Information **2015**, *6*, 633-649; doi:10.3390/info6040633

OPEN ACCESS

information

ISSN 2078-2489

www.mdpi.com/journal/information

Article

An Enhanced Quantum-Behaved Particle Swarm Optimization Based on a Novel Computing Way of Local Attractor

Pengfei Jia, Shukai Duan * and Jia Yan

College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China;
E-Mails: jiapengfei200609@126.com (P.J.); yanjia119@163.com (J.Y.)

* Author to whom correspondence should be addressed; E-Mail: duansk@swu.edu.cn;
Tel./Fax: +86-23-6825-1252.

Academic Editor: Willy Susilo

Received: 14 July 2015 / Accepted: 6 October 2015 / Published: 13 October 2015

Abstract: Quantum-behaved particle swarm optimization (QPSO), a global optimization method, is a combination of particle swarm optimization (PSO) and quantum mechanics. It has a great performance in the aspects of search ability, convergence speed, solution accuracy and solving robustness. However, the traditional QPSO still cannot guarantee the finding of global optimum with probability 1 when the number of iterations is limited. A novel way of computing the local attractor for QPSO is proposed to improve QPSO's performance in global searching, and this novel QPSO is denoted as EQPSO during which we can guarantee the particles are diversiform at the early stage of iterations, and have a good performance in local searching ability at the later stage of iteration. We also discuss this way of computing the local attractor in mathematics. The results of test functions are compared between EQPSO and other optimization techniques (including six different PSO and seven different optimization algorithms), and the results found by the EQPSO are better than other considered methods.

Keywords: QPSO; optimization algorithm; local attractor; global optimum

1. Introduction

QPSO (Quantum-behaved Particle Swarm Optimization) is a novel optimization method, which is proposed based on the PSO (Particle Swarm Optimization) algorithm and quantum mechanics. The QPSO algorithm has been greatly improved in the aspects of search ability, convergence speed, solution accuracy and solving robustness, and it can also overcome the shortcoming that the standard PSO (SPSO) cannot guarantee the global convergence [1,2]. For this reason, QPSO has been widely used in bio-medicine, antenna design, combinatorial optimization, signal processing, neural networks and other fields [3–11].

Besides QPSO, some other algorithms such as genetic algorithm (GA), evolution strategy with covariance matrix adaptation (CMA-ES) [12], krill herd (KH) [13–15], monarch butterfly optimization (MBO) [16], harmony search (HS) [17,18] and artificial plant optimization algorithm (APOA) [19] are also proposed by researchers to solve optimization problems.

The disadvantage of a QPSO algorithm is that it cannot find the global optimum with probability 1 when the number of iterations is limited. The number of total iterations is usually a real number and cannot be set as infinity during the practical application. Therefore, we propose a novel way to compute the local attractor of QPSO to improve the global searching ability of QPSO when the number of iterations is limited. In the rest of this paper, we will firstly introduce the basic theory of SPSO and QPSO, then the proposed way of computing the local attractor is expressed and analyzed. Finally, the enhanced QPSO will be tested by the test functions, and its results are compared with six other PSO algorithms and seven optimization algorithms.

2. QPSO-Based Optimization Algorithm

2.1. Standard Particle Swarm Optimization

PSO is an evolutionary computation algorithm based on swarm intelligence theory. The algorithm comes from the simulation of the bird predation behavior, and its emphases lie in the cooperation and competition between individuals. Because of its rapidity of calculation and ease of implementation, PSO algorithms have been successfully applied in system identification, neural network training, fuzzy system control and other application fields.

In PSO [20], each candidate solution is called a particle and all of the particles are seen to have no quality and volume. Each particle knows its best position, which is called the local optimum. The best position of all the particles is called the global optimum. Suppose a population including *M* particles moves in a *D* dimensional space with a certain velocity, then PSO will update the speed and position according to Equations (1) and (2).

$$\begin{aligned} V_{id}(t) = V_{id}(t-1) + c_1 r_1 (pbest_{id} - X_{id}(t-1)) \\ + c_2 r_2 (gbest_{gd} - X_{id}(t-1)) \end{aligned} \tag{1}$$

$$X_{id}(t) = X_{id}(t-1) + V_{id}(t) \tag{2}$$

where $i = 1, 2, \cdots M$, c_1 and c_2 are the learning factors, and generally, $c_1 = c_2 = 1$, r_1 and r_2 are the random numbers uniformly distributed between 0 and 1.

To improve the optimization ability, Shi [21,22] put forward a PSO algorithm with an inertial weight ω which can be decreased linearly from 1 to 0.1. Then Equation (1) can be changed to Equation (3).

$$V_{id}(t) = \omega V_{id}(t-1) + c_1 r_1 (pbest_{id} - X_{id}(t-1)) \\ + c_2 r_2 (gbest_{gd} - X_{id}(t-1)) \quad (3)$$

The algorithm expressed by Equations (2) and (3) is commonly referred to as the standard PSO (SPSO). Compared to PSO, the SPSO has a larger search space in the initial period and can obtain more precise results in the final period. It is worth mentioning here that although PSO has a relatively simple structure and runs very fast, this algorithm cannot guarantee the global convergence.

2.2. Quantum-Behaved Particle Swarm Optimization

QPSO is a new PSO algorithm and is inspired by the consideration of quantum mechanics with the PSO algorithms. It is superior to the traditional PSO algorithm not only in search ability but also in its accuracy. Particles of this model based on delta potential well can appear in any point of search space with a certain probability, and the QPSO algorithm can overcome the defect of the SPSO algorithm, that it cannot guarantee the global convergence with probability 1.

In the quantum space, a particle's velocity and position cannot be determined at the same time. Therefore, the state of a particle must be depicted by wave function $\psi(X,t)$. The physical meaning of $\psi(X,t)$ is that the probability density of a particle's appearance in a certain position can be determined using the formula $|\psi(X,t)|^2$, and then the probability distribution function can be obtained. As for the probability distribution function, through a Monte Carlo stochastic simulation method [23], the particle's position can be updated according to Equation (4).

$$X_{id} = p_{id} \pm {}^{L}\!/_{2} \ln({}^{1}\!/_{u}) \quad u \sim U(0,1) \quad (4)$$

where $U(0,1)$ means a random number uniformly distributed between 0 and 1. p_{id} is the local attractor and it can be defined as

$$p_{id} = \beta P_{id} + (1-\beta) P_{gd} \quad \beta \sim U(0,1) \quad (5)$$

where $P_i = (P_{i1}, P_{i2}, \cdots P_{id})$ is the best location of the *i*-th particle, $P_g = (P_{g1}, P_{g2}, \cdots P_{gd})$ is the best location of all the particles; and the parameter *L* can be evaluated by Equation (6).

$$L = 2\alpha \cdot |mbest_d - X_{id}| \quad (6)$$

where *mbest* is the average optimal position of all the particles [24], and it can be computed by Equation (7).

$$mbest = \frac{1}{M}\sum_{i=1}^{M} pbest_i = (\frac{1}{M}\sum_{i=1}^{M} pbest_{i,1}, \frac{1}{M}\sum_{i=1}^{M} pbest_{i,2}, ..., \frac{1}{M}\sum_{i=1}^{M} pbest_{i,d}) \quad (7)$$

Therefore,

$$X_{id} = p_{id} \pm \alpha |mbest_{id} - X_{id}| \times \ln(\frac{1}{u}) \quad u \sim U(0,1) \quad (8)$$

where α is a parameter of the QPSO algorithm called the contraction-expansion coefficient. In this paper, $\alpha = 0.5 + 0.5 \times (L_c - C_c) / L_c$, where L_c is the total number of iterations, and C_c is the current number of iterations.

