Information Sciences 579 (2021) 231–250

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A novel hybrid particle swarm optimization using adaptive strategy

Rui Wang, Kuangrong Hao *, Lei Chen, Tong Wang, Chunli Jiang

College of Information Science and Technology, Donghua University, Engineering Research Center of Digitized Textile and Apparel Technology, Ministry of Education, Shanghai 201620, China

ARTICLE INFO

Article history:
Received 3 February 2021
Received in revised form 27 July 2021
Accepted 30 July 2021
Available online 3 August 2021

Keywords:
Particle swarm optimization
Elite and dimensional learning
Adaptive strategy
Competitive substitution mechanism

ABSTRACT

Particle swarm optimization (PSO) has been employed to solve numerous real-world problems because of its strong optimization ability and easy implementation. However, PSO still has some shortcomings in solving complicated optimization problems, such as premature convergence and poor balance between global exploration and local exploitation. A novel hybrid particle swarm optimization using adaptive strategy (ASPSO) is developed to address associated difficulties. The contribution of ASPSO is threefold: (1) a chaotic map and an adaptive position updating strategy to balance exploration behavior and exploitation nature in the search progress; (2) elite and dimensional learning strategies to enhance the diversity of the population effectively; (3) a competitive substitution mechanism to improve the accuracy of solutions. Based on various functions from CEC 2017, the numerical experiment results demonstrate that ASPSO is significantly better than the other 16 optimization algorithms. Furthermore, we apply ASPSO to a typical industrial problem, the optimization of melt spinning progress, where the results indicate that ASPSO performs better than other algorithms.

1. Introduction

All kinds of real-world problems that exist in engineering, social and physical sciences often can be transformed into optimization problems [1]. With the increasing complexity of actual optimization problems, it is too difficult to use traditional optimization techniques to solve [2]. Therefore, the optimization methods have attracted many researchers' interest in the past few years, especially *meta*-heuristic optimization ones, for example, particle swarm optimization (PSO) [3], Grey Wolf Optimizer (GWO) [4], and artificial bee colony (ABC) algorithm [5]. Many tasks, such as feature selection [6] and data clustering [7], use these optimization algorithms. PSO is preferred and the most popular among these algorithms due to its strong optimization ability and simplicity in implementation [8].

As an efficient and intelligent optimization algorithm, PSO has received wide attention in the research field. PSO and its variants provide solutions close to the optimum, and the performances have been verified in data clustering [7] and various types of real-world problems [9]. However, PSO poses great challenges because of easily falling into the local optimum and premature convergence, especially over multimodal fitness landscapes. To end this, substantial amounts of modified versions of PSO were proposed [10–16], which can be roughly divided into four categories [17].

* Corresponding author.
E-mail address: krhao@dhu.edu.cn (K. Hao).

https://doi.org/10.1016/j.ins.2021.07.093

- Parameter setting. The proper parameters such as inertia weight ω and two acceleration coefficients c_1 and c_2 have significant effects on the convergence of the solution progress. Concerning inertia weight, several modified inertia weights, such as random, linearly decreasing [18], chaotic dynamic weight [16], and nonlinear time-varying [19], have been used to speed up the convergence rate of PSO. They stated that nonlinear time-varying and chaotic dynamic weight usually have better performance. As for two acceleration coefficients, time-varying acceleration coefficients were adopted to control the local search efficiently [20].
- Neighborhood topology. Neighborhood topology controls exploration and exploitation according to information-sharing mechanisms. Researchers have devised different neighborhood topologies that include wheel, ring [21], and Von Neumann topology. Mendes and Kennedy [22] introduced a fully informed PSO (FIPSO), which entirely used the information of the personal best positions of all topological neighbors to guide the movement of particles. Parsopoulos and Vrahatis [23] proposed a unified version (UPSO), which cleverly combined global and local PSO to synthesize their exploration and exploitation capabilities. Instead of using a fixed neighborhood topology, Nasir et al. [24] proposed a dynamic neighbor learning PSO (DNLPSO), which used a few novel strategies to select exemplar particles to update the velocity. Tanweer et al. [15] presented a new dynamic mentoring and self-regulation-based particle swarm optimization (DMeSR-PSO) algorithm using the concept of mentor and mentee.
- Learning strategy. PSO adopts different learning strategies to control exploration and exploitation that have attracted considerable attention. Liang et al. [10] presented a comprehensive learning PSO (CLPSO). CLPSO incorporated a novel learning strategy into PSO whereby all other particles' personal best information was used to update a given particle's velocity. This strategy preserved the diversity of the population and effectively avoided premature convergence. Some researchers proposed some variants of CLPSO to balance exploration and exploitation [25–28]. Nandar et al. [25] proposed the heterogeneous comprehensive learning PSO, which divided the swarm population into two subpopulations: exploration and the other to focus on exploitation. Zhang et al. [26] presented an enhanced comprehensive PSO, which used local optima topology to enlarge the particle's search space and increase the convergence speed with a certain probability. Xu et al. [27] proposed a dimensional learning PSO algorithm, in which each particle learned from the personal best experience via a dimensional learning strategy. Wang et al. [28] presented an improved-PSO algorithm, using comprehensive learning and dynamic multi-swarm strategy to construct the exploitation subpopulation exemplar and design the exploration subpopulation exemplar, respectively. Li et al. [29] proposed a multi-population cooperative PSO algorithm, which employed a multidimensional comprehensive learning strategy to improve the accuracy of solutions.
- Hybrid versions. Hybridizing PSO with other evolutionary algorithms is another focus of researchers. PSO borrowed the ideas from the genetic operators, such as selection, crossover, and mutation [13,14]. Furthermore, differential evolution [30], sine cosine algorithm [31], and ant colony optimization [32] have been introduced into PSO to solve optimization problems.

The PSO variants mentioned above have been successfully applied to solve the optimization problems in reality. However, with the increasing complexity of actual multimodal and high-dimensional optimization problems, existing algorithms cannot guarantee the great diversity and efficiency of the solutions.

To overcome the above limitations, this paper develops a novel hybrid particle swarm optimization using adaptive strategy named ASPSO. The main contributions are summarized as follows. We introduce the chaotic map to tune inertia weight ω to keep the balance between the exploration behavior and exploitation nature in the search progress. Elite and dimensional learning strategies are designed to replace the personal and global learning strategy, which enhances the diversity of the population and effectively avoids premature convergence. An adaptive position update strategy is used to improve the position quality of the next generation effectively further to balance exploration and exploitation in the search process. Finally, a competitive substitution mechanism is presented to improve the accuracy of ASPSO solutions.

This paper is structured as follows. Section 2 reviews the basic PSO. Section 3 illustrates the detailed process of the proposed ASPSO algorithm. Section 4 presents results and discussions about the proposed approach with other algorithms. In Section 5, we apply ASPSO to the engineering problem of the melt spinning process. Finally, a short conclusion is given in Section 6.

2. Particle swarm optimization (PSO)

PSO is a swarm intelligent optimization algorithm inspired by bird flocking and fish schooling [3]. In PSO, each particle represents a candidate solution with the velocity and position vectors. When searching in a D-dimensional space, a particle i is reprensed by the position $X_i^d = [x_i^1, x_i^2, \cdots, x_i^D]$ with a velocity $V_i^d = [v_i^1, v_i^2, \cdots, v_i^D]$. The velocity and the position are updated by the following formulas [18]:

$$V_i^d(t+1) = \omega(t)V_i^d(t) + c_1 * r_1 * (pbest_i^d(t) - X_i^d(t)) + c_2 * r_2 * (gbest^d(t) - X_i^d(t)) \quad (1)$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \quad (2)$$

$$\omega(t) = \omega_{max} - \frac{(\omega_{max} - \omega_{min})}{T_{max}} \cdot t \tag{3}$$

where N is the number of particles in the whole population, D refers to the dimension of each particle. ω is the inertia weight, r_1 and r_2 are random variables in the interval of $[0, 1]$, c_1 and c_2 are two positive acceleration coefficients and usually set as $c_1 = c_2 = 2$, $pbest_i^d(t) = (pbest_i^1, pbest_i^2, \ldots, pbest_i^D)$ is the personal best position of the i-th particle and $gbest^d(t) = (gbest^1, gbest^2, \ldots, gbest^D)$ is the global best position in the population. $\omega_{max} = 0.9$, $\omega_{min} = 0.4$. t and T_{max} are the current iteration and maximum iteration, respectively.

3. The proposed ASPSO algorithm

This section illustrates the proposed ASPSO algorithm in detail, as shown in Fig. 1. Inertia weight with chaotic is introduced in Section 3.1. Elite and dimensional learning strategies are described in Section 3.2. Adaptive position update strategy and competitive substitution mechanism are presented in Section 3.3 and Section 3.4, respectively.

3.1. Inertia weight with chaotic

The inertia weight ω plays a critical role in balancing exploration and exploitation in the search progress [33]. Therefore, the proper selection of parameter ω is important. Generally, linear inertia weight is adopted, but most practical scenarios in

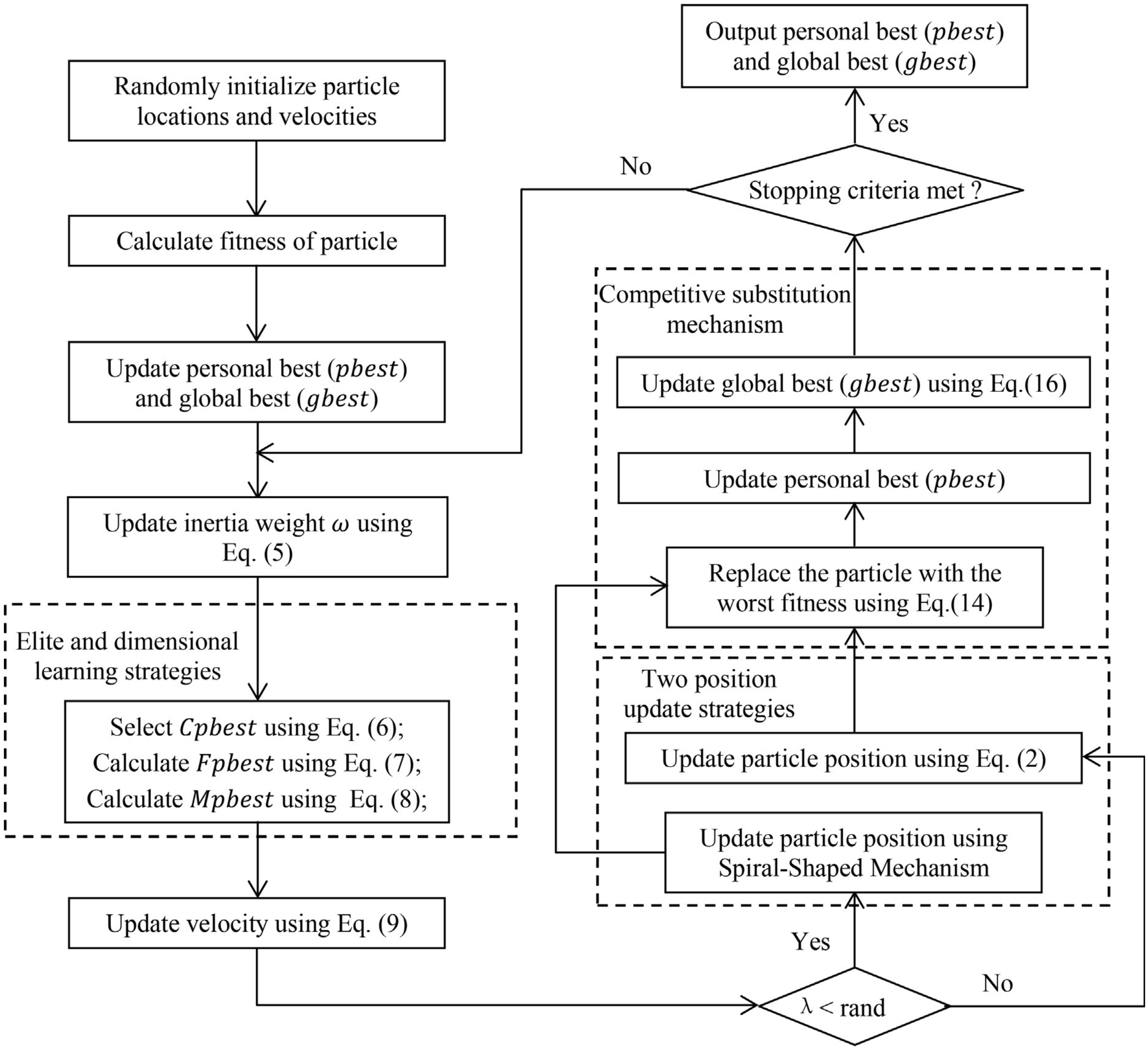


Fig. 1. The proposed ASPSO algorithm.

the real-world are complex nonlinear systems. The chaotic map has the characteristics of randomness, ergodicity, and sensitivity [34]. The algorithm named C-PSO employs the chaotic map, a nonlinear map, to adjust ω. The formula of ω is as follows:

$$x_t = A \cdot x_{t-1} \cdot (1 - x_{t-1}) x_t \in (0, 1) \tag{4}$$

$$\omega(t) = (\omega_{max} - \omega_{min}) \cdot \frac{(T_{max} - t)}{T_{max}} + \omega_{min} \cdot x_t \tag{5}$$

where A = 4.

