

Article

Quantum-Behaved Particle Swarm Optimization with Weighted Mean Personal Best Position and Adaptive Local Attractor

Shouwen Chen

School of Mathematics and Finance, Chuzhou University, Chuzhou 239000, China; cshouwen@163.com;
Tel.: +86-0550-3510583

Received: 26 October 2018; Accepted: 4 January 2019; Published: 10 January 2019



Abstract: Motivated by concepts in quantum mechanics and particle swarm optimization (PSO), quantum-behaved particle swarm optimization (QPSO) was proposed as a variant of PSO with better global search ability. In this paper, a QPSO with weighted mean personal best position and adaptive local attractor (ALA-QPSO) is proposed to simultaneously enhance the search performance of QPSO and acquire good global optimal ability. In ALA-QPSO, the weighted mean personal best position is obtained by distinguishing the difference of the effect of the particles with different fitness, and the adaptive local attractor is calculated using the sum of squares of deviations of the particles' fitness values as the coefficient of the linear combination of the particle best known position and the entire swarm's best known position. The proposed ALA-QPSO algorithm is tested on twelve benchmark functions, and compared with the basic Artificial Bee Colony and the other four QPSO variants. Experimental results show that ALA-QPSO performs better than those compared method in all of the benchmark functions in terms of better global search capability and faster convergence rate.

Keywords: quantum-behaved particle swarm optimization; weighted mean personal best position; adaptive local attractor

1. Introduction

Particle swarm optimization (PSO) is a stochastic population-based algorithm proposed by Kennedy and Eberhart [1], which is motivated by the intelligent collective behavior of some animals such as flocks of birds or schools of fish. In PSO, each particle is regarded as a potential solution. All particles have fitness values and velocities, and they fly in a multidimensional search space by learning from the historical information, which contains their memories of the personal best positions and knowledge of the global best position in the groups during the search process. PSO can be easily implemented and is computationally inexpensive, and has few parameters to adjust. For its superiority, PSO has rapidly developed with applications in solving real-world optimization problems in recent years [2,3]. However, PSO is easily trapped into the local optima, and premature convergence appears when it is applied to complex multimodal problems [4]. Many attempts have been made to improve the performance of the PSO [5]. Inspired by quantum mechanics and trajectory analysis of PSO [6], Sun et al. [7,8] proposed a variant of PSO, which is named quantum-behaved PSO (QPSO). Unlike PSO, QPSO needs no velocity vectors for particles, and also has fewer parameters to adjust, making it easier to implement. Since QPSO was proposed, it has attracted much attention and different variants of QPSO have been proposed to enhance the performance from different aspects and successfully applied to solve a wide range of continuous optimization problems [9–14]. In general, most current QPSO variants can be classified into three categories [15]: the improvement based on operators from other evolutionary algorithms, hybrid search methods, and cooperative methods.

Though those strategies have improved the performance of QPSO in terms of convergence speed and global optimality, it is rather difficult to improve the global search capability and accelerate the rate of convergence simultaneously. In classic QPSO, both the mean personal best position and the local attractor have a great influence on the performance of the algorithm. On one hand, the former is simply the average on the personal best position of all particles, which ignores the difference of the effect of the particles with different fitness on guiding particles to search global optimal solutions. Thus, it is not conducive to improve the global search ability of QPSO. On the other hand, the local attractor for a particle can be obtained as the weighted sum of its personal and global best positions. It has been found that there are few improvements concentrating on the local attractors in QPSO. A novel quantum-behaved particle swarm optimization with Gaussian distributed local attractor point (GAQPSO) is proposed by Sun et al. [16]. In GAQPSO, the local attractor is subject to Gaussian distribution whose mean value is the original local attractor that is defined in classic QPSO. An enhanced QPSO based on a novel computing way of local attractor (EQPSO) is introduced by Jia et al. [17]. In EQPSO, the local attractor is the weighted sum of particle personal and global best positions, using the function designed by the current and total numbers of iterations as the weight. This calculation method cannot monitor the population diversity in real time. Therefore it is not conducive to improve the global search ability of QPSO either. In general, diversity [18] is an important aspect of population-based optimization methods since it influences their performance, and diversity is closely linked to the tradeoff of exploration–exploitation. High diversity facilitates exploration, which is usually required during the initial iterations of the optimization algorithm. Low diversity is indicative of exploitation of a small area of the search space, desired during the latter part of the optimization process. Monitoring the diversity of QPSO populations to construct local attractors to guide particles optimization, thereby improving the algorithm’s ability to search global optimum and accelerating the algorithm’s convergence rate, this practice is rarely reported. In this paper, in order to balance the global and local searching abilities, we propose a set of weighted coefficients that can distinguish the fitness of particles to calculate the mean personal best position, and a novel way of computing the local attractor, furthermore, a new kind of quantum-behaved particle swarm optimization with weighted mean personal best position and adaptive local attractor is designed for numerical optimization. Experimental results show that our proposed method is effective.

The remainder of the paper is structured as follows. A brief introduction of PSO, two versions of QPSO and population diversity of PSO are presented in Section 2. The proposed QPSO with weighted mean personal best position and adaptive local attractor (ALA-QPSO) is given in Section 3. Section 4 provides the experimental results on benchmark functions. Some concluding remarks are given in Section 5.

2. Background

2.1. Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic population-based algorithm proposed by Kennedy and Eberhart [1], which is motivated by intelligent collective behavior of some animals such as flocks of birds or schools of fish. The candidate solutions for PSO are called particles. The movements of the particles are guided by their own best known position called *pbest* and the entire swarm’s best known position called *gbest*. The position and velocity of the *i*–th particle is updated according to (1) and (2).

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

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

In (1) and (2), $X_i^t = (x_{i1}^t, \dots, x_{id}^t, \dots, x_{iD}^t)$ and $V_i^t = (v_{i1}^t, \dots, v_{id}^t, \dots, v_{iD}^t)$ are the position and velocity vector of *D*-dimensional space for particle *i* at iteration *t*, respectively. c_1 and c_2 are positive constants which control the influence of *pbest*_{*i*} and *gbest* in the search process, respectively. r_1 and r_2

are random values between 0 and 1. Inertia weight, ω , which was proposed by Shi and Eberhart [19], plays a significant role in balancing the algorithm’s global and local search ability. The fitness value of each particle’s position is determined by a fitness function. PSO is usually executed by repeatedly computing (1) and (2) until a specified number of iterations have reached or the velocity updates are close to zero during the iterations. The pseudo code for the PSO algorithm can be referred to the description proposed by Tian et al. [4].

2.2. Quantum-Behaved Particle Swarm Optimization

Trajectory analysis by Clerc and Kennedy [6] demonstrated that convergence of PSO may be achieved if each particle converges to its local attractor, $LA_i^t = (la_{i1}^t, \dots, la_{id}^t, \dots, la_{iD}^t)$, defined as follows,

$$la_{id}^t = \frac{c_1 r_1 pbest_{id}^t + c_2 r_2 gbest_d^t}{c_1 r_1 + c_2 r_2} \tag{3}$$

or

$$la_{id}^t = \varphi pbest_{id}^t + (1 - \varphi) gbest_d^t, \varphi \sim U(0,1) \tag{4}$$

for $1 \leq d \leq D$, where t is the iteration number, φ is a random number uniformly distributed on $(0, 1)$, that is $\varphi \sim U(0,1)$, $pbest_i$ represents the best historical position found by particle i , and $gbest$ is the current global best position found by the swarm.