2.3. Novel Computing Way of Local Attractor

It has been proved that QPSO can find the global optimum when the number of total iterations tends towards infinity [25]. The number of total iterations is usually a real number, such as one thousand or one million, and this number cannot be set as infinity during the practical application. So the global searching ability of QPSO is limited when the number of iterations is limited. To give QPSO better performance in global searching ability in this case, we must guarantee that the particles are diversiform at the early stage of iterations, and have good performance in local searching ability at the later stage of iterations. A novel way of computing the local attractor is proposed to achieve the above scheme, and its equation is shown as Equation (9).

$$p_{id} = \frac{L_c - C_c}{L_c} \beta P_{id} + \frac{C_c}{L_c}(1-\beta) P_{gd} \quad \beta \sim U(0,1) \tag{9}$$

We find that the coefficients of P_{id} and P_{gd} are different from Equation (5). In this way, we can let the experience of each particle (P_{id}) have more influence on particles when they update their next position at the beginning of iterations, and the experience of other particles (P_{gd}) has more influence on particles when they update their next position at the later stage of iterations. For simplicity, we define $\beta_1 = \frac{L_c - C_c}{L_c}$, $\beta_2 = \frac{C_c}{L_c}(1-\beta)$. Because $\beta \sim U(0,1)$, we can find $\beta_1 \sim U(0, \frac{L_c - C_c}{L_c})$, and $\beta_2 \sim U(0, \frac{C_c}{L_c})$. Then we will discuss the probability of $\beta_1 > \beta_2$.

(1) If $C_c < 0.5 L_c$, then $\frac{L_c - C_c}{L_c} > \frac{C_c}{L_c}$. The range of β_1, β_2 is shown in Figure 1 below.

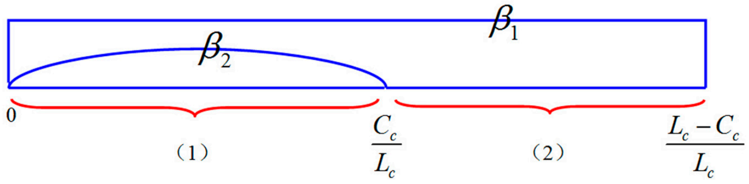

Figure 1. Range of β_1, β_2 when $\frac{L_c - C_c}{L_c} < \frac{C_c}{L_c}$.

There are two kinds of situations which can let $\beta_1 > \beta_2$: (a) β_1, β_2 both fall into area (1), at which point the probability of $\beta_1 > \beta_2$ is $\frac{C_c}{2(L_c - C_c)}$; (b) β_1 falls into area (2), at which point the probability of $\beta_1 > \beta_2$ is $\frac{L_c - 2C_c}{L_c - C_c}$. So the probability of $\beta_1 > \beta_2$ is $\frac{2L_c - 3C_c}{2(L_c - C_c)}$ when $C_c < 0.5 L_c$.

(2) If $C_c > 0.5 L_c$, then $\frac{L_c - C_c}{L_c} > \frac{C_c}{L_c}$. The range of β_1, β_2 is shown in Figure 2.

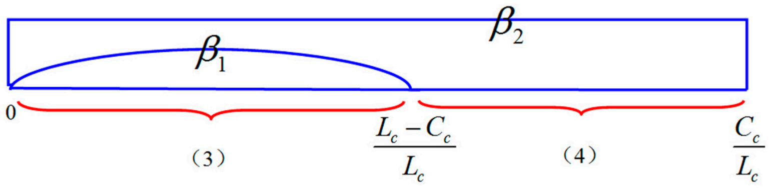

Figure 2. Range of β_1, β_2 when $\frac{L_c - C_c}{L_c} > \frac{C_c}{L_c}$.

If $\beta_1 > \beta_2$, then β_2 must fall into area (3), and the probability of $\beta_1 > \beta_2$ is $\frac{L_c - C_c}{2C_c}$ when $C_c > 0.5L_c$.

Above all, the probability of $\beta_1 > \beta_2$ is

$$p(\beta_1 > \beta_2) = \begin{cases} \frac{2L_c - 3C_c}{2(L_c - C_c)} & 0 \le C_c < 0.5L_c \\ \frac{L_c - C_c}{2C_c} & 0.5L_c \le C_c \le L_c \end{cases} \tag{10}$$

Let $R = \frac{C_c}{L_c}$ and Equation (10) can be rewritten as

$$p(\beta_1 > \beta_2) = \begin{cases} \frac{2 - 3R}{2(1 - R)} & 0 \le R < 0.5 \\ \frac{1 - R}{2R} & 0.5 \le R \le 1 \end{cases} \tag{11}$$

The probability of $\beta_1 > \beta_2$ with two different ways of computing the local attractor is shown in Figure 3, where the computing ways of method 1 and 2 are Equations (5) and (9), respectively.

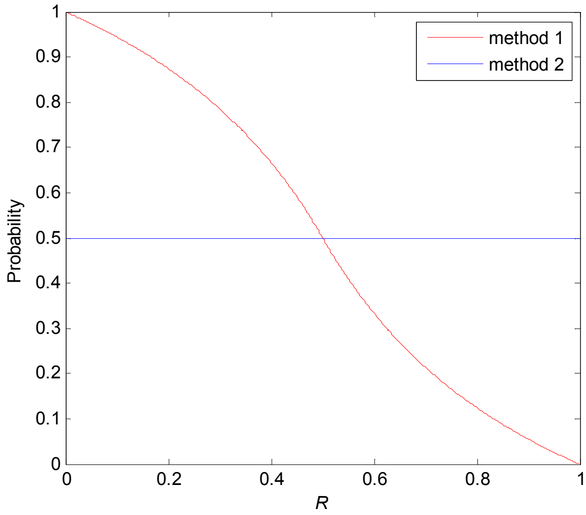

Figure 3. Probability of $\beta_1 > \beta_2$ with different computing ways of local attractor.

It can be found that P_{id} has more influence on particles when they update their next position at the beginning of iterations and P_{gd} has more influence on particles when they update their next position at the later stage of iterations in the proposed QPSO. Meanwhile, this influence of P_{id} and P_{gd} on particles of the traditional QPSO is random, and the random probability value is 0.5 during all iterations. For simplicity, we define the novel QPSO proposed in this paper as the enhanced QPSO (EQPSO), and the traditional QPSO as the standard QPSO (SQPSO).

Finally, the EQPSO algorithm can be described as the following procedure:

Step 1: Initialize the particles X and let $X_{id} = P_{id}$;
Step 2: Update P_{id} according to the fitness function and P_{gd} among each particles' best positions;
Step 3: Compute $mbest$ according to Equation (7);
Step 4: Compute p_{id} according to Equation (9);
Step 5: Compute the new position X_{id} according to Equation (8);
Step 6: Repeat steps 2–5 until the algorithm satisfies the end condition.

