	0.0000e+0	0.0000e+0	0.0000e+0	1.0807e-3
	Std	0.00e+0	0.00e+0	0.00e+0	4.18e-3
UF6	Mean	3.5474e-2	3.6327e-2	6.8209e-3	4.3732e-2
	Std	1.15e-2	1.82e-2	1.04e-2	7.63e-3
UF7	Mean	4.4619e-1	4.4779e-1	3.8362e-1	4.4982e-1
	Std	1.21e-2	1.29e-2	1.83e-2	1.20e-2
UF8	Mean	2.6789e-1	1.9281e-1	2.7062e-1	2.8351e-1
	Std	2.55e-2	2.15e-2	1.89e-2	1.31e-2
UF9	Mean	5.9759e-1	5.9395e-1	5.7772e-1	6.0437e-1
	Std	1.80e-2	1.65e-2	1.92e-2	1.48e-2
UF10	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
	Std	0.00e+0	0.00e+0	0.00e+0	0.00e+0
DTLZ1	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
	Std	0.00e+0	0.00e+0	0.00e+0	0.00e+0
DTLZ2	Mean	3.8836e-1	5.3658e-1	5.1301e-1	5.5118e-1
	Std	2.40e-2	3.48e-3	5.85e-3	1.87e-3
DTLZ3	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
	Std	0.00e+0	0.00e+0	0.00e+0	0.00e+0
DTLZ4	Mean	4.8425e-1	4.5759e-1	4.1713e-1	4.8668e-1
	Std	2.08e-2	2.34e-2	3.26e-2	1.87e-2
DTLZ5	Mean	1.7460e-1	1.9841e-1	1.8787e-1	1.9861e-1
	Std	1.03e-2	3.27e-4	2.10e-3	3.95e-4
DTLZ6	Mean	1.8388e-1	2.0082e-1	2.0071e-1	2.0078e-1
	Std	9.76e-3	2.12e-4	3.59e-4	2.95e-4
DTLZ7	Mean	2.3186e-3	2.7507e-1	2.8268e-1	2.8109e-1
	Std	1.78e-2	3.79e-3	7.24e-4	1.27e-4
Best/All		0	2	2	15

Table 8 Sensitivity analysis of parameter m in IGD indicators

	$m=60$	$m=80$	$m=100$	$m=120$	$m=140$
ZDT1	2.8497e-3	2.8692e-3	2.8246e-3	2.9073e-3	2.9077e-3
ZDT2	3.6060e-3	3.6885e-3	3.5633e-3	3.5415e-3	3.5718e-3
ZDT6	2.7779e-3	2.6528e-3	2.3143e-3	2.8429e-3	2.6586e-3
UF4	5.5681e-2	5.5768e-2	5.5194e-2	5.5906e-2	5.5546e-2
UF6	5.1852e-1	5.2528e-1	4.8692e-1	5.2202e-1	5.2097e-1
UF7	1.0266e-1	9.8236e-2	8.8484e-2	1.0004e-1	1.0041e-1
DTLZ6	5.3898e-3	4.5604e-3	4.4021e-3	5.7623e-3	4.4949e-3
DTLZ7	4.4966e-2	4.5496e-2	4.3578e-2	4.5478e-2	4.4707e-2
Best/all	0	0	7	1	0

Table 9 Sensitivity analysis of parameter W in IGD indicators

	$W=0.2$	$W=0.3$	$W=0.4$	$W=0.5$	$W=0.6$
ZDT1	3.0502e-3	2.9991e-3	2.8246e-3	2.8902e-3	2.9271e-3
ZDT2	3.6885e-3	3.7318e-3	3.5633e-3	3.7049e-3	3.6390e-3
ZDT6	2.1805e-3	2.3911e-3	2.3143e-3	2.7586e-3	2.6038e-3
UF4	5.5927e-2	5.5938e-2	5.5194e-2	5.5684e-2	5.6260e-2
UF6	4.9513e-2	4.8256e-2	4.8692e-2	5.1705e-2	5.2188e-2
UF7	8.7753e-2	8.8394e-2	8.8484e-2	1.0270e-1	9.8562e-2
DTLZ6	4.1304e-3	5.2204e-3	4.4021e-3	4.8685e-3	4.8833e-3
DTLZ7	4.7841e-2	4.6103e-2	4.3578e-2	4.4806e-2	4.5426e-2
Best/all	3	1	4	0	0