Inspired by quantum mechanics and the trajectory analysis of PSO, two versions of quantum-behaved PSO (QPSO), called QPSO with delta potential well model (QDPSPSO) [7] and a revised QPSO (RQPSO) [8], were proposed one after another by Sun et al. in 2004.

In QDPSPSO, the position of particle i at iteration t is updated according to (5) by employing the Monte Carlo method.

$$x_{id}^{t+1} = \begin{cases} la_{id}^t + \alpha |la_{id}^t - x_{id}^t| \ln(1/u), & \text{if } rand \geq 0.5 \\ la_{id}^t - \alpha |la_{id}^t - x_{id}^t| \ln(1/u), & \text{otherwise} \end{cases} \tag{5}$$

wherein, both u and $rand$ are random numbers uniformly distributed on $(0, 1)$ and α is a positive real number, named the contraction–expansion coefficient, which can be set as $\alpha = 0.5 + 0.5(T - t)/T$ to balance the global and local searching ability of QDPSPSO, wherein parameter t and T are the current and maximum iteration numbers, respectively.

In RQPSO, a global point, denoted as $mbest = (mbest_1, \dots, mbest_d, \dots, mbest_D)$ and called the mean personal best position, is introduced to enhance the global searching ability of QPSO. The global point corresponding to the t –th iteration is computed by Equation (6).

$$mbest^t = \frac{1}{S} \left(\sum_{i=1}^S pbest_{i1}^t, \dots, \sum_{i=1}^S pbest_{id}^t, \dots, \sum_{i=1}^S pbest_{iD}^t \right), d = 1, 2, \dots, D \tag{6}$$

wherein, parameter t is the current iteration number and S is the number of particles.

The position of particle i at iteration t in RQPSO is updated as shown Equation (7).

$$x_{id}^{t+1} = \begin{cases} la_{id}^t + \alpha |mbest_d^t - x_{id}^t| \ln(1/u), & \text{if } rand \geq 0.5 \\ la_{id}^t - \alpha |mbest_d^t - x_{id}^t| \ln(1/u), & \text{otherwise} \end{cases} \tag{7}$$

wherein, α and u have the same meaning as the corresponding parameters in (5), and $mbest_d^t$ is the mean personal best position of the population for the d –th dimension at the t –th iteration.

2.3. Population Diversity

The population diversity of PSO is useful for measuring and dynamically adjusting the algorithm’s ability of best path exploration [18]. Lu et al. [20] proposed a method to measure the population diversity using (8).

$$\sigma^2(t) = \sum_{i=1}^S \left(\frac{f_i^{(t)} - f_{avg}^{(t)}}{F} \right)^2, \quad f_{avg}^{(t)} = \frac{1}{S} \sum_{i=1}^S f_i^{(t)} \quad (8)$$

In (8), $\sigma^2(t)$ is the sum of squares of deviations of the particles’ fitness values, S stands for the swarm size, $f_i^{(t)}$ is the fitness of the i -th particle at the t -th iteration, $f_{avg}^{(t)}$ is the average fitness of the swarm at the t -th iteration, and F is the normalized calibration factor to confine $\sigma^2(t)$. Lu et al. [20] defined F as (9).

$$F = \begin{cases} \max |f_i^{(t)} - f_{avg}^{(t)}|, & \text{if } \max |f_i^{(t)} - f_{avg}^{(t)}| > 1 \\ 1, & \text{otherwise} \end{cases} \quad (9)$$

3. Quantum-Behaved Particle Swarm Optimization with Weighted Mean Personal Best Position and Adaptive Local Attractor (ALA-QPSO)

3.1. Weighted Mean Personal Best Position

In RQPSO, the mean personal best position of the population (*mbest*) is tracked by particle i during the searching process of the particle. An equal weight coefficient is used to construct the linear combination of each particle best position, which does not distinguish the difference of the effect of the particles with different fitness on guiding particle i to search global optimal solutions. Some useful information hidden in particles’ personal best positions may be lost, which is not conducive to improve the global search ability of QPSO algorithm. For a minimum optimization problem, the elite are the particles which are corresponding to the minimum objective function value. The smaller the objective function value is, the better the particles’ fitness. In societal settings, the elite contribute more towards qualitative improvements. Such an analogy is drawn into the mean personal best position calculation by assigning larger weights to particles having better fitness and smaller weights to comparatively poorer performing particles. In this paper, a weighted mean personal best position, which is calculated by Equations (10) and (11) accordingly to the feedback of fitness values of the particles, can be obtained for guiding particle i to find global optimal solutions for a minimum optimization problem.

$$r_i(t) = \begin{cases} \frac{1}{S-1} \left(1 - \frac{f_{obj}^i(t)}{\sum_{k=1}^S f_{obj}^k(t)} \right), & \text{if } \sum_{k=1}^S f_{obj}^k(t) \neq 0 \\ \frac{1}{S}, & \text{others} \end{cases} \quad (10)$$

$$mpbest^t = \sum_{i=1}^S r_i(t) pbest_i^t = \left(\sum_{i=1}^S r_i(t) pbest_{i1}^t, \dots, \sum_{i=1}^S r_i(t) pbest_{id}^t, \dots, \sum_{i=1}^S r_i(t) pbest_{iD}^t \right) \quad (11)$$

In (10) and (11), t is the current iteration number, $f_{obj}^i(t)$ is the objective function value corresponding to the i -th particle at the t -th iteration, S is the number of particles, and $r_i(t)$ is the coefficient of the best position of the i -th particle at the t -th iteration and is used to construct the weighted mean personal best position.

From (10), it can be deduced that the sum of $r_i(t)$ is 1 and $r_i(t)$ is between 0 and 1 at iteration t . When the sum of the objective function value corresponding to all particles is 0, the coefficient $r_i(t)$ is $1/S$. Otherwise, the smaller $f_{obj}^i(t)$ is, the larger $r_i(t)$ is. That is to say, when constructing a weighted mean personal best position to guide particles for finding optimal solutions for a minimize optimization problem, the larger a particle fitness is, the more important the particle best position is. Thus, the difference of the effect of the particles with different fitness is distinguished.

3.2. Adaptive Local Attractor

Trajectory analysis in [6] show that each particle in the PSO converges to its local attractor. From Equations (3) or (4), the local attractor combines the particle best known position ($pbest_i$) and the entire swarm’s best known position ($gbest$). So, it is very useful to find an efficient way to combine the good information hidden in these two best known positions.

Generally, in population-based optimization methods, during the early stages of the optimization, it is desirable to encourage the individuals to wander through the entire search space, without gathering around local optima. On the other hand, during the latter stages, it is very important to enhance convergence toward the global optima, to find the optimum solution efficiently. Moreover, diversity is an important aspect of population-based optimization methods since it influences their performance, and diversity is closely linked to the exploration–exploitation tradeoff. High diversity facilitates exploration, which is usually required during the initial iterations of the optimization algorithm. A low diversity is indicative of exploitation of a small area of the search space, desired during the latter part of the optimization process. Furthermore, the experience of each particle have more influence on particles when they update their next position at the beginning of iterations, and the experience of other particles has more influence on particles when they update their next position at the later stage of iterations.