3. Results and Discussion

3.1. Comparison of Different PSO Algorithms

In this section, we compare results of test functions found by EQPSO with six different PSO methods. Sphere Model, Generalized Rastrigin, Griewank, Ackley, Alpine, Schwefel's Problem and Generalized Rosenbrock were used as the test functions. The detailed information of these seven functions is shown in Table 1. Sphere Model is a nonlinear symmetric unimodal function, and its different dimensions are separable. Most algorithms can easily find the global optimum of Sphere Model, and it can be used for testing the optimization precision of EQPSO. Generalized Rastrigin, Griewank, Ackley, Alpine, Schwefel's Problem and Generalized Rosenbrock are complex and have many local minimums, so they are employed to test the global searching ability of EQPSO in this paper.

Table 1. Standard test functions.

Functions	Mathematical Expression	Global Minimum
Sphere Model	$f_1(x)=\sum_{i=1}^{n} x_i^2$, where $-100 \le x_i \le 100$	$\min(f_1)=f_1(0,0,\cdots,0)=0$
Generalized Rastrigin	$f_2(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$, where $-5.12 \le x_i \le 5.12$	$\min(f_2)=f_2(0,0,\cdots,0)=0$
Griewank	$f_3(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$, where $-500 \le x_i \le 500$	$\min(f_3)=f_3(0,0,\cdots,0)=0$
Ackley	$f_4(x)=-20\exp(-0.2\times\sqrt{\frac{1}{30}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{30}\sum_{i=1}^{n}\cos 2\pi x_i)+20+e$, where $-32 \le x_i \le 32$	$\min(f_4)=f_4(0,0,\cdots,0)=0$
Alpine	$f_5(x)=\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$, where $-10 \le x_i \le 10$	$\min(f_5)=f_5(0,0,\cdots,0)=0$
Schwefel's Problem	$f_6(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$, where $-10 \le x_i \le 10$	$\min(f_6)=f_6(0,0,\cdots,0)=0$
Generalized Rosenbrock	$f_7(x)=\sum_{i=1}^{n}[100(x_i+1-x_i^2)^2+(x_i-1)^2]$, where $-30 \le x_i \le 30$	$\min(f_7)=f_7(0,0,\cdots,0)=0$

Besides EQPSO, PSO, SPSO, and SQPSO, the enhanced QPSO proposed in paper [6,7,11] (for simplicity, these three methods are denoted as M1, M2 and M3) are also used to optimize the seven test functions.

PSO uses Equation (1) to update particles' speed, while SPSO uses Equation (3). SQPSO employs Equation (5) to compute the local attractor, while EQPSO employs Equation (9). PSO, SPSO, SQPSO and EQPSO all initialize their particle swarm by random numbers uniformly distributed, and other sets of these four methods are as introduced in Section 2. The set of M1, M2 and M3 was the same as [6,7,11]. The dimension of the all seven test functions was 30, and each program was repeated 10 times. All programs were run on MATLAB R2009a which was installed at a computer with a Windows 7 operating system and 4 Intel (R) Core (TM) i5-3470 CPUs @ 3.2GHz. The minimum value and its mean value found by each optimization algorithm during the 10 running times were used to evaluate its performance. All results are shown in Tables 2–8.

Table 2. Optimized results of the Sphere Model function found by different PSO algorithms.

Particles	**20**			**40**			**80**		
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
PSO	1.6441	0.8441	0.4893	0.9618	0.2800	0.1398	0.2838	0.0988	0.0363
	0.8564	0.4999	0.2937	0.5715	0.1682	0.0759	0.2083	0.0531	0.0184
SPSO	0.7261	0.3672	0.2391	0.1312	0.0628	0.0220	0.0109	0.0014	3.9598×10^{-4}
	0.4624	0.1436	0.1009	0.0561	0.0258	0.0092	0.0039	6.3394×10^{-4}	4.1215×10^{-5}
SQPSO	2.5633	1.0165	0.3628	1.3290	0.3267	0.0691	0.6697	0.0688	0.0066
	1.7973	0.5651	0.1783	0.6007	0.1829	0.0308	0.4505	0.0271	0.0023
M1	1.2343×10^{-7}	6.4803×10^{-9}	4.5401×10^{-11}	6.1377×10^{-8}	1.6919×10^{-9}	3.0023×10^{-11}	1.3427×10^{-7}	7.0411×10^{-9}	2.1708×10^{-11}
	1.3762×10^{-8}	1.7252×10^{-10}	2.5644×10^{-12}	5.9569×10^{-9}	5.6898×10^{-11}	5.1446×10^{-12}	9.7467×10^{-9}	1.5518×10^{-10}	2.8007×10^{-12}
M2	1.5605×10^{-7}	2.5670×10^{-9}	9.4584×10^{-11}	4.9876×10^{-8}	1.4890×10^{-9}	5.5934×10^{-11}	8.7657×10^{-8}	1.4832×10^{-9}	2.6456×10^{-11}
	2.7035×10^{-8}	4.1998×10^{-10}	4.0927×10^{-12}	1.6057×10^{-8}	8.3158×10^{-11}	1.0805×10^{-11}	2.8367×10^{-8}	3.5118×10^{-10}	2.9924×10^{-12}
M3	1.1710×10^{-3}	2.7697×10^{-4}	9.6107×10^{-8}	6.2409×10^{-4}	1.7757×10^{-6}	7.3520×10^{-8}	9.9449×10^{-5}	2.4029×10^{-5}	2.2679×10^{-7}
	2.7035×10^{-5}	6.1339×10^{-7}	1.2988×10^{-8}	3.0888×10^{-5}	1.7354×10^{-7}	1.1750×10^{-8}	1.9037×10^{-5}	2.5322×10^{-7}	5.7069×10^{-9}
EQPSO	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Note: there are two numbers in each cell. The second number is the minimum value found by each optimization method during the 10 running times, and the first number is the mean value of the 10 minimum value.

Table 3. Optimized results of the Generalized Rastrigin function found by different PSO algorithms.

Particles	**20**			**40**			**80**		
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
PSO	12.6982	7.5865	5.5879	6.0211	5.5924	5.4869	7.9318	6.9536	6.7100
	4.3357	3.7038	3.5364	2.2309	1.9919	1.9899	4.1269	3.0257	3.0257
SPSO	8.0665	5.8705	4.9350	4.7761	4.3778	4.3315	6.8265	5.5614	5.3427
	3.9084	3.2631	1.2684	1.9899	1.9899	1.9899	2.4574	2.3509	0.0199
SQPSO	5.5515	4.0321	3.1779	4.3447	3.2834	3.2834	5.7808	5.0256	4.3040
	3.5126	2.102	1.0388	0.9950	0.9950	0.9950	2.1503	1.9929	2.0035

Table 3. *Cont.*

Particles		**20**			**40**			**80**	
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
M1	0.0609	2.7337×10^{-4}	9.0049×10^{-5}	5.4290×10^{-6}	2.6125×10^{-5}	5.4374×10^{-8}	3.2645×10^{-5}	1.4798×10^{-7}	1.0201×10^{-7}
	2.0564×10^{-8}	4.7390×10^{-11}	8.0593×10^{-12}	1.5582×10^{-8}	2.9045×10^{-10}	3.4266×10^{-12}	2.8414×10^{-8}	2.0407×10^{-10}	2.7871×10^{-12}
M2	0.1494	6.7643×10^{-4}	8.1401×10^{-5}	1.6174×10^{-4}	4.4490×10^{-6}	5.3705×10^{-8}	4.8611×10^{-5}	4.3624×10^{-7}	1.6225×10^{-7}
	3.9666×10^{-7}	1.4207×10^{-10}	4.3480×10^{-11}	5.2413×10^{-8}	1.4959×10^{-9}	5.5120×10^{-12}	8.0935×10^{-8}	1.4349×10^{-9}	5.8833×10^{-12}
M3	0.5327	8.0681×10^{-1}	1.1260×10^{-1}	3.8695×10^{-2}	4.9710×10^{-2}	7.2574×10^{-2}	1.0967×10^{-2}	2.0896×10^{-2}	5.7961×10^{-2}
	8.9381×10^{-4}	1.6325×10^{-6}	9.2113×10^{-8}	1.9217×10^{-4}	8.5606×10^{-7}	1.3543×10^{-8}	8.8096×10^{-5}	3.2663×10^{-7}	7.5069×10^{-9}
EQPSO	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Table 4. Optimized results of the Griewank function found by different PSO algorithms.