3.2. Elite and dimensional learning strategies

The basic PSO adopts personal and global learning strategies to guide the particle's velocity and position update. That is, all particles take advantage of the swarm's best experience ($pbest_i^d$ and $gbest^d$) to accelerate the solution progress [27]. However, this strategy might lead to getting trapped in a local optimal when solving multimodal functions [10]. To end this, we introduce elite and dimensional learning strategies. In the elite learning strategy, the particles learn from other outstanding individuals to enhance the diversity of the population.

During the search progress, each particle i will learn from four different $pbest_i^d$ particles that are randomly selected from the population (we will discuss the number of learning particles M in Section 4.2.2). Then, the personal best particle i is compared with the chosen four particles, and the particle with best fitness value will be kept as the personal best($Fpbest_i^d$). The learning strategy is expressed by:

$$Cpbest(t) = argmin\{f(pbest_a^d(t)), f(pbest_b^d(t)), \cdots, f(pbest_d^d(t))\} \quad a \neq b \neq c \neq d \tag{6}$$

$$Fpbest_i^d(t) = \begin{cases} Cpbest(t), & f(Cpbest(t)) < f(pbest_i^d(t)) \\ pbest_i^d(t), & otherwise \end{cases} \tag{7}$$

where $f(\cdot)$ is the fitness function.

The overemphasis of $gbest^d$ will rapidly cause the population's diversity, and we use the dimensional learning method to solve this potential problem. By promoting communications between particles in the dimensional aspect, the mean value provides additional information to increase diversity and improve search efficiency. A global particle denoted as $Mpbest^d$, is defined as follows:

$$Mpbest^d(t) = \left(\frac{1}{N}\sum\nolimits_{i=1}^{N} pbest_i^1(t), \frac{1}{N}\sum\nolimits_{i=1}^{N} pbest_i^2(t), \cdots, \frac{1}{N}\sum\nolimits_{i=1}^{N} pbest_i^D(t)\right) \tag{8}$$

Above all, we change the velocity update equation to:

$$V_i^d(t+1) = \omega(t) V_i^d(t) + c_1 * r_1 * \left(Fpbest_i^d(t) - X_i^d(t)\right) + c_2 * r_2 * \left(Mpbest^d(t) - X_i^d(t)\right) \tag{9}$$

3.3. Adaptive position update strategy

The basic PSO cannot effectively balance exploration and exploitation in the search process. The position update law makes the particles always move toward the previous best position, reducing the ability to search neighborhoods around the known optimal solution [35]. A spiral-shaped mechanism is introduced as a local search operator around the known optimal solution region [36]. Inspired by that, we propose the adaptive position update strategy to generate particle positions based on local exploitation or global exploration, which are expressed by

$$\lambda = \frac{exp(f(X_i^d(t)))}{exp\left(\frac{1}{N}\sum_{i=1}^{N} f(X_i^d(t))\right)} \tag{10}$$

$$X_i^d(t+1) = \begin{cases} D_1 \cdot exp(b \cdot l) \cdot cos(2\pi l) + gbest^d(t), \lambda < r \text{ and} \\ X_i^d(t) + V_i^d(t+1), \quad otherwise \end{cases} \tag{11}$$

where $D_1 = \left|gbest^d(t) - X_i^d\right|$ represents the distance between the current best location and the i-th particle, b is a constant to control the shape of the logarithmic spiral, l is a random number, $l \in [-1, 1]$.

In each iteration, a ratio λ is obtained by calculating the fitness value of the current particle and the corresponding average fitness value. If λ is small, the particle is close to the optimal position and needs to enhance local exploitation ability. On the contrary, the particle is in a poor position and will be updated to improve the global exploration ability to discourage premature convergence.

3.4. Competitive substitution mechanism

A competitive substitution mechanism is introduced to improve the performance of PSO, called CS-PSO. The worst particle ($WX_i^d(t)$) will be substituted in each iteration, defined as

$$WX_i^d(t) = argmax\left\{f(X_1^d(t)),f(X_2^d(t)),...,f(X_N^d(t))\right\} \tag{12}$$

Let $pbest_e^d$ and $pbest_f^d$ be the personal best positions of two particles randomly selected from the population. $NX_i^d(t)$ refers to the new position of i-th particle, which is defined as Eq. (13). The substitution mechanism is defined as Eq. (14).

$$NX_i^d(t) = gbest^d(t) + r_3 \cdot (pbest_e^d(t) - pbest_f^d(t)), \quad e \neq f \neq i \in [1,2,...,N] \tag{13}$$

$$WX_i^d(t) = \begin{cases} NX_i^d(t), & f(NX_i^d(t)) < f(WX_i^d(t)) \\ WX_i^d(t), & otherwise \end{cases} \tag{14}$$

where $r_3 \in (0,1)$ is a random number.

In the search process, all particles in the population will learn from the $gbest^d$ particle, thus the $gbest^d$ has a significant influence on the population. In a complex search environment, once the $gbest^d$ falls into a local optimum, the remaining particles tend to converge to the sub-optimal region, leading to premature convergence. Therefore, a disturbance strategy is incorporated into ASPSO to help the $gbest^d$ escape from the local optimum. To minimize the time wasted on poor directions, we set a condition to trigger the disturbance strategy if the$gbest^d$ does not update the value after ten iterations. The disturbance strategy is given below:

$$Nbest(t) = r_4 \cdot gbest^d(t) + (1 - r_4) \cdot (gbest^d(t) - pbest_i^d(t)), \quad i \in [1,2,\cdots,N] \tag{15}$$

$$gbest^d(t) = \begin{cases} Nbest(t), & f(Nbest(t)) < f(gbest^d(t)) \\ gbest^d(t), & otherwise \end{cases} \tag{16}$$

where $r_4 \in (0,1)$ is a random parameter.

4. Experimental verification and analysis

4.1. Benchmark functions and comparison algorithm

The performance of the proposed ASPSO is tested in the CEC2017 benchmark functions [37]. Among thirty functions, F2 is excluded in this experimentation because it shows unstable behavior, especially in high dimensions. The benchmark functions are divided into four categories: unimodal functions (F1–F3), simple multimodal functions (F4 - F10), hybrid functions (F11–F20), and composition functions (F21–F30), as stated in Table 1.

To validate the performances of the ASPSO, we chose eight representative PSO variations and eight state-of-the-art evolutionary algorithms. The PSO variants and evolutionary algorithms adopt the recommended parameter of their original literature and are presented in Table 2. The population size, dimension, and limit iterations are set on the same page as 50, 30, and 1000. All algorithms are run independently thirty times on each benchmark function. The nonparametric statistical tests, i.e., the Wilcoxon signed-rank test and Friedman test [38], are used to make the comparison more convincing.

4.2. Characteristic test

4.2.1. Effects of ASPSO components

The main components of the ASPSO algorithm are (1) chaotic map, (2) elite and dimensional learning strategies, (3) adaptive position update strategy, and (4) competitive substitution mechanism. To demonstrate the effectiveness of each component, the algorithms named C-PSO, ED + A-PSO, CS-PSO, and ASPSO are tested and compared with PSO. The mean values of thirty runs are presented in Table 3. The best results obtained by the five algorithms are shown in bold. The "Best" row accounts for the times that the corresponding algorithm produces the best solution.

On all tested functions, PSO, C-PSO, ED + A-PSO, CS-PSO, and ASPSO exhibit the best performance on zero, zero, one, three, and twenty-six functions, respectively. ASPSO performs significantly better than PSO on twenty-nine functions. C-PSO performs significantly better and significantly worse than PSO on twenty-eight functions and one function, respectively. ED + A-PSO performs significantly better and significantly worse than PSO on twenty-one functions and eight functions, respectively. CS-PSO performs significantly better than and equivalently to PSO on twenty-eight functions and one function, respectively. The experimental results show that each strategy is effective on the test functions.

Fig. 2 shows the convergence curves of the five algorithms on unimodal function F1, multimodal function F9, hybrid function F17, and composition function F24. Fig. 2(a) clearly shows that CS-PSO has better search accuracy than the other three

Table 1
Details of CEC2017 benchmark functions.

	NO.	Functions	Search Ranges	F^*=$F(x^*)$
Unimodal functions	1	Shifted and Rotated Bent Cigar Function	$[-100,100]^D$	100
	3	Shifted and Rotated Zakharov Function	$[-100,100]^D$	300
Simple Multimodal functions	4	Shifted and Rotated Rosenbrock's Function	$[-100,100]^D$	400
	5	Shifted and Rotated Rastrigin's Function	$[-100,100]^D$	500
	6	Shifted and Rotated Expanded Scaffer's F6 Function	$[-100,100]^D$	600
	7	Shifted and Rotated Lunacek Bi-Rastrigin Function	$[-100,100]^D$	700
	8	Shifted and Rotated Non-Continuous Rastrigin's Function	$[-100,100]^D$	800
	9	Shifted and Rotated Levy Function	$[-100,100]^D$	900
	10	Shifted and Rotated Schwefel's Function	$[-100,100]^D$	1000
Hybrid functions	11	Hybrid Function 1 (N = 3)	$[-100,100]^D$	1100
	12	Hybrid Function 2 (N = 3)	$[-100,100]^D$	1200
	13	Hybrid Function 3 (N = 3)	$[-100,100]^D$	1300
	14	Hybrid Function 4 (N = 4)	$[-100,100]^D$	1400
	15	Hybrid Function 5 (N = 4)	$[-100,100]^D$	1500
	16	Hybrid Function 6 (N = 4)	$[-100,100]^D$	1600
	17	Hybrid Function 6 (N = 5)	$[-100,100]^D$	1700
	18	Hybrid Function 6 (N = 5)	$[-100,100]^D$	1800
	19	Hybrid Function 6 (N = 5)	$[-100,100]^D$	1900
	20	Hybrid Function 6 (N = 6)	$[-100,100]^D$	2000
Composition functions	21	Composition Function 1 (N = 3)	$[-100,100]^D$	2100
	22	Composition Function 2 (N = 3)	$[-100,100]^D$	2200
	23	Composition Function 3 (N = 4)	$[-100,100]^D$	2300
	24	Composition Function 4 (N = 4)	$[-100,100]^D$	2400
	25	Composition Function 5 (N = 5)	$[-100,100]^D$	2500
	26	Composition Function 6 (N = 5)	$[-100,100]^D$	2600
	27	Composition Function 7 (N = 6)	$[-100,100]^D$	2700
	28	Composition Function 8 (N = 6)	$[-100,100]^D$	2800
	29	Composition Function 9 (N = 3)	$[-100,100]^D$	2900
	30	Composition Function 10 (N = 3)	$[-100,100]^D$	3000

Note: x^* stands for the global optima. $F(\cdot)$ is the fitness value. *F2 has been excluded because it shows unstable behavior.

Table 2
Parameter settings of various algorithms.