Equation (8) shows that $\sigma^2(t)$ belongs to the interval between 0 and S . When all particles locate in the same position, $\sigma^2(t)$ is zero. That stands for the swarm aggregation degree at its strongest. Contrarily, $\sigma^2(t)$ is S on the condition that all absolute differences between the current fitness of all particles and their average fitness equal to one. Thus, the sum of squares of deviations of the particles’ fitness values generally shows a decreasing trend with the number of iterations increasing.

Considering those concerns, in this paper, we propose a novel way of computing the local attractor to achieve the above scheme, and its equation is shown as Equation (12). The objective of this development is to enhance the global search in the early part of the optimization and to encourage the particles to converge toward the global optima at the end of the search. Then, the position of particle i at iteration t is updated as shown (13) in our proposed QPSO with weighted mean personal best position and Adaptive Local Attractor (ALA-QPSO).

$$Ala_{id}^t = \varphi \frac{\sigma^2(t)}{S} pbest_{id}^t + (1 - \varphi) \left(1 - \frac{\sigma^2(t)}{S} \right) gbest_d^t, \varphi \sim U(0, 1), d = 1, 2, \dots, D \quad (12)$$

$$x_{id}^{t+1} = \begin{cases} Ala_{id}^t + \alpha |mpbest_d^t - x_{id}^t| \ln(1/u), & \text{if } rand \geq 0.5 \\ Ala_{id}^t - \alpha |mpbest_d^t - x_{id}^t| \ln(1/u), & \text{otherwise} \end{cases} \quad (13)$$

In (12), $\sigma^2(t)$ is the sum of squares of deviations of the particles’ fitness values at iteration t , S is the swarm size, and φ is a random number uniformly distributed on $(0, 1)$. In (13), α and u have the same meaning as the corresponding parameters in (5), $mpbest_d^t$ is the weighted mean personal best position for the d –th dimension at iteration t , and its calculation equation is shown as (11); Ala_{id}^t is the adaptive local attractor of particle i for the d –th dimension at iteration t .

3.3. Quantum-Behaved Particle Swarm Optimization with Adaptive Local Attractor (ALA-QPSO)

Applying the weighted mean personal best position in (11) and the adaptive local attractor in (12) to QPSO, the proposed QPSO with weighted mean personal best position and Adaptive Local Attractor (ALA-QPSO) is described in Algorithm 1 below.

Algorithm 1 ALA-QPSO

1. Initialize the parameters including swarm size S , maximum iteration count T .
 2. Initialize particles in the population with random position vectors.
 3. Calculate $pbest$ for each particle and $gbest$ for the swarm.
 4. Calculate population diversity $\sigma^2(t)$ using (8).
 5. Update the weighted mean personal best position $mpbest^t$ using (11).
 6. Update the adaptive local attractor for each particle using (12).
 7. Update the position for each particle using (13).
 8. If terminating condition is not meet, go to step 3.
-

4. Experiments and Discussion

4.1. Benchmark Functions

In this section, twelve classical benchmark functions are listed in Table 1, which are used to verify the performance of the ALA-QPSO algorithm. These benchmark functions are widely adopted in numerical optimization methods [21]. In this paper, the twelve benchmark functions are divided into three groups. The first group includes five unimodal functions ($f_1 \sim f_5$). The second group includes four complex multimodal functions ($f_6 \sim f_9$), where f_7 owns a large number of local optima so that it is difficult to find the global optimization values, f_8 has many minor local optima, f_9 includes linkage components among variables so it is difficult to reach theoretical optimal values. The third group includes three rotated multimodal functions ($f_{10} \sim f_{12}$). In Table 1, D denotes the solution space dimension, the global optimal solution (x^*) and the global optimal value ($f(x^*)$) of twelve classical benchmark functions are given in column 5 and column 6, respectively.

4.2. Experimental Settings

The benchmark functions defined in the previous subsection are used to test the performance of ALA-QPSO. Five algorithms are used to compare for benchmark functions: Artificial bee colony (ABC) [22], revised QPSO (RQPSO) [8], QPSO with Gaussian distributed attractor (GAQPSO) [16], an enhanced QPSO based on a novel computing way of local attractor (EQPSO) [17], and an improved QPSO with weighted mean best position (WQPSO) [23].

For all compared algorithms, the population size is set to 20 [22,24]. For ABC, the limit number of iterations which a food source cannot be improved (Trial limit) is set to 100 [25]. For all kinds of QPSO, the contraction–expansion coefficient (α) decreases linearly from 1.0 to 0.5 during the search process, the local attractors are calculated by Equation (4) except for ALA-QPSO.

Two group experiments are tested. Firstly, setting the convergence criterion as reaching the maximum number of iterations, the mean (Mean) and the standard deviation (SD) of the final fitness value at the end of iteration over multiple runs are compared to test the difference of these compared algorithms solution accuracy and stability. Secondly, setting the convergence criterion as not exceeding the maximum number of iterations and the value of the objective function reaching the allowable precision, the success rate (SR), average convergence iteration number (AIN), and average computational time (Time) over multiple runs are compared for testing the difference of these compared algorithms convergence speed. To obtain an unbiased CPU time comparison, all experiments are conducted using MATLAB R2018a on a personal computer with an Intel Core i7-7500U 2.7 GHz CPU and 16 GB memory.

4.3. Results and Discussions

4.3.1. Comparison of the Solution Accuracy and Stability

In the experiments, the first comparison of tested functions in Table 1 is conducted on 30, 50, and 100 dimensions, and the maximum iteration number (T) is set 10,000, 10,000, and 2000, respectively. All compared algorithms terminate calculation when the maximum iteration number is reached. The results are shown in Tables 2–4 in terms of the mean and standard deviation of the solutions obtained in the 30 independent runs by each algorithm. Note that the mean of solutions indicates the solution quality of the algorithms, and the standard deviation represents the stability of the algorithms.

Table 1. High dimensional classical benchmark functions.

Group	Name	Test Function ¹	Search Space	Global opt. x^*	$f(x^*)$
Unimodal	Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	$[0]^D$	0
	Schwefel's 2.22	$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$	$[0]^D$	0
	Schwefel's 1.2	$f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^D$	$[0]^D$	0
	Schwefel's 2.21	$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	$[-100, 100]^D$	$[0]^D$	0
	Step	$f_5(x) = \sum_{i=1}^D ([x_i + 0.5])^2$	$[-100, 100]^D$	$[-0.5]^D$	0
Multimodal	Rosenbrock	$f_6(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-30, 30]^D$	$[1]^D$	0
	Rastrigin	$f_7(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	$[0]^D$	0
	Ackley	$f_8(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right) + 20 + e$	$[-32, 32]^D$	$[0]^D$	0
	Griewank	$f_9(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$	$[0]^D$	0
Rotated	Rotated Griewank	$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^D y_i^2 - \prod_{i=1}^D \cos \left(\frac{y_i}{\sqrt{i}} \right) + 1, y = M * x, M$ is an orthogonal matrix.	$[-600, 600]^D$	$[0]^D$	0
	Rotated Weierstrass	$f_{11}(x) = \sum_{i=1}^D \sum_{k=0}^{20} \left[\frac{1}{2^k} \cos(3^k(2\pi y_i + \pi)) \right] - D \sum_{k=0}^{20} \left[\frac{1}{2^k} \cos(3^k \pi) \right], y = M * x, M$ is an orthogonal matrix.	$[-0.5, 0.5]^D$	$[0]^D$	0
	Rotated Rastrigin	$f_{12}(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10), y = M * x, M$ is an orthogonal matrix.	$[-5.12, 5.12]^D$	$[0]^D$	0

¹ The orthogonal rotation matrix M is generated by the Salomon's method and used to calculate $f_{10} \sim f_{12}$.