Table 10 Sensitivity analysis of parameter CI , $C2$ in IGD indicators

	$CI=C2=1$	$CI=C2=2$	$CI=C2=3$	$CI=C2=4$	$CI=C2=5$
ZDT1	3.0575e-3	2.8246e-3	2.9385e-3	2.6895e-3	2.4932e-3
ZDT2	1.2618e-1	3.5633e-3	3.7674e-3	3.6452e-3	3.0769e-3
ZDT6	3.1727e-2	2.3143e-3	2.1070e-3	2.1211e-3	2.1953e-3
UF4	5.9166e-2	5.5194e-2	5.5322e-2	5.5285e-2	5.6195e-2
UF6	5.0565e-1	4.8692e-1	7.0648e-1	9.6356e-1	1.0510e+0
UF7	8.8946e-2	8.8484e-2	1.4571e-1	2.2804e-1	2.4169e-1
DTLZ6	5.0301e-2	4.4021e-3	4.2919e-3	4.8606e-3	3.3390e-3
DTLZ7	5.1555e-1	4.3578e-2	4.6552e-2	4.5914e-2	4.6134e-2
Best/all	0	4	1	0	3

4.11 Parameter Sensitivity Analysis

The aim of this section is to explore in depth the effect of the key parameter m on the number of sparks mentioned in the paper and how the core parameters W , CI , and $C2$ in the adaptive inertia weights and learning factor formulations affect the performance mechanism of FAMOPSO. Specifically, the value of parameter m has a direct impact on the output effectiveness of the algorithm: if m is set too small, the algorithm will only generate a small number of fbest, which may lead to the algorithm converging to the local optimum prematurely and missing the opportunity to explore the globally optimal solution; on the contrary, if m is too large, it may lead to the redundancy of the search process, which

may in turn reduce the learning efficiency and convergence speed. Given that the inertia weights and learning factors in this paper are dynamically adjusted by an adaptive strategy, their intrinsic mechanisms are quite complex and difficult to be analyzed directly. Therefore, we focus on the basic parameters W , CI , and $C2$ in the formula of inertia weights and learning factors (i.e., Eqs. 21–23), and through in-depth analyses of their sensitivities, we aim to reveal the specific effects of these parameters on the performance of the algorithm. The reasonable configuration of these parameters is of vital importance to enhance the overall performance of FAMOPSO. In this study, a systematic sensitivity analysis of the above four parameters will be carried out to provide solid