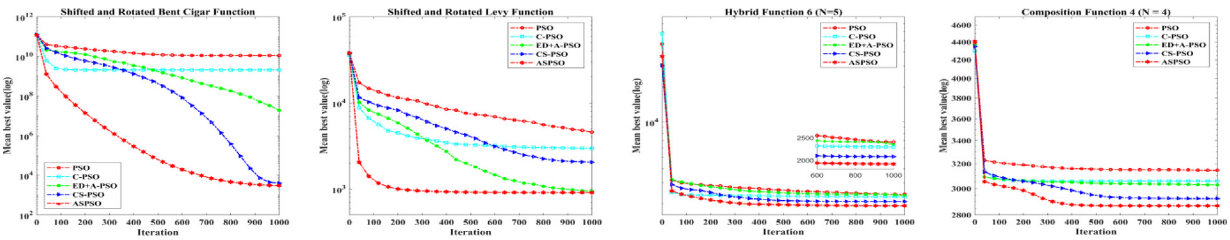
algorithm	Parameter settings	year	Refs.
PSO	ω = 0.9 ~ 0.4, c_1 = c_2 = 2	1998	[18]
FDR_PSO	ω = 0.4 ~ 0.9, c_1 = c_2 = 1, c_3 = 2	2003	[39]
CLPSO	ω = 0.9 ~ 0.4, c = 1.49445, m = 7	2006	[10]
DNLPSO	ω = 0.9 ~ 0.4, c_1 = c_2 = 1.49445	2012	[24]
LIPSO	c = 2, N = 3	2013	[40]
HCLPSO	ω = 0.99 ~ 0.2, c_1= 2.5 ~ 0.5, c_2 = 0.5 ~ 2.5, c = 3 ~ 1.5	2015	[25]
EPSO		2017	[41]
TCSPSO	ω = 0.9 ~ 0.4, c_1 = c_2 = 2	2019	[42]
ABC	*limit* = 100, *size of employed-bee* = $N/2$	2007	[5]
BBO	*pMutation* = 0.1	2008	[43]
BSA	*mixrate* = 1.0	2013	[44]
ISA	$\lambda = 0.01(UB - LB)$	2014	[45]
CSA	*pa* = 0.25	2014	[46]
GWO	a = 2 ~ 0, b = 1	2014	[4]
VSA	x = 0.1, *ginv* = $(1/x)$**gammaincinv*$(x,1)$	2015	[47]
MVO	WEP = 0.2 ~ 1, TDR = 0.6 ~ 0	2016	[48]

methods. From Fig. 2(b), it can be seen that ED + A-PSO has the best performance for this benchmark function. Fig. 2(c) shows that C-PSO has the fastest convergence speed initially. Fig. 2(d) illustrates that CS-PSO has a faster convergence speed and better search accuracy when using F24. However, the performance of PSO combined with any single strategy on these test functions is not as good as that of ASPSO.

Wilcoxon signed-rank test is used to evaluate the performance of ASPSO and its peers with a significance level of 5%, i.e., $\alpha = 0.05$. The symbols "+", "−" and "=" indicate that PSO performs significantly better than, markedly worse than, and ties with the compared algorithm, respectively. Table 4 shows that these several strategies have improved the performance of PSO. The Friedman test in Table 5 shows that the competition substitution mechanism has the most excellent effect on improving PSO performance among these several strategies.

Table 3
The test results of PSO-based algorithms for CEC2017 benchmark functions.

Func.	PSO	C-PSO	ED + A-PSO	CS-PSO	ASPSO
F1	1.11E + 10	2.06E + 09	1.91E + 07	3.98E + 03	**3.02E + 03**
F3	8.46E + 04	5.29E + 04	5.54E + 04	**3.29E + 03**	4.08E + 03
F4	1.50E + 03	7.79E + 02	5.08E + 02	**4.90E + 02**	5.08E + 02
F5	6.58E + 02	6.19E + 02	7.00E + 02	6.20E + 02	**5.43E + 02**
F6	6.19E + 02	6.06E + 02	6.06E + 02	6.04E + 02	**6.01E + 02**
F7	9.79E + 02	8.63E + 02	9.49E + 02	8.87E + 02	**7.76E + 02**
F8	9.44E + 02	9.12E + 02	1.00E + 03	9.16E + 02	**8.44E + 02**
F9	4.57E + 03	2.96E + 03	9.44E + 02	2.05E + 03	**9.12E + 02**
F10	4.80E + 03	4.65E + 03	8.25E + 03	4.35E + 03	**4.32E + 03**
F11	1.65E + 03	1.38E + 03	1.34E + 03	1.24E + 03	**1.20E + 03**
F12	9.42E + 08	1.41E + 08	3.50E + 06	4.49E + 05	**3.50E + 05**
F13	1.69E + 08	5.92E + 06	6.32E + 04	1.72E + 04	**1.66E + 04**
F14	3.09E + 05	1.89E + 05	1.73E + 05	4.95E + 03	**4.62E + 03**
F15	1.30E + 05	4.57E + 04	1.85E + 06	8.60E + 03	**5.25E + 03**
F16	2.96E + 03	2.84E + 03	3.10E + 03	2.51E + 03	**2.33E + 03**
F17	2.41E + 03	2.29E + 03	2.35E + 03	2.08E + 03	**1.91E + 03**
F18	2.81E + 06	1.46E + 06	5.71E + 06	1.19E + 05	**9.32E + 04**
F19	1.63E + 07	5.48E + 06	2.86E + 04	1.17E + 04	**5.82E + 03**
F20	2.36E + 03	2.48E + 03	2.57E + 03	2.36E + 03	**2.20E + 03**
F21	2.45E + 03	2.43E + 03	2.50E + 03	2.40E + 03	**2.34E + 03**
F22	6.06E + 03	5.18E + 03	3.77E + 03	3.18E + 03	**2.30E + 03**
F23	2.97E + 03	2.87E + 03	2.87E + 03	2.76E + 03	**2.69E + 03**
F24	3.15E + 03	3.06E + 03	3.03E + 03	2.92E + 03	**2.87E + 03**
F25	3.20E + 03	2.98E + 03	2.91E + 03	2.90E + 03	**2.89E + 03**
F26	6.85E + 03	5.68E + 03	5.66E + 03	4.31E + 03	**3.86E + 03**
F27	3.34E + 03	3.29E + 03	**3.21E + 03**	3.25E + 03	3.22E + 03
F28	4.21E + 03	3.80E + 03	3.36E + 03	**3.23E + 03**	**3.23E + 03**
F29	4.24E + 03	3.91E + 03	4.11E + 03	3.76E + 03	**3.56E + 03**
F30	6.00E + 06	1.19E + 06	4.51E + 05	1.34E + 04	**8.39E + 03**
Best	0	0	1	3	26



(a) F1 (b) F9 (c) F17 (d) F24

Fig. 2. Comparison of performances for the tested functions (F1, F9, F17, and F24).

Table 4
Wilcoxon signed-rank test of PSO-based algorithms for CEC2017 benchmark functions.

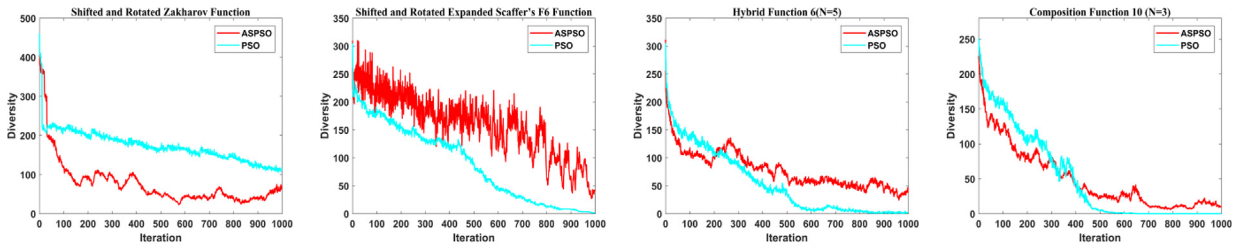
PSO vs.	C-PSO	ED + A-PSO	CS-PSO	ASPSO
R+	424	333	435	435
R-	11	102	0	0
+	1	7	0	0
–	25	17	28	29
=	3	5	1	0
p-value	2.05E–07	1.13E–02	3.73E–09	3.73E–09

Table 5
Friedman test of PSO-based algorithms for CEC2017 benchmark functions.

	PSO	C-PSO	ED + A-PSO	CS-PSO	ASPSO
Friedman rank	4.93	2.48	3.90	2.38	1.31
Final rank	5	3	4	2	1
p-value	8.04E-11				

Table 6
The rank values produced by ASPSO using different numbers of M.

	M = 2	M = 3	M = 4	M = 5	M = 6
Total rank	74	72	52	67	62
Final rank	5	4	1	3	2



(a) F3 (b) F6 (c) F19 (d) F30

Fig. 3. Diversity curves.

4.2.2. Parameter M

Parameter M determines the ability of the particle to learn from other outstanding individuals and has an important implication on the solutions. Thus, different values of M are selected and executed thirty times based on the CEC2017 benchmark functions, and the results are shown in Table 6. The total rank is the sum of the rankings of various algorithms on the test function. The row of "Total rank" denotes the sum of the ranking of various algorithms on the test functions.

A smaller number of learning particles will reduce opportunities for particles to learn from other excellent particles, resulting in reduced population diversity. For example, ASPSO with M = 2 produces the worst performance for the benchmark functions. Similar experiments are conducted for M = 5 and 6. The large value will increase the computational burden, but the experimental results are not ideal. Based on the simulation results summarized in Table 6, M = 4 is reported as an appropriate parameter setting.

4.2.3. Analysis of diversities

ASPSO is proposed to balances the exploration and exploitation in the search progress. The diversity evaluates the ability of exploitation and exploration accurately [28] and is measured by

$$diversity(N) = \frac{1}{N}\sum\nolimits_{i=1}^{N}\sqrt{\sum\nolimits_{j=1}^{D}\left(X_i^d - \overline{X^d}\right)^2} \tag{17}$$

$$\overline{X^d} = \frac{1}{N}\sum\nolimits_{i=1}^{N} X_i^d(t) \tag{18}$$

where X_i^d is the d-th dimension of particle i and $\overline{X^d}$ is the d-th dimension of the mean of the population.

The lack of diversity makes particles more likely to fall into local optima and lead to premature convergence. As shown in Fig. 3, the unimodal (F3), multimodal (F6), hybrid (F19), and composition (F30) functions are selected for the diversity comparison of ASPSO and PSO. The parameter settings remain the same as Section 4.1. On F3, the diversity of ASPSO is lower than that of PSO, but it is full of fluctuations throughout the iteration process, which is conducive to enhancing local exploitation. On F6, F19, and F30, ASPSO maintains the high diversity as expected due to the adoption of nonlinear inertia weight and adaptive position update strategy. These strategies enable ASPSO to maintain a better balance between exploration and exploitation, thereby reducing the probability of tending to local optimal. From Table 7, it is clear that ASPSO has a higher convergence accuracy than PSO.

4.3. Performance evaluation of ASPSO

4.3.1. Qualitative analysis of ASPSO

We select four functions for qualitative analysis of ASPSO. The iteration limit is set to 200, the number of particles 5, and D = 2. Some simulation graphs are presented in Figs. 4–7 include the convergence curve and trajectory of the first particle in two dimensions. The trajectory curve fluctuates violently in the initial stage, which indicates that the particles search for the optimal solution. Then, it tends to stabilize, which states that the particles reach the global or local optimum. Figs. 4–7 shows that ASPSO has a high-speed convergence rate on these four functions, finding the global optimum after a few iterations. The experimental results show that the ASPSO algorithm has good performance on these functions.

Table 7
Comparisons of experimental results between ASPSO with some well-known variants of PSO.

