

RESEARCH ARTICLE

Particle Swarm Optimization Algorithm With Adaptive Two-Population Strategy

MENGLING ZHAO, HAONAN ZHAO¹, AND MENG ZHAO¹

College of Science, Xi'an University of Science and Technology, Xi'an 710054, China

Corresponding author: Haonan Zhao (1595759214@qq.com)

This work was supported in part by the National Natural Science Foundation of China Youth Program under Grant 11801436 and Grant 11601417, and in part by the Shaanxi Province Natural Basic Research Program Project under Grant 2018JM1047.

ABSTRACT The particle swarm optimization (PSO) algorithm is a swarm intelligence (SI) algorithm used to solve optimization problems. Owing to its advantages in simplicity, using only a few parameters, PSO has become one of the most popular optimization algorithms. However, the single structure of PSO leads to challenges in finding the appropriate optima, resulting in low convergence accuracy. To solve the defects of PSO, it is necessary to increase the diversity of the populations involved as well as enhance the ability of the algorithm to develop locally. In this study, we propose a PSO algorithm with an adaptive two-population strategy (PSO-ATPS), which adaptively divides a population into two groups representing excellent and ordinary populations. Inspired by animal hunting behavior, a new velocity–position update method is proposed for the general population. A velocity update formulation with decreasing inertia weights based on logistic chaotic mapping is applied to the excellent population. The algorithm increases the diversity of the population by continuously changing the search strategy of the particles. In addition, a new neighborhood search strategy (oscillation strategy) is proposed, in which a particle searches randomly in its own adaptive neighborhood when its motion is stagnant and updates the particle position using an elite strategy. Among several optimization strategies, PSO-ATPS achieved first place in 7, 8, and 9 groups of tests involving 10 test functions in 3 dimensions, indicating the accuracy and effectiveness of PSO-ATPS. The results show that the performance of PSO-ATPS is competitive, and many improvements developed for PSO can be applied to PSO-ATPS, demonstrating the potential for further development.

INDEX TERMS Adaptive two-population strategy, elite strategy, logistic chaotic mapping, oscillation strategy, particle swarm optimization.

I. INTRODUCTION

Computation is a scientific method that connects theory and reality. A large number of real-world problems can be transformed into theoretical problems through analysis and modeling. The solutions to these abstract theoretical problems require computation, and algorithms are a general term for the class of computational methods [1]. Optimization algorithms are computational methods that are used to solve optimization problems. The scope of optimization problems is very broad, and many real problems in engineering, science, and economics can be classified as optimization problems [2]. For optimization problems with few param-

eters and low dimensionality, researchers have proposed computational methods based on problem gradients, which are often called traditional optimization algorithms. These algorithms have high accuracy and fast computational speed. However, if the problem to be optimized is complex and high-dimensional, traditional optimization algorithms are not useful. With the remarkable increase in the availability of data in this era, the problems we face have become more complex and higher-dimensional, motivating the development of meta-inspired algorithms to address such complex and high-dimensional optimization problems.

Meta-inspired algorithms are a class of intelligent optimization algorithms provoked by natural phenomena and biological behavior. Many excellent meta-inspired algorithms have been proposed in recent years, such as Henry

The associate editor coordinating the review of this manuscript and approving it for publication was Felix Albu¹.

TABLE 1. Examples of research on optimization algorithms.

Author	Year	Algorithm	Problems	Background
Rizk M. Rizk-Allah et al. [3]	2022	CO-SMA	Single-objective optimization	Minimizing the cost of energy for wind turbines at high altitudes
Murugan Ramachandran et al. [4]	2022	MGOA-IHHO	Multiobjective optimization	Combined heat and power economic dispatch (CHPED)
Shankar Thawkar et al. [5]	2021	BOAALO	Feature selection	Design and development of a computer system for breast cancer detection
Gehad Ismail Sayed et al. [6]	2021	BES	Parameter optimization	A predictive model for melanoma using SqueezeNet.
Sanjoy Chakraborty et al. [7]	2021	mWOAPR	Image segmentation.	Segmentation of COVID-19 chest X-ray images to determine the patient’s condition.
Jie Xing et al. [8]	2021	QGBWOA	Image segmentation.	Segmentation of COVID-19 images to determine the condition of the patient’s lungs
Jayashree Piri et al. [9]	2021	MOQBHHO	Feature selection	Feature selection of medical data to help doctors determine the condition of patients

gas solubility optimization [10], Archimedes optimization algorithm [11], Rime optimization algorithm [12], social network search [13], black widow optimization [14], and coronavirus herd immunity optimizer [15]. According to the source of inspiration, meta-inspired algorithms can be broadly classified as those motivated by physics, chemistry, and biology. Meanwhile, optimization algorithms inspired by biological behavior are known as swarm intelligence optimization algorithms. There are many novel and excellent SI optimization algorithms that have been proposed based on the premise that the behavior of organisms is guided by group behavior and that this biological intelligence exists only in the group and not in the individual organisms. Many SI optimization algorithms, such as genetic algorithm (GA) [16], sparrow search algorithm [17], whale optimization algorithm (WOA) [18], harris hawk optimizer [19], particle swarm optimization (PSO) [20], slime mould algorithm [21], honey badger algorithm [22], and colony predation algorithm [23], utilize the information shared in the population to perform the search. These meta-inspired algorithms have spawned many new algorithms applied to real-world optimization problems, and some examples of the research are listed in Table 1.

The idea of PSO originates from the process of foraging in a population containing social properties such as a flock of birds or colony of bees. In these social groups, individual members share information when performing foraging tasks. Examples of this sharing are leaving a scent or vibrating wings at a certain frequency. Thus, members of the group have information about other members at all times; in PSO, this information is reduced to information about the position of any particular individual, and information regarding the optimal position is communicated to all individuals. All members of the group not only receive the optimal information from other individuals but also record their optimal

positions, and the acceleration of the individuals is based on these two optimal positions. The PSO model is divided into two parts: the velocity update equation and the position update equation. Therefore, PSO essentially involves the summation of vectors.

II. RELATED WORK

We first describe the framework of the original PSO algorithm. Then, we present some representative improved versions of the PSO algorithm, which can be divided into parameter improvement, topology improvement, strategy combination with other algorithms, and practical problem solving.

A. THEORY OF PSO

The PSO algorithm was proposed by Kennedy and Eberhart in 1995 [20], and it has shown great research potential in solving optimization problems. The PSO model consists of two parts, namely the velocity model and the position model. In dimensional space, the solutions to the problem are matrixes with the corresponding dimensions. Thus, the position and the velocity of PSO can be represented by matrixes of size $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$ and $V_i = [v_{i,1}, v_{i,2}, \dots, v_{i,D}]$ of dimension D of the problem solution space, where $x_{i,j}$ denotes the value of the position X_i in dimension j and $v_{i,k}$ denotes the value of velocity V_i in dimension k . The operations of the algorithm are also matrix operations, and the velocity and position update models of the standard PSO are represented as [24]

$$\begin{cases} v_{i,j}^{t+1} = \omega v_{i,j}^t + c_1 r_1 (Gbest_j^t - x_{i,j}^t) + c_2 r_2 (Pbest_{i,j}^t - x_{i,j}^t) \\ x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \end{cases} \tag{1}$$

where $j = 1, 2, \dots, D$ and c_1 and c_2 are the acceleration coefficients controlling the particle deflection, which are constants in the original PSO. Parameters r_1 and r_2 are random numbers taking the value of $[0, 1]$ and following the Gaussian distribution, and t is the number of current iterations. $Gbest_j^t$ denotes the value of the j th dimension of the globally optimal particle in the previous t iterations, and $Pbest_{i,j}^t$ denotes the value of the j th dimension of the optimal position of particle i in the previous t iterations.

B. VARIANTS OF PSO

1) PARAMETER IMPROVEMENT

Parameter optimization in PSO mainly focuses on the inertia weight coefficients ω and acceleration coefficients c_1, c_2 . Because these coefficients control the migration of the particles, the fluctuation of the coefficients affects the search area of the particles and the convergence speed of the algorithm, whereby a suitable combination of parameters can achieve a balance between population diversity and the convergence speed of the algorithm.

To enhance the local search and escape from the local PSO optimum, Chen et al. [25] used a sine mapping scheme to describe the inertia weight and sine mapping and cosine mapping schemes to describe the acceleration coefficients and, respectively. Considering that sine mapping demonstrates good ergodicity as well as nonrepeatability and randomness, it can be used to improve the population diversity of PSO. Ghasemi et al. [26] proposed PPSO, which removes the inertial part of the velocity from the PSO and uses adaptive phase angles to control the motion of the particles. Feng et al. [27] proposed the chaotic descending inertia weight PSO (CDI-WPSO), which randomly decreases the inertia weight of the PSO by logistic chaos mapping, and Duan et al. [28] initialized the particles using logistic chaos mapping, demonstrating that the chaos mapping PSO has an advantage over the standard PSO in a global search. Tian et al. [29] used the Sigmoid function to update the acceleration coefficients of the particles and added a slow-change function to the position update formulation to ensure that the algorithm does not converge prematurely. In the later stage of the search, a Gaussian variation strategy was adopted for poorly adapted particles. Kassoul et al. [30] and Moazen et al. [31] normalized the fitness of the particles to adjust the parameters, with each particle given a set of unique parameter combinations. Sedighzadeh et al. [32] proposed using dynamic inertia weights via automatic adjustment and introduced two terms in the velocity update model to increase the diversity of the PSO population.

2) TOPOLOGY IMPROVEMENT

To improve PSO, changes can also be made in the topology consideration. In standard PSO, the motion of the particle is influenced by two factors, social cognition and self-cognition. In the velocity update of the particle, the former causes the particle to converge toward the optimal position of the

population, whereas the latter allows the population to perform a more comprehensive search of the solution space. That is, at this time, individual particles receive information regarding the optimal position of the population and the individual. This relationship can cause problems such as slow convergence, premature convergence, and low convergence accuracy.

To solve the problem that the PSO algorithm can become trapped near a local optimum, Liu and Nishi [33] proposed a new update formula for Pbest, which expands the search range of particles by searching adjacent positions of Pbest to avoid premature convergence to some extent. Lee et al. [34] added a random noise term to the velocity update formula of PSO and proposed the repulsive method based on the repulsion theory applied to the PSO algorithm. This algorithm causes the particles to adopt repulsive and attractive strategies for Gbest in the development and search phases, respectively. Mousavirad and Rahnamayan [35] added a centroid of all particles to the velocity update formula of PSO to promote faster algorithm convergence [36] and utilizes an average dimensional learning strategy to allow the particles to escape when they fall into a local optimum with information from other dimensions. Meanwhile, [37] and [38] utilize the learning strategy to modify the PSO topology by changing the source of the particle information, which allows the particles to diversify their search in the solution space.