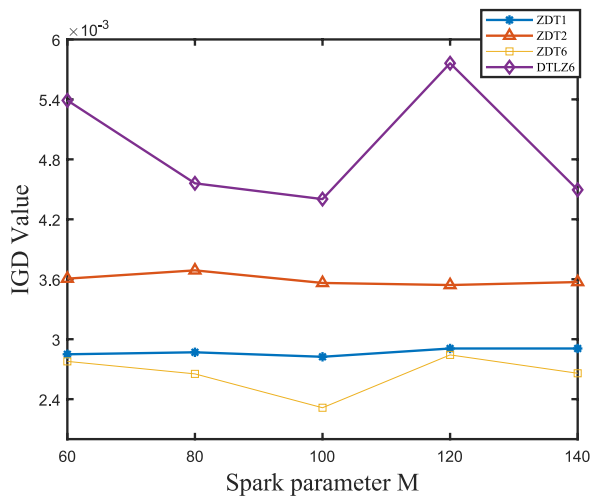


Fig. 12 IGD line graph for different values of m

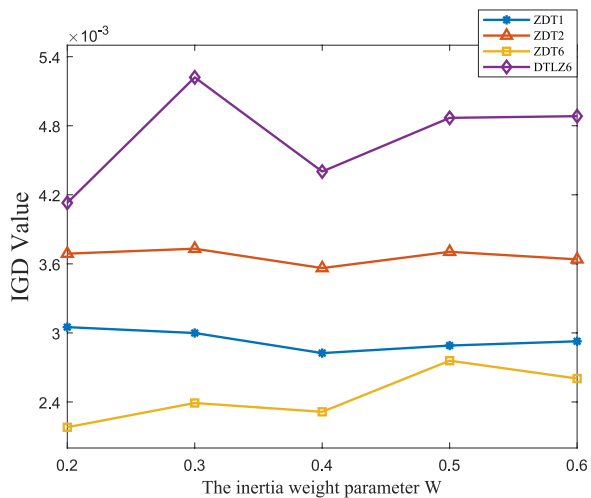
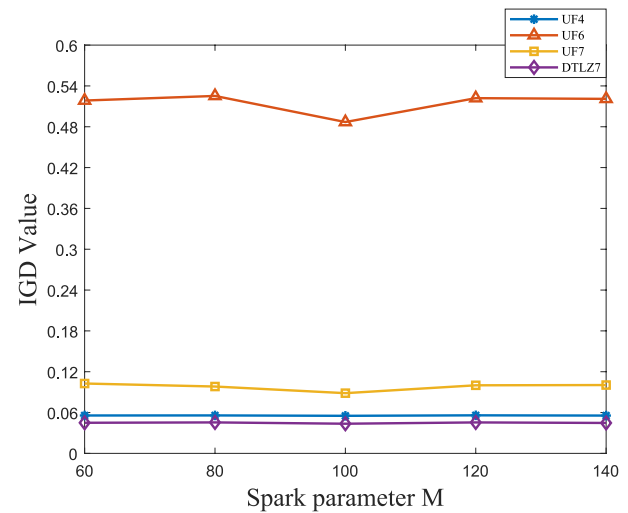


Fig. 13 IGD line graph for different values of W

theoretical support and practical guidance for the parameter selection of FAMOPSO.

Tables 8, 9 and 10 show in detail the IGD value performance of FAMOPSO under different parameters when dealing with test problems ZDT1, ZDT2, ZDT6, UF4, UF6, UF7, DTLZ6, and DTLZ7. These tests cover three different parameter configurations, specifically the number of fireworks M (taking values of 60, 80, 100, 120, and 140, in that order), the inertia weight base parameter W (0.2, 0.3, 0.4, 0.5, and 0.6, in that order), and the base parameter for the learning factor $C1 = C2$ (2, 3, 4, 5, and 6, in that order). To highlight the best performance, the best IGD values for each test problem are bolded in the table. For each test problem,

the algorithm was run independently 30 times to ensure the reliability and stability of the results.

Analyzing the data in Table A, it can be seen that FMOPSO obtained the best IGD values on 7 out of the 8 test functions when $m = 100$. This indicates that the performance of the algorithm is optimal at that particular parameter setting. Looking further at Tables B and C, we find that the algorithm similarly achieves optimal performance on 4 test problems for $W = 0.4$ as well as $C1 = C2 = 2$. This result further validates the importance of specific parameter combinations to enhance the performance of the algorithm. To demonstrate more intuitively the trend of the IGD values of FMOPSO under different parameter settings, we plotted