Table 2. Mean and standard deviations of the function values ($S = 20; D = 30; T = 10,000$).

f	Criteria	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO
f_1	Mean	3.2115×10^{-16}	0	4.3697×10^{-62}	3.0586×10^{-59}	1.0489×10^{-295}	0
	SD	6.4479×10^{-17}	0	2.3700×10^{-61}	9.3210×10^{-59}	0	0
f_2	Mean	6.5535×10^{-16}	0	2.6295	1.8088×10^{-36}	1.9995×10^{-183}	0
	SD	7.4422×10^{-17}	0	9.6839×10^{-1}	9.8507×10^{-36}	0	0
f_3	Mean	2.5595×10^4	0	9.4301×10^{-2}	6.8640×10^{-2}	5.8909×10^{-11}	0
	SD	3.4939×10^3	0	8.8509×10^{-2}	7.5320×10^{-2}	1.5317×10^{-10}	0
f_4	Mean	3.1831	0	4.8093×10^{-8}	1.3798×10^{-6}	6.8607×10^{-47}	0
	SD	2.1421	0	6.9539×10^{-8}	2.0440×10^{-6}	3.0515×10^{-46}	0
f_5	Mean	2.0000×10^{-1}	0	0	0	0	0
	SD	4.0684×10^{-1}	0	0	0	0	0
f_6	Mean	2.7875×10^1	2.7256×10^1	3.2247×10^1	3.3247×10^1	4.1283×10^1	2.7188×10^1
	SD	2.4684×10^1	3.4765×10^{-2}	2.1339×10^1	2.2535×10^1	3.0141×10^1	2.0527×10^{-1}
f_7	Mean	1.3148×10^2	0	1.3565×10^1	1.5995×10^1	6.2577×10^1	0
	SD	3.9721×10^1	0	3.7272	4.1657	2.8038×10^1	0
f_8	Mean	1.6520×10^{-14}	1.8356×10^{-15}	7.6383×10^{-15}	6.6909×10^{-15}	5.8620×10^{-15}	1.3619×10^{-15}
	SD	3.2788×10^{-15}	1.5283×10^{-15}	2.8908×10^{-15}	1.8027×10^{-15}	1.0840×10^{-15}	1.7413×10^{-15}
f_9	Mean	8.5117×10^{-17}	0	1.1383×10^{-2}	1.3693×10^{-2}	4.1262×10^{-3}	0
	SD	8.5915×10^{-17}	0	1.6879×10^{-2}	1.4073×10^{-2}	7.9803×10^{-3}	0
f_{10}	Mean	1.4624×10^{-3}	0	9.8340×10^{-3}	8.7590×10^{-3}	1.6271×10^{-2}	0
	SD	4.1486×10^{-3}	0	1.5979×10^{-2}	1.5123×10^{-2}	2.3173×10^{-2}	0
f_{11}	Mean	3.9661×10^1	0	1.5813×10^1	2.2942×10^1	3.4979×10^1	0
	SD	1.0377	0	3.3834	7.3531	4.7694	0
f_{12}	Mean	2.3683×10^2	0	4.3726×10^1	9.0690×10^1	1.7675×10^2	0
	SD	1.2881×10^1	0	2.3307×10^1	5.9659×10^1	1.6948×10^1	0

Mean: mean of objective values. SD: standard deviation.

Table 2 shows the results on 30 dimension problems. According to the results, ALA-QPSO can find better average objective function values than those of other algorithms on f_6 and f_8 , both ALA-QPSO and EQPSO can find the global optimum on the remaining functions. On the other hand, the standard deviations of the ALA-QPSO algorithm are smaller than those of ABC, GAQPSO, RQPSO, and WQPSO algorithms on all tested functions. Although the standard deviation of ALA-QPSO is larger than that of EQPSO on f_6 and f_8 , the ALA-QPSO algorithm provides smaller average objective function values than those of EQPSO on these two functions.

Table 3 shows that the results on 50 dimension problems. From Table 3, it can be got that both ALA-QPSO and EQPSO can find the global optimum on $f_1 \sim f_3, f_5, f_7$ and $f_9 \sim f_{12}$, and can find equal average objective function value on f_8 . ALA-QPSO obtains better average objective function values than other algorithms on the remaining functions. The standard deviation of ALA-QPSO is less than or equal to that of EQPSO on all the test functions. ALA-QPSO and EQPSO achieve higher accuracy and stronger stability than other algorithms on all test functions.

Table 4 shows that the results on 100 dimension problems. From Table 4, one can see that both ALA-QPSO and EQPSO provide a solution to the true optimum values for six out of the twelve benchmark functions (f_5, f_7 and $f_9 \sim f_{12}$). ALA-QPSO can find better average objective function values than those of other algorithms on $f_1 \sim f_4$ and f_6 . Overall, the standard deviation of ALA-QPSO and EQPSO is smaller than that of other algorithms.

Table 3. Mean and standard deviations of the function values ($S = 20; D = 50; T = 10,000$).

f	Criteria	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO
f_1	Mean	3.6199×10^{-4}	0	6.7392×10^{-29}	6.0020×10^{-23}	1.1225×10^{-187}	0
	SD	1.3183×10^{-3}	0	1.5822×10^{-28}	1.6548×10^{-22}	0	0
f_2	Mean	2.0161×10^{-15}	0	3.2628×10^1	1.8119×10^{-15}	1.8498×10^{-122}	0
	SD	3.4824×10^{-16}	0	3.9216	6.1161×10^{-15}	4.8665×10^{-122}	0
f_3	Mean	9.4472×10^4	0	2.1336×10^2	6.1731×10^2	1.0466×10^2	0
	SD	8.6811×10^3	0	6.1093×10^1	3.2722×10^2	2.0518×10^2	0
f_4	Mean	6.0318×10^1	4.9407×10^{-324}	2.1843×10^{-2}	2.4619×10^{-1}	2.7663×10^{-19}	0
	SD	9.1391	0	9.5378×10^{-3}	9.7297×10^{-2}	9.2674×10^{-19}	0
f_5	Mean	2.2833×10^1	0	0	3.3333×10^{-2}	0	0
	SD	4.5945	0	0	1.8257×10^{-1}	0	0
f_6	Mean	6.2760×10^6	4.7372×10^1	5.3485×10^1	5.8728×10^1	8.3258×10^1	4.7255×10^1
	SD	2.7022×10^6	2.5471×10^{-1}	2.1756×10^1	2.7264×10^1	4.4773×10^1	2.1455×10^{-1}
f_7	Mean	3.9881×10^2	0	3.0224×10^1	3.9652×10^1	1.9921×10^2	0
	SD	3.3527×10^1	0	9.7823	1.1177×10^1	5.6608×10^1	0
f_8	Mean	1.6808×10^{-6}	2.6645×10^{-15}	9.9654×10^{-14}	6.0201×10^{-13}	7.1646×10^{-15}	2.6645×10^{-15}
	SD	3.3123×10^{-6}	0	1.2662×10^{-13}	7.5045×10^{-13}	2.4567×10^{-15}	0

Table 3. Cont.