3) HYBRID PSO

Some studies have combined PSO with theories from other metaheuristic algorithms. For example, applying related strategies from the genetic algorithm method, [39] combined the PSO algorithm with the crossover method in GA to improve the diversity of the population by achieving crossover with the position of random particles. In [40], a crossover and variation strategy is applied to some particles in PSO (a value of is suggested in the study) to ensure diversity in the population. Chen et al. [41] combined artificial bee colony (ABC) with PSO, borrowing the lead, follow, and scout strategies of ABC to diversify the PSO population behavior. Singh et al. [42] combined the salp swarm algorithm with PSO to search the solution space by setting different position update formulas for leaders and followers. Hu et al. [43] and Khan and Ling [44] combined the gravitational search algorithm (GSA) with PSO to help PSO escape from local optimum solutions, using the excellent global search capability of GSA.

4) PROBLEM SOLVING WITH PSO

Owing to its simple structure and good plasticity, PSO and its variants have been applied to many real-world problems.

In [45], the proposed quadratic binary PSO is used to solve the scheduling problem of smart home appliances. In [46], [47], enhanced leader particle swarm optimization (PSO) and time-varying acceleration coefficients PSO are proposed and applied to parameter estimation for

TABLE 2. Symbol Description.

Symbols	Description
ω	Inertia weights
c_1	Group learning factor
c_2	Personal learning factor
D	Dimensionality of particles
N	Number of particles
ϕ	Classification factor
z	Logistic chaotic mapping
t	Number of iterations of the algorithm
$iter_{max}$	Maximum number of iterations of the algorithm
k	Adaptive functions
φ	Coefficients of the control neighborhood
Gbest	Global optimal position
Pbest	Personal optimal position

photovoltaic cells and modules, respectively. In [48], PSO is used to plan the flight path of a vehicle, with the optimization resulting in the reduction of the number of collisions of the vehicle. In [49], a new PSO algorithm is proposed to solve the microgrid unit combination problem. In [50], the hybrid binary continuous PSO algorithm is used to solve the coordinated control problem of the generator sets. In [51], PSO and some variants of PSO are used to solve the Flexible AC Transmission Systems optimization problem for power systems. In [52], the fuzzy expected value model problem is solved by combining stochastic PSO with a back propagation neural network. In [53], PSO is used to reduce the operating cost of a power system. In [54], the opposition-based PSO algorithm is proposed by considering the learning strategy of the opposition to PSO, which is used to solve the distribution and dispatching problem of active distribution networks.

III. PARTICLE SWARM OPTIMIZATION ALGORITHM WITH ADAPTIVE TWO-POPULATION STRATEGY

As one of the more well-known optimization algorithms, PSO is very influential and flexible. However, owing to its simple and easily adjustable model, PSO can easily get trapped into local optimality when facing many high-dimensional complex problems. As PSO is a mathematical model that abstracts the predatory behavior of a flock of birds, the population of PSO can be used to perform more than one type of search behavior. To improve the solution quality as well as convergence speed of PSO, this chapter proposes a PSO algorithm with an adaptive two-population strategy (PSO-ATPS). The logic of this algorithm is to divide the population into two subpopulations by fitness sorting, with the two populations executing different search strategies. The size of the two populations can be dynamically adjusted by an adaptive function to achieve a balance between the algorithm in exploitation and search. An explanation of the notation used in optimization models is presented in Table 2.

This chapter is divided into six parts: (1) The first part introduces a new simple classification function that helps us classify the individuals in the population into two subpopulations according to fitness size. (2) The second part introduces the velocity update formula for each of the two classified subpopulations and introduces a new velocity update formula to enhance the exploitation ability as well as the search ability of the population. (3) The third part introduces the position update formulas for each of the two classified subpopulations and introduces a new position update method with the help of a vector addition feature. (4) The fourth part introduces a new neighborhood search strategy that can help particles trapped in local optima to escape. (5) The fifth part presents the general framework of PSO-ATPS. (6) The last part analyzes and explains the time complexity of PSO-ATPS.

A. ADAPTIVE FUNCTIONS TO BE USED FOR CLASSIFICATION

To apply PSO-ATPS, the particles in population POP must first be sorted, and if the minimization problem (maximization problem) is solved, the fitness is sorted from low to high (high to low). The sorted population is called POPnew and is adaptively divided into excellent and ordinary populations by the function shown in (2), with the functional expression

$$\phi = \left\lceil \text{random} * N * \frac{t}{iter_{max}} \right\rceil \quad (2)$$

where *random* is a Gaussian-distributed random number within the interval [0,1], *N* is the size of the population, *t* is the current number of iterations of the algorithm, and *iter_{max}* is the maximum number of iterations of the algorithm set.

B. FORMULA FOR THE NEW UPDATED VELOCITY

Before introducing the velocity update formula, we first need to classify POPnew into excellent population POPnew1 and ordinary population POPnew2, according to the classification function. When the particle's ordinal number *i* is less than the number of classification ϕ , this particle belongs to POPnew 1, while other particles that do not satisfy the condition belong to POPnew2.

We provide a velocity update formula for each of these two subgroups

$$\begin{cases} v_{ij}^{t+1} = \omega^t v_{ij}^t + c_1 r_1 (Gbest_j^t - x_{ij}^t) \\ \quad + c_2 r_2 (Pbest_{ij}^t - x_{ij}^t) & \text{if } i \leq \phi \\ v_{ij}^{t+1} = Gbest_j^t + r_3 * Levy \\ \quad * (2 * r_4 * Gbest_j^t - x_{ij}^t) & \text{if } i > \phi \end{cases} \quad (3)$$

where r_1 is a random number in the interval [-1,1] and r_4 is a random number in the interval [0,1]. Parameter ω^t is the decreasing inertia weight using the logistic chaos mapping [27]

$$z^{t+1} = 4 * z^t * (1 - z^t) \quad (4)$$

$$\omega^t = \frac{\omega_{max} - \omega_{min}}{iter_{max}} (iter_{max} - t) + \omega_{min} z^t \quad (5)$$

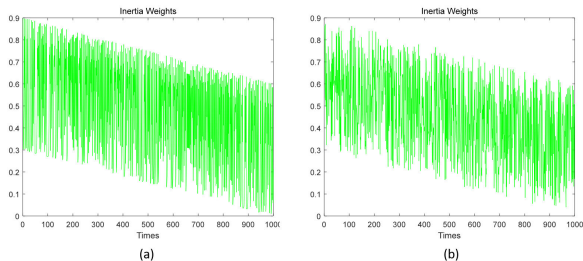


FIGURE 2. Logistic chaotic mapping control of inertia weights and random control of falling inertia weights.

$\omega_{max} = 0.9$ and $\omega_{min} = 0.6$. Fig. 2. (b) shows the iterative images of the inertia weights for random number rand with the same parameter settings. It is evident that the inertia weights are more uniformly distributed with the help of the logistic chaos mapping; thus, the logistic chaos mapping is suitable.

Levy is a random number obeying the Levy distribution, with the Levy distribution defined by the following equation [55]

$$L(x, \gamma, \mu) = \sqrt{\frac{\gamma}{2\pi}} \exp\left[-\frac{\gamma}{2(x-\mu)}\right] \frac{1}{(x-\mu)^{3/2}}, \quad 0 < \mu < x \quad (6)$$

Here, parameter γ is used to control the shape of the distribution curve, and parameter μ is used to control the position of the distribution curve.

After Fourier transform, the Levy distribution becomes

$$F(k) = \exp[-\alpha|k|^\beta] \quad (7)$$

where α is a parameter in the interval [1, 1] and Levy index β is a parameter in the interval (0,2). A typical value for β is 1.5. Obtaining random numbers obeying the Levy distribution requires the use of Mantegna’s equation, which has the following form,

$$S = \frac{u}{|v|^{1/\beta}} \quad (8)$$

where S is a random number generated according to Mantegna’s equation, obeying the Levy distribution, and both u and v obey the Gaussian distribution of the form

$$u \sim N(0, \sigma^2), v \sim N(0, 1) \quad (9)$$

$$\sigma = \left[\frac{\Gamma(1+\beta) \sin(\frac{\beta\pi}{2})}{\Gamma(\frac{1+\beta}{2}) \beta 2^{(\frac{\beta-1}{2})}} \right] \quad (10)$$

The random walk based on the step size generated by this method is referred to as Levy Flight [56], [57], which describes most of the random-motion trajectories in nature. Compared with Brownian motion, Levy Flight makes large step spans, a feature that permits the escape from local optima in stochastic search. Fig. 3.(a) shows the path of Levy Flight for Levy index $\beta = 1.5$ with 10^3 cycles. Fig. 3.(a) shows that Levy Flight makes large stepwise jumps, which is advantageous for global stochastic search. Fig. 3.(b) shows a plot of

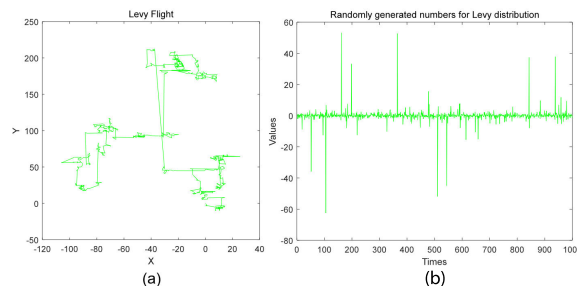


FIGURE 3. Levy Flight diagram and randomly generated numbers obeying the Levy distribution.

the randomly generated numbers obeying the Levy distribution for $\beta = 1.5$ and 10^3 cycles are executed, from which we can see that the Levy random numbers are not uniform; rather, this randomness is more in line with biological reality.

C. FORMULA FOR THE NEW UPDATE POSITION

As the position update essentially involves a summation of vectors, we classify the position parameters of the two subgroups based on the standard PSO. For POPnew1, the position update formula of the standard PSO is followed, as the form of its velocity update formula does not differ from that of the standard PSO. The velocity update formula of POPnew2 is taken from the biopredator algorithm; thus, the form of the position update formula needs to be changed to

$$\begin{cases} x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} & \text{if } i \leq \phi \\ x_{i,j}^{t+1} = (x_{i,j}^t + v_{i,j}^t) / 2 & \text{if } i > \phi \end{cases} \quad (11)$$

The newly introduced update formulas of velocity and position of POPnew2 cause particles with poor adaptation to randomly search in the neighborhood of the optimum solution, which can enhance the efficiency of particle utilization and strengthen the ability of the local search, improving the accuracy of the solution. From the velocity update formula of POPnew2, velocity can be regarded as a position in (3).