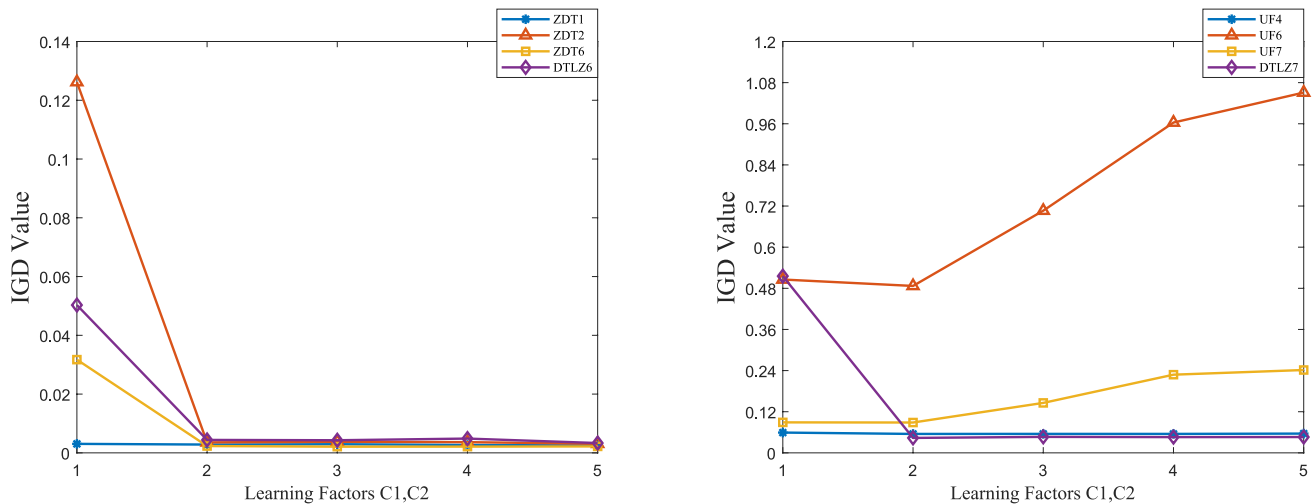


Fig. 14 IGD line graph for different values of C1, C2

the data in Tables 8, 9 and 10 as line graphs 12, 13 and 14. These graphs clearly reveal how different parameter configurations affect the IGD results of the algorithm, which in turn helps us to better understand the mechanism of the parameter choices' impact on the algorithm's performance.

4.12 Friedman Rank Test

In the preceding chapter, a comprehensive comparison was conducted between FAMOPSO, MOPSOs, and MOEAs with the adoption of IGD and HV indicators. Additionally, the performance differences between FAMOPSO and other algorithms were illustrated in an intuitive manner. To provide a more comprehensive evaluation of the performance of FAMOPSO, this chapter introduces the Friedman rank test [16] to facilitate a detailed statistical analysis of the data presented in Tables 2, 3, 4, and 5. This analysis aims to provide a more objective comparison of the overall performance of FAMOPSO with that of the selected comparison algorithm.

Table 11 shows the ordering of IGD indicators and the comprehensive results of the Friedman rank test for each algorithm when it is used to solve the ZDT, UF, and DTLZ test problems. Based on the test findings, FAMOPSO is ranked first in the Friedman rank test because its IGD index on the ZDT and UF series problems is significantly better than that of the other comparative algorithms. In the overall rating of the Friedman test, FAMOPSO continues to rank higher even though its performance in the DTLZ series is not entirely satisfactory. Next, the algorithm ranking data based on the HV index is shown in Table 12. The results demonstrate that FAMOPSO also achieved first place in the

Friedman rank test of ZDT and UF test problems, which indicates its efficacy in addressing MOPs.

The detailed analysis of the IGD and HV indicators, along with the statistical verification of the Friedman rank test, clearly demonstrates that the overall performance of FAMOPSO is superior to that of the other ten comparison algorithms in the selected diversified test problems. This outcome serves to illustrate the considerable aptitude of FAMOPSO in addressing intricate MOPs.

5 Summarizing Work on FAMOPSO

5.1 The Discussion of FAMOPSO in Practical Application Areas is Missing

This paper has primarily concentrated on the technical aspects and performance of the FAMOPSO algorithm, with less emphasis on its practical applications in non-technical domains such as management and decision-making. To enhance the practicality and breadth of the research, future work will further investigate the application of FAMOPSO to MOPs. For example, in the field of logistics, FAMOPSO can be used to optimize goods distribution routes and warehouse layouts. In the field of economics, the algorithm can be applied to asset allocation or market strategy analysis. To apply the FAMOPSO algorithm to multi-objective optimization problems in different domains, it is necessary to adjust the parameter settings or introduce specific domain knowledge in accordance with the particular nature and constraints of the problem under consideration. In the future, we will present case studies

Table 11 Friedman rank test for IGD values for all algorithms

Algorithm	ZDT1-4,6		UF1-10		DTLZ1-7		Overall	
	Friedman test	Rank	Friedman test	Rank	Friedman test	Rank	Friedman test	Rank
MMOPSO	4.4	2	3.8	2	4.5	3	4.16	2
MOPSOCD	5.5	6	9.8	10	8.57	11	8.43	11
NMPSO	5.1	4	5.2	4	5.5	6	5.27	5
SMPSO	5.8	7	9.2	11	7.5	8	7.89	10
dMOPSO	7.2	8	4.9	3	5.29	4	5.55	6
MOEADD	7.2	8	5.3	5	3.36	1	5.11	4
NSGAIII	8.6	10	5.5	6	5.43	5	6.18	7
SPEAR	5.4	5	5.6	7	4	2	5.05	3
CMOEAD	8.4	9	6.3	8	8.29	10	7.41	9
IMMOEA	4.8	3	7.8	9	7.71	9	7.09	8
FAMOPSO	3.6	1	2.6	1	5.86	7	3.86	1