<i>f</i>	Criteria	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO
<i>f</i> ₉	Mean	4.1901×10^{-4}	0	2.3000×10^{-3}	4.6762×10^{-3}	1.5135×10^{-3}	0
	SD	2.2474×10^{-3}	0	4.4326×10^{-3}	9.5542×10^{-3}	3.4264×10^{-3}	0
<i>f</i> ₁₀	Mean	1.3995	0	1.9316×10^{-2}	2.1549×10^{-2}	1.4702×10^{-1}	0
	SD	8.3632×10^{-1}	0	9.5536×10^{-3}	1.2197×10^{-2}	1.0084×10^{-1}	0
<i>f</i> ₁₁	Mean	7.4129×10^1	0	3.4137×10^1	5.9507×10^1	6.9111×10^1	0
	SD	1.2208	0	6.4357	1.1104×10^1	5.8322	0
<i>f</i> ₁₂	Mean	5.3144×10^2	0	1.3132×10^2	2.5030×10^2	3.7453×10^2	0
	SD	1.9377×10^1	0	5.3289×10^1	9.9465×10^1	3.1666×10^1	0

Mean: mean of objective values. SD: standard deviation.

Table 4. Mean and standard deviations of the function values (*S* = 20; *D* = 100; *T* = 2000).

<i>f</i>	Criteria	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO
<i>f</i> ₁	Mean	5.6282×10^4	8.9938×10^{-270}	1.2873×10^1	7.8668×10^1	6.8659×10^{-16}	4.9774×10^{-283}
	SD	9.1711×10^3	0	9.3089	4.2189×10^1	1.0448×10^{-15}	0
<i>f</i> ₂	Mean	2.4596×10^{16}	1.6177×10^{-150}	3.6510×10^3	1.9578	5.3771×10^{-13}	4.0840×10^{-155}
	SD	8.1969×10^{16}	8.7797×10^{-150}	1.7832×10^4	5.7540×10^{-1}	4.7000×10^{-13}	7.1024×10^{-155}
<i>f</i> ₃	Mean	4.4087×10^5	1.6281×10^{-171}	2.5833×10^4	3.9155×10^4	7.3538×10^4	8.8599×10^{-172}
	SD	4.5889×10^4	0	4.1772×10^3	6.8323×10^3	1.4797×10^4	0
<i>f</i> ₄	Mean	9.1993×10^1	3.6553×10^{-108}	2.0536×10^1	2.9877×10^1	1.7931×10^1	1.9923×10^{-108}
	SD	1.2479	8.9262×10^{-108}	2.5961	3.4435	9.7999	3.7736×10^{-108}
<i>f</i> ₅	Mean	5.2736×10^4	0	7.1800×10^1	1.7167×10^2	0	0
	SD	1.2741×10^4	0	7.1461×10^1	1.0108×10^2	0	0
<i>f</i> ₆	Mean	2.0590×10^8	9.8213×10^1	2.0340×10^3	2.2630×10^4	9.7107×10^1	9.8063×10^1
	SD	6.1190×10^7	2.4313×10^{-1}	1.0493×10^3	2.1261×10^4	4.4917×10^{-1}	3.7079×10^{-1}
<i>f</i> ₇	Mean	1.1560×10^3	0	1.5749×10^2	2.4022×10^2	6.7970×10^2	0
	SD	6.9418×10^1	0	2.8214×10^1	4.8421×10^1	1.2108×10^2	0
<i>f</i> ₈	Mean	1.7282×10^1	2.6645×10^{-15}	1.4915	2.7322	2.8350×10^{-9}	2.6645×10^{-15}
	SD	7.5513×10^{-1}	0	5.3252×10^{-1}	5.0710×10^{-1}	1.8879×10^{-9}	0
<i>f</i> ₉	Mean	4.6645×10^2	0	1.0690	1.6927	4.9550×10^{-3}	0
	SD	9.1110×10^1	0	1.3023×10^{-1}	3.6188×10^{-1}	1.2937×10^{-2}	0
<i>f</i> ₁₀	Mean	2.1488×10^3	0	7.0063	1.3661×10^1	5.0429×10^{-2}	0
	SD	3.8195×10^2	0	2.6842	6.1557	1.3579×10^{-1}	0
<i>f</i> ₁₁	Mean	1.6597×10^2	0	6.9628×10^1	8.3404×10^1	1.2400×10^2	0
	SD	1.8829	0	7.5849	6.3336	2.7652×10^1	0
<i>f</i> ₁₂	Mean	1.8132×10^3	0	5.7674×10^2	8.8909×10^2	9.7741×10^2	0
	SD	6.7997×10^1	0	1.2077×10^2	1.5362×10^2	5.4787×10^1	0

Mean: mean of objective values. SD: standard deviation.

To present the total comparison on performance between ALA-QPSO and other algorithms, Table 5 shows the detailed results from the non-parametric Friedman test [26,27]. The last column in this table denotes the corresponding measured *p*-values, which suggest the significant differences between the compared algorithms at the 0.05 significance level. From Table 5, it can be obtained that there is a significant difference of accuracy and stability among the compared algorithms. Since the Friedman test assigns the lowest rank to the best performing algorithm, the conclusion can be obtained that the ALA-QPSO algorithm has the highest precision and the strongest stability among the compared algorithms.

Table 5. Average ranking of algorithms for test functions, as obtained by the Friedman test.

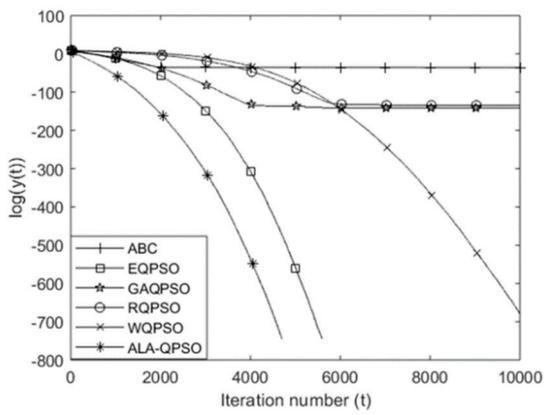
<i>D</i>	<i>T</i>	Criteria	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO	<i>p</i> -Value
30	10,000	Mean	5.17	1.71	4.17	4.33	4.08	1.54	7.02×10^{-8}
		SD	4.83	1.71	4.42	4.50	3.67	1.88	1.18×10^{-6}
50	10,000	Mean	5.67	1.63	3.71	4.58	3.88	1.54	3.16×10^{-9}
		SD	5.00	1.67	3.96	5.00	3.79	1.58	3.85×10^{-8}
100	2000	Mean	6.00	1.83	3.75	4.58	3.42	1.42	7.45×10^{-10}
		SD	5.25	1.58	4.17	4.67	3.83	1.50	1.70×10^{-8}

Moreover, Figures 1 and 2 present the converging curves for the twelve selected functions. Note that the logarithmic of the mean of the best object value according to the *t*-th iteration, which is denoted $\log(y(t))$, is as *y*-coordinate and the iteration number is as *x*-coordinate. Since the logarithmic function is a negative infinity (not shown in the figures) when the obtained best object value is zero, some curves stop after a certain number of function evaluations. From Figures 1 and 2, it can be

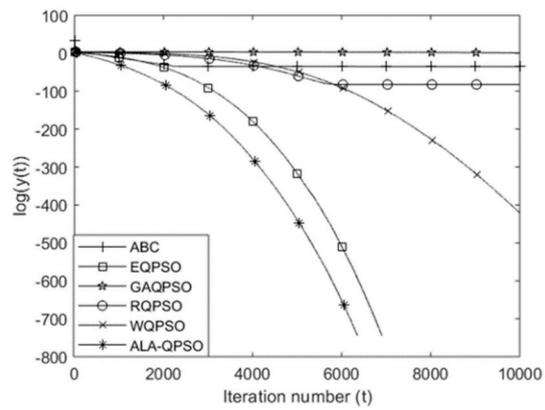
obtained that both ALA-QPSO and EQPSO can find the global optimum of $f_1 \sim f_5$, f_7 and $f_9 \sim f_{12}$ for 30 dimension problems, also, ALA-QPSO requires fewer iterations than EQPSO; ALA-QPSO can find smaller average objective function values than those of other algorithms on f_6 and f_8 for 30 dimension problems. The convergence curves of ALA-QPSO for all the functions sharply dropped off at a certain point during the early stage of optimization.