If the velocity or position of the particle exceeds the maximum (minimum) bound during the iteration, (12) is used to adjust the particle in POPnew1, and (13) is used to adjust the particle in POPnew2

$$\begin{cases} v_i^t = v_{min} + r * (v_{max} - v_{min}) & \text{if } v_i^t \notin [v_{min}, v_{max}] \\ x_i^t = x_{min} + r * (x_{max} - x_{min}) & \text{if } x_i^t \notin [x_{min}, x_{max}] \end{cases} \quad (12)$$

$$\begin{cases} x_i^t = x_{min} + r^2 * (v_{max} - v_{min}) & \text{if } x_i^t < x_{min} \\ x_i^t = x_{min} + \sqrt{r} * (v_{max} - v_{min}) & \text{if } x_i^t > x_{max} \end{cases} \quad (13)$$

where v_{min} is the lower bound of the velocity, v_{max} is the upper bound of the velocity, x_{max} is the upper bound of the solution space, and x_{min} is the lower bound of the solution space.

D. OSCILLATION STRATEGY

To solve the PSO trapping problem, a new adaptive neighborhood search strategy is introduced. Inspired by the ABC optimization algorithm, we use the neighborhood to help

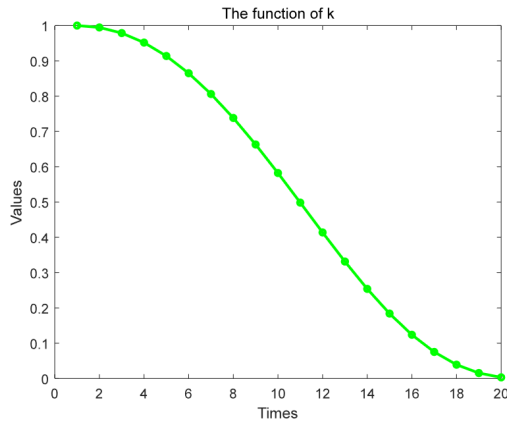


FIGURE 4. Trajectory of parameter k .

particles with constant fitness to escape the optimization trap, applying an oscillation strategy,

$$x_{i,j}^t = x_{i,j}^t + k * Levy * \frac{x_{max} - x_{min}}{\varphi * random} \quad (14)$$

where φ is the neighborhood control parameter, and k is the adaptive function from the following expression,

$$k = \left(1 - \frac{t}{iter_{max}}\right)^{\frac{2t}{iter_{max}}} \quad (15)$$

Fig. 4 shows an image of the function k . The image shows that at low cycles, k at the maximum number of iterations is concave, which allows the particle to oscillate in a large range. Meanwhile, at a high number of cycles, k at the maximum number of iterations is convex, allowing the particle to oscillate in a small range, causing the search to achieve a balance. When the fitness of the particle is equal to its historical optimal fitness, the particle executes the oscillation strategy and compares the new solution to its historical optimal solution. If the result is better, the solution is updated; otherwise, the current position of the particle remains unchanged.

E. FRAMEWORK OF THE PROPOSED ALGORITHM

Fig. 1 shows the flowchart of the new algorithm, and Table 3 shows the corresponding pseudocode. The pseudocode and flowchart can help the reader organize the relationship between the components.

F. ALGORITHM COMPLEXITY ANALYSIS

The time complexity of PSO is $O(DN)$ (D and N are the dimensionality and population size, respectively.). As opposed to PSO, PSO-ATPS contains classification function ϕ , inertia weight ω^t , and an oscillation strategy. The time complexity of the classification function ϕ is $O(1)$ and that of the inertia weights ω^t is $O(D)$. Pertaining to oscillation strategy, two extreme cases are considered: no oscillation strategy and POPnew1 executes all oscillation strategies. The time complexity of the two cases is $O(1)$ and $O(DN)$, respectively. Therefore, the time complexity of the worst case of

TABLE 3. PSO-ATPS Algorithm.

Input: $\omega_{max}, \omega_{min}, x_{max}, x_{min}, c_1, c_2, \beta, iter_{max}, \varphi, z^0$
Initialization: Position X , Velocity V , Classification factor ϕ
While $t < iter_{max}$ do
Calculate the fitness $f(x_i)$ of all the particles in the population POP and rank the particles according to their fitness from largest to smallest;
Update the individual optimal position $Pbest$ and the group optimal position $Gbest$;
The classification factor ϕ was calculated according to equation (2);
The populations were classified into POPnew1 and POPnew2 according to the classification factor ϕ ;
Update the inertia weights ω^t according to Equation (4)(5);
Update the speed of POPnew1 according to equation (3);
Update the position of POPnew1 according to Equation (11);
If $f(x_i) = Pbest_i^t$ then
Execute the oscillation strategy according to equation (15)(14);
End If
Processing of out-of-bounds particles according to Equation (12);
Update the speed of POPnew2 according to equation (3);
Update the position of POPnew2 according to Equation (11);
Processing of out-of-bounds particles according to Equation (13);
Find the optimal solution $f(x_i)$ as the solution of this iteration;
End While
Output $Gbest$

PSO-ATPS is also $O(DN)$, and PSO-ATPS has the same time complexity as PSO.

Typically, PSO-ATPS takes more time on simple functions than PSO, while it takes nearly the same amount of time on complex multi-peaked functions as PSO. This is because oscillatory strategies are more likely to be executed on simple functions and less often on multi-peaked functions.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section uses seven algorithms for comparison with PSO-ATPS and conducts numerical experiments on a test set consisting of 15 benchmark test functions. Three of them are related to PSO algorithms: PSO, CDIWPSO, and PPSO. Four of them are intelligent optimization algorithms widely used in recent years: sine cosine algorithm (SCA) [58], WOA [18], grey wolf optimizer (GWO) [59], and marine predators algorithm (MPA) [60].

A. NUMERICAL BENCHMARK FUNCTIONS

PSO is chosen as the comparison algorithm, as the strategy adopted by PSO-ATPS for POPnew1 is inspired by PSO. CDIWPSO is chosen as the comparison algorithm because both it and PSO-ATPS employ decreasing inertia weights

TABLE 4. Unimodal test functions.

Name	Function	Initial range	F _{min}
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]^d$	0
Schwefel_2_22	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]^d$	0
Schwefel_1_2	$f_3(x) = \sum_{i=1}^d \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^d$	0
Schwefel_2_21	$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	$[-100, 100]^d$	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1} \left[100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \right]$	$[-30, 30]^d$	0

TABLE 5. Multimodal test functions.

Name	Function	Initial range	F _{min}
Rastrigin	$f_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^d$	0
Ackley	$f_7(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	0	
Griewank	$f_8(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right)$	$[-600, 600]^d$	0
Levy	$f_9(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 \left[1 + 10 \sin^2(\pi w_i + 1) + (w_d - 1)^2 \right]$ $+ (w_d - 1)^2 [1 + \sin^2(2\pi w_d)]$ <p>Where</p> $w_i = 1 + \frac{x_i - 1}{4}$	$[-10, 10]^d$	0
Schwefel	$f_{10}(x) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{ x_i })$	$[-500, 500]^d$	0

influenced by chaotic mapping. PPSO is chosen as the comparison algorithm because of its unique way of updating the velocity, and changes are made in PSO-ATPS for POPnew2. SCA is chosen as one of the comparison algorithms for PSO-ATPS because it belongs to the group intelligence optimization algorithm. In addition, the search range is controlled by the triangular function, and the strategy of neighborhood search is also used in PSO-ATPS. The WOA and GWO algorithms are chosen as comparison algorithms because of their relationship with PSO-ATPS regarding the location update approach of the algorithm, and as one of the most well-known SI optimization algorithms, comparison with them can better display the advantages and disadvantages of PSO-ATPS. MPA is chosen as a comparison algorithm because the adaptive process of PSO-ATPS is taken from MPA, and as one

of the latest well-known SI optimization algorithms, MPA can provide us with many new ideas. The parameters of all algorithms are indicated in Table 9.

Test functions are divided into three categories. The first is unimodal test functions, which have only one extreme value in the solution space. The second category is multi-peaked functions, which have many extreme points in the solution space, i.e., they have many local optima, and these problems are very complex, whereby the optimization algorithm can easily descend into the local optima. The third category is test functions with fixed dimensionality. For the sake of fairness, all algorithms are set with the same population size $N = 100$ and the same maximum number of iterations $iter_{max} = 10^3$. All tests are executed 25 times, from which the mean, standard deviation, and optimal fitness values are recorded. For

TABLE 6. Fixed-dimension test functions.

Name	Function	Initial range	dim	F _{min}
Cross-in-Tray	$f_{11}(x) = -0.0001 \left(\left \sin(x_1) \sin(x_2) \exp \left(\left 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right \right) \right + 1 \right)^{0.1}$	$[-10, 10]^d$	2	-2.06261
Drop-Wave	$f_{12}(x) = -\frac{1 + \cos \left(\frac{12\sqrt{x_1^2 + x_2^2}}{0.5(x_1^2 + x_2^2) + 2} \right)}$	$[-5.12, 5.12]^d$	2	-1
Eggholder	$f_{13}(x) = -(x_2 + 47) \sin \left(\sqrt{ x_2 + \frac{x_1}{2} + 47 } \right)$ $-x_1 \sin \left(\sqrt{ x_1 - (x_2 + 47) } \right)$	$[-512, 512]^d$	2	-959.6407
Shubert	$f_{14}(x) = \prod_{j=1}^2 \left(\sum_{i=1}^5 i \cos((i+1)x_j + i) \right)$	$[-5.12, 5.12]^d$	2	-186.7309
Six-Hump Camel	$f_{15}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^d$	2	-1.0316285

TABLE 7. Ranking of test means for different ω_{min} .

ω_{min}	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Ave	7.92E-04	7.73E-04	4.13E-04	1.65E-04	2.90E-04	1.38E-04	9.94E-05	2.02E-04
Rank	8	7	6	3	5	2	1	4

TABLE 8. Ranking of test means for different φ .

φ	10	20	30	40	50	60	70	80	90	100
Ave	2.61E-04	8.29E-05	1.11E-04	2.44E-04	1.45E-04	6.44E-05	1.60E-04	2.39E-04	2.41E-04	1.49E-04
Rank	10	2	3	9	4	1	6	7	8	5

TABLE 9. Algorithm parameter setting.

Algorithm	Number	iter _{max}	dim	Parameter settings
PSO-ATPS	100	1000	10,30,50	$\omega_{max}=0.9, \omega_{min} = 0.6, c_1 = c_2 = 1.49618, \varphi = 25$
PSO	100	1000	10,30,50	$\omega = 0.7298, c_1 = c_2 = 1.49618$
CDIWPSO	100	1000	10,30,50	$\omega_{max}=0.9, \omega_{min} = 0.4, c_1 = c_2 = 2$
PPSO	100	1000	10,30,50	c_1 and c_2 are given by the trigonometric functions
SCA	100	1000	10,30,50	$a=2$
WOA	100	1000	10,30,50	$\beta=1$
GWO	100	1000	10,30,50	a decreases linearly from 2 to 0
MPA	100	1000	10,30,50	$P=0.5, F_{ADS}=0.2$

the first two types of functions, the performance of the algorithms for number of dimensions dim=10, dim=30 and dim=50 are examined to determine whether the algorithms will have a sharp decay in accuracy owing to the increase in dimensionality.