Particles		**20**			**40**			**80**	
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
PSO	0.1002	0.0511	0.0332	0.0583	0.0237	0.0137	0.0249	0.0093	0.0042
	0.0732	0.0272	0.0206	0.0400	0.0133	0.0112	0.0148	0.0061	0.0024
SPSO	0.0458	0.0272	0.0158	0.0107	0.0038	0.0029	0.0016	3.4054×10^{-4}	8.3607×10^{-5}
	0.0215	0.0128	0.0072	0.0054	0.0013	0.0012	4.4367×10^{-4}	1.2080×10^{-4}	3.1961×10^{-5}
SQPSO	0.1317	0.0433	0.0205	0.0667	0.0151	0.0052	0.0306	0.0062	8.7017×10^{-4}
	0.0850	0.0275	0.0081	0.0367	0.0087	0.0018	0.0188	0.0012	1.3873×10^{-4}
M1	4.3802×10^{-9}	1.3854×10^{-10}	2.4538×10^{-12}	3.9419×10^{-9}	1.4175×10^{-9}	4.8735×10^{-11}	8.4938×10^{-8}	2.7257×10^{-10}	2.0276×10^{-12}
	8.1498×10^{-10}	6.5957×10^{-12}	9.5704×10^{-14}	1.6868×10^{-10}	1.1181×10^{-11}	3.2141×10^{-13}	5.2482×10^{-10}	1.5978×10^{-11}	3.4417×10^{-13}
M2	6.8394×10^{-9}	1.8461×10^{-10}	1.2017×10^{-11}	1.1083×10^{-8}	2.2452×10^{-8}	1.0303×10^{-11}	1.1995×10^{-7}	3.1639×10^{-10}	6.6436×10^{-12}
	4.3524×10^{-10}	1.4072×10^{-11}	1.5965×10^{-13}	4.8845×10^{-10}	2.0792×10^{-11}	4.9039×10^{-13}	5.4717×10^{-10}	1.7831×10^{-11}	7.6550×10^{-13}
M3	1.4325×10^{-7}	2.1973×10^{-10}	5.9936×10^{-11}	1.9127×10^{-7}	5.8875×10^{-10}	7.6208×10^{-11}	6.6305×10^{-8}	6.9551×10^{-9}	1.0071×10^{-10}
	1.1876×10^{-9}	2.0809×10^{-11}	9.8377×10^{-13}	8.1657×10^{-10}	6.5016×10^{-11}	1.8130×10^{-13}	2.7275×10^{-9}	5.3252×10^{-11}	3.2663×10^{-12}
EQPSO	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Table 5. Optimized results of the Ackley function found by different PSO algorithms.

Particles		**20**			**40**			**80**	
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
PSO	2.5623	2.6098	1.9049	2.0805	1.6090	1.3993	1.6511	0.9438	1.0636
	2.1335	1.7159	1.5390	1.5095	1.0413	0.8074	0.9987	0.2065	0.3235
SPSO	2.1057	1.9863	1.6675	1.6441	1.3633	1.3631	1.1669	1.4441	0.7958
	1.4326	1.324	1.0799	0.7471	0.2686	0.6554	0.0826	0.0426	0.0212
SQPSO	2.8525	2.0558	1.4556	2.3059	1.6019	0.4848	1.9968	0.4809	0.1158
	2.2745	1.5366	0.9074	1.6426	0.5011	1.1067	1.1180	0.2082	0.0244
M1	2.0271×10^{-4}	4.2164×10^{-5}	5.8044×10^{-6}	1.4847×10^{-4}	3.5473×10^{-5}	2.9210×10^{-6}	1.4986×10^{-4}	2.1964×10^{-5}	1.0458×10^{-8}
	7.3464×10^{-5}	1.0555×10^{-5}	9.4110×10^{-7}	7.7522×10^{-5}	7.9518×10^{-6}	1.5560×10^{-6}	6.3927×10^{-5}	8.4422×10^{-6}	4.0880×10^{-9}
M2	2.3179×10^{-4}	3.8744×10^{-5}	8.6453×10^{-6}	1.8161×10^{-4}	3.2453×10^{-5}	4.3471×10^{-6}	1.4951×10^{-4}	2.8008×10^{-5}	1.4904×10^{-8}
	9.3839×10^{-5}	2.0169×10^{-5}	1.0569×10^{-6}	9.1389×10^{-5}	1.3333×10^{-5}	1.5869×10^{-6}	7.6068×10^{-5}	9.6436×10^{-6}	4.8599×10^{-9}

Table 5. *Cont.*

Particles	**20**			**40**			**80**		
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
M3	1.8620×10^{-1}	4.7290×10^{-2}	4.3239×10^{-4}	1.8255×10^{-1}	3.1622×10^{-2}	4.7492×10^{-3}	1.6163×10^{-1}	2.6017×10^{-2}	5.5176×10^{-5}
	1.0703×10^{-1}	2.0169×10^{-2}	1.5221×10^{-4}	1.0083×10^{-1}	1.1978×10^{-2}	1.1828×10^{-3}	1.0100×10^{-1}	9.3215×10^{-3}	6.1040×10^{-5}
EQPSO	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}
	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}	3.5527×10^{-15}

Table 6. Optimized results of the Alpine function found by different PSO algorithms.