Table 12 Friedman rank test for HV values for all algorithms

Algorithm	ZDT 1–4,6		UF 1–10		DTLZ 1–7		Overall	
	Friedman test	rank	Friedman test	rank	Friedman test	rank	Friedman test	rank
MMOPSO	4.1	2	2.9	2	4.86	3	3.8	2
MOPSOCD	5.1	5	9.1	9	8.14	9	7.89	11
NMPSO	4.5	4	4.7	3	4.14	1	4.48	3
SMPSO	5.3	6	9.1	9	6.14	6	7.3	9
dMOPSO	4.2	3	7.3	8	7.14	8	6.55	7
MOEADD	7.6	8	6.1	5	6.29	7	6.5	6
NSGAIII	7.6	8	5.45	4	4.71	2	5.7	4
SPEAR	9.9	10	6.25	6	6	5	7	8
CMOEAD	6.2	7	6.1	5	5.14	4	5.82	5
IMMOEA	8.4	9	6.4	7	8.57	10	7.55	10
FAMOPSO	3.1	1	2.6	1	4.86	3	3.43	1

demonstrating the effectiveness and potential value of the algorithm in solving real-world MOPs.

5.2 Limitations of FAMOPSO

Despite the excellent performance demonstrated by FAMOPSO, it is not without certain inherent limitations. The first issue is that the algorithm may be susceptible to a phenomenon known as ‘dimensional catastrophe’ when confronted with high-dimensional optimization challenges. This not only impedes the rate of convergence but also predisposes the search process to converge on a local optimum rather than a global optimum prematurely. Secondly, in real-time optimization scenarios, the computational complexity and response time of FAMOPSO may become significant limitations to its widespread application, particularly in highly time-sensitive environments. Ultimately, while this paper provides a comprehensive analysis of the technical aspects and performance

indicators of the FAMOPSO algorithm, it does not delve deeply into its non-technical application value, particularly in the exploration of its potential in practical application scenarios such as management decision-making and strategic planning. In our follow-up work, we will conduct further research in these areas to comprehensively evaluate and promote the application and development of the FAMOPSO algorithm in a wider range of practices.

5.3 Summary of FAMOPSO's Innovations

In this paper, we innovatively propose the FAMOPSO algorithm, which greatly enriches the diversity of the population and significantly accelerates the convergence process through the subtle integration of the FA for generating the fbest that leads the evolution of the population, the well-designed size-double archiving mechanism, as well as the introduction of Active particles based on the fusion of information and designing the adaptive parameters,

which are strategies that greatly enrich the diversity of the population and significantly accelerate the convergence process. Experimental data show that FAMOPSO outperforms existing MOPSO variants and MOEAs on several standard test functions, demonstrating its superior ability to solve MOPs.

5.4 The Future of Work

Although FAMOPSO shows significant application potential in the field of MOPs, it still faces many challenges to achieve its wide application across industries and domains. Further research should concentrate on the development of more sophisticated algorithmic enhancement strategies, the creation of bespoke variants of the algorithm for particular application contexts, and the deepening of the algorithm's integration with the practical application environment. Specifically, further explore the optimization of FA parameter settings to enhance the exploration capability and efficiency of FAMOPSO; secondly, consider combining the adaptive parameter tuning mechanism with other intelligent optimization techniques, such as machine learning or deep learning, to achieve more fine-grained dynamic tuning; and lastly, in view of the computational resource constraints of the practical applications, study how to reduce the computational complexity of FAMOPSO and improve its practicality and scalability. By continuously optimizing and improving the FAMOPSO and its application framework, it is anticipated that more efficient and practical solutions for solving complex MOPs will be provided, which will in turn promote further development and innovation in this field.