The information of the 15 benchmark test functions is presented in Table 4, Table 5, and Table 6. Five of the test functions in Table 4 are unimodal test functions, which are tested on this test set to determine the local search capability of the algorithm. In Table 5, five multimodal test functions are

TABLE 10. Accuracy comparison for 10-dimensional problems.

Function	Criterion	PSO-ATPS	PSO	CDIWPSO	PPSO	SCA	WOA	GWO	MPA
f1	Ave	0	2.93E-50	8.78E-15	4.81E-22	2.52E-36	0	0	2.47E-82
	Std	0	1.06E-49	2.69E-14	8.71E-22	6.16E-36	0	0	1.10E-81
	Best	0	6.10E-61	2.45E-27	8.43E-26	1.24E-43	0	0	8.16E-87
f2	Ave	0	2.65E-02	8.70E-04	4.46E-07	1.98E-24	0	1.36E-98	6.56E-45
	Std	0	6.29E-02	1.30E-03	6.98E-07	2.76E-24	0	6.06E-98	8.10E-45
	Best	0	4.12E-05	7.57E-06	3.16E-09	1.47E-28	0	0	9.02E-47
f3	Ave	0	1.02E-41	5.91E-09	9.28E-16	1.72E-16	9.20E-03	2.61E-86	1.79E-45
	Std	0	5.10E-41	2.14E-08	2.16E-15	8.44E-16	2.18E-02	1.18E-85	4.25E-45
	Best	0	6.97E-56	1.21E-18	4.53E-19	1.15E-23	7.27E-08	2.77E-95	4.09E-48
f4	Ave	0	1.41E-01	3.67E-02	9.76E-07	3.65E-13	1.80E-03	1.13E-57	1.42E-33
	Std	0	1.73E-01	5.17E-02	1.50E-06	7.20E-13	9.00E-03	2.08E-57	2.02E-33
	Best	0	3.60E-03	1.10E-03	2.29E-09	1.10E-17	1.13E-24	8.70E-62	1.12E-35
f5	Ave	3.12E-02	8.10E+00	7.55E+00	3.18E-01	6.76E+00	4.62E+00	5.86E+00	5.55E-07
	Std	5.58E-02	2.39E+01	2.06E+01	1.10E+00	2.88E-01	9.86E-01	5.93E-01	2.95E-07
	Best	2.70E-20	2.18E-06	1.93E-05	4.73E-20	6.29E+00	4.49E-02	5.13E+00	1.42E-07
f6	Ave	0	1.20E+01	3.90E+00	6.82E-15	0	0	0	0
	Std	0	5.12E+00	1.64E+00	1.01E-14	0	0	0	0
	Best	0	3.97E+00	9.95E-01	0	0	0	0	0
f7	Ave	8.88E-16	1.05E+00	4.58E-01	1.06E-11	4.01E-15	3.44E-15	4.44E-15	8.88E-16
	Std	0	9.17E-01	7.89E-01	1.32E-11	1.17E-15	1.92E-15	0	0
	Best	8.88E-16	7.99E-15	7.64E-12	4.62E-13	8.88E-16	8.88E-16	4.44E-15	8.88E-16
f8	Ave	0	2.40E-01	1.57E-01	1.41E-02	3.45E-02	3.47E-02	1.60E-02	0
	Std	0	1.57E-01	8.17E-02	3.84E-02	1.15E-01	8.22E-02	3.26E-02	0
	Best	0	2.71E-02	4.68E-02	0	0	0	0	0
f9	Ave	8.26E-06	6.11E-02	8.27E-17	3.11E-20	2.68E-01	4.54E-06	3.61E-02	7.41E-11
	Std	8.76E-06	1.06E-01	3.08E-16	8.81E-20	6.25E-02	4.97E-06	5.20E-02	3.80E-11
	Best	6.73E-11	0	4.10E-27	9.17E-25	1.61E-01	4.14E-07	4.03E-07	3.00E-11
f10	Ave	1.22E+03	1.05E+03	2.77E+02	7.62E+01	1.82E+03	4.06E+02	1.17E+03	3.01E+02
	Std	4.35E+02	2.63E+02	1.52E+02	1.32E+02	1.41E+02	5.21E+02	2.61E+02	1.38E+02
	Best	4.73E+02	5.72E+02	1.27E-04	1.27E-04	1.52E+03	1.32E-04	6.90E+02	1.18E+02
Rank		1	8	7	5	6	4	3	2

tested on this test set to determine the global search capability and local development capability of the algorithm. Table 6 shows the five fixed-dimension test functions.

B. PARAMETER SENSITIVITY ANALYSIS

As the selection of parameters affects the performance of the algorithm, we perform a sensitivity analysis of the parameters

TABLE 11. Accuracy comparison for 30-dimensional problems.

Function	Criterion	PSO-ATPS	PSO	CDIWPSO	PPSO	SCA	WOA	GWO	MPA
f1	Ave	0	2.57E+02	7.10E+00	4.11E-08	7.59E-04	0	1.30E-84	6.69E-56
	Std	0	1.75E+02	5.06E+00	7.17E-08	2.20E-03	0	4.83E-84	1.04E-55
	Best	0	5.21E+01	1.03E+00	3.20E-09	3.06E-08	0	5.80E-88	2.29E-58
f2	Ave	0	8.74E+00	1.30E+00	1.70E-03	1.27E-06	0	3.11E-49	4.91E-31
	Std	0	2.64E+00	4.71E-01	1.70E-03	2.42E-06	0	3.91E-49	7.09E-31
	Best	0	5.20E+00	4.20E-01	4.19E-05	4.72E-10	0	2.71E-50	3.80E-33
f3	Ave	0	2.36E+03	3.66E+02	1.61E-04	9.93E+02	4.14E+03	9.89E-27	8.31E-19
	Std	0	9.45E+02	1.59E+02	2.04E-04	1.44E+03	3.77E+03	3.02E-26	1.60E-18
	Best	0	1.21E+03	1.28E+02	2.20E-06	5.38E+00	2.17E+01	1.29E-30	3.63E-28
f4	Ave	0	1.85E+01	7.56E+00	4.60E-03	9.00E+00	5.51E+00	9.49E-22	1.56E-21
	Std	0	3.89E+00	1.70E+00	1.28E-02	7.55E+00	1.20E+01	1.21E-21	1.10E-21
	Best	0	1.22E+01	3.20E+00	1.29E-07	7.64E-01	3.17E-11	2.23E-23	2.33E-22
f5	Ave	1.13E-01	1.85E+04	2.52E+02	2.22E+01	5.85E+01	2.58E+01	2.64E+01	2.27E+01
	Std	2.21E-01	1.58E+04	1.64E+02	8.75E-01	1.49E+02	3.02E-01	6.12E-01	2.85E-01
	Best	3.22E-22	1.77E+03	7.46E+01	2.06E+01	2.72E+01	2.51E+01	2.52E+01	2.23E+01
f6	Ave	0	6.95E+01	4.69E+01	1.86E-08	7.38E+00	0	2.27E-15	0
	Std	0	1.62E+01	1.12E+01	2.63E-08	2.08E+01	0	1.13E-14	0
	Best	0	4.55E+01	2.88E+01	9.62E-11	8.53E-10	0	0	0
f7	Ave	8.88E-16	7.41E+00	3.35E+00	4.05E-05	1.18E+01	3.73E-15	1.08E-14	4.15E-15
	Std	0	1.14E+00	6.61E-01	2.08E-05	9.29E+00	2.05E-15	3.40E-15	9.83E-16
	Best	8.88E-16	5.54E+00	2.40E+00	3.97E-06	2.96E-05	8.88E-16	7.99E-15	8.88E-16
f8	Ave	0	3.08E+00	1.02E+00	2.40E-03	4.22E-02	1.40E-03	1.00E-03	0
	Std	0	9.35E-01	5.19E-02	5.20E-03	8.82E-02	5.00E-03	2.90E-03	0
	Best	0	1.76E+00	8.81E-01	4.84E-09	1.15E-07	0	0	0
f9	Ave	1.38E-04	6.61E+00	3.51E+00	2.28E-06	1.99E+00	3.39E-02	8.59E-01	4.10E-03
	Std	3.77E-04	3.25E+00	2.11E+00	5.68E-06	1.72E-01	5.84E-02	1.10E-01	1.78E-02
	Best	5.97E-12	1.61E+00	3.65E-01	2.21E-09	1.69E+00	1.57E-04	6.36E-01	1.81E-05
f10	Ave	5.21E+03	5.71E+03	2.73E+03	9.05E+02	8.34E+03	8.60E+02	6.18E+03	3.67E+03
	Std	1.08E+03	4.77E+02	5.16E+02	1.10E+03	2.09E+02	1.23E+03	8.43E+02	4.84E+02
	Best	3.38E+03	4.66E+03	1.59E+03	3.87E-04	7.99E+03	7.52E-02	5.02E+03	2.79E+03
Rank		1	8	6	5	7	3	4	2

in PSO-ATPS in this section. The adjustable parameters in PSO-ATPS are the minimum value of inertia weight ω_{\min} and

neighborhood control parameter φ . The values of ω_{\min} that can be selected are, and 0.8. Thus, we test these eight groups

TABLE 12. Accuracy comparison for 50-dimensional problems.