Func.	Criteria	PSO	FDR_PSO	CLPSO	DNLPSO	LIPSO	HCLPSO	EPSO	TCSPSO	ASPSO
F1	Mean	1.11E + 10	4.03E + 03	1.61E + 07	1.35E + 08	7.83E + 03	**2.64E + 03**	3.64E + 03	2.63E + 04	3.02E + 03
	Std	6.54E + 09	4.01E + 03	5.24E + 06	5.33E + 07	2.03E + 04	2.98E + 03	3.84E + 03	6.13E + 04	**1.66E + 03**
	Rank	9	4	7	8	5	1	3	6	2
F3	Mean	8.46E + 04	1.80E + 04	1.15E + 05	5.84E + 04	9.26E + 04	1.29E + 04	3.56E + 04	2.30E + 04	**4.08E + 03**
	Std	2.13E + 04	7.50E + 03	1.61E + 04	1.94E + 04	1.81E + 04	4.23E + 03	1.07E + 04	5.61E + 03	**1.70E + 03**
	Rank	7	3	9	6	8	2	5	4	1
F4	Mean	1.50E + 03	4.85E + 02	5.85E + 02	**4.63E + 02**	5.91E + 02	4.90E + 02	4.88E + 02	6.40E + 02	5.08E + 02
	Std	8.69E + 02	2.84E + 01	1.58E + 01	4.31E + 01	6.80E + 01	2.97E + 01	2.58E + 01	9.78E + 01	**1.52E + 01**
	Rank	9	2	6	1	7	4	3	8	5
F5	Mean	6.58E + 02	5.67E + 02	6.45E + 02	6.90E + 02	5.57E + 02	5.51E + 02	5.71E + 02	6.07E + 02	**5.43E + 02**
	Std	3.94E + 01	2.14E + 01	**1.32E + 01**	2.33E + 01	1.76E + 01	1.63E + 01	1.51E + 01	2.16E + 01	1.66E + 01
	Rank	8	4	7	9	3	2	5	6	1
F6	Mean	6.19E + 02	**6.00E + 02**	6.03E + 02	6.08E + 02	6.08E + 02	6.01E + 02	6.02E + 02	6.05E + 02	6.01E + 02
	Std	7.44E + 00	**3.59E-01**	5.47E-01	1.56E + 00	3.64E + 00	5.81E-01	1.55E + 00	3.48E + 00	3.89E-01
	Rank	7	1	4	6	6	2	3	5	2
F7	Mean	9.79E + 02	7.88E + 02	8.96E + 02	9.73E + 02	7.95E + 02	8.00E + 02	8.02E + 02	8.59E + 02	**7.76E + 02**
	Std	1.19E + 02	1.93E + 01	1.68E + 01	2.22E + 01	2.11E + 01	1.88E + 01	1.98E + 01	3.01E + 01	**1.37E + 01**
	Rank	9	2	7	8	3	4	5	6	1
F8	Mean	9.44E + 02	8.53E + 02	9.48E + 02	9.85E + 02	8.58E + 02	8.50E + 02	8.60E + 02	8.92E + 02	**8.44E + 02**
	Std	3.33E + 01	1.44E + 01	1.59E + 01	2.58E + 01	1.39E + 01	1.59E + 01	**1.24E + 01**	2.43E + 01	1.75E + 01
	Rank	7	3	8	9	4	2	5	6	1
F9	Mean	4.57E + 03	9.18E + 02	3.12E + 03	1.22E + 03	1.37E + 03	9.13E + 02	1.03E + 03	1.77E + 03	**9.12E + 02**
	Std	2.16E + 03	2.76E + 01	5.42E + 02	2.67E + 02	3.89E + 02	1.51E + 01	7.75E + 01	7.27E + 02	**1.29E + 01**
	Rank	9	3	8	5	6	2	4	7	1
F10	Mean	4.80E + 03	4.28E + 03	6.31E + 03	8.32E + 03	**3.79E + 03**	4.21E + 03	5.39E + 03	4.96E + 03	4.32E + 03
	Std	8.31E + 02	5.41E + 02	**2.95E + 02**	3.11E + 02	3.99E + 02	7.65E + 02	1.32E + 03	6.98E + 02	6.59E + 02
	Rank	5	3	8	9	1	2	7	6	4
F11	Mean	1.65E + 03	**1.20E + 03**	1.59E + 03	1.41E + 03	1.36E + 03	**1.20E + 03**	1.21E + 03	1.29E + 03	**1.20E + 03**
	Std	3.21E + 02	3.70E + 01	1.54E + 02	6.97E + 01	1.41E + 02	3.39E + 01	3.97E + 01	8.79E + 01	**3.30E + 01**
	Rank	7	1	6	5	4	1	2	3	1
F12	Mean	9.42E + 08	**1.29E + 05**	1.87E + 07	2.33E + 07	2.47E + 06	5.06E + 05	2.43E + 05	1.43E + 07	3.50E + 05
	Std	1.37E + 09	**8.03E + 04**	6.46E + 06	1.21E + 07	6.65E + 06	4.64E + 05	1.96E + 05	5.29E + 07	2.08E + 05
	Rank	9	1	7	8	5	4	2	6	3
F13	Mean	1.69E + 08	1.17E + 04	3.69E + 06	1.68E + 06	**5.12E + 03**	1.29E + 04	1.36E + 04	1.33E + 04	1.66E + 04
	Std	4.73E + 08	8.50E + 03	2.69E + 06	1.16E + 06	**3.97E + 03**	1.01E + 04	1.20E + 04	1.16E + 04	1.33E + 04
	Rank	9	2	8	7	1	3	5	4	6
F14	Mean	3.09E + 05	1.66E + 04	1.21E + 05	7.98E + 04	7.40E + 04	2.20E + 04	2.98E + 04	4.14E + 04	**4.62E + 03**
	Std	6.34E + 05	1.73E + 04	9.62E + 04	7.70E + 04	6.95E + 04	2.01E + 04	3.27E + 04	4.10E + 04	**2.97E + 03**
	Rank	9	2	8	7	6	3	4	5	1
F15	Mean	1.30E + 05	6.65E + 03	7.14E + 04	3.88E + 05	**3.05E + 03**	5.47E + 03	6.41E + 03	9.49E + 03	5.25E + 03
	Std	9.24E + 04	6.02E + 03	5.48E + 04	2.58E + 05	**1.55E + 03**	3.75E + 03	6.01E + 03	8.44E + 03	5.55E + 03
	Rank	8	5	7	9	1	3	4	6	2
F16	Mean	2.96E + 03	2.37E + 03	2.60E + 03	3.12E + 03	2.32E + 03	**2.30E + 03**	2.43E + 03	2.81E + 03	2.33E + 03
	Std	2.68E + 02	2.91E + 02	**1.59E + 02**	2.89E + 02	1.66E + 02	2.58E + 02	2.54E + 02	4.27E + 02	2.44E + 02
	Rank	8	4	6	9	2	1	5	7	3
F17	Mean	2.41E + 03	1.94E + 03	1.95E + 03	2.21E + 03	1.97E + 03	**1.86E + 03**	2.00E + 03	2.08E + 03	1.91E + 03
	Std	2.47E + 02	1.12E + 02	**8.34E + 01**	1.58E + 02	1.14E + 02	1.07E + 02	1.55E + 02	1.80E + 02	1.71E + 02
	Rank	9	3	4	8	5	1	6	7	2
F18	Mean	2.81E + 06	2.74E + 05	5.68E + 05	2.05E + 06	2.43E + 05	3.01E + 05	6.60E + 05	6.46E + 05	**9.32E + 04**
	Std	4.01E + 06	1.73E + 05	3.88E + 05	1.52E + 06	2.21E + 05	2.57E + 05	9.66E + 05	5.42E + 05	**7.02E + 04**
	Rank	9	3	5	8	2	4	7	6	1
F19	Mean	1.63E + 07	7.86E + 03	7.75E + 04	1.82E + 05	**2.97E + 03**	8.68E + 03	5.79E + 03	1.27E + 04	5.82E + 03
	Std	2.67E + 07	8.36E + 03	7.95E + 04	1.11E + 05	**2.01E + 03**	6.72E + 03	4.34E + 03	1.21E + 04	3.46E + 03
	Rank	9	4	7	8	1	5	2	6	3
F20	Mean	2.36E + 03	2.28E + 03	2.35E + 03	2.59E + 03	2.31E + 03	2.26E + 03	2.31E + 03	2.41E + 03	**2.20E + 03**
	Std	1.56E + 02	1.13E + 02	**8.67E + 01**	1.50E + 02	1.05E + 02	9.44E + 01	1.31E + 02	1.66E + 02	1.16E + 02
	Rank	6	3	5	8	4	2	4	7	1
F21	Mean	2.45E + 03	2.36E + 03	2.45E + 03	2.48E + 03	2.36E + 03	2.35E + 03	2.36E + 03	2.40E + 03	**2.34E + 03**
	Std	3.41E + 01	1.59E + 01	**1.39E + 01**	2.78E + 01	1.45E + 01	1.68E + 01	1.78E + 01	3.29E + 01	1.40E + 01
	Rank	5	3	5	6	3	2	3	4	1
F22	Mean	6.06E + 03	3.38E + 03	3.55E + 03	7.83E + 03	2.53E + 03	**2.30E + 03**	**2.30E + 03**	3.08E + 03	**2.30E + 03**
	Std	1.51E + 03	1.71E + 03	8.78E + 02	3.37E + 03	7.16E + 02	**8.38E-01**	1.21E + 00	1.59E + 03	9.79E-01
	Rank	6	4	5	7	2	1	1	3	1
F23	Mean	2.97E + 03	2.71E + 03	2.81E + 03	2.84E + 03	2.75E + 03	2.71E + 03	2.73E + 03	2.86E + 03	**2.69E + 03**
	Std	9.45E + 01	1.73E + 01	1.86E + 01	4.46E + 01	3.98E + 01	**1.44E + 01**	2.39E + 01	7.27E + 01	1.48E + 01
	Rank	8	2	5	6	4	2	3	7	1
Func.	Criteria	PSO	FDR_PSO	CLPSO	DNLPSO	LIPSO	HCLPSO	EPSO	TCSPSO	ASPSO

(*continued on next page*)

Table 7 (*continued*)

Func.	Criteria	PSO	FDR_PSO	CLPSO	DNLPSO	LIPSO	HCLPSO	EPSO	TCSPSO	ASPSO
F24	Mean	3.15E + 03	2.89E + 03	3.02E + 03	3.03E + 03	2.89E + 03	2.88E + 03	2.90E + 03	3.01E + 03	**2.87E + 03**
	Std	7.93E + 01	1.72E + 01	**1.30E + 01**	2.94E + 01	5.65E + 01	2.17E + 01	2.97E + 01	6.07E + 01	1.72E + 01
	Rank	8	3	6	7	3	2	4	5	1
F25	Mean	3.20E + 03	**2.89E + 03**	2.95E + 03	2.90E + 03	2.93E + 03	**2.89E + 03**	2.90E + 03	2.94E + 03	**2.89E + 03**
	Std	2.97E + 02	1.01E + 01	1.20E + 01	1.29E + 01	2.42E + 01	**2.33E + 00**	1.90E + 01	2.94E + 01	7.49E + 00
	Rank	6	1	5	2	3	1	2	4	1
F26	Mean	6.85E + 03	3.88E + 03	5.23E + 03	5.52E + 03	**3.85E + 03**	**3.85E + 03**	4.06E + 03	4.83E + 03	3.86E + 03
	Std	8.46E + 02	7.55E + 02	**2.92E + 02**	3.72E + 02	8.87E + 02	5.66E + 02	1.04E + 03	1.39E + 03	4.23E + 02
	Rank	8	3	6	7	1	1	4	5	2
F27	Mean	3.34E + 03	3.24E + 03	3.26E + 03	**3.20E + 03**	3.31E + 03	3.23E + 03	3.25E + 03	3.39E + 03	3.22E + 03
	Std	7.42E + 01	1.52E + 01	8.16E + 00	**1.83E-04**	2.74E + 01	1.16E + 01	2.57E + 01	3.38E + 01	1.04E + 01
	Rank	8	4	6	1	7	3	5	9	2
F28	Mean	4.21E + 03	**3.19E + 03**	3.40E + 03	3.29E + 03	3.32E + 03	3.22E + 03	3.20E + 03	3.33E + 03	3.23E + 03
	Std	1.07E + 03	4.22E + 01	2.37E + 01	1.04E + 01	8.04E + 01	1.59E + 01	3.12E + 01	9.24E + 01	**9.24E + 00**
	Rank	9	1	8	5	6	3	2	7	4
F29	Mean	4.24E + 03	3.55E + 03	3.83E + 03	3.92E + 03	3.86E + 03	**3.52E + 03**	3.67E + 03	3.84E + 03	3.56E + 03
	Std	4.92E + 02	1.40E + 02	8.46E + 01	1.96E + 02	1.55E + 02	**1.11E + 02**	1.97E + 02	2.58E + 02	1.30E + 02
	Rank	9	2	6	8	7	1	4	5	3
F30	Mean	6.00E + 06	8.86E + 03	7.90E + 05	4.09E + 05	9.65E + 04	9.12E + 03	8.66E + 03	9.86E + 04	**8.39E + 03**
	Std	7.39E + 06	2.77E + 03	4.48E + 05	5.09E + 05	1.42E + 05	2.68E + 03	**1.73E + 03**	1.41E + 05	1.92E + 03
	Rank	9	3	8	7	5	4	2	6	1
Total Rank	229	79	187	194	115	68	111	166	58	
Final Rank	9	3	7	8	5	2	4	6	1	

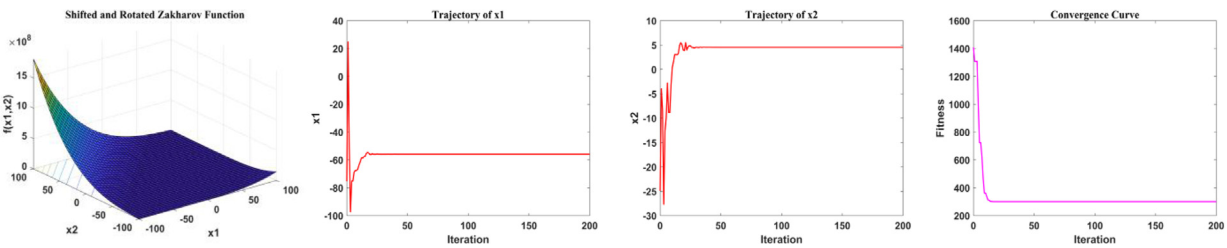


Fig. 4. Qualitative results for the tested function F3.