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 62062071), and the Qiankehe Platform Talent ZZSG [2024] 014.

Author contributions Yansong Zhang: methodology, software, investigation, validation, writing—original manuscript. Yanmin Liu: project conceptualization, funding acquisition, methodology, project management, writing, review. Xiaoyan Zhang and Qian Song: data management, validation, writing review, and editing.

Data availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflict of interest The authors certify that they have no competing financial interests or personal relationships that could have influenced the work presented in this paper.

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/>.

References

1. Li, K., Chen, R.: Batched data-driven evolutionary multi-objective optimization based on manifold interpolation. *IEEE Trans. Evol. Comput. Evol. Comput.* **27**(1), 126–140 (2022)
2. Pescador-Rojas, M., Coello, C.A.C.: Collaborative and adaptive strategies of different scalarizing functions in MOEA/D. *IEEE Congr. Evol. Comput. (CEC)* **2018**, 1–8 (2018)
3. Hu, Y., Zhang, Y., Gong, D.: Multi-objective particle swarm optimization for feature selection with fuzzy cost. *IEEE Trans. Cybern.* **51**, 874–888 (2021)
4. Moubayed, N.A., Petrovski, A., Mccall, J.: D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. *Evol. Comput.. Comput.* **22**, 47–77 (2014)
5. Li, M., Yang, S., Liu, X.: Pareto or non-Pareto: bi-criterion evolution in multi-objective optimization. *IEEE Trans. Evol. Comput. Evol. Comput.* **20**, 645–665 (2016)
6. Feng, W., Gong, D., Yu, Z.: Multi-objective evolutionary optimization based on online perceiving Pareto front characteristics. *Inf. Sci.* **581**, 912–931 (2021)
7. bin Mohd Zain, M.Z., Kanesan, J., Chuah, J.H., Dhanapal, S., Kendall, G.: A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization. *Appl. Soft Comput. Comput.* **70**, 680–700 (2018)
8. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput. Evol. Comput.* **18**(4), 577–601 (2013)
9. Liu, J., Liu, J.: Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems. *Appl. Soft Comput. Comput.* **74**, 167–189 (2019)
10. Hancer, E.: A new multi-objective differential evolution approach for simultaneous clustering and feature selection. *Eng. Appl. Artif. Intell. Artif. Intell.* **87**, 103307 (2020)
11. Fister, I., Perc, M., Ljubic, K., Kamal, S.M., Iglesias, A., Fister, I.: Particle swarm optimization for automatic creation of complex graphic characters. *Chaos Solitons Fractals* **73**, 29–35 (2015)
12. Yang, C., Ding, J., Jin, Y., Chai, T.: Offline data-driven multiobjective optimization: knowledge transfer between surrogates and generation of final solutions. *IEEE Trans. Evol. Comput. Evol. Comput.* **24**, 409–423 (2020)
13. Zhang, J., Zhang, C., Chu, T., Perc, M.: Resolution of the stochastic strategy spatial prisoner's dilemma by means of particle swarm optimization. *PLoS ONE* **6**(7), e21787 (2011)
14. Taleizadeh, A.A., Niaki, S.T.A., Aryanezhad, M.-B., Shafii, N.: A hybrid method of fuzzy simulation and genetic algorithm to optimize constrained inventory control systems with stochastic replenishments and fuzzy demand. *Inf. Sci.* **220**, 425–441 (2013)