Function	Criterion	PSO-ATPS	PSO	CDIWPSO	PPSO	SCA	WOA	GWO	MPA
f1	Ave	0	2.87E+03	9.44E+01	1.01E-05	9.65E+00	0	4.36E-62	3.24E-50
	Std	0	9.51E+02	3.60E+01	7.60E-06	1.26E+01	0	1.03E-61	5.33E-50
	Best	0	1.31E+03	3.03E+01	2.10E-07	1.21E-02	0	5.61E-64	3.20E-53
f2	Ave	0	3.48E+01	7.77E+00	7.90E-03	1.60E-03	0	2.07E-36	4.90E-28
	Std	0	7.62E+00	3.55E+00	5.70E-03	2.80E-03	0	1.12E-36	5.67E-28
	Best	0	2.02E+01	2.90E+00	1.30E-03	1.43E-05	0	8.22E-37	1.30E-31
f3	Ave	0	1.06E+04	4.49E+03	2.80E-03	2.06E+04	4.49E+04	4.04E-14	2.50E-13
	Std	0	2.99E+03	1.44E+03	4.70E-03	9.25E+03	2.15E+04	9.93E-14	7.59E-13
	Best	0	3.38E+03	2.35E+03	1.14E-05	9.83E+03	5.21E+03	6.07E-18	2.53E-18
f4	Ave	0	3.13E+01	2.01E+01	5.60E-03	4.79E+01	4.82E+01	5.17E-14	3.72E-19
	Std	0	5.00E+00	3.40E+00	7.10E-03	1.03E+01	3.45E+01	5.63E-14	2.20E-19
	Best	0	2.13E+01	1.42E+01	5.04E-05	2.53E+01	1.90E-03	7.46E-15	1.35E-19
f5	Ave	1.38E-01	5.50E+05	4.47E+03	4.49E+01	1.55E+05	4.61E+01	4.62E+01	4.43E+01
	Std	2.67E-01	1.97E+05	1.92E+03	9.24E-01	4.49E+05	2.67E-01	7.83E-01	4.18E-01
	Best	1.02E-16	2.22E+05	1.49E+03	4.34E+01	2.85E+02	4.58E+01	4.51E+01	4.37E+01
f6	Ave	0	1.95E+02	1.18E+02	5.07E-06	4.87E+01	0	6.82E-15	0
	Std	0	2.23E+01	2.61E+01	8.89E-06	4.61E+01	0	1.88E-14	0
	Best	0	1.46E+02	7.34E+01	5.89E-10	4.76E-02	0	0	0
f7	Ave	8.88E-16	1.18E+01	5.71E+00	5.56E-04	1.49E+01	3.16E-15	1.83E-14	4.44E-15
	Std	0	9.20E-01	8.48E-01	3.33E-04	8.63E+00	2.26E-15	3.38E-15	0
	Best	8.88E-16	9.88E+00	4.27E+00	9.93E-05	1.54E-02	8.88E-16	1.50E-14	4.44E-15
f8	Ave	0	2.71E+01	1.91E+00	8.42E-04	9.08E-01	4.60E-03	7.41E-04	0
	Std	0	6.24E+00	3.81E-01	2.70E-03	2.85E-01	1.61E-02	2.60E-03	0
	Best	0	1.65E+01	1.31E+00	2.08E-07	7.01E-02	0	0	0
f9	Ave	2.20E-04	1.93E+01	1.28E+01	1.33E-01	7.57E+00	6.34E-02	2.08E+00	2.61E+00
	Std	4.27E-04	6.24E+00	4.36E+00	2.19E-01	6.63E+00	7.47E-02	2.90E-01	1.63E-01
	Best	5.84E-15	7.98E+00	3.67E+00	6.71E-06	3.91E+00	3.00E-03	1.54E+00	2.81E-02
f10	Ave	1.04E+04	1.13E+03	6.24E+03	1.50E+03	1.55E+04	6.89E+02	1.15E+04	7.43E+03
	Std	1.05E+03	8.36E+02	5.91E+02	1.03E+03	3.76E+02	6.90E+02	1.10E+03	7.23E+02
	Best	8.03E+03	9.77E+03	5.14E+03	2.38E+02	1.47E+04	2.87E-02	1.02E+04	5.97E+03
Rank		1	7	6	5	8	3	4	2

TABLE 13. Fixed-dimension test functions.

Function	dim	Criterion	PSO-ATPS	PSO	CDIWPSO	PPSO	SCA	WOA	GWO	MPA
f11	2	Ave	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626
		Std	9.50E-16	9.06E-16	9.06E-16	9.06E-16	2.48E-06	9.74E-12	8.72E-10	9.06E-16
		Best	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626
f12	2	Ave	-1	-1	-1	-1	-1	-0.9898	-1	-1
		Std	0	0	0	0	0	2.39E-02	0	0
		Best	-1	-1	-1	-1	-1	-1	-1	-1
f13	2	Ave	-958.0395	-958.2314	-958.8414	-951.8333	-959.6403	-959.6407	-922.2373	-959.6407
		Std	1.1043	1.3005	1.1279	21.5785	7.11E-04	1.73E-13	60.2658	4.64E-13
		Best	-959.6345	-959.6329	-959.6398	-959.6407	-959.6407	-959.6407	-959.6407	-959.6407
f14	2	Ave	-186.7309	-186.7309	-186.7309	-186.7309	-186.7009	-186.7309	-186.7309	-186.7309
		Std	3.62E-14	2.65E-14	2.32E-14	2.52E-14	0.0421	4.06E-07	9.80E-06	5.18E-14
		Best	-186.7309	-186.7309	-186.7309	-186.7309	-186.7273	-186.7309	-186.7309	-186.7309
f15	2	Ave	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
		Std	4.88E-16	6.79E-16	6.79E-16	6.64E-16	5.10E-06	4.38E-13	1.68E-09	6.72E-16
		Best	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316

of parameters on f9 25 times each and record the average value. Here, f9 is selected as a test function from Table 5:

$$\begin{aligned}
 f_9(x) = & \sin^2(\pi w_1) \\
 & + \sum_{i=1}^{d-1} (w_i - 1)^2 \left[1 + 10 \sin^2(\pi w_i + 1) + (w_d - 1)^2 \right] \\
 & + (w_d - 1)^2 \left[1 + \sin^2(2\pi w_d) \right] \quad (16)
 \end{aligned}$$

The test function dimension D is set to 30, and the maximum number of iterations $iter_{max}$ is set to 10^3 . Table 7 presents the test results, showing that the parameter ω_{min} has a small effect, and the maximum difference in the average value is only 6.92E-04. Our recommended choice for ω_{min} is 0.6 to 0.7.

For the neighborhood control parameter φ , we selected 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 for comparison. Table 8 shows the average values of these 10 groups of parameters at f9 in 30 dimensions, and each group of experiments was conducted 25 times. The effect of the parameter φ is very small, with a maximum average difference of only 1.97E-04. We recommend the selection of 20 to 30 dimensions.

C. ACCURACY COMPARISON

Tables 10, 11, and 12 show the test results of the comparison algorithms for unimodal test functions and multimodal test functions in 10, 30, and 50 dimensions for the mean, standard deviation, and optimal values of the results from 25 independent experiments. The results of the best algorithm of comparison algorithm for each test function are shown in bold [61].

The results in Table 10 show that for the 10-dimensional test functions, PSO-ATPS obtained the theoretically optimal

solutions for f1, f4 and f6, f8; PPSO obtained the best result for f9, f10; SCA found the theoretically optimal solution for f6; WOA found the theoretically optimal solution on f1, f2, and f6; GWO found the theoretically optimal solution for f1 and f6; and MPA found the theoretically optimal solution for f5, f8. Among the test functions in 10 groups, PSO-ATPS achieved first place in seven groups of tests with an overall ranking of 1, PPSO achieved first place in two groups of tests with an overall ranking of 5, SCA achieved first place in one group of tests with an overall ranking of 6, WOA achieved first place in two groups of tests with an overall ranking of 4, GWO achieved first place in two groups of tests with an overall ranking of 3, MPA achieved first place in four groups of tests with an overall ranking of 2, PSO ranked as 8, and CDIWPSO ranked as 7.

The results in Table 11 show that for the 30-dimensional test functions, PSO-ATPS achieved first place for f1 to f8 and obtained the theoretically optimal solution on all groups of tests except f5, PPSO achieved first place for f9, WOA found the theoretically optimal solution for f1 and f2 and MPA achieved first place for f6 and f8. Among the 10 groups of tested functions, PSO-ATPS achieved first place in eight groups of tests, ranked first. PPSO achieved first place in one group of tests and ranked fifth. WOA achieved first place in three groups of tests and ranked second. PSO ranked eighth. The CDIWPSO ranking is 6, the SCA ranking is 7, and the GWO ranking is 4.

The results in Table 12 show that PSO-ATPS achieved first place on all 50-dimensional test functions from f1 to f9. WOA found theoretically optimal solutions for f1, f2, and f6, and achieved first place for f10. MPA achieved first place for f6 and f8. Among the 10 groups of test functions, PSO-ATPS achieved first place in nine groups of tests, ranked first. WOA

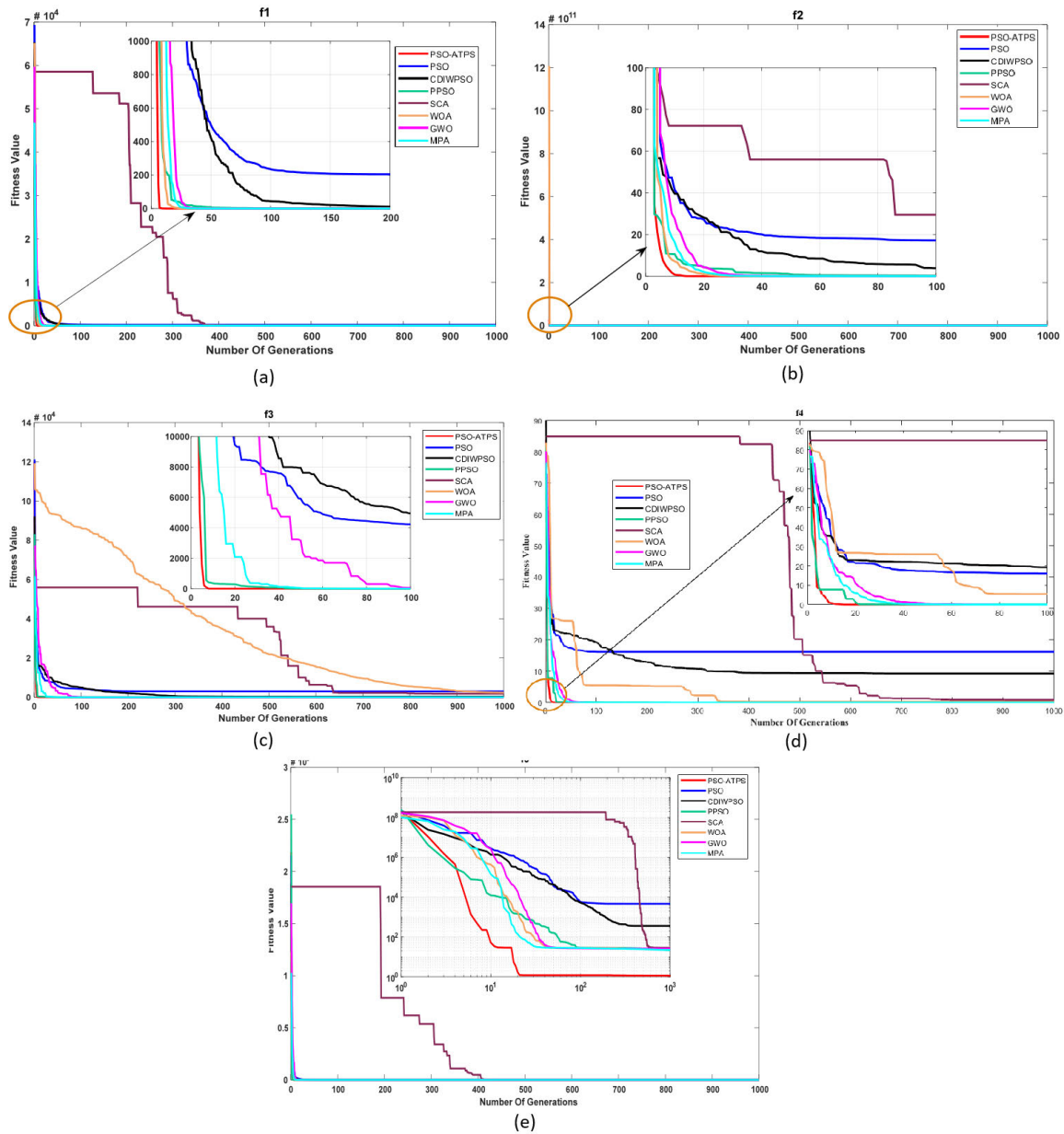


FIGURE 5. Convergence curves of competing algorithms for unimodal test functions in 30-dimensional space.

achieved first place in four groups of tests and ranked third. MPA achieved first place in two groups of tests and ranked second. PSO ranked seventh, and CDIWPSO ranked sixth. The PPSO ranking is 5, the SCA ranking is 8, and the GWO ranking is 4.