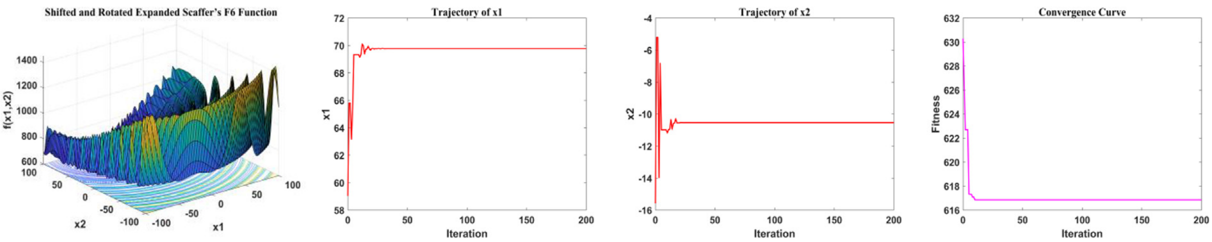


Fig. 5. Qualitative results for the tested function F6.

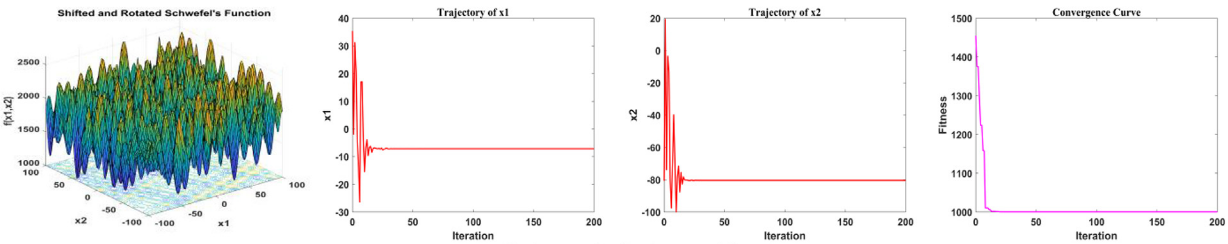


Fig. 6. Qualitative results for the tested function F10.

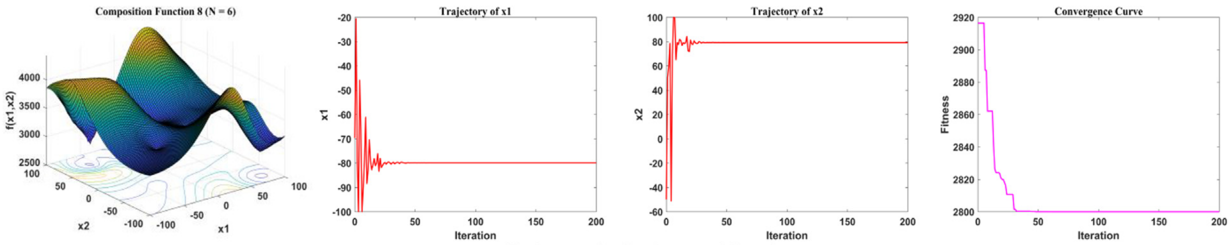


Fig. 7. Qualitative results for the tested function F28.

4.3.2. Comparison test with eight PSO variants

To show the competitiveness of the proposed ASPSO, eight representative PSO variants are chosen to compare. Table 7 shows the comparison results obtained by the nine algorithms for the 30-dimensional test problems in CEC2017. Two indicators are adopted include mean value (Mean) and standard deviation (Std). The best result among the nine algorithms is indicated in boldface. The total rank of each algorithm is the sum of the ranks on all 29 benchmark functions.

For unimodal functions (F1 and F3), ASPSO achieves the best performance on F3, while HCLPSO obtains the best solution on F1. According to their ranks, ASPSO and HCLPSO are equally competitive. Fig. 8(a) shows that ASPSO has a faster convergence speed and higher convergence accuracy on F3. For simple multimodal functions (F4 - F10), DNLPSO, FDR_PSO, and LIPSO obtain the best solution on F4, F6, and F10, respectively. ASPSO ranks first on F5 and F7 - F9 and second on function F6. A reason is that ASPSO uses elite and dimensional learning strategies to effectively maintain the diversity of the population. The convergence curves on F5 are shown in Fig. 8(b).

For hybrid functions (F11 - F20), ASPSO is excellent on F11, F14, F18, and F20. But ASPSO doesn't find the best solution for the other functions. Among them, ASPSO is worse than LIPSO and HCLPSO on F15 and F17, respectively. ASPSO ranking on F12, F13, F16, and F19 are third, sixth, third and third, respectively. The convergence curves on F14, F18, and F20 are shown in Fig. 8(c)–(e), respectively. For composition functions (F21 - F30), HCLPSO achieves the best results on F26 and F29, whereas DNLPSO performs the best on F27, and FDR_PSO offers the best results on F28. In addition, ASPSO has excellent performance on 6 out of 10 test functions, i.e., F21- F25, and F30. The reason is that ASPSO can jump out from local optima by competitive substitution mechanism. The adaptive position update strategy results in that particle may adaptively adjust its search direction to a more excellent particle when particle searches in a complex space. The convergence curves on F21, F22, and F30 are shown in Fig. 8(f)–(h), respectively. Above all, depending on the final rank, the proposed ASPSO demonstrates the highest performance among all compared algorithms, obtaining 15 best Mean values out of the 29 selected benchmark functions.

4.3.3. Comparison test with eight state-of-the-art evolutionary algorithms

The performance of the proposed ASPSO algorithm is compared with eight state-of-the-art evolutionary algorithms, namely, ABC [5], BBO [43], BSA [44], ISA [45],CSA [46], GWO [4], VSA [47] and MVO [48][62]. The parameter settings of the comparison algorithm are listed in Table 2. The comparison results and simulation diagrams are shown in Table 8 and Fig. 9, respectively.

For unimodal functions (F1 and F3), ASPSO cannot find the best solution on them. ASPSO ranking on F1 and F3 are third and second, respectively. Fig. 9(a) shows the convergence curve of ASPSO on F1. For simple multimodal functions (F4 - F10), the ASPSO algorithm ranks first on F5, and F7 – 9, and second on F6. CSA obtains the best solution on F4, whereas ABC achieves the best solution on functions F6 and F10. Overall, ASPSO achieved the best overall performance on multimodal problems. The convergence curves on F8 are shown in Fig. 9(b). For hybrid and composition functions (F11 - F30), ASPSO achieves the best performance on 12 out of 20 test functions, i.e., F11, F12, F16 - F18, F20 - F24, F29, and F30. CSA has excellent performance on F14, F15, F19, F25, and F27. In addition, BBO, ABC, and VSA perform the best on F13, F26, and F28, respectively. Therefore, ASPSO still obtains the best overall performance on hybrid and composition functions. The convergence curves on F12, F16, F18, F21, F24, and F30 are shown in Fig. 9(c)–(h), respectively.

According to the "No Free Lunch" theorem, "any elevated performance over one class of problems is offset by performance over another class" [49]. ASPSO performs better on multimodal, hybrid, and composition functions from the statistical results, except unimodal functions.

4.4. Statistical analysis of the results

The Wilcoxon signed-rank test in Table 9 shows that ASPSO is significantly better than any other algorithms on major benchmark functions with a level of significance $\alpha = 0.05$. The Friedman test is used to compare the comprehensive performance of each algorithm for the 30-D test problems in CEC2017. Table 10 gives the results of the Friedman test. The closer p-value is to 0, which indicates that these algorithms have significant differences in test problems. Among all seventeen comparison algorithms, ASPSO ranks first.

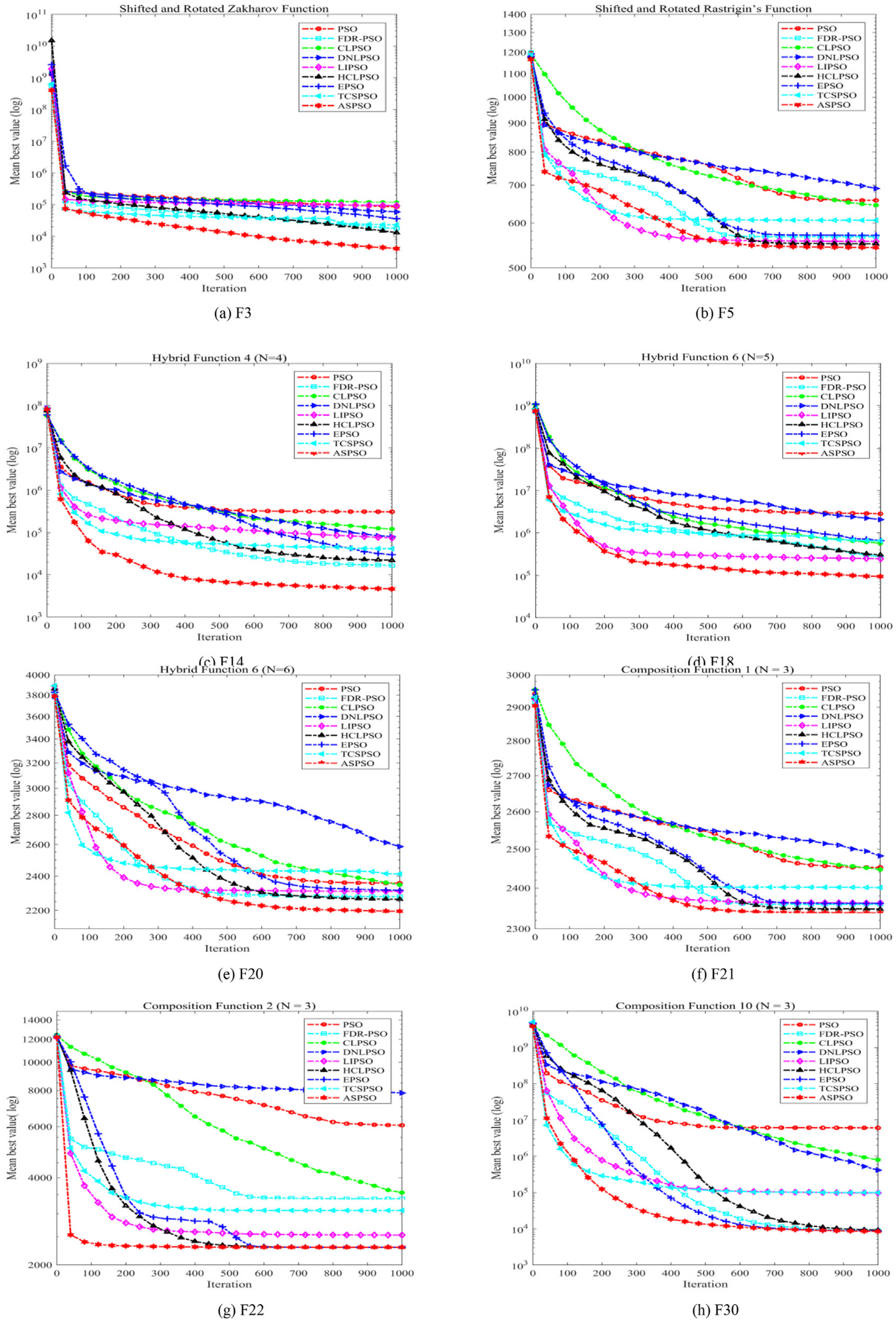


Fig. 8. The convergence curves of ASPSO and the other PSOs on CEC2017 benchmark functions.

Table 8
Comparisons of experimental results between ASPSO with eight state-of-the-art evolutionary algorithms.