15. Han, H.G., Zhang, L., Liu, H.X., Qiao, J.F.: Multi-objective design of fuzzy neural network controller for wastewater treatment process. *Appl. Soft Comput. Comput.* **67**, 467–478 (2018)
16. Lu, J., Zhang, J., Sheng, J.: Enhanced multi-swarm cooperative particle swarm optimizer. *Swarm Evol. Comput.* **69**, 100989 (2022)
17. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, IEEE, pp. 39–43 (1995)
18. Cheng, S.X., Zhan, H., Yao, H.Q., Fan, H.Y., Liu, Y.: Large-scale many-objective particle swarm optimizer with fast convergence based on alpha-stable mutation and logistic function. *Appl. Soft Comput. Comput.* **99**, 106947 (2021)
19. Coello, C.A.C., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput. Evol. Comput.* **8**(3), 256–279 (2004)
20. Wu, B.L., Hu, W., Hu, J.J., Yen, G.G.: Adaptive multi-objective particle swarm optimization based on evolutionary state estimation. *IEEE Trans. Cybern.* **51**, 3738–3751 (2021)
21. Raquel, C.R., Naval, P.C.: An effective use of crowding distance in multi-objective particle swarm optimization. In: *GECCO 2005—Genet. Evol. Comput. Conf.*, pp. 257–264 (2005)
22. Yuan, Y.Q., Sun, J., Zhou, D.M.: Multi-objective random drift particle swarm optimization algorithm with adaptive grids. In: *2016 IEEE Congr. Evol. Comput.*, IEEE, pp. 2064–2070 (2016)
23. Yang, S.X., Li, M.Q., Liu, X.H., Zheng, J.H.: A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput. Evol. Comput.* **17**, 721–736 (2013)
24. Hu, W., Yen, G.G.: Adaptive multi-objective particle swarm optimization based on parallel cell coordinate system. *IEEE Trans. Evol. Comput. Evol. Comput.* **19**, 1–18 (2015)
25. Cui, Y., Meng, X., Qiao, J.: A multi-objective particle swarm optimization algorithm based on two-archive mechanism. *Appl. Soft Comput. Comput.* **119**, 108532 (2022)
26. Wu, B.L., Hu, W., He, Z.N., Jiang, M., Yen, G.G.: A many-objective particle swarm optimization based on virtual pareto front. In: *IEEE. Cong. Evol. Comput. (CEC)*, pp. 78–85 (2018)
27. Luo, J.P., Huang, X.W., Yang, Y., Li, X., Wang, Z.K., Feng, J.Q.: A many-objective particle swarm optimizer based on indicator and direction vectors for many-objective optimization. *Inf. Sci.* **514**, 166–202 (2020)
28. Ishibuchi, H., Setoguchi, Y., Masuda, H., Nojima, Y.: Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Trans. Evol. Comput. Evol. Comput.* **21**, 169–190 (2017)
29. Kouka, N., BenSaid, F., Fdhila, R., Fourati, R., Hussain, A., Alimi, A.M.: A novel approach of many-objective particle swarm optimization with cooperative agents based on an inverted generational distance indicator. *Inf. Sci.* **623**, 220–241 (2023)
30. Garcia, I.C., Coello, C.A.C., Arias-Montano, A.: MOPSOhv: a new hypervolume-based multi-objective particle swarm optimizer. In: *IEEE. Cong. Evol. Comput. (CEC)*, pp. 266–273 (2014)
31. Li, Y., Zhang, Y., Hu, W.: Adaptive multi-objective particle swarm optimization based on virtual Pareto front. *Inf. Sci.* **625**, 206–236 (2023)
32. Han, H., Lu, W., Qiao, J.: An adaptive multi-objective particle swarm optimization based on multiple adaptive methods. *IEEE Trans. Cybern.* **47**(9), 2754–2767 (2017)
33. Zhou, Y., Kang, J.H., Guo, H.N.: Many-objective optimization of feature selection based on two-level particle cooperation. *Inf. Sci.* **532**, 91–109 (2020)
34. Lin, Q.Z., Liu, S.B., Zhu, Q.L., Tang, C.Y., Song, R.Z., Chen, J.Y., Coello, C.A.C., Wong, K.C., Zhang, J.: Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Trans. Evol. Comput. Evol. Comput.* **22**, 32–46 (2018)
35. Martín, D., Rosete, A., Alcalá-Fdez, J., Herrera, F.: A new multi-objective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules. *IEEE Trans. Evol. Comput. Evol. Comput.* **18**(1), 54–69 (2014)
36. Helwig, S., Branke, J., Mostaghim, S.: Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Trans. Evol. Comput. Evol. Comput.* **17**(2), 259–271 (2013)