The results in Table 13 show that for the fixed-dimension test functions, PSO-ATPS obtained the theoretically optimal solution for f11, f12, f14, and f15, and obtained fourth place for f13. MPA found the theoretically optimal solution for all five problems.

The above results show that PSO-ATPS has strong local exploitation and global search capability, and because the population of PSO-ATPS is dynamically updated, it does not present a sharp decline in accuracy based on the increase in dimensions. In addition, the experimental results show that

the results of PSO increase sharply with an increase in dimensionality, and PSO-ATPS solves this problem and enhances the applicability of PSO. In tests on unimodal test functions, PSO-ATPS performs very well, which demonstrates that the new strategy shows significant improvement in the accuracy of the algorithm.

D. STABILITY ANALYSIS

On the 10-dimensional test set, the standard deviation of the results of 25 independent experiments of PSO-ATPS for f1 to f4 and f6 to f8 is all 0, indicating that PSO-ATPS is very stable in these tests. The standard deviation of PSO-ATPS for f5 is slightly higher than that of MPA but smaller than that of the other algorithms, and the PSO-ATPS standard deviation for f9 is better than that of PSO, SCA, and GWO. The PSO-ATPS

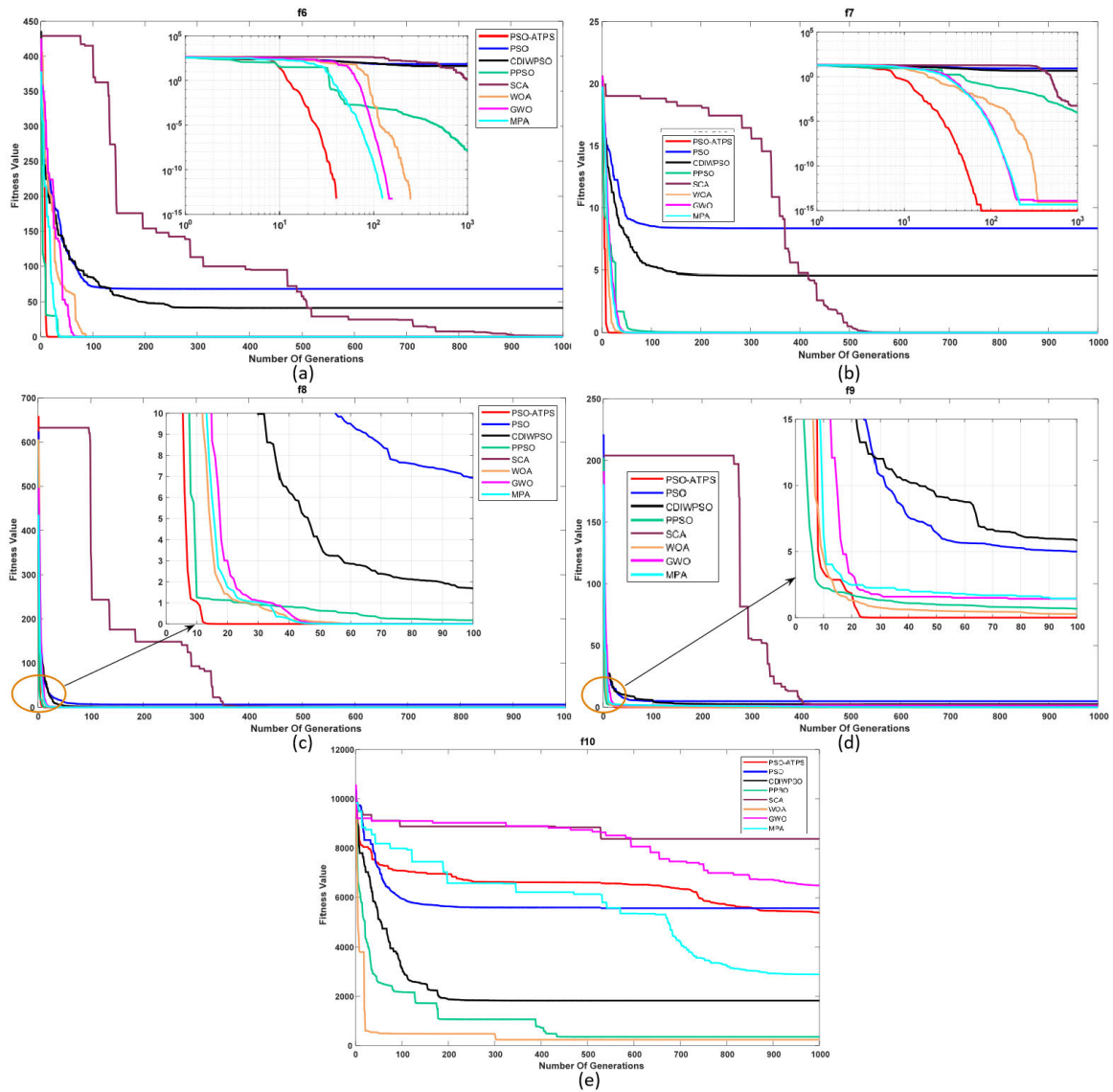


FIGURE 6. Convergence curves of competing algorithms for multimodal test functions in 30-dimensional space.

standard deviation for f10 is only better than that of WOA.

On the 30-dimensional test set, PSO-ATPS outperforms the other algorithms in terms of standard deviation of results for 25 independent experiments on f1 to f8, and only slightly outperforms PPSO on f9 but outperforms the other algorithms. The standard deviation on f10 is only better than that of WOA.

On the 50-dimensional test set, PSO-ATPS outperforms other comparative algorithms in terms of the standard deviation of results for 25 independent experiments on f1 to f9 and outperforms GWO in terms of standard deviation on f10.

In the test results of the fixed-dimension test functions, the standard deviation of the experimental results of PSO-ATPS achieved rankings of 5th, 1st, 4th, 4th and 1st place in the tests of f11 to f15, respectively. Thus, the stability of the algorithm

of PSO-ATPS algorithm is excellent, which is caused by the mechanism of PSO. It can be seen that the standard deviation of PSO and its variants in the fixed-dimension test functions is generally better than that of the other intelligent optimization algorithms.

E. COMPARISON OF CONVERGENCE SPEED

Fig. 5 shows the convergence curves of the compared algorithms on the 30-dimensional unimodal test functions. In Fig. 5. (a) to Fig. 5. (d), a part of the graph is enlarged to observe the convergence speed of the algorithm more intuitively because the algorithm converges too fast. Meanwhile, in Fig. 5. (e), a logarithmic coordinate system is used to clearly show the difference in convergence speed. From Fig. 5, it is clear that PSO-ATPS converges fastest on all five sets of test functions, indicating that PSO-ATPS demonstrates good convergence. Fig. 6 shows the convergence curves of

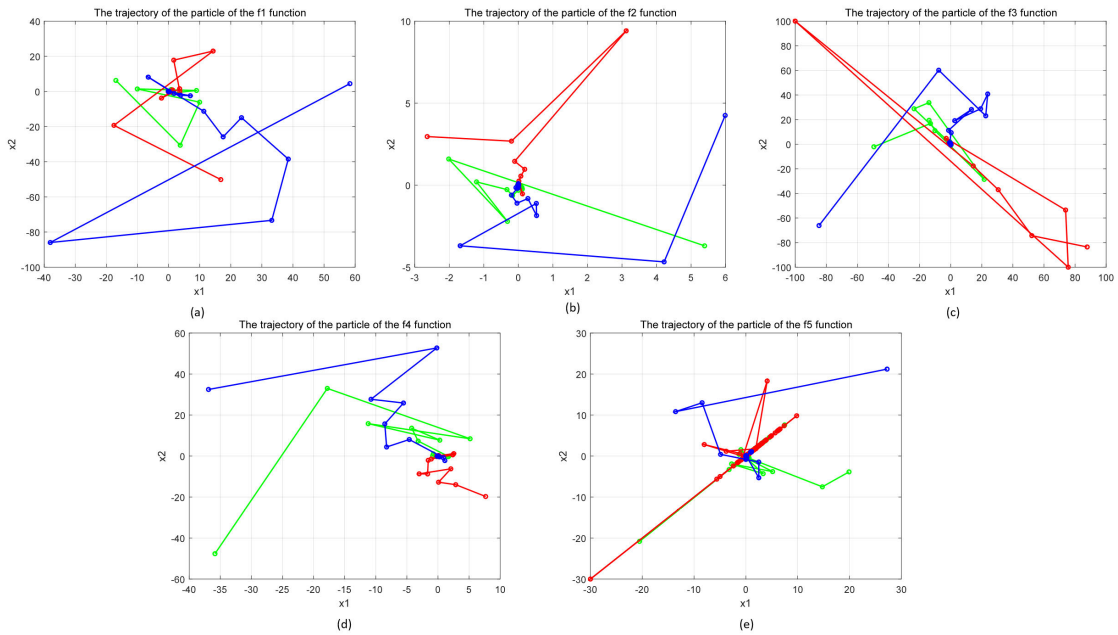


FIGURE 7. Two-dimensional trajectories of PSO-ATPS particles for unimodal test functions.