Func.	Criteria	ABC	ISA	BSA	BBO	CSA	GWO	VSA	MVO	ASPSO
f1	Mean	**2.98E + 03**	2.30E + 08	9.35E + 04	9.61E + 05	1.00E + 10	1.81E + 09	3.00E + 03	2.66E + 05	3.02E + 03
	Std	2.52E + 03	4.85E + 07	6.36E + 04	2.59E + 03	6.58E + 08	1.42E + 09	2.77E + 03	7.35E + 04	**1.66E + 03**
	Rank	1	7	4	6	9	8	2	5	3
f3	Mean	1.31E + 05	2.11E + 05	6.70E + 04	7.80E + 03	1.14E + 05	4.92E + 04	4.30E + 03	**4.07E + 02**	4.08E + 03
	Std	2.07E + 04	4.51E + 04	1.42E + 04	8.01E + 03	1.76E + 04	9.87E + 03	2.37E + 03	**7.12E + 01**	1.70E + 03
	Rank	8	9	6	4	7	5	3	1	2
f4	Mean	4.77E + 02	5.88E + 02	5.19E + 02	4.90E + 02	**4.71E + 02**	5.66E + 02	5.03E + 02	4.92E + 02	5.08E + 02
	Std	1.69E + 01	3.91E + 01	**8.72E + 00**	2.85E + 01	2.05E + 01	4.38E + 01	2.51E + 01	1.43E + 01	1.52E + 01
	Rank	2	9	7	3	1	8	5	4	6
f5	Mean	6.11E + 02	9.07E + 02	6.11E + 02	5.65E + 02	6.76E + 02	6.00E + 02	6.09E + 02	5.99E + 02	**5.43E + 02**
	Std	1.84E + 01	6.30E + 01	**1.36E + 01**	1.87E + 01	2.25E + 01	2.45E + 01	3.59E + 01	2.04E + 01	1.66E + 01
	Rank	6	8	6	2	7	4	5	3	1
f6	Mean	**6.00E + 02**	6.92E + 02	6.01E + 02	6.02E + 02	6.58E + 02	6.07E + 02	6.18E + 02	6.17E + 02	6.01E + 02
	Std	**1.33E-02**	1.04E + 01	3.51E-01	1.79E + 00	6.92E + 00	3.36E + 00	9.19E + 00	8.74E + 00	3.89E-01
	Rank	1	8	2	3	7	4	6	5	2
f7	Mean	8.26E + 02	1.95E + 03	8.52E + 02	8.12E + 02	9.07E + 02	8.80E + 02	8.70E + 02	8.47E + 02	**7.76E + 02**
	Std	1.56E + 01	2.57E + 02	1.58E + 01	2.58E + 01	2.23E + 01	5.25E + 01	5.92E + 01	2.94E + 01	**1.37E + 01**
	Rank	3	9	5	2	8	7	6	4	1
f8	Mean	9.21E + 02	1.14E + 03	9.14E + 02	8.61E + 02	9.64E + 02	8.89E + 02	9.02E + 02	9.14E + 02	**8.44E + 02**
	Std	1.73E + 01	6.37E + 01	**1.26E + 01**	1.74E + 01	2.44E + 01	3.13E + 01	2.62E + 01	4.16E + 01	1.75E + 01
	Rank	6	8	5	2	7	3	4	5	1
f9	Mean	3.65E + 03	2.27E + 04	1.09E + 03	1.24E + 03	7.62E + 03	1.85E + 03	2.92E + 03	4.56E + 03	**9.12E + 02**
	Std	1.02E + 03	4.95E + 03	1.08E + 02	2.80E + 02	1.87E + 03	6.87E + 02	1.37E + 03	4.06E + 03	**1.29E + 01**
	Rank	6	9	2	3	8	4	5	7	1
f10	Mean	**3.92E + 03**	7.48E + 03	5.22E + 03	4.59E + 03	5.19E + 03	4.56E + 03	4.54E + 03	4.31E + 03	4.32E + 03
	Std	3.45E + 02	4.96E + 02	2.79E + 02	6.44E + 02	**2.67E + 02**	1.26E + 03	6.71E + 02	4.02E + 02	6.59E + 02
	Rank	1	9	8	6	7	5	4	2	3
f11	Mean	2.71E + 03	2.24E + 03	1.23E + 03	1.27E + 03	1.24E + 03	1.82E + 03	1.28E + 03	1.35E + 03	**1.20E + 03**
	Std	7.04E + 02	6.05E + 02	2.63E + 01	6.55E + 01	**1.77E + 01**	5.92E + 02	4.20E + 01	5.91E + 01	3.30E + 01
	Rank	9	8	2	4	3	7	5	6	1
f12	Mean	2.13E + 06	7.53E + 07	1.75E + 06	1.08E + 06	7.89E + 09	8.15E + 07	5.58E + 06	9.91E + 06	**3.50E + 05**
	Std	8.37E + 05	4.38E + 07	7.92E + 05	7.91E + 05	3.95E + 09	9.39E + 07	4.66E + 06	7.98E + 06	**2.08E + 05**
	Rank	4	7	3	2	9	8	5	6	1
f13	Mean	1.38E + 05	9.36E + 06	3.67E + 04	**1.25E + 04**	5.69E + 08	1.59E + 07	7.91E + 04	1.45E + 05	1.66E + 04
	Std	1.28E + 05	2.83E + 06	3.29E + 04	**1.01E + 04**	1.85E + 09	5.39E + 07	4.29E + 04	1.01E + 05	1.33E + 04
	Rank	5	7	3	1	9	8	4	6	2
f14	Mean	3.62E + 05	9.26E + 05	5.68E + 03	1.04E + 05	**1.54E + 03**	5.25E + 05	2.47E + 04	2.33E + 04	4.62E + 03
	Std	2.17E + 05	1.04E + 06	3.88E + 03	8.73E + 04	**2.03E + 01**	7.34E + 05	2.37E + 04	1.95E + 04	2.97E + 03
	Rank	7	9	3	6	1	8	5	4	2
f15	Mean	3.65E + 04	1.82E + 06	6.03E + 03	5.27E + 03	**2.89E + 03**	4.51E + 05	7.20E + 04	6.94E + 04	5.25E + 03
	Std	2.58E + 04	5.57E + 05	3.59E + 03	4.41E + 03	**3.78E + 02**	9.60E + 05	3.97E + 04	5.02E + 04	5.55E + 03
	Rank	5	9	4	3	1	8	7	6	2
f16	Mean	2.45E + 03	4.27E + 03	2.73E + 03	2.56E + 03	2.83E + 03	2.47E + 03	2.57E + 03	2.54E + 03	**2.33E + 03**
	Std	1.91E + 02	4.77E + 02	**1.52E + 02**	2.01E + 02	1.53E + 02	2.93E + 02	2.94E + 02	3.27E + 02	2.44E + 02
	Rank	2	9	7	5	8	3	6	4	1
f17	Mean	2.06E + 03	2.94E + 03	2.04E + 03	2.12E + 03	2.11E + 03	2.00E + 03	2.03E + 03	2.12E + 03	**1.91E + 03**
	Std	9.80E + 01	3.53E + 02	1.01E + 02	2.44E + 02	**9.05E + 01**	1.38E + 02	2.09E + 02	2.08E + 02	1.71E + 02
	Rank	5	8	4	7	6	2	3	7	1
f18	Mean	5.61E + 05	3.94E + 06	1.51E + 05	6.89E + 05	1.70E + 05	1.05E + 06	3.59E + 05	3.21E + 05	**9.32E + 04**
	Std	2.87E + 05	2.78E + 06	1.34E + 05	5.98E + 05	8.01E + 04	1.25E + 06	1.89E + 05	2.00E + 05	**7.02E + 04**
	Rank	6	9	2	7	3	8	5	4	1
f19	Mean	6.16E + 04	1.43E + 07	9.82E + 03	1.06E + 04	**2.46E + 03**	8.67E + 05	9.77E + 05	8.41E + 05	5.82E + 03
	Std	4.78E + 04	7.65E + 06	5.30E + 03	1.20E + 04	**3.70E + 02**	1.01E + 06	7.27E + 05	8.21E + 05	3.46E + 03
	Rank	5	9	3	4	1	7	8	6	2
f20	Mean	2.37E + 03	3.07E + 03	2.38E + 03	2.68E + 03	2.58E + 03	2.39E + 03	2.41E + 03	2.51E + 03	**2.20E + 03**
	Std	1.12E + 02	2.49E + 02	**9.67E + 01**	2.39E + 02	1.01E + 02	1.53E + 02	1.48E + 02	1.85E + 02	1.16E + 02
	Rank	2	9	3	8	7	4	5	6	1
f21	Mean	2.39E + 03	2.67E + 03	2.41E + 03	2.38E + 03	2.46E + 03	2.40E + 03	2.41E + 03	2.41E + 03	**2.34E + 03**
	Std	6.49E + 01	5.92E + 01	2.76E + 01	1.92E + 01	4.32E + 01	2.96E + 01	3.32E + 01	2.94E + 01	**1.40E + 01**
	Rank	3	7	5	2	6	4	5	5	1
f22	Mean	3.16E + 03	8.17E + 03	2.78E + 03	4.23E + 03	5.91E + 03	4.69E + 03	2.45E + 03	5.16E + 03	**2.30E + 03**
	Std	1.46E + 03	2.07E + 03	1.26E + 03	2.03E + 03	1.58E + 03	1.85E + 03	8.41E + 02	1.57E + 03	**9.79E-01**
	Rank	4	9	3	5	8	6	2	7	1
f23	Mean	2.75E + 03	3.50E + 03	2.76E + 03	2.76E + 03	2.82E + 03	2.76E + 03	2.78E + 03	2.75E + 03	**2.69E + 03**
	Std	2.83E + 01	1.72E + 02	1.53E + 01	2.62E + 01	2.30E + 01	5.54E + 01	4.50E + 01	4.09E + 01	**1.48E + 01**
	Rank	2	6	3	3	5	3	4	2	1
Func.	Criteria	ABC	ISA	BSA	BBO	CSA	GWO	VSA	MVO	ASPSO

(*continued on next page*)

Table 8 (*continued*)

Func.	Criteria	ABC	ISA	BSA	BBO	CSA	GWO	VSA	MVO	ASPSO
f24	Mean	2.93E + 03	3.52E + 03	2.95E + 03	2.92E + 03	2.95E + 03	2.93E + 03	2.92E + 03	2.92E + 03	**2.87E + 03**
	Std	1.78E + 02	1.43E + 02	**1.55E + 01**	3.39E + 01	1.96E + 01	5.56E + 01	3.10E + 01	2.92E + 01	1.72E + 01
	Rank	3	5	4	2	4	3	2	2	1
f25	Mean	2.89E + 03	2.99E + 03	2.90E + 03	2.89E + 03	**2.88E + 03**	2.97E + 03	2.90E + 03	2.89E + 03	2.89E + 03
	Std	5.04E + 00	3.87E + 01	5.77E + 00	**2.16E + 00**	2.23E + 00	3.19E + 01	1.46E + 01	1.14E + 01	7.49E + 00
	Rank	2	5	3	2	1	4	3	2	2
f26	Mean	**3.27E + 03**	1.02E + 04	4.45E + 03	4.85E + 03	4.53E + 03	4.70E + 03	4.84E + 03	4.59E + 03	3.86E + 03
	Std	6.21E + 02	2.04E + 03	6.06E + 02	6.94E + 02	6.44E + 02	**2.61E + 02**	3.82E + 02	5.92E + 02	4.23E + 02
	Rank	1	9	3	8	4	6	7	5	2
f27	Mean	3.22E + 03	3.69E + 03	3.23E + 03	3.31E + 03	**3.20E + 03**	3.25E + 03	3.25E + 03	3.22E + 03	3.22E + 03
	Std	5.41E + 00	2.64E + 02	4.07E + 00	2.89E + 01	**6.05E-05**	2.06E + 01	2.66E + 01	1.36E + 01	1.04E + 01
	Rank	2	6	3	5	1	4	4	2	2
f28	Mean	3.25E + 03	3.39E + 03	3.29E + 03	**3.22E + 03**	3.30E + 03	3.40E + 03	**3.22E + 03**	3.24E + 03	3.23E + 03
	Std	1.40E + 01	4.47E + 01	1.22E + 01	1.91E + 01	**5.68E-05**	8.08E + 01	2.34E + 01	3.67E + 01	9.24E + 00
	Rank	4	7	5	1	6	8	1	3	2
f29	Mean	3.64E + 03	5.45E + 03	3.70E + 03	3.86E + 03	4.14E + 03	3.82E + 03	3.89E + 03	3.92E + 03	**3.56E + 03**
	Std	**9.40E + 01**	3.90E + 02	1.06E + 02	2.38E + 02	1.48E + 02	1.55E + 02	2.48E + 02	1.99E + 02	1.30E + 02
	Rank	2	9	3	5	8	4	6	7	1
f30	Mean	3.50E + 04	1.88E + 07	3.21E + 04	1.17E + 04	2.73E + 04	8.87E + 06	2.69E + 06	3.58E + 06	**8.39E + 03**
	Std	1.33E + 04	1.22E + 07	1.59E + 04	3.67E + 03	1.76E + 04	8.95E + 06	2.15E + 06	1.88E + 06	**1.92E + 03**
	Rank	5	9	4	2	3	8	6	7	1
Total Rank	112	232	115	113	155	161	133	133	48	
Final Rank	2	8	4	3	6	7	5	5	1	

Table 9
Wilcoxon signed-rank test for CEC2017 benchmark functions with a significance level of α = 0.05.