37. Zhang, X., Tian, Y., Cheng, R., Jin, Y.: An efficient approach to non-dominated sorting for evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput. Evol. Comput.* **19**(2), 201–213 (2015)
38. Cheng, S., Zhao, L., Jiang, X.: An effective application of bacteria quorum sensing and circular elimination in MOPSO. *IEEE/ACM Trans. Comput. Biol. Bioinform. Comput. Biol. Bioinform.* **14**(1), 56–63 (2017)
39. Feng, Q., Li, Q., Chen, P., Wang, H., Xue, Z., Yin, L., Ge, C.: Multi-objective particle swarm optimization algorithm based on adaptive angle division. *IEEE Access* **7**, 87916–87930 (2019)
40. Zhang, Y., Wei Gong, D., Hua Zhang, J.: Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing* **103**, 172–185 (2013)
41. Li, L., Li, G., Chang, L.: A many-objective particle swarm optimization with grid dominance ranking and clustering. *Appl. Soft Comput. Comput.* **96**, 106661 (2020)
42. Tan Y., Zhu Y.: Fireworks algorithm for optimization. In: *Advances in Swarm Intelligence: First International Conference, ICSI 2010, Beijing, China, June 12–15, 2010, Proceedings, Part I*, pp. 355–364. Springer, Berlin Heidelberg (2010)
43. Li, M., Yang, S., Liu, X.: Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Trans. Evol. Comput. Evol. Comput.* **18**(3), 348–365 (2013)
44. Raquel, C.R., Naval, P.C.: An effective use of crowding distance in multiobjective particle swarm optimization. In: *GECCO 2005—Genet. Evol. Comput. Conf.*, pp. 257–264 (2005)
45. Zitzler, E., Deb, K., Thiele, L.: Comparison of multi-objective evolutionary algorithms: empirical results. *Evol. Comput. Comput.* **8**(2), 173–195 (2000)
46. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multi-objective optimization test instances for the CEC 2009 special session and competition. *Mech Eng New York* **264**, 1–30 (2008)
47. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. In: *Evol. Mult. Opt. London*, pp. 105–145 (2005)
48. Lin, Q.Z., Li, J.Q., Du, Z.H., Chen, J.Y., Ming, Z.: A novel multi-objective particle swarm optimization with multiple search strategies. *Eur. J. Oper. Res. Oper. Res.* **247**, 732–744 (2015)
49. Raquel, C.R., Naval, Jr P.C.: An effective use of crowding distance in multi-objective particle swarm optimization. In: *Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, 257–264 (2005)
50. Nebro, A.J., Durillo, J.J., Nieto, G., Coello, C.A.C., Luna, F., Alba, E.: SMPSO: a new pso-based metaheuristic for multi-objective optimization. In: *2009 IEEE Symp. Comput. Intell. Multi-Criteria Decis.*, pp. 66–73 (2009)
51. Martínez, S.Z., Coello, C.A.C.: A multi-objective particle swarm optimizer based on decomposition. In: *Genet. Evol. Comput. Conf. GECCO'11*, pp. 69–76 (2011)
52. Li, K., Deb, K., Zhang, Q.F., Kwong, S.: An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput. Evol. Comput.* **19**, 694–716 (2015)
53. Jiang, S.Y., Yang, S.X.: A strength pareto evolutionary algorithm based on reference direction for multiobjective and

- many-objective optimization. *IEEE Trans. Evol. Comput. Evol. Comput.* **21**, 329–346 (2017)
54. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput. Evol. Comput.* **18**(4), 602–622 (2013)
55. Cheng, R., Jin, Y., Narukawa, K., et al.: A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling. *IEEE Trans. Evol. Comput. Evol. Comput.* **19**(6), 838–856 (2015)
56. Tian, Y., Cheng, R., Zhang, X.Y., Jin, Y.C.: PlatEMO: A Matlab platform for evolutionary multi-objective optimization. *IEEE Comput. Intell. Mag. Comput. Intell. Mag.* **12**, 73–87 (2017)
57. Zhou, A.M., Jin, Y.C., Zhang, Q.F., Sendhoff, B., Tsang, E.: Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In: 2006 IEEE Int Conf Evol Comput, pp. 892–899. <https://doi.org/10.1109/CEC.2006.1688406> (2006)
58. While, L., Hingston, P., Barone, L., Huband, S.: A faster algorithm for calculating hypervolume. *IEEE Trans. Evol. Comput. Evol. Comput.* **10**(1), 29–38 (2006). <https://doi.org/10.1109/TEVC.2005.851275>

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