International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 3, Issue 11 (June 2014) PP: 45-51

An Improved Quantum-behaved Particle Swarm Optimization Algorithm Based on Chaos Theory Exerting to Particle Position

Pan Dazhi[1], He Min[2], Chen Youjun[3], Yang Shuang[4]
College of Mathematic and Information, China West Normal University, Nanchong 637009, China

Abstract: *In this paper, we propose an improved quantum-behaved particle swarm optimization (QPSO), introducing chaos theory into QPSO and exerting logistic map to every particle position X(t) at a certain probability. In this improved QPSO, the logistic map is used to generate a set of chaotic offsets and produce multiple positions around X(t). According to their fitness, the particle's position is updated. In order to further enhance the diversity of particles, mutation operation is introduced into and acts on one dimension of the particle's position. What's more, the chaos and mutation probabilities are carefully selected. Through several typical function experiments, its result shows that the convergence accuracy of the improved QPSO is better than those of QPSO, so it is feasible and effective to introduce chaos theory and mutation operation into QPSO.*
Keywords: *Particle swarm optimization, Quantum-behaved particle swarm optimization, Chaos theory, Logistic Map, Mutation operation comma*

I. INTRODUCTION

Particle swarm optimization technique is considered as one of the modern heuristic algorithms for optimization introduced by James Kennedy and Eberhart[1,2] in 1995. It is based on the social behavior metaphor[1] and a population-based optimization technique. As an alternative tool of genetic algorithms, it has gained lots of attention from various of optimal control system applications. PSO is a stochastic search technique which has less memory requirement, meanwhile it can compute more effectively and easily than other evolutionary algorithms. In the iterative process, PSO has memory, i.e., every particle remembers its best solution as well as the group's best solution. So PSO is well suited to tackle dynamical problems. But it's very easy to fall into the local extreme point, which is called as premature.

In 2004, QPSO is proposed by J.Sun[3], which is an improvement of standard PSO algorithm from quantum mechanics. In QPSO, particles state equations are structured by wave function, and each particle state is described by the attractor p(t) and delta trap characteristic length L(t), which is determined by the mean-optimal position(MP). Because MP enhances the cooperation between particles and particles' waiting effort for each other, QPSO can prevent particles trapping into local minima. But the convergence speed of QPSO is slow and convergence accuracy is still low. So many improved methods for QPSO have been proposed. Yang[4] proposed a hybrid quantum-behaved particle swarm optimization based on cultural algorithm and differential evolution, which improved the QPSO's performance. Wang[5] introduces Gaussian disturbance into QPSO, which can effectively prevent the stagnation of the particles and make them escape the local optimum easily. Su[6] integrates simulated annealing into QPSO, and the improved QPSO can avoid the default of QPSO to fall into local extremum. Long[7] propose two improved QPSO algorithms integrating selection mechanism, which is exerted on the global best position to improve the search ability of the QPSO algorithm. Zhou[8] propose a revised QPSO (RQPSO) technique with a novel iterative equation, which helps prevent the evolutionary algorithms' tendency to be easily trapped into local optima as a result of rapid decline in diversity.

In this paper, we introduce chaos theory into QPSO and let logistic map generate a set of chaotic offsets for every particle at a certain probability, which are exerted on X(t) to produce multiple positions for updating each particle position. To enhance further the diversity of the population, the mutation operator in genetic algorithm is acted on particle position. Then the feasibility and effectiveness of the improved QPSO are examined by several typical functions.

II. REVIEW OF QPSO ALGORITHM AND CHAOS THEORY

2.1 QPSO algorithm

QPSO is a complex nonlinear system, and accords to state superimposed principle. Hence, quantum system possesses more states, and is an indeterminate system without determinate tracks, thus each particle can appear in arbitrary position in seeking space according to some probability, which is favorable in terms of the global convergence and shaking off the local extremum. In swarm, each particle has a position vector ($X_i(t)$) and a current local optimal position($P_i(t)$) encountered by oneself, and the swarm has a current global optimal position($G(t)$) encountered by the whole swarm.