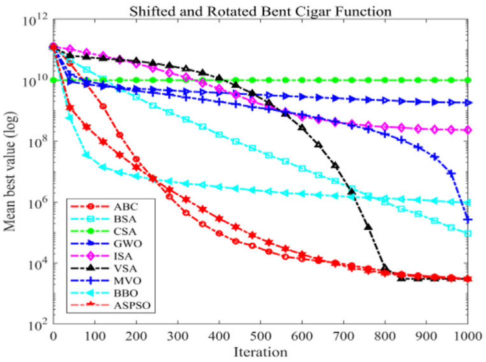
Group A Results on the 30-dimensional functions from Table 7								
ASPSO vs.	PSO	FDR_PSO	CLPSO	DNLPSO	LIPSO	HCLPSO	EPSO	TCSPSO
R+	0	111	3	7	73	140	94	21
R-	435	324	432	428	362	295	341	414
+	29	15	28	27	21	13	18	27
–	0	4	0	2	3	5	4	0
=	0	10	1	0	5	11	7	2
p-value	3.73E–09	2.03E–02	1.86E–08	7.08E–08	1.17E–03	4.63E–02	6.45E–03	1.67E–06
Group B Results on the 30-dimensional functions from Table 8								
ASPSO vs.	ABC	ISA	BSA	BBO	CSA	GWO	VSA	MVO
R+	46	0	1	32	66	0	6	27
R-	389	435	434	403	369	435	429	408
+	22	29	26	21	23	27	25	25
–	4	0	1	1	6	0	0	2
=	3	0	2	7	0	2	4	2
p-value	6.84E–05	3.73E–09	7.45E–09	1.03E–05	6.07E–04	3.73E–09	5.22E–08	4.70E–06

Table 10
The Friedman test for CEC2017 benchmark functions.

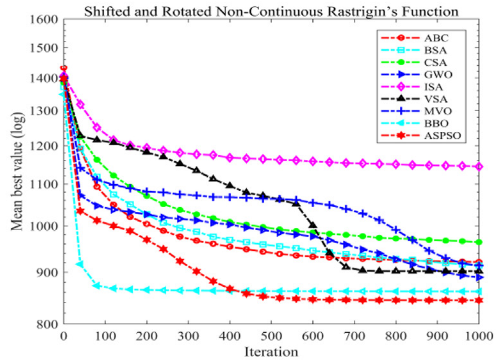
	PSO	FDR_PSO	CLPSO	DNLPSO	LIPSO	HCLPSO	EPSO	TCSPSO	ABC
Friedman Rank	14.62	5.45	12.17	14.10	6.59	3.13	6.10	10.03	8.66
Final Rank	16	3	14	15	5	2	4	11	8
	ISA	BSA	BBO	CSA	GWO	VSA	MVO	ASPSO	
Friedman Rank	15.07	7.55	7.79	10.13	11.38	9.93	8.72	1.66	
Final Rank	17	6	7	12	13	10	9	1	
p-value	1.50E-10								

5. Application in the optimization of melt spinning progress

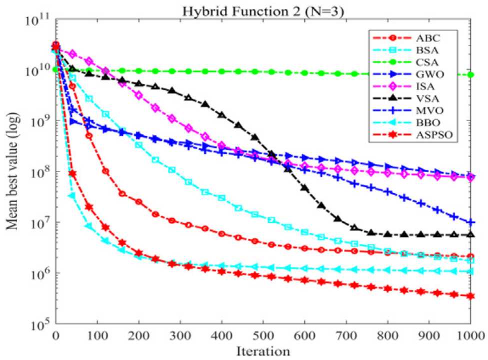
In this part, we apply the proposed ASPSO algorithm to optimize the melt spinning process, a typical complex real-world optimization problem. Melt spinning is one of the traditional techniques to produce producing methods of polymer fibers. Its principle is to feed high polymer raw materials into a screw extruder, send it to a heating zone by a rotating screw, and then



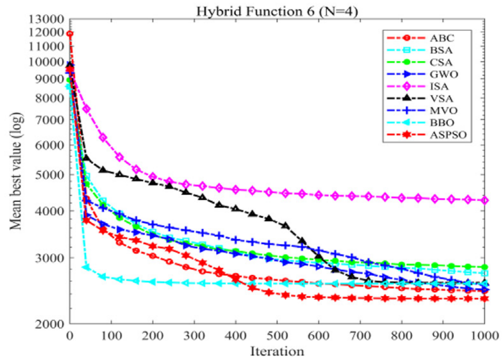
(a) F1



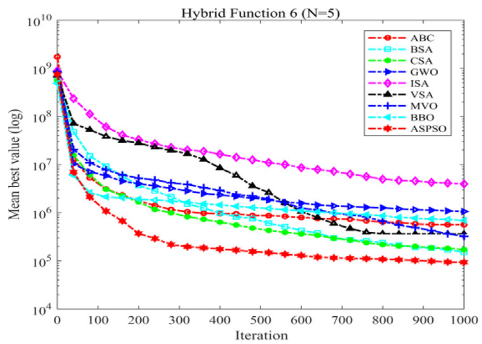
(b) F8



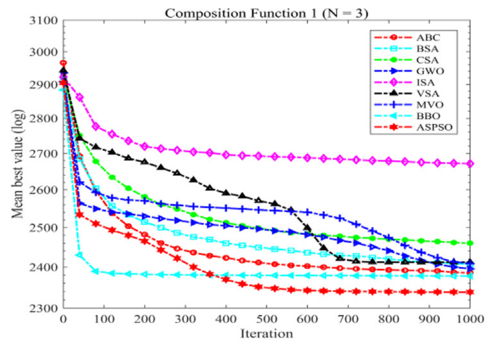
(c) F12



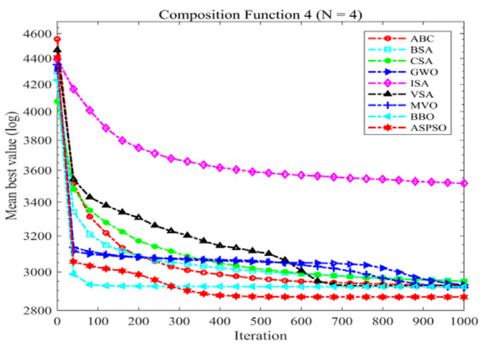
(d) F16



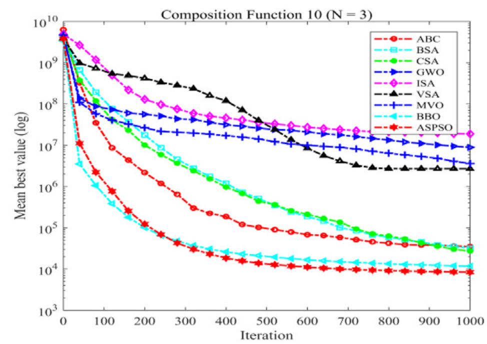
(e) F18



(f) F21



(g) F24



(h) F30

Fig. 9. The convergence curves of ASPSO and eight state-of-the-art evolutionary algorithms on CEC2017 benchmark functions.

send it to a metering pump after extrusion. Many varieties of synthetic fibers, polyester, cotton, and polypropylene are all produced by melt spinning.

The melt spinning process is a crucial link during the whole fiber production [50]. The polymer is melted in the screw extruder and then sent to the spinning position, sent to the spinning assembly by the metering pump, and extruded from the capillary holes of the spinneret. As shown in Fig. 10, the polymer melt exits the spinneret with a radius R_0, extrusion velocity v_0, temperature T_0 and rheological force F_0. The melt is cooled by a transverse stream of quench air at the temperature T_a with a velocity v_a when it passes through the cooling medium. Then the fiber is pulled downward by the take-up machine at a drawdown velocity v_L which is significantly higher than the extrusion velocity v_0. The z-axis is the direction from the spinneret to the take-up wheel. The r-axis is the direction from the fiber center to the outside. Details of the melt spinning process can be found in [50]. The crystallinity, orientation, and fiber uniformity are directly affected by this process. In other words, this process significantly involves the physical properties of the as-spun fibers and constrains the quality of the final products. Thus, for better production quality, a detailed understanding of the melt spinning process is crucial.

By studying the physical relationship between variables and process parameters, the melt spinning process can be described by

$$W = \rho v_z \pi {R_z}^2 \tag{19}$$

$$\pi R_z^2 \bar{N}_{1z} - \pi R_0^2 N_{10} - \pi \int_0^z c_f \rho_a R_z v_z^2 dz - W(v_z - v_0) = 0 \tag{20}$$

$$\frac{\partial T}{\partial z} = \frac{k_p}{v_z \rho c_p}\left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r}\right) + \frac{\Delta H_f^0}{c_p}\frac{dx}{dz} \tag{21}$$

$$K_i \tau_{izz} + \lambda_i\left[v_z \frac{d\tau_{izz}}{dz} - 2(1-\xi_i)\frac{dv_z}{dz}\tau_{izz}\right] = 2g_i\lambda_i\frac{dv_z}{dz} \tag{22}$$

$$K_i \tau_{irr} + \lambda_i\left[v_z \frac{d\tau_{irr}}{dz} + (1-\xi_i)\frac{dv_z}{dz}\tau_{irr}\right] = -g_i\lambda_i\frac{dv_z}{dz} \tag{23}$$

$$\frac{d\theta}{dz} = \frac{nK}{v_z}(1-\theta)[-ln(1-\theta)]^{\frac{n-1}{n}} \tag{24}$$

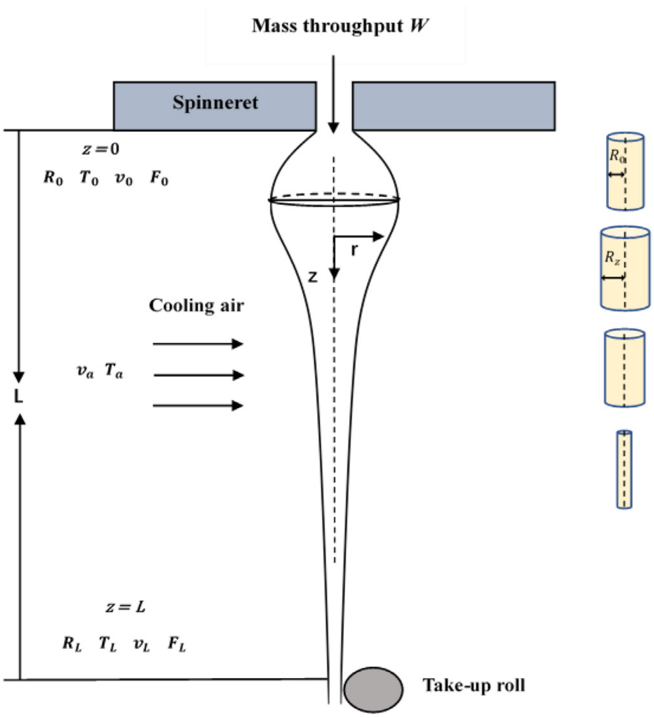
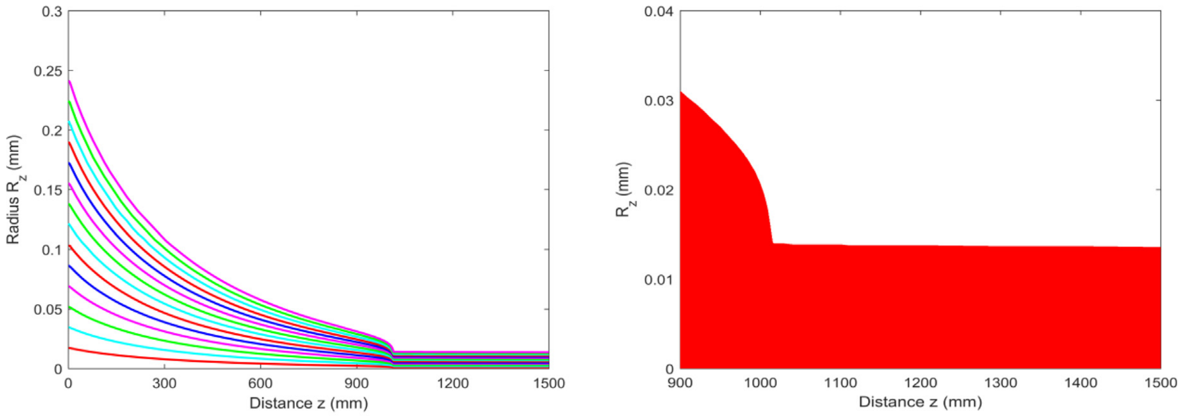


Fig. 10. Schematic illustration of melt spinning progress.



(a) (b)

Fig. 11. The profile of radius R_z along the spinline.

The melt spinning model is quite complicated, containing one partial differential equation, fifteen ordinary differential equations (Eqs. (22) and (23) include seven equations, respectively). Due to the complexity of the model, the model gives a computational burden on the solution process.

5.1. Numerical computation results

Fig. 11(a) shows the radius curve of the fiber from the spinneret to the take-up wheel. The different curves are the different layer positions inside the fiber relating to different radiuses. Fig. 11(b) shows the necking phenomenon of the fiber. The radius of the fiber decreases rapidly in a short interval, and then the radius remains unchanged. The radius of the fiber remains unchanged after the point z = 1020 mm. The point after which the radius of the fiber remains constant is called the freezing point (Z_0).