4.3.2. Comparison of the Convergence Speed and Reliability

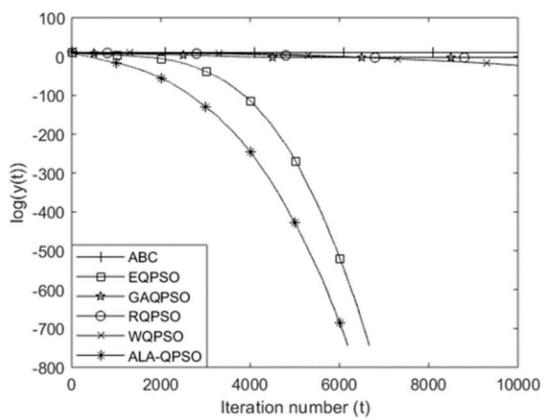
Each algorithm stops when the maximum number of iterations is arrived at or the value of the objective function is less than or equal to its target accuracy threshold. Wherein, the accuracy threshold for f_6 and f_8 is set as 2.8×10^1 and 5.0×10^{-15} , respectively, and for the others is set as 1.0×10^{-50} , the maximum iteration number is set as 10,000. Each algorithm was executed 30 times. Three comparing indexes are calculated for the twelve functions with 30 dimensions and are reported in Tables 6–8, including the success rate (SR), which is the percentage of the running number of experiments with the final population best solution meeting the accuracy threshold; the average convergence iteration number (AIN), which is the average number of iterations when the algorithm reaches the termination condition over multiple runs; and the average computational time (Time), which is the average of the running time that the algorithm meets the termination condition established for this comparison over multiple runs. Note that some of the compared algorithms may converge to the threshold for the tested functions, while the other may not. The minimum number of iterations required for the objective function meeting an accuracy threshold is used to calculate the AIN, while not meeting; the maximum number of iterations is used for the calculation.



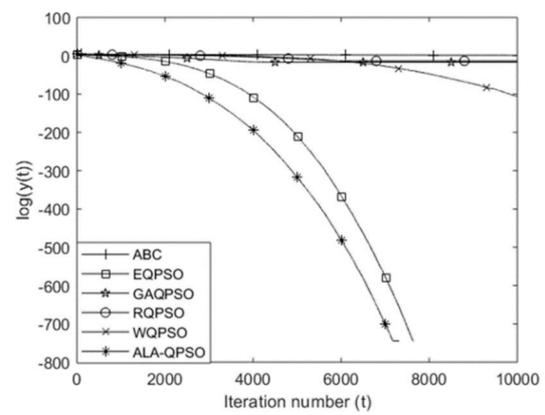
(a) f_1



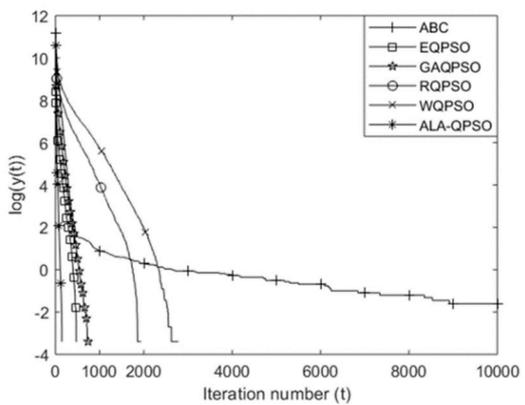
(b) f_2



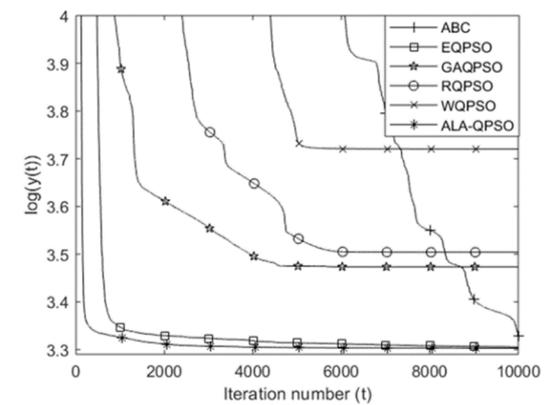
(c) f_3



(d) f_4



(e) f_5



(f) f_6

Figure 1. Convergence graphs of six algorithms on 30-D benchmark functions. $f_1 \sim f_6$: $y(t)$ represents the mean of the best object value according to the t -th.