Particles	**20**			**40**			**80**		
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
PSO	0.0013	7.0809×10^{-4}	6.7859×10^{-4}	5.4561×10^{-4}	2.1766×10^{-4}	2.9181×10^{-4}	2.9190×10^{-4}	2.2766×10^{-4}	1.1943×10^{-4}
	1.4526×10^{-4}	3.6689×10^{-5}	4.1992×10^{-5}	1.7372×10^{-4}	5.5540×10^{-5}	3.8497×10^{-5}	2.6377×10^{-5}	5.1664×10^{-5}	4.5208×10^{-5}
SPSO	0.0022	7.3980×10^{-4}	8.9023×10^{-4}	9.6133×10^{-4}	5.4579×10^{-4}	4.8548×10^{-4}	5.6289×10^{-4}	2.1799×10^{-4}	1.7122×10^{-4}
	9.8492×10^{-5}	5.4569×10^{-5}	1.1949×10^{-5}	5.3114×10^{-6}	3.3547×10^{-5}	7.9522×10^{-6}	6.4584×10^{-5}	1.0359×10^{-5}	3.8545×10^{-6}
SQPSO	0.0143	0.0063	0.0061	0.0036	0.0031	0.0019	0.0031	0.0017	0.0010
	1.6962×10^{-4}	3.3031×10^{-5}	2.3869×10^{-4}	1.1016×10^{-4}	6.2983×10^{-5}	2.6630×10^{-4}	1.4311×10^{-4}	2.7915×10^{-5}	2.5984×10^{-5}
M1	1.8608×10^{-7}	1.3972×10^{-8}	2.6450×10^{-11}	7.1641×10^{-8}	6.6746×10^{-10}	3.5401×10^{-8}	7.7803×10^{-9}	2.3994×10^{-10}	1.0419×10^{-11}
	2.8334×10^{-11}	2.2204×10^{-16}	5.5511×10^{-17}	2.3989×10^{-13}	1.0230×10^{-14}	1.7764×10^{-14}	1.0359×10^{-12}	1.990×10^{-14}	4.4409×10^{-16}
M2	1.2585×10^{-6}	1.0236×10^{-7}	4.5142×10^{-13}	4.8673×10^{-5}	3.8113×10^{-8}	2.0704×10^{-7}	2.1529×10^{-7}	1.9118×10^{-7}	3.2466×10^{-8}
	2.3168×10^{-10}	1.1102×10^{-15}	2.2206×10^{-16}	3.0183×10^{-10}	1.3156×10^{-13}	3.8825×10^{-16}	5.4900×10^{-10}	6.2728×10^{-14}	9.8810×10^{-15}
M3	3.0681×10^{-2}	8.3576×10^{-5}	4.2667×10^{-9}	7.5882×10^{-5}	6.2528×10^{-6}	4.3311×10^{-7}	7.0333×10^{-3}	1.7644×10^{-6}	3.2466×10^{-7}
	1.2865×10^{-8}	4.0240×10^{-10}	1.1102×10^{-13}	4.1666×10^{-6}	6.0830×10^{-8}	1.7875×10^{-11}	6.7802×10^{-8}	2.4242×10^{-10}	6.0830×10^{-8}
EQPSO	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Table 7. Optimized results of the Schwefel's Problem function found by different PSO algorithms.

Particles	**20**			**40**			**80**		
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
PSO	5.6638×10^{-4}	1.5892×10^{-4}	2.2988×10^{-5}	4.0478×10^{-4}	4.9989×10^{-6}	4.3501×10^{-5}	3.8480×10^{-5}	2.0497×10^{-7}	1.1667×10^{-6}
	6.8503×10^{-5}	1.1801×10^{-5}	2.5200×10^{-8}	6.9970×10^{-6}	7.8797×10^{-9}	5.9474×10^{-7}	5.7091×10^{-7}	1.5086×10^{-8}	1.8273×10^{-7}
SPSO	1.9251×10^{-5}	2.0635×10^{-5}	1.3080×10^{-5}	1.5901×10^{-5}	2.9800×10^{-6}	3.8669×10^{-6}	2.4663×10^{-6}	2.1393×10^{-6}	1.6282×10^{-6}
	1.4706×10^{-6}	5.0909×10^{-6}	2.1169×10^{-6}	1.9596×10^{-6}	1.2431×10^{-7}	2.111×10^{-7}	1.6972×10^{-8}	2.7569×10^{-8}	4.8958×10^{-9}
SQPSO	4.0874×10^{-4}	2.1921×10^{-4}	1.2638×10^{-4}	2.0573×10^{-4}	6.1381×10^{-5}	4.1455×10^{-5}	6.9722×10^{-5}	6.2645×10^{-5}	3.4666×10^{-5}
	4.7434×10^{-5}	5.4540×10^{-6}	2.5055×10^{-6}	4.3894×10^{-5}	6.4015×10^{-6}	2.1093×10^{-6}	1.2494×10^{-5}	9.2323×10^{-6}	2.8307×10^{-6}
M1	5.6981×10^{-12}	2.5339×10^{-18}	9.3601×10^{-22}	8.8922×10^{-14}	3.0140×10^{-19}	9.8431×10^{-21}	9.1446×10^{-16}	6.6840×10^{-20}	7.3108×10^{-34}
	1.9443×10^{-15}	4.4085×10^{-30}	1.3082×10^{-43}	1.0195×10^{-15}	9.0262×10^{-33}	1.5374×10^{-48}	8.5965×10^{-18}	1.3921×10^{-34}	6.6827×10^{-45}
M2	8.4145×10^{-11}	2.6179×10^{-18}	3.4991×10^{-20}	2.0636×10^{-13}	8.7243×10^{-19}	5.1265×10^{-20}	2.6896×10^{-15}	1.4726×10^{-20}	3.6702×10^{-20}
	9.4579×10^{-15}	5.2842×10^{-29}	3.4261×10^{-40}	1.7643×10^{-15}	1.2441×10^{-30}	8.1405×10^{-47}	3.9470×10^{-17}	4.1385×10^{-32}	1.1838×10^{-44}
M3	9.9860×10^{-9}	7.4953×10^{-16}	3.5571×10^{-16}	9.3567×10^{-10}	2.6529×10^{-16}	7.2096×10^{-17}	3.0135×10^{-12}	1.3079×10^{-15}	3.9001×10^{-17}
	6.5071×10^{-11}	9.0021×10^{-24}	3.4261×10^{-37}	4.2655×10^{-13}	3.6111×10^{-27}	7.2354×10^{-41}	2.4677×10^{-13}	7.5319×10^{-29}	4.0303×10^{-41}
EQPSO	2.5546×10^{-19}	3.2685×10^{-19}	3.9020×10^{-20}	5.3782×10^{-19}	2.5603×10^{-20}	1.2063×10^{-20}	8.3867×10^{-34}	4.3086×10^{-49}	2.1549×10^{-51}
	3.5873×10^{-31}	2.3113×10^{-56}	4.5433×10^{-92}	4.7913×10^{-19}	2.4727×10^{-60}	6.4142×10^{-90}	2.1460×10^{-34}	1.4186×10^{-63}	2.1766×10^{-94}

Table 8. Optimized results of the Generalized Rosenbrock function found by different PSO algorithms.

Particles	20			40			80		
Iterations	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**	**1000**	**2000**	**3000**
PSO	14.0820	6.9217	5.9551	6.0812	5.1278	4.2665	5.1579	5.0835	3.0416
	7.7815	4.8018	1.7819	1.3719	3.5548	0.7336	3.3651	1.6381	0.0486
SPSO	7.4839	7.3913	7.3204	7.4344	7.4063	7.3672	7.6472	7.3979	7.5233
	7.1661	7.0657	6.9552	6.6722	7.1152	6.9455	6.7529	6.4032	6.4121
SQPSO	4.1649	5.7284	4.5535	5.7841	7.8350	2.3123	3.1037	2.1716	1.7536
	1.1123	0.9931	0.3690	0.5311	0.8699	0.0745	0.0491	0.1512	0.0467
M1	3.7539	3.5003	2.6244	2.1098	0.4345	0.0150	1.9711×10^{-5}	7.3442×10^{-8}	1.1838×10^{-8}
	0.1306	0.1975	0.3908	1.5283×10^{-4}	4.0287×10^{-8}	1.8431×10^{-12}	7.9672×10^{-25}	3.1923×10^{-27}	3.1702×10^{-29}
M2	3.9761	4.0357	3.4834	1.0713	0.5714	0.2179	9.3065×10^{-8}	4.0549×10^{-10}	1.5138×10^{-11}
	0.4548	0.5157	1.3480	0.0027	1.3731×10^{-7}	2.1565×10^{-11}	1.1516×10^{-24}	2.3693×10^{-28}	2.3247×10^{-29}
M3	5.1747	4.1321	5.3282	1.3111	0.6469	0.0266	1.0219×10^{-7}	9.7145×10^{-6}	1.3633×10^{-11}
	0.3702	0.6407	0.5069	0.0089	3.6672×10^{-3}	1.8315×10^{-7}	2.4738×10^{-20}	2.5591×10^{-25}	1.9752×10^{-23}
EQPSO	3.6340	3.2169	2.9860	0.7790	0.1639	0.0150	1.9013×10^{-8}	9.1212×10^{-31}	0
	0.0199	0.0088	0.5001	3.8339×10^{-5}	2.1950×10^{-8}	1.8431×10^{-12}	7.1663×10^{-29}	0	0