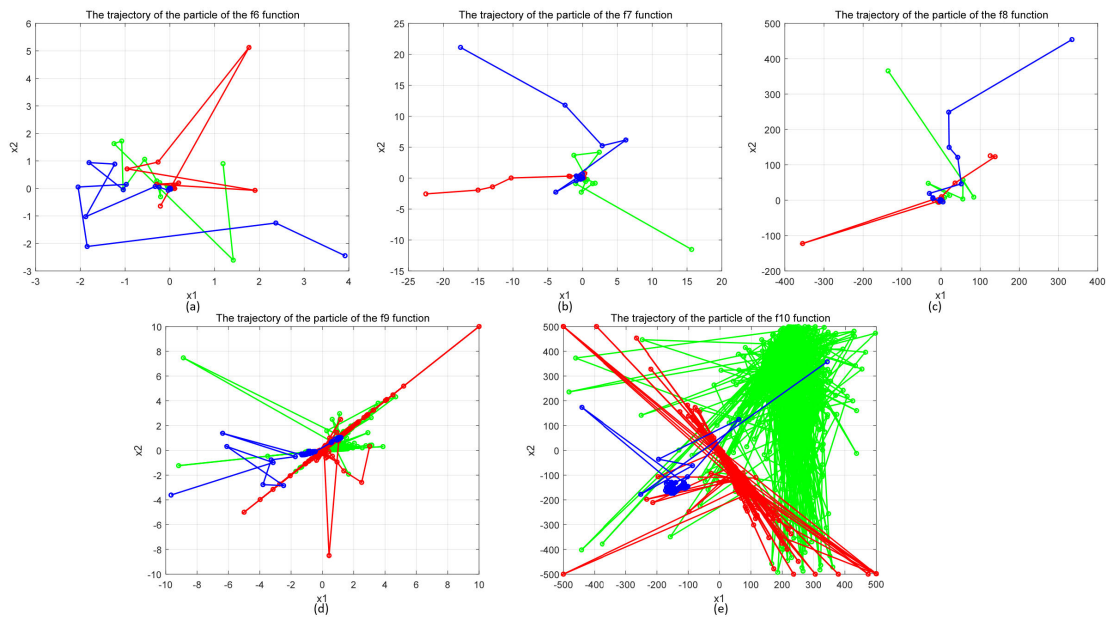


FIGURE 8. Two-dimensional trajectories of PSO-ATPS particles for multimodal test functions.

all compared algorithms on 30-dimensional multimodal test functions. In Fig. 6. (a) and Fig. 6. (b), the curves are shown in the logarithmic coordinate system, Fig. 6. (c) and Fig. 2. (d) show enlarged plots of certain intervals, and Fig. 6(e) is not adjusted because the curves are evident. From (a), (b), and (c) in Fig. 6, PSO-ATPS converges faster than other comparison algorithms on f6 and f8, and Fig. 6(d) shows that PSO-ATPS converges slightly slower than PPSO on f9 but faster than other comparison algorithms. Fig. 6. (e) clearly shows that WOA converges fastest for f10.

To have a more intuitive feeling of particle motion in PSO-ATPS, Fig. 7 to Fig. 10 show the two- and three-dimensional

particle motion trajectories of PSO-ATPS for unimodal test functions and multimodal test functions, respectively. The trajectories of Fig. 7. (a) to Fig. 7. (e) and Fig. 9. (a) to Fig. 9. (e) show that the particles search the solution space using large steps at the beginning of the algorithm. As the iterative step size undergoes irregular changes, the step size decreases to exploit the global optimum. The trajectories of the particles on the unimodal test functions shrink steadily toward the global optimum. In Fig. 8. (a) to Fig. 8. (d) and Fig. 10. (a) to Fig. 10. (d), the particles search efficiently in the global range and converge quickly to the global optimum. As the strategies employed by the particles can change at

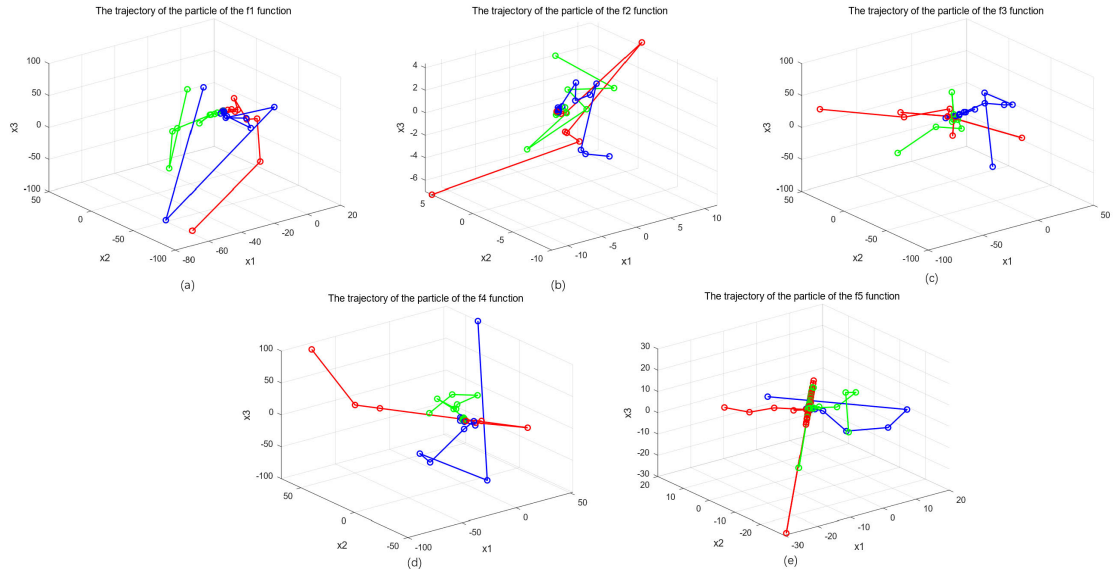


FIGURE 9. Three-dimensional trajectories of PSO-ATPS particles for unimodal test functions.

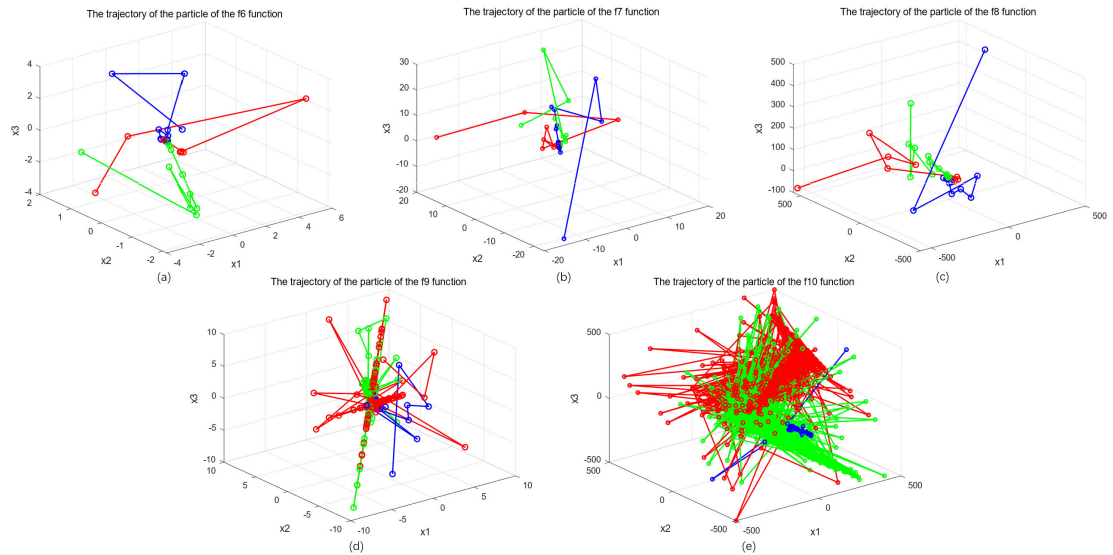


FIGURE 10. Three-dimensional trajectories of PSO-ATPS particles for multimodal test functions.

any time, each particle is either searching or developing at any given iteration, and this property of the particles is well-suited for optimization in high-dimensional complex problems. In Fig. 10. (e), three particles fall into the local optimum, but the search for the solution space is very comprehensive.

The above comparison shows that PSO-ATPS has excellent optimization accuracy with negligible errors and a very fast convergence rate. Therefore, PSO-ATPS is an effective optimization algorithm that can dynamically balance search and development.

F. DISCUSSION

This study has potential limitations. With respect to the use of an oscillation strategy, we include the trigger condition that the strategy is triggered when the solution after the present

iteration is equal to its historical optimal solution. This strategy is intended to remove the particles when they are trapped in a local optimum; however, it is continuously triggered when the algorithm has searched for a global optimum. The convergence speed of PSO-ATPS is very fast, and this leads to the oscillation strategy performing useless work after the algorithm converges to the optimal solution, which increases the time complexity of the algorithm. We believe that the solution to this problem is to propose new conditions or add new conditions so that the oscillation strategy is no longer executed after the algorithm converges, reducing the time complexity of the algorithm.

V. CONCLUSION

Inspired by biological predation behavior, we propose a new variant of PSO, PSO-ATPS, which is used to solve the

problem of prematurely falling into a local optimum, causing poor convergence accuracy of the PSO algorithm. We classify the original population according to the degree of adaptation and implement different strategies for different types of particles. To verify the performance of PSO-ATPS, we compare PSO-ATPS on a classical function test set with some newly proposed well-known optimization algorithms. The experimental results show that PSO-ATPS has great potential in treating this problem, providing good comparative results.

The main research contributions of this study are as follows. (1) address the shortcomings of the premature behavior of the PSO algorithm, we propose a new algorithmic structure, increasing the population diversity by shifting the particles between two subpopulations. The number of individuals in the two subpopulations is controlled using an adaptive classification function. (2) We propose a new oscillation strategy based on the elite strategy, which is used to solve the problem of poor solution accuracy of PSO in high-dimensional complex functions. (3) We propose a new concise velocity–position update formulation.

Based on the Occam's Razor principle, we provide a simple velocity–position update formulation, which makes PSO-ATPS very flexible. For PSO, a large number of improved versions and new strategies of metaheuristic algorithms can be easily mixed with PSO-ATPS. The capability of a single optimization algorithm is always limited; thus, it may be better to integrate the strategies of optimization algorithms. Interested readers can borrow the ideas from this paper to introduce some excellent strategies, such as the spiral search strategy of WOA and the crossover and variational strategies of GA. These strategies can be triggered under suitable conditions, making the algorithm more applicable to the situation presented.