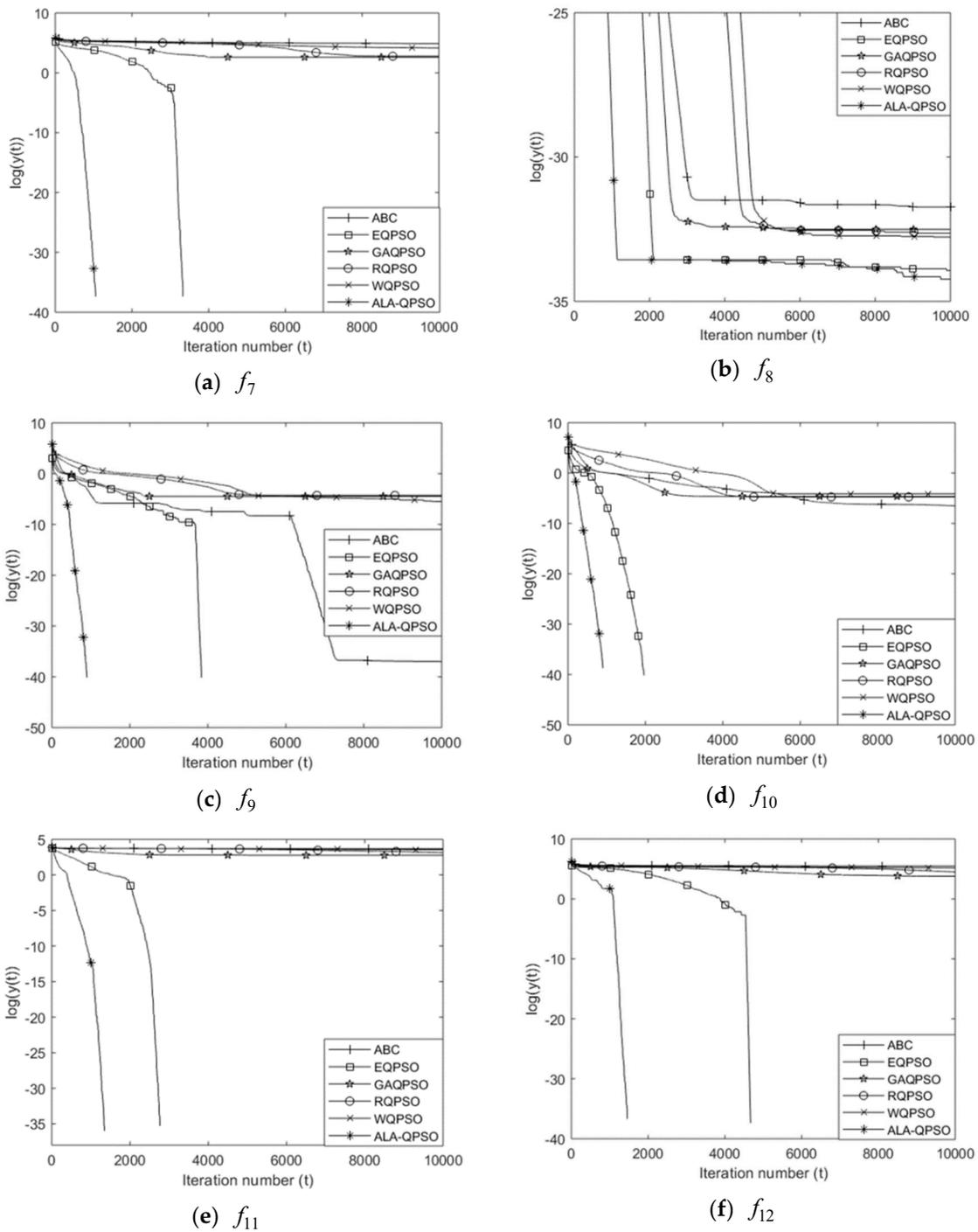


Figure 2. Convergence graphs of six algorithms on 30-D benchmark functions. $f_7 \sim f_{12}$: $y(t)$ represents the mean of the best object value according to the t -th.

Table 6 presents the success rate of the six compared algorithms on the tested functions with 30-Dimension. Both ALA-QPSO and EQPSO can obtain 100% success rate on all tested functions. The success rate of the two algorithms is higher than those of other algorithms for ten out of the twelve benchmark functions. The results indicate that ALA-QPSO and EQPSO show the best reliability among all the compared algorithms.

Table 7 presents the average convergence iteration numbers of all the compared algorithms on the tested functions with 30-Dimension. The average convergence iteration number of ALA-QPSO is the minimum among those of six compared algorithms for all tested functions. The results show that

ALA-QPSO needs the minimum number of convergence iteration to reach the accuracy threshold and that the optimize speed of ALA-QPSO is faster than those of other compared algorithms for all the tested functions.

Table 6. The success rate of six algorithms ($S = 20; D = 30; T = 10,000$).

SR (%)	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO
f_1	0.0	100.0	100.0	100.0	100.0	100.0
f_2	0.0	100.0	0.0	0.0	100.0	100.0
f_3	0.0	100.0	0.0	0.0	0.0	100.0
f_4	0.0	100.0	0.0	0.0	13.3	100.0
f_5	90.0	100.0	100.0	100.0	100.0	100.0
f_6	83.3	100.0	83.3	83.3	90.0	100.0
f_7	0.0	100.0	0.0	0.0	0.0	100.0
f_8	0.0	100.0	0.0	0.0	10.0	100.0
f_9	26.7	100.0	50.0	46.7	66.7	100.0
f_{10}	0.0	100.0	0.0	0.0	3.3	100.0
f_{11}	0.0	100.0	0.0	0.0	0.0	100.0
f_{12}	0.0	100.0	0.0	0.0	0.0	100.0

Table 7. The average convergence iteration number of six algorithms ($S = 20; D = 30; T = 10,000$).

AIN	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO
f_1	1.0000×10^4	2.6734×10^3	3.5675×10^3	5.4186×10^3	5.6027×10^3	1.6216×10^3
f_2	1.0000×10^4	3.3180×10^3	1.0000×10^4	1.0000×10^4	6.4183×10^3	2.4406×10^3
f_3	1.0000×10^4	3.9438×10^3	1.0000×10^4	1.0000×10^4	1.0000×10^4	2.7628×10^3
f_4	1.0000×10^4	4.0493×10^3	1.0000×10^4	1.0000×10^4	9.9972×10^3	3.0298×10^3
f_5	4.7939×10^3	4.0820×10^2	5.6233×10^2	1.7512×10^3	2.3136×10^3	1.1700×10^2
f_6	3.9238×10^3	1.7044×10^3	2.5791×10^3	3.7521×10^3	4.1808×10^3	8.8953×10^2
f_7	1.0000×10^4	2.5510×10^3	1.0000×10^4	1.0000×10^4	1.0000×10^4	8.5687×10^2
f_8	1.0000×10^4	2.0492×10^3	1.0000×10^4	1.0000×10^4	9.9056×10^3	1.0859×10^3
f_9	8.7063×10^3	2.3023×10^3	6.0576×10^3	7.3058×10^3	7.5273×10^3	8.3030×10^2
f_{10}	1.0000×10^4	1.8583×10^3	1.0000×10^4	1.0000×10^4	9.9930×10^3	8.4923×10^2
f_{11}	1.0000×10^4	2.3043×10^3	1.0000×10^4	1.0000×10^4	1.0000×10^4	1.1307×10^3
f_{12}	1.0000×10^4	3.7039×10^3	1.0000×10^4	1.0000×10^4	1.0000×10^4	8.9340×10^2

Table 8 presents the computational time of the six compared algorithms. It can be observed that the computational time of ALA-QPSO is the minimum among those compared algorithms. Since both ALA-QPSO and EQPSO can converge to the threshold for all tested functions, the computational time demonstrates the superiority of the proposed algorithm to some extent.

Table 8. The computational time of six algorithms ($S = 20; D = 30; T = 10,000$).

Time (S)	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO
f_1	5.3178	1.4530	2.1226	2.6127	4.3395	0.8115
f_2	5.0471	2.7620	8.4172	5.1515	4.0062	1.2683
f_3	7.3963	4.3318	6.1789	6.2239	6.0860	1.8055
f_4	6.3163	3.4015	4.8744	5.0325	4.8261	1.4174
f_5	3.1655	0.3873	0.2949	0.7712	1.0777	0.0748
f_6	2.7310	1.6312	1.3465	1.7113	2.1007	0.4562
f_7	5.2583	2.1870	4.9350	4.5979	4.5153	0.4413
f_8	5.0777	2.0003	4.3977	4.5203	4.8249	0.5535
f_9	3.5959	2.2915	2.7715	3.4466	3.6319	0.4417
f_{10}	6.1578	2.7307	7.6677	7.0954	7.8027	0.6347
f_{11}	24.0117	8.4135	23.0703	18.4729	23.8336	2.4332
f_{12}	7.7361	2.5692	6.4690	6.6778	6.9953	0.6512

To present the total comparison on convergence speed and reliability between ALA-QPSO and other algorithms, Table 9 shows the detailed results from the Friedman test. From Table 9, it can be obtained that there is a significant difference of convergence speed and reliability among those compared algorithms. Since the Friedman test assigns the lowest rank to the best performing algorithm, the conclusion can be obtained that the ALA-QPSO algorithm has the fastest convergence speed and the best reliability among all compared algorithms.

Table 9. Average ranking of competitor algorithms for the twelve benchmark functions, as obtained by the Friedman test ($S = 20$; $D = 30$; $T = 10,000$).