5.2. Melt spinning process optimization based on ASPSO

Through studying the mechanism of the melt spinning process, it is found that the position of the freezing point will have a significant influence on the performance of the fiber. The outcome of this point is the result of all the process parameters, such as mass throughput W, spinneret orifice radius R_0, initial temperature T_0, initial rheological force F_0, velocity of quench air v_a, quench air temperature T_a, take-up speed v_L, spinline length L, and so on. Meanwhile, this position directly reflects the degree of coagulation of the fiber, which is related to various fiber properties, such as strength, evenness, viscoelasticity, elongation at break, etc. Therefore, it is an excellent choice to take the position of the freezing point as the optimization target.

Different from the maximum and minimum optimization, the position optimization of the freezing point Z_0 is optimization with a specified value, that is, $Z_0 = Z_{set}$, where Z_{set} is the set desired target. But through the transformation, this problem can be transformed into a minimum problem as $|Z_0 - Z_{set}| = 0$. The output result Z_0 is related to multiple input parameters, so this system can be regarded as a multiple input single output system. Through multiple tests of the system, there will be multiple different input combinations that can get the same output result, so this optimization problem is multimodal. At the same time, its solution should lie on one or more extreme value intervals rather than independent extreme values. These results are a set of the same optimal solution for decision-makers to choose.

The mechanism of melt spinning is complicated, and many factors affect the position of the freezing point. Through research, we select several parameters that greatly influence the position as input variables. The input variables are as follows: (1) initial temperature T_0; (2) quench air temperature T_a; (3) velocity of quench air v_a; (4) initial rheological force F_0. In this experiment, the optimization goal is set to Z_{set} = 1060 mm. The position optimization of the freezing point during the melt spinning process is tested for six algorithms. The population size and number of iterations are 50 and 100, respectively.

The results in Table 11 show that the performance of ASPSO is better than the other five algorithms. Fig. 12 shows the convergence curves, from which we can see that ASPSO quickly converges to the optimization goal. Due to the complexity of the melt spinning model, the calculation is very time-consuming. In the iterative process, the total time cost of one fitness calculation for 50 particles is about 283 s. ASPSO quickly found the optimization goal after twenty-five iterations, which is a great help for this optimization task. Table 12 lists the best solutions, which can be chosen by decision-makers according to the actual situation. Meanwhile, multiple solutions could also be analyzed to discover the internal mechanism of the melt spinning progress.

Table 11
Comparisons of experimental results between ASPSO with five algorithms.

Criteria	FDR_PSO	LIPSO	HCLPSO	EPSO	BSA	ASPSO
Mean	0.597	0.955	0.932	1.565	0.926	**0.413**
Std	**1.051**	2.431	2.855	4.399	2.035	1.570
Rank	2	5	4	6	3	1

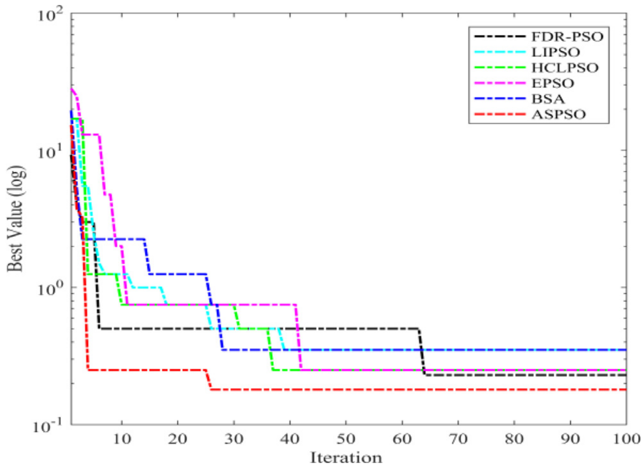


Fig. 12. The convergence curves of six algorithms on the melt spinning process.

Table 12
Six best solutions of Z_0 optimizaiton.

T_0(℃)	T_a(℃)	v_a(m/s)	F_0(N)	Z_{set}(mm)
308.21	23.89	0.104	1.984 E−4	1060
309.36	23.92	0.125	1.978 E−4	1060
306.84	25.20	0.082	1.983 E−4	1060
307.41	24.64	0.092	1.986 E−4	1060
308.76	22.14	0.107	1.971 E−4	1060
305.48	24.67	0.052	1.972 E−4	1060

6. Conclusions and future work

In this research, we integrated four strategies into PSO and obtained the ASPSO algorithm. In ASPSO, to better balance exploration behavior and exploitation nature, a chaotic map and an adaptive position updating strategy are proposed. Meanwhile, elite and dimensional learning strategies are devised to effectively enhance the diversity of the population and avoid premature convergence. Finally, a competitive substitution mechanism is presented to improve the accuracy of ASPSO for complex optimization problems. We use the CEC2017 benchmark functions comprising unimodal, simple multimodal, hybrid, and composition functions to test the performance of ASPSO. Experimental results show that ASPSO is significantly better than the other 16 state-of-the-art algorithms for most test functions. The application of melt spinning progress optimization shows that the optimization effect of the ASPSO algorithm is better than other algorithms. It is worth noting that the proposed ASPSO does not perform satisfactorily on some unimodal functions and some multimodal functions. The reasons are complex, and there must be some potential mechanisms worthy of detailed study in the future. Combining other excellent learning strategies with particle swarm optimization may be investigated in the future to address this problem. Our follow-up studies will also include applying the proposed optimization algorithm to other complex practical engineering problems.

CRediT authorship contribution statement

Rui Wang: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Writing - review & editing. **Kuangrong Hao:** Funding acquisition, Supervision, Writing - review & editing. **Lei Chen:** Data curation, Investigation. **Tong Wang:** Data curation, Investigation. **Chunli Jiang:** Formal analysis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Key Research and Development Plan from Ministry of Science and Technology (2016YFB0302701), the Fundamental Research Funds for the Central Universities (2232021A-10), National Natural Science Foundation of China (no. 61903078), Natural Science Foundation of Shanghai (19ZR1402300, 20ZR1400400), and Fundamental Research Funds for the Central Universities and Graduate Student Innovation Fund of Donghua University (CUSF-DH-D-2021050). In addition, we are grateful to Chenwei Zhao for her valuable comments that helped us improve this paper.

References

[1] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, X.-L. Shen, A hybrid particle swarm optimization algorithm using adaptive learning strategy, Inf. Sci. 436 (2018) 162–177.
[2] X. Zhang, X. Wang, Q. Kang, J. Cheng, Differential mutation and novel social learning particle swarm optimization algorithm, Inf. Sci. 480 (2019) 109–129.
[3] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, IEEE, 1995, pp. 1942-1948..
[4] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[5] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471.
[6] K. Mistry, L. Zhang, S.C. Neoh, C.P. Lim, B. Fielding, A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition, IEEE Trans. Cybern. 47 (2016) 1496–1509.
[7] M. Alswaitti, M. Albughdadi, N.A.M. Isa, Density-based particle swarm optimization algorithm for data clustering, Expert Syst. Appl. 91 (2018) 170–186.
[8] A.P. Engelbrecht, Computational intelligence: an introduction, John Wiley & Sons, 2007.
[9] K.M. Ang, W.H. Lim, N.A.M. Isa, S.S. Tiang, C.H. Wong, A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems, Expert Syst. Appl. 140 (2020) 112882.
[10] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (2006) 281–295.
[11] H.-R. Liu, J.-C. Cui, Z.-D. Lu, D.-Y. Liu, Y.-J. Deng, A hierarchical simple particle swarm optimization with mean dimensional information, Appl. Soft Comput. 76 (2019) 712–725.
[12] K. Chen, B. Xue, M. Zhang, F. Zhou, Novel chaotic grouping particle swarm optimization with a dynamic regrouping strategy for solving numerical optimization tasks, Knowl.-Based Syst. 194 (2020) 105568.
[13] P. Dziwinski, L. Bartczuk, A New Hybrid Particle Swarm Optimization and Genetic Algorithm Method Controlled by Fuzzy Logic, IEEE Trans. Fuzzy Syst. 28 (6) (2020) 1140–1154.
[14] A. Lin, W. Sun, H. Yu, G. Wu, H. Tang, Global genetic learning particle swarm optimization with diversity enhancement by ring topology, Swarm Evol. Comput. 44 (2019) 571–583.
[15] M.R. Tanweer, S. Suresh, N. Sundararajan, Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems, Inf. Sci. 326 (2016) 1–24.
[16] K.e. Chen, F. Zhou, A. Liu, Chaotic dynamic weight particle swarm optimization for numerical function optimization, Knowl.-Based Syst. 139 (2018) 23–40.
[17] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inf. Sci. 294 (2015) 182–202.
[18] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of the1998 IEEE International Conference on Evolutionary Computation, IEEE, 1998, pp. 69-73..
[19] C. Yang, W. Gao, N. Liu, C. Song, Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight, Appl. Soft Comput. 29 (2015) 386–394.
[20] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.
[21] J. Zou, Q. Deng, J. Zheng, S. Yang, A close neighbor mobility method using particle swarm optimizer for solving multimodal optimization problems, Inf. Sci. 519 (2020) 332–347.
[22] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (3) (2004) 204–210.
[23] K.E. Parsopoulos, UPSO: A unified particle swarm optimization scheme, Lecture Series on Computer and Computational, Science 1 (2004) 868–873.
[24] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Inf. Sci. 209 (2012) 16–36.
[25] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.
[26] Z.A. Kai, Q.H. B, Y.Z. A, Enhancing comprehensive learning particle swarm optimization with local optima topology, Inf. Sci. 471 (2019) 1–18.
[27] G. Xu, Q. Cui, X. Shi, H. Ge, Z.-H. Zhan, H.P. Lee, Y. Liang, R. Tai, C. Wu, Particle swarm optimization based on dimensional learning strategy, Swarm Evol. Comput. 45 (2019) 33–51.
[28] S. Wang, G. Liu, M. Gao, S. Cao, A. Guo, J. Wang, Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators, Inf. Sci. 540 (2020) 175–201.
[29] Wei Li, Xiang Meng, Ying Huang, Zhang-Hua Fu, Multipopulation cooperative particle swarm optimization with a mixed mutation strategy, Inf. Sci. 529 (2020) 179–196.
[30] S. Wang, Y. Li, H. Yang, Self-adaptive mutation differential evolution algorithm based on particle swarm optimization, Appl. Soft Comput. 81 (2019) 105496.
[31] Ke Chen, Fengyu Zhou, Lei Yin, Shuqian Wang, Yugang Wang, Fang Wan, A hybrid particle swarm optimizer with sine cosine acceleration coefficients, Inf. Sci. 422 (2018) 218–241.
[32] Vinita Jindal, Punam Bedi, An improved hybrid ant particle optimization (IHAPO) algorithm for reducing travel time in VANETs, Appl. Soft Comput. 64 (2018) 526–535.

[33] Jianhua Liu, Yi Mei, Xiaodong Li, An analysis of the inertia weight parameter for binary particle swarm optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 666–681.
[34] Amir H. Gandomi, Xin-She Yang, Chaotic bat algorithm, Journal of Computational Science 5 (2) (2014) 224–232.
[35] Ke Chen, Feng-Yu Zhou, Xian-Feng Yuan, Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection, Expert Syst. Appl. 128 (2019) 140–156.
[36] Seyedali Mirjalili, Andrew Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[37] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization., (2016)..
[38] Joaquín Derrac, Salvador García, Daniel Molina, Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[39] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in, in: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003, pp. 174–181.
[40] B.Y. Qu, Ponnuthurai Nagaratnam Suganthan, Swagatam Das, A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization, IEEE Trans. Evol. Comput. 17 (3) (2013) 387–402.
[41] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55 (2017) 533–548.
[42] X. Zhang, H. Liu, T. Zhang, Q. Wang, Y. Wang, L. Tu, Terminal crossover and steering-based particle swarm optimization algorithm with disturbance, Appl. Soft Comput. 85 (2019) 105841.
[43] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (2008) 702–713.
[44] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (2013) 8121–8144.
[45] A.H. Gandomi, Interior search algorithm (ISA): a novel approach for global optimization, ISA Trans. 53 (2014) 1168–1183.
[46] Xin-She Yang, Suash Deb, Cuckoo search: recent advances and applications, Neural Comput. Appl. 24 (1) (2014) 169–174.
[47] Berat Doğan, Tamer Ölmez, A new metaheuristic for numerical function optimization: Vortex Search algorithm, Inf. Sci. 293 (2015) 125–145.
[48] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Abdolreza Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[49] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[50] Wolfgang Dietz, Polyester fiber spinning analyzed with multimode Phan Thien-Tanner model, J. Nonnewton. Fluid Mech. 217 (2015) 37–48.