It can be found from Tables 2–8 that the best results are acquired by EQPSO, which proves the global searching ability of EQPSO has been greatly improved with the help of the novel way of computing the local attractor. EQPSO can find the global minimum of Sphere Model, Generalized Rastrigin, Griewank, Alpine and Generalized Rosenbrock, and the result found by EQPSO is better than other six algorithms although they all do not find the global optimum of Ackley and Schwefel's Problem. We can also find that the EQPSO of either 20 or 40 particles obtains the best result, which illustrates that the EQPSO can acquire a good result under the condition that the population is small.

The convergence speeds of different optimization algorithms are shown in Figures 4–7 (the swarm size was 80 and the number of iterations was 3000) when used to optimize the Sphere Model, Generalized Rastrigin, Griewank and Ackley functions. It can be found that the convergence speed of EQPSO is faster than other considered PSO methods.

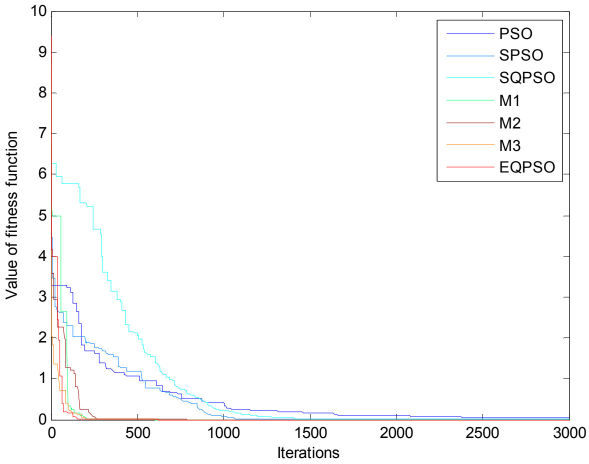

Figure 4. Convergence speed of different PSO algorithms when used to optimize the Sphere Model function.

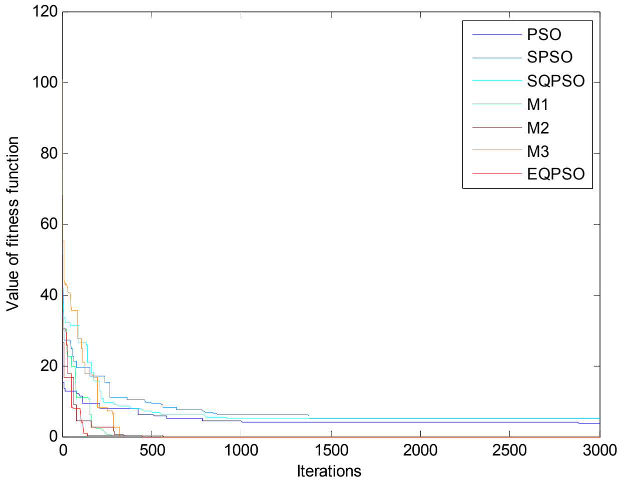

Figure 5. Convergence speed of different PSO algorithms when used to optimize the Generalized Rastrigin function.

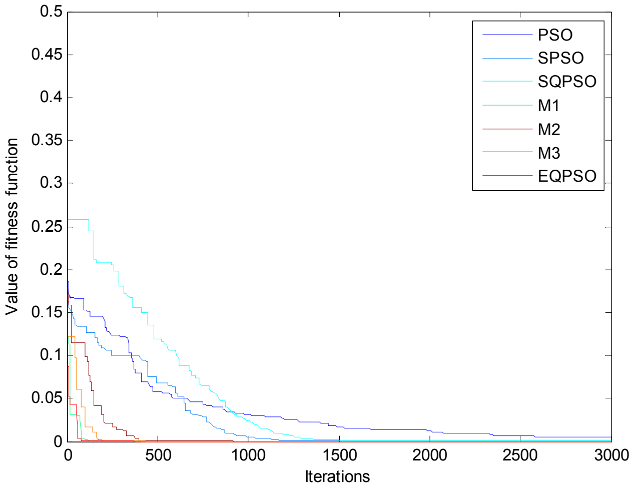

Figure 6. Convergence speed of different PSO algorithms when used to optimize the Griewank function.

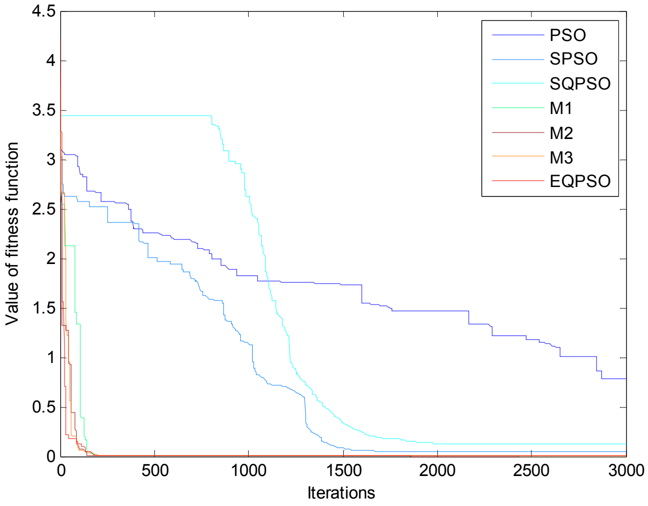

Figure 7. Convergence speed of different PSO algorithms when used to optimize the Ackley function.

EQPSO and the other considered PSO methods were also used to optimize the problems with constraints, and three functions with constraints of paper [26] (g07, g09 and g10, which are shown in Table 9) are used as the test functions.

Table 9. Test functions with constraints.