Criteria	ABC	EQPSO	GAQPSO	RQPSO	WQPSO	ALA-QPSO	<i>p</i> -Value
Time	5.50	2.17	3.67	4.17	4.50	1.00	7.94×10^{-9}
AIN	5.08	2.00	4.17	4.50	4.25	1.00	6.88×10^{-10}
SR	5.08	1.79	4.42	4.50	3.42	1.79	3.90×10^{-9}

5. Conclusions

In this paper, a variant of QPSO, namely ALA-QPSO, is proposed to simultaneously enhance the search performance of QPSO and acquire good global optimal ability. Firstly, a weight parameter is introduced to distinguish the difference of the effect of the particles with different fitness, and is used to obtain the weighted mean personal best position of the population. Secondly, a linear combination of the particle best known position and the entire swarm's best known position is designed to form the adaptive local attractor by using the sum of squares of deviations of the particles' fitness values as the linear combination coefficient. The objective of this development is to enhance the global search in the early part of the optimization and to encourage the particles to converge toward the global optima at the end of the search. Finally, the proposed ALA-QPSO algorithm was tested on twelve benchmark functions, and compared with the basic Artificial Bee Colony and the other four QPSO variants. Experimental results show that ALA-QPSO performs better than the compared methods in all of the benchmark functions in terms of better global search capability and faster convergence rate. We conclude that distinguishing the fitness of particles to calculate the weighted mean personal best position and monitoring the diversity of QPSO population to construct adaptive local attractor to guide particles optimization, is feasible. Although the proposed ALA-QPSO exhibited superior performance in the experimental results reported in the previous subsections, it is applicable only to the unconstrained problems in continuous search space. More modifications need to be done to further extend the applicability of the proposed ALA-QPSO to a more general class of optimization problems, including discrete, multi-objective, constrained, as well as dynamic optimization problems. Our further work will focus on using some other strategies to construct more effective local attractor for the QPSO algorithm and applying ALA-QPSO to the real-world optimization problems.

Funding: This research and the APC were funded by the Natural Science Research of the Education Department of Anhui Province, China (Grant no. KJ2016A522).

Acknowledgments: The authors would like to thank the editor and reviewers for their helpful suggestions and valuable comments on a previous version of the present paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks, Nagoya, Japan, 4–6 October 1995; pp. 1942–1948. [\[CrossRef\]](#)
2. Zhang, Y.D.; Wang, S.H.; Ji, G.L. A comprehensive survey on particle swarm optimization algorithm and its applications. *Math. Probl. Eng.* **2015**, *2015*, 38. [\[CrossRef\]](#)

3. Zhang, Y.D.; Wang, S.H.; Sui, Y.X.; Yang, M.; Liu, B.; Cheng, H.; Sun, J.D.; Jia, W.J.; Phillips, P.; Gorriz, J. Multivariate approach for Alzheimer's disease detection using stationary wavelet entropy and predator-prey particle swarm optimization. *J. Alzheimers Dis.* **2018**, *65*, 855–869. [[CrossRef](#)] [[PubMed](#)]
4. Tian, D.P.; Shi, Z.Z. MPSO: Modified particle swarm optimization and its applications. *Swarm Evol. Comput.* **2018**, *41*, 49–68. [[CrossRef](#)]
5. Nickabadi, A.; Ebadzadeh, M.M.; Safabakhsh, R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl. Soft Comput.* **2011**, *11*, 3658–3670. [[CrossRef](#)]
6. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolut. Comput.* **2002**, *6*, 58–73. [[CrossRef](#)]
7. Sun, J.; Xu, W.B.; Feng, B. A global search strategy of quantum-behaved particle swarm optimization. In Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 1–3 December 2004; pp. 111–116.
8. Sun, J.; Feng, B.; Xu, W.B. Particle swarm optimization with particles having quantum behavior. In Proceedings of the 2004 IEEE Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; pp. 326–331.
9. Zhang, Y.D.; Ji, G.L.; Yang, J.Q.; Wang, S.H.; Dong, Z.C. Preliminary research on abnormal brain detection by wavelet-energy and quantum-behaved PSO. *Biomed. Eng.* **2016**, *61*, 431–441. [[CrossRef](#)]
10. Li, L.L.; Jiao, L.C.; Zhao, J.Q.; Shang, R.H.; Gong, M.G. Quantum-behaved discrete multi-objective particle swarm optimization for complex network clustering. *Pattern Recogn.* **2017**, *63*, 1–14. [[CrossRef](#)]
11. Feng, Z.K.; Niu, W.J.; Cheng, C.T. Multi-objective quantum-behaved particle swarm optimization for economic environmental hydrothermal energy system scheduling. *Energy* **2017**, *131*, 165–178. [[CrossRef](#)]
12. Fang, W.; Sun, J.; Wu, X.; Palade, V. Adaptive web QoS controller based on online system identification using quantum-behaved particle swarm optimization. *Soft Comput.* **2015**, *19*, 1715–1725. [[CrossRef](#)]
13. Sing, M.R.; Mahapatra, S.S. A quantum behaved particle swarm optimization for flexible job shop scheduling. *Comput. Ind. Eng.* **2016**, *93*, 36–44. [[CrossRef](#)]
14. Tang, D.; Cai, Y.; Xue, Y. A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems. *Inf. Sci.* **2014**, *289*, 162–189. [[CrossRef](#)]
15. Yang, Z.L.; Wu, A.; Min, H.Q. An improved quantum-behaved particle swarm optimization algorithm with elitist breeding for unconstrained optimization. *Comput. Intell. Neurosci.* **2015**, *2015*, 12. [[CrossRef](#)] [[PubMed](#)]
16. Sun, J.; Fang, W.; Palade, V.; Wu, X.J.; Xu, W.B. Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. *Appl. Math. Comput.* **2011**, *218*, 3763–3775. [[CrossRef](#)]
17. Jia, P.; Duan, S.; Yan, J. An enhanced quantum-behaved particle swarm optimization based on a novel computing way of local attractor. *Information* **2015**, *6*, 633–649. [[CrossRef](#)]
18. Han, F.; Liu, Q. A diversity-guided hybrid particle swarm optimization algorithm based on gradient search. *Neurocomputing* **2014**, *137*, 234–240. [[CrossRef](#)]
19. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE Congress on Computational Intelligence, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
20. Lu, Z.S.; Hou, Z.R.; Du, J. Particle swarm optimization with adaptive mutation. *Front. Electr. Electron. Eng. China* **2006**, *1*, 99–104. [[CrossRef](#)]
21. Chen, K.; Zhou, F.Y.; Liu, A.L. Chaotic dynamic weight particle swarm optimization for numerical function optimization. *Knowl-Based Syst.* **2018**, *139*, 23–40. [[CrossRef](#)]
22. Karaboga, D.; Akay, B. A modified Artificial Bee Colony algorithm for real-parameter optimization. *Inf. Sci.* **2012**, *192*, 120–142.
23. Xi, M.L.; Sun, J.; Xu, W.B. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. *Appl. Math. Comput.* **2008**, *205*, 751–759. [[CrossRef](#)]
24. Zhang, L.P.; Yu, H.J.; Hu, S.X. Optimal choice of parameters for particle swarm optimization. *J. Zhejiang Univ. Sci.* **2005**, *6*, 528–534.
25. Ye, W.X.; Feng, W.Y.; Fan, S.H. A novel multi-swarm particle swarm optimization with dynamic learning strategy. *Appl. Soft Comput.* **2017**, *61*, 832–843. [[CrossRef](#)]

26. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]
27. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]



© 2019 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).