REFERENCES

- [1] M. Jain, V. Saihjal, N. Singh, and S. B. Singh, "An overview of variants and advancements of PSO algorithm," *Appl. Sci.*, vol. 12, no. 17, p. 8392, Aug. 2022.
- [2] A. Thakkar and K. Chaudhari, "A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 2133–2164, Jun. 2021.
- [3] R. M. Rizk-Allah, A. E. Hassanien, and D. Song, "Chaos-opposition-enhanced slime mould algorithm for minimizing the cost of energy for the wind turbines on high-altitude sites," *ISA Trans.*, vol. 121, pp. 191–205, Feb. 2022.
- [4] M. Ramachandran, S. Mirjalili, M. Nazari-Heris, D. S. Parvathysankar, A. Sundaram, and C. A. R. Charles Gnanakkan, "A hybrid grasshopper optimization algorithm and Harris hawks optimizer for combined heat and power economic dispatch problem," *Eng. Appl. Artif. Intell.*, vol. 111, May 2022, Art. no. 104753.
- [5] S. Thawkar, S. Sharma, M. Khanna, and L. K. Singh, "Breast cancer prediction using a hybrid method based on butterfly optimization algorithm and ant lion optimizer," *Comput. Biol. Med.*, vol. 139, Dec. 2021, Art. no. 104968.
- [6] G. I. Sayed, M. M. Soliman, and A. E. Hassanien, "A novel melanoma prediction model for imbalanced data using optimized SqueezeNet by bald eagle search optimization," *Comput. Biol. Med.*, vol. 136, Sep. 2021, Art. no. 104712.
- [7] S. Chakraborty, A. K. Saha, S. Nama, and S. Debnath, "COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction," *Comput. Biol. Med.*, vol. 139, Dec. 2021, Art. no. 104984.
- [8] J. Xing, H. Zhao, H. Chen, R. Deng, and L. Xiao, "Boosting whale optimizer with quasi-oppositional learning and Gaussian barebone for feature selection and COVID-19 image segmentation," *J. Bionic Eng.*, vol. 20, no. 2, pp. 797–818, Mar. 2023.
- [9] J. Piri and P. Mohapatra, "An analytical study of modified multi-objective Harris hawk optimizer towards medical data feature selection," *Comput. Biol. Med.*, vol. 135, Aug. 2021, Art. no. 104558.
- [10] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Gener. Comput. Syst.*, vol. 101, pp. 646–667, Dec. 2019.
- [11] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems," *Int. J. Speech Technol.*, vol. 51, no. 3, pp. 151–155, Mar. 2021.
- [12] H. Su, D. Zhao, A. A. Heidari, L. Liu, X. Zhang, M. Mafarja, and H. Chen, "RIME: A physics-based optimization," *Neurocomputing*, vol. 532, pp. 183–214, May 2023.
- [13] H. Bayzidi, S. Talatahari, M. Sarace, and C.-P. Lamarche, "Social network search for solving engineering optimization problems," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–32, Sep. 2021.
- [14] V. Hayyolalam and A. A. Pourhaji Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103249.
- [15] M. A. Al-Betar, Z. A. A. Alyasseri, M. A. Awadallah, and I. A. Doush, "Coronavirus herd immunity optimizer (CHIO)," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5011–5042, May 2021.
- [16] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
- [17] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci., Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020.
- [18] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
- [19] A. A. Heidari, S. Mirjalili, H. Farris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
- [20] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Dec. 1995, pp. 1942–1948.
- [21] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.
- [22] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey badger algorithm: New metaheuristic algorithm for solving optimization problems," *Math. Comput. Simul.*, vol. 192, pp. 84–110, Feb. 2022.
- [23] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *J. Bionic Eng.*, vol. 18, no. 3, pp. 674–710, May 2021.
- [24] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput., IEEE World Congr. Comput. Intell.*, May 1998, pp. 69–73.
- [25] K. Chen, F. Zhou, L. Yin, S. Wang, Y. Wang, and F. Wan, "A hybrid particle swarm optimizer with sine cosine acceleration coefficients," *Inf. Sci.*, vol. 422, pp. 218–241, Jan. 2018.
- [26] M. Ghasemi, E. Akbari, A. Rahimnejad, S. E. Razavi, S. Ghavidel, and L. Li, "Phasor particle swarm optimization: A simple and efficient variant of PSO," *Soft Comput.*, vol. 23, no. 19, pp. 9701–9718, Oct. 2019.
- [27] Y. Feng, G.-F. Teng, A.-X. Wang, and Y.-M. Yao, "Chaotic inertia weight in particle swarm optimization," in *Proc. 2nd Int. Conf. Innov. Comput., Inf. Control (ICICIC)*, Sep. 2007, p. 475.
- [28] Y. Duan, N. Chen, L. Chang, Y. Ni, S. V. N. S. Kumar, and P. Zhang, "CAPSO: Chaos adaptive particle swarm optimization algorithm," *IEEE Access*, vol. 10, pp. 29393–29405, 2022.
- [29] D. Tian, X. Zhao, and Z. Shi, "Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization," *Swarm Evol. Comput.*, vol. 51, Dec. 2019, Art. no. 100573.

- [30] K. Kassoul, S. B. Belhaouari, and N. Cheikhrouhou, "Dynamic cognitive-social particle swarm optimization," in *Proc. 7th Int. Conf. Autom., Robot. Appl. (ICARA)*, Feb. 2021, pp. 200–205.
- [31] H. Moazen, S. Molaei, L. Farzinavash, and M. Sabaei, "PSO-ELPM: PSO with elite learning, enhanced parameter updating, and exponential mutation operator," *Inf. Sci.*, vol. 628, pp. 70–91, May 2023.
- [32] D. Sedighzadeh, E. Masehian, M. Sedighzadeh, and H. Akbaripour, "GEPESO: A new generalized particle swarm optimization algorithm," *Math. Comput. Simul.*, vol. 179, pp. 194–212, Jan. 2021.
- [33] Z. Liu and T. Nishi, "Adaptive heterogeneous particle swarm optimization with comprehensive learning strategy," *J. Adv. Mech. Design, Syst., Manuf.*, vol. 16, no. 4, pp. 1–15, 2022.
- [34] K. H. Lee, S. W. Baek, and K. W. Kim, "Inverse radiation analysis using repulsive particle swarm optimization algorithm," *Int. J. Heat Mass Transf.*, vol. 51, nos. 11–12, pp. 2772–2783, Jun. 2008.
- [35] S. J. Mousavirad and S. Rahnamayan, "CenPSO: A novel center-based particle swarm optimization algorithm for large-scale optimization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 2066–2071.
- [36] G. Xu, Q. Cui, X. Shi, H. Ge, Z.-H. Zhan, H. P. Lee, Y. Liang, R. Tai, and C. Wu, "Particle swarm optimization based on dimensional learning strategy," *Swarm Evol. Comput.*, vol. 45, pp. 33–51, Mar. 2019.
- [37] Z. Meng, Y. Zhong, G. Mao, and Y. Liang, "PSO-sono: A novel PSO variant for single-objective numerical optimization," *Inf. Sci.*, vol. 586, pp. 176–191, Mar. 2022.
- [38] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, and P. N. Suganthan, "A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization," *Inf. Sci.*, vol. 209, pp. 16–36, Nov. 2012.
- [39] S. Molaei, H. Moazen, S. Najjar-Ghabel, and L. Farzinavash, "Particle swarm optimization with an enhanced learning strategy and crossover operator," *Knowl.-Based Syst.*, vol. 215, Mar. 2021, Art. no. 106768.
- [40] L. Shi, J. Gong, and C. Zhai, "Application of a hybrid PSO-GA optimization algorithm in determining pyrolysis kinetics of biomass," *Fuel*, vol. 323, Sep. 2022, Art. no. 124344.
- [41] X. Chen, H. Tianfield, and W. Du, "Bee-foraging learning particle swarm optimization," *Appl. Soft Comput.*, vol. 102, Apr. 2021, Art. no. 107134.
- [42] N. Singh, S. B. Singh, and E. H. Houssein, "Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions," *Evol. Intell.*, vol. 15, pp. 1–34, Sep. 2022.
- [43] H. Hu, X. Cui, and Y. Bai, "Two kinds of classifications based on improved gravitational search algorithm and particle swarm optimization algorithm," *Adv. Math. Phys.*, vol. 2017, pp. 1–7, Sep. 2017.
- [44] T. A. Khan and S. H. Ling, "A novel hybrid gravitational search particle swarm optimization algorithm," *Eng. Appl. Artif. Intell.*, vol. 102, Jun. 2021, Art. no. 104263.
- [45] A. R. Jordehi, "Binary particle swarm optimisation with quadratic transfer function: A new binary optimisation algorithm for optimal scheduling of appliances in smart homes," *Appl. Soft Comput.*, vol. 78, pp. 465–480, May 2019.
- [46] A. R. Jordehi, "Enhanced leader particle swarm optimisation (ELPSO): An efficient algorithm for parameter estimation of photovoltaic (PV) cells and modules," *Sol. Energy*, vol. 159, pp. 78–87, Jan. 2018.
- [47] A. R. Jordehi, "Time varying acceleration coefficients particle swarm optimisation (TVACPSO): A new optimisation algorithm for estimating parameters of PV cells and modules," *Energy Convers. Manage.*, vol. 129, pp. 262–274, Dec. 2016.
- [48] G. Kumar, A. Anwar, A. Dikshit, A. Poddar, U. Soni, and W. K. Song, "Obstacle avoidance for a swarm of unmanned aerial vehicles operating on particle swarm optimization: A swarm intelligence approach for search and rescue missions," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 44, no. 2, p. 56, Feb. 2022.
- [49] A. R. Jordehi, "An improved particle swarm optimisation for unit commitment in microgrids with battery energy storage systems considering battery degradation and uncertainties," *Int. J. Energy Res.*, vol. 45, no. 1, pp. 727–744, Jan. 2021.
- [50] A. R. Jordehi, "A mixed binary-continuous particle swarm optimisation algorithm for unit commitment in microgrids considering uncertainties and emissions," *Int. Trans. Electr. Energy Syst.*, vol. 30, no. 11, Nov. 2020, Art. no. e12581.
- [51] A. R. Jordehi, J. Jasni, N. I. A. Wahab, and M. Z. A. Abd Kadir, "Particle swarm optimisation applications in FACTS optimisation problem," in *Proc. IEEE 7th Int. Power Eng. Optim. Conf. (PEOCO)*, Jun. 2013, pp. 193–198.
- [52] N. Xiao, "An algorithm for solving FEVM problem based on SPSO algorithm," *Evol. Intell.*, vol. 15, no. 4, pp. 2421–2426, Dec. 2022.
- [53] S. Zhang, "Optimal control of hybrid energy storage system of new energy power generation system based on improved particle swarm algorithm," *J. Interconnection Netw.*, vol. 22, no. 3, Sep. 2022, Art. no. 2145004.
- [54] A. R. Jordehi, "Particle swarm optimisation with opposition learning-based strategy: An efficient optimisation algorithm for day-ahead scheduling and reconfiguration in active distribution systems," *Soft Comput.*, vol. 24, no. 24, pp. 18573–18590, Dec. 2020.
- [55] Z. Xie, X. Huang, and W. Liu, "Subpopulation particle swarm optimization with a hybrid mutation strategy," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–19, Feb. 2022.
- [56] D. Brockmann, L. Hufnagel, and T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, no. 7075, pp. 462–465, Jan. 2006.
- [57] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.
- [58] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
- [59] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [60] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.
- [61] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.



MENGLING ZHAO received the bachelor's degree in applied mathematics from the Mathematics Department, North-West University, in July 1998, and the Ph.D. degree from Xidian University, Xi'an, China. She is currently the Vice Professor of science with the Xi'an University of Science and Technology, China. She has teaching experience of 25 years. She has completed five scientific research projects. She has published more than 20 articles in various journals. Her current research interests include optimization algorithms and mathematical methods in data mining.



HAONAN ZHAO is currently pursuing the M.Sc. degree with the Faculty of Science, Xi'an University of Science and Technology. His current research interests include intelligent optimization algorithms and deep learning.



MENG ZHAO is currently pursuing the M.Sc. degree with the Faculty of Science, Xi'an University of Science and Technology. Her current research interests include computer science and mathematics.

...