Functio	Mathematical Expression	Global Minimum
g07	$f_8(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 +$ $(x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$ subject to: $g_1(x) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0$ $g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$ $g_3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$ $g_4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0$ $g_5(x) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$ $g_6(x) = x_1^2 + x(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0$ $g_7(x) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0$ $g_8(x) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$ where $-10 \leq x_i \leq 10\ (i = 1,2,\cdots,10)$	$\min(f_8) = f_8(2.171996,$ $2.363683, 8.773926,$ $5.095984, 0.9906548,$ $1.430574, 1.321644,$ $9.828726, 8.280092,$ $8.375927) = 24.3062091$
g09	$f_9(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 +$ $x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$ subject to: $g_1(x) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0$ $g_2(x) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$ $g_3(x) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$ $g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$ where $-10 \leq x_i \leq 10\ (i = 1,2,\cdots,7)$	$\min(f_9) = f_9(2.330499,$ $1.951372, -0.4775414,$ $4.365726, -0.6244870,$ $1.038131, 1.594227)$ $= 680.6300573$
g10	$f_{10}(x) = x_1 + x_2 + x_3$ suject to: $g_1(x) = -1 + 0.0025(x_4 + x_6) \leq 0$ $g_2(x) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$ $g_3(x) = -1 + 0.01(x_8 - x_5) \leq 0$ $g_4(x) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0$ $g_5(x) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$ $g_6(x) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$ where $100 \leq x_1 \leq 10000, 1000 \leq x_i \leq 10000\ (i = 2,3),$ $10 \leq x_i \leq 1000\ (i = 4,5,\cdots,8)$	$\min(f_{10}) = f_{10}(579.19,$ $1360.13, 5109.92,$ $182.0174, 295.5985,$ $217.9799, 286.40,$ $395.5979) = 7049.25$

The mechanism proposed in paper [26] was adopted to help the considered PSO methods of this paper to solve the functions with constraints. This mechanism is used to select the leaders, and it is based both on the feasible solutions and on the fitness value of a particle. In this mechanism, when two feasible particles are compared, the particle which has the highest fitness value wins. If one particle is not feasible and the other one is feasible, then the feasible particle wins. If both particles are infeasible, the particle that has the lowest fitness value wins. The idea is to choose as a leader the particle that, even when infeasible, lies closer to the feasible region. More detailed information about this mechanism can be found in paper [18]. When the seven considered methods were used to optimize these three functions, the size of their particle swarm was 80, and the maximum number of iterations was 3000. The results are shown in Table 10.

Table 10. Optimized results of g07, g09 and g10 found by different PSO algorithms.

	g07	**g09**	**g10**
PSO	26.8630	685.2511	7512.6585
	25.4369	684.5264	7058.5644
SPSO	26.3571	685.2355	7489.2385
	25.3321	682.8701	7056.6452
SQPSO	26.9852	685.7819	7498.3160
	25.8752	684.2511	7053.8519
M1	25.0162	684.6238	7369.3470
	24.6835	681.0064	7056.0559
M2	25.1302	685.6650	7396.5825
	24.6973	681.4274	7057.8687
M3	25.5647	686.6652	7450.6183
	25.1960	682.2697	7059.7984
EQPSO	24.4080	681.5307	7145.6589
	24.3090	680.6331	7051.0049

We can find from Table 10 that the results found by EQPSO are better than other PSO methods for functions g07, g09 and g10.

3.2. Comparison of Different Optimization Algorithms

In this section, we compare EQPSO with CMA-ES, GA, KH, an enhanced KH of paper [21], MBO, HS and APOA. All these methods were used to optimize the test functions of Table 1. For GA, KH, an enhanced KH of paper [21] (denoted as KH-E), MBO, HS, APOA and EQPSO, the population size is 20 and the maximal value of iterations is 1000. Other parameters of each algorithm were as follows: CMA-ES: SIGMA is a parameter which can determine the initial coordinate wise standard deviations for the search, and we set SIGMA to one third of the initial search region. GA: the selecting function was stochastic uniform, the crossover function was Intermediate, and the mutation function was Gaussian. The crossover fraction was 0.8. KH-E: The setting of KH-E was the same as paper [21]. MBO: The BAR value was equal to the percentage of population for MBO, and we randomly divided the whole population into population1 and population2. HS: Harmony memory considering rate was 0.95 and pitch adjustment rate was 0.3. APOA: the value of phototropism operator was 0.1. All programs were run ten times, the best result and mean result were used to evaluate the eight methods, and the results are shown in Table 11.

Table 11. Optimization results of different optimization methods.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
CMA-ES	1.3485×10^{-15}	9.8501	0.0017	0.1155	2.6214	3.3525×10^{-10}	2.5101×10^{-15}
	7.4881×10^{-16}	3.9798	1.8874×10^{-15}	3.2028×10^{-11}	2.1760×10^{-14}	3.4148×10^{-14}	1.0652×10^{-15}
GA	4.3537	12.2978	0.1908	3.2453	4.5903	0.1696	4.3999
	2.1249	7.6093	0.1368	2.9185	3.3934	3.2352×10^{-5}	0.5610
KH	2.5923	1.5810	20.5440	2.9902	1.9149	2.4876	3.6905
	0.0652	0.0242	20.4087	1.1703	8.7010×10^{-5}	1.6511	1.6508
KH−E	0.0041	0.1829	2.4236	0.0027	1.2341×10^{-2}	0.0027	0.1204
	8.5871×10^{-6}	4.5122×10^{-4}	2.2548	1.1790×10^{-4}	0.0410×10^{-10}	1.5855×10^{-4}	8.6122×10^{-4}
MBO	1.5505×10^{2}	197.2550	1.7072×10^{2}	12.9117	0.0441	8.2794×10^{5}	4.7425×10^{4}
	0.4331	1.1103	12.4189	0.0013	5.5036×10^{-5}	0.0034	46.3762
APOA	1.8645×10^{1}	8.3188	6.6463×10^{-1}	3.5394	4.5291×10^{-2}	2.7639×10^{-1}	1.1939×10^{1}
	1.4024×10^{1}	7.8942	4.6362×10^{-1}	3.5140	3.3480×10^{-3}	4.2181×10^{-3}	1.0967×10^{1}
HS	4.7139×10^{-2}	3.7631	1.1711×10^{1}	2.1087×10^{1}	1.3010×10^{-1}	2.3984×10^{2}	5.4177×10^{1}
	4.4334×10^{-2}	1.8259	1.0330×10^{1}	2.0906×10^{1}	1.7142×10^{-2}	2.0715	5.0345×10^{1}
EQPSO	0	0	0	3.5527×10^{-15}	0	2.5546×10^{-19}	3.6340
	0	0	0	3.5527×10^{-15}	0	3.5873×10^{-31}	0.0199

As we can see from Table 11, for function f_7, the results found by CMA-ES are better than EQPSO, and other than that the results of EQPSO are better than other methods.

3.3. Statistical Test

Finally, the sign test, which is known as a statistical test method, was adopted to verify the significance of the results found by EQPSO. In this method, the overall performance of algorithms is based on number of cases on which an algorithm is the overall winner. A detailed introduction to the sign test can be seen in paper [27]. Table 12 shows the critical number of wins needed to achieve both $\alpha = 0.05$ and $\alpha = 0.1$ levels of significance. An algorithm is significantly better than another if it performs better on at least the cases presented in each row of Table 12.

Table 12. Critical values for sign test.

Cases	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
$\alpha = 0.05$	5	6	7	7	8	9	9	10
$\alpha = 0.1$	5	6	6	7	7	8	9	9

The minimum value of 10 test functions (including all functions of Tables 1 and 9) found by PSO, SPSO, SQPSO, M1, M2, M3 and EQPSO (the size of particle swarm was 80 and the maximum number of iterations was 3000) were analyzed by sign test, and the results are shown in Table 13. The minimum value of 7 test functions (all functions of Table 1) found by CMA-ES, GA, KH, KH-E, MBO, APOA, HS and EQPSO were also analyzed by sign test, and the results are shown in Table 14.

Table 13. Results of sign test performed on EQPSO (different PSO algorithms).

EQPSO	PSO	SPSO	SQPSO	M1	M2	M3
wins	10	10	10	10	10	10
loses	0	0	0	0	0	0
Detected differences	$\alpha = 0.05$					

Table 14. Results of sign test performed on EQPSO (different optimization algorithms).

EQPSO	CMA-ES	GA	KH	KH-E	MBO	APOA	HS
wins	6	7	7	7	7	7	7
loses	1	0	0	0	0	0	0
Detected differences	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.05$	$\alpha = 0.05$	$\alpha = 0.05$	$\alpha = 0.05$	$\alpha = 0.05$

As we can see, in the $\alpha = 0.05$ level EQPSO shows a significant improvement over the other six PSO methods, GA, KH, KH-E, MBO, APOA and HS. For CMA-ES, EQPSO shows a significant improvement in the $\alpha = 0.1$ level.

4. Conclusions

The QPSO algorithm shows ideal performance in the aspects of search ability, convergence speed, solution accuracy and solving robustness. However, it cannot be guaranteed that the traditional QPSO can find the global optimum when the number of iterations is limited.

We propose a novel way of computing the local attractor of QPSO to improve the global searching ability of QPSO in this case. With this proposed computing way, we can guarantee the particles are diversiform at the early stage of iterations, and ensure good performance in local searching ability at the later stage of iterations. Results of the test functions have proved that the proposed method has better performance in global searching ability.

In this paper, we improve the optimization performance of EQPSO through controlling the iterative process, as one can find that the initial distribution of particles also influences the optimization ability of EQPSO. We will focus our research on finding an effective way to initialize the particles of EQPSO in our future studies, and we believe the optimization performance will be more ideal through this research.

Acknowledgments

This work is supported by Program for New Century Excellent Talents in University (No. (2013) 47), National Natural Science Foundation of China (Nos. 61372139, 61101233, 60972155), Fundamental Research Funds for the Central Universities (No. XDJK2015C073), Science and Technology personnel training program Fund of Chongqing (No. Cstc2013kjrc-qnrc40011) and Fundamental Research Funds for the Central Universities (No. SWU115009).

Author Contributions

Pengfei Jia wrote this paper; Shukai Duan provided the analysis results on the performance test of the proposed QPSO, and prepared most of the figures for the paper; Jia Yan was involved in all research activities, provided many valuable experimental results and offered valuable suggestions. All authors have discussed about and contributed to the manuscript. All authors have read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Van den Bergh, F.; Engelbrecht, A.P. A new locally convergent particle swarm optimizer. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia, 6–9 October 2002; pp. 96–101.
2. Van den Bergh, F. An Analysis of Particle Swarm Optimizers. Ph.D. Thesis, University of Pretoria, Pretoria, South Africa, 2001.
3. Coelho, L.D.S. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst. Appl.* **2010**, *37*, 1676–1683.
4. Ch, S.; Anand, N.; Panigrahi, B.K.; Mathur, S. Streamflow forecasting by SVM with quantum behaved particle swarm optimization. *Neurocomputing* **2012**, *4*, 18–23.
5. Liu, H.; Yang, G.; Song, G. MIMO radar array synthesis using QPSO with normal distributed contraction-expansion factor. *Procedia Eng.* **2011**, *15*, 2449–2453.
6. Sun, J.; Chen, W.; Fang, W.; Wun, X.; Xu, W. Gene expression data analysis with the clustering method based on an improved quantum-behaved Particle Swarm Optimization. *Eng. Appl. Artif. Intell.* **2012**, *25*, 376–391.
7. Mariani, V.C.; Duck, A.R.K.; Guerra, F.A.; Coelho, L.D.S.; Rao, R.V. A chaotic quantum-behaved particle swarm approach applied to optimization of heat exchangers. *Appl. Therm. Eng.* **2012**, *42*, 119–128.
8. Zou, H.; Liang, D.; Zeng, J.; Feng, L. Quantum-behaved particle swarm optimization algorithm for the reconstruction of fiber Bragg grating sensor strain profiles. *Opt. Commun.* **2012**, *285*, 539–545.
9. Shayeghi, H.; Shayanfar, H.A.; Jalilzadeh, S.; Safari, A. Tuning of damping controller for UPFC using quantum particle swarm optimizer. *Energy Convers. Manag.* **2010**, *51*, 2299–2306.
10. Jia, P.; Tian, F.; He, Q.; Fan, S.; Liu, J.; Yang, S.X. Feature Extraction of Wound Infection Data for Electronic Nose Based on a Novel Weighted KPCA. *Sens. Actuators B Chem.* **2014**, *201*, 555–566.
11. Jia, P.; Tian, F.; Fan, S.; He, Q.; Feng, J.; Yang, S.X. A novel sensor array and classifier optimization method of electronic nose based on enhanced quantum-behaved particle swarm optimization. *Sen. Rev.* **2014**, *34*, 304–311.
12. Kern, S.; Müller, S.D.; Hansen, N.; Büche, D.; Ocenasek, J.; Koumoutsakos, P. Learning probability distributions in continuous evolutionary algorithms-a comparative review. *Nat. Comput.* **2004**, *3*, 77–112.
13. Wang, G.; Gandomi, A.H.; Alavi, A.H.; Hao, G. Hybrid krill herd algorithm with differential evolution for global numerical optimization. *Neural Comput. Appl.* **2014**, *25*, 297–308.
14. Guo, L.; Wang, G.; Gandomi, A.H.; Alavi, A.H.; Duan, H. A new improved krill herd algorithm for global numerical optimization. *Neurocomputing* **2014**, *138*, 392–402.

15. Wang, G.; Guo, L.; Gandomi, A.H.; Hao, G.; Wang, H. Chaotic krill herd algorithm. *Inf. Sci.* **2014**, *274*, 17–34.
16. Wang, G.; Deb, S.; Cui, Z. Monarch butterfly optimization. *Neural Comput. Appl.* **2015**, doi:10.1007/s00521-015-1923-y.
17. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68.
18. Wang, G.; Gandomi, A.H.; Zhao, X.; Chu, H.C.E. Hybridizing harmony search algorithm with cuckoo search for global numerical optimization. *Soft Comput.* **2014**, 1–13, doi:10.1007/s00500-014-1502-7.
19. Yu, B.; Cui, Z.; Zhang, G. Artificial plant optimization algorithm with correlation branches. *J. Bioinform. Intell. Control* **2013**, *2*, 146–155.
20. Kennedy, J.; Elbert, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, USA, 27 November–1 December 1995; pp. 1942–1948.
21. Shi, Y.; Eberhart, R.C. A modified particle swarm optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
22. Shi, Y.; Eberhart, R.C. Empirical study of particle swarm optimization. In Proceedings of the Congress on Evolutionary Computation, Washington, DC, USA, 6–9 July 2000; pp. 1945–1950.
23. Sun, J.; Feng, B.; Xu, W. 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.
24. Xi, M.; Sun, J.; Xu, W. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. *Appl. Math. Comput.* **2008**, *205*, 751–759.
25. Sun, J.; Fang, W.; Wu, X.; Xu, W. *Quantum Particle Swarm Optimization: Principle and Application*; Tsinghua University Press: Beijing, China, 2011.
26. Pulido, G.T.; Coello, C.A. A constraint-handing mechanism for particle swarm optimization. In Proceeding of the 2004 IEEE Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; pp. 1396–1403.
27. Derrac, J.; Garcia, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical as a methodology for computing evolutionary and swarm intelligence algorithm. *Swarm Evolut. Comput.* **2011**, *1*, 3–18.