In QPSO algorithm, the particle's velocity vector ($V_i(t)$) is removed and the position($X_i(t)$) of each particle can be updated with eq.(1-3).

$$p_{id}(t) = \varphi(t)P_{id}(t) + (1-\varphi(t))G_d(t), d = 1,2,..., D \tag{1}$$

$$MP(t) = (\sum_{i=1}^{M} P_{i1}(t)/M, \sum_{i=1}^{M} P_{i2}(t)/M, ..., \sum_{i=1}^{M} P_{iD}(t)/M) \tag{2}$$

$$X_{id}(t+1) = p_{id}(t) \pm \alpha |MP_d(t) - X_{id}(t)| \times \ln(1/u_{id}(t)) \tag{3}$$

where $p_i(t) = (p_{i1}(t), p_{i2}(t),..., p_{iD}(t))$ is the *i*th particle's attractor, M expresses the colony size, $\varphi(t)$ and $u_{id}(t)$ are Uniformly distributed random numbers with a scope from zero to one, α is the compressing-expansive factor, which is used to control the convergence speed.

2.2 Chaos theory

The chaotic state is a very common phenomenon in nonlinear systems and exists widely in the natural and social phenomena, their behavior is complex and pseudo-random. But the chaotic process, which seems to be confusion, is not entirely chaotic and exists the inherent fine regularity.

The logistic map is a very simple chaotic system , which is applied widely. Its definition is as following.

$$z_{k+1} = \mu z_k (1 - z_k) \tag{4}$$

Where z_k is a real-valued sequence and their values are between zero and one, μ is chaotic parameter.

When the value of μ is 3.571448, logistic map begins to enter a chaotic state. Some studies have shown that logistic map has been in a chaotic state when μ values between 3.571448 and 4, and the interval [3.571448,4] known as the chaotic region of logistic map. For any given initial value z_0, the sequence $\{z_k\}$ has chaotic characteristics. According to eq.(4), the value of z_0 can't be taken from the set $A = \{0,0.25,0.5,0.75,1\}$ when μ is 4, and the items of set A are called fixed-points.

III. MPROVED QPSO WITH LOGISTIC MAP

In QPSO algorithm, the waiting effect among particles with MP can prevent it prematurely to trap into local minima, but its convergence speed is slow and convergence accuracy is still low. In order to further improve the convergence accuracy of QPSO, we let logistic map generate a set of chaotic offsets, which is exerted on each particle's current position *X(t)*. Based on these offsets, a set of positions are produced around *X(t)* and are used to update each particle's position. The improved QPSO with chaos theory exerting on *X(t)* is denoted as CX-QPSO.

3.1 Logistic map exerting on particle's position

In QPSO, each particle generates their own next-generation iterative position according to MP and its attractor $p_i(t)$, so the iterative speed will be slowed down. In addition, the iterative precision of QPSO is not very high. In order to better control the particle iterative process, the chaos theory is introduced in QPSO. Firstly, generating the *i*th particle's initial iterative position $X_i^0(t)$ based on eq.(3); Secondly, generating chaotic offset sequence $\{\Delta_h(t), h = 1,2,..., H\}$ with logistic map; Thirdly, constituting $\{X_i^h(t), h = 1,2,..., H\}$ based on chaotic offset sequence; finally, determining the *i*th particle's iterative position based on fitness values of $\{X_i^h(t), h = 0,1,..., H\}$.

During chaotic thinking is integrated into the CX-QPSO algorithm, the *i*th particle's current position $X_i^0(t)$ is not looked as the initial value of the chaotic sequence, but to let the logistic chaotic system generate a chaotic offset sequence $\{\Delta_h(t), h = 1,2,..., H\}$. Suppose the problem is D-dimensional, and let P_C be the chaotic probability and C_α be chaotic offset factor, the chaotic operator is as following.

Chaotic Operator:

Step 1 Set the number of the *i*th particle's positions: $N_i=1$;

Step 2 Generate r_c randomly between zero and one, if $r_c < P_C$ then

Step 2.1 Use the random function to generate *D* random numbers in [0,1], and random constitute a D-dimensional vector $R_1(t)$ at the *t*th generation during the iterative process according to eq.(5).

$$R_1(t) = \{r_{11}(t), r_{12}(t), ..., r_{1D}(t)\}, r_{id} \in (0,1), d = 1,..., D \quad (5)$$

Step2.2 Let $R_1(t)$ as the initial value, and generate the chaotic sequence $\{R_h(t) = (r_{h1}(t),..., r_{hD}(t)), h = 1,..., H\}$ by logistic map based on eq.(6).

$$r_{hd}(t) = 4 \times r_{h-1,d}(t) \times (1 - r_{h-1,d}(t)), d = 1,..., D \quad (6)$$

Step 2.3 Let $C_\alpha = (C_{\alpha 1}, C_{\alpha 2},..., C_{\alpha D})$ act on $\{R_h(t), h = 1,..., H\}$, and generate the offset sequence $\{\Delta_h(t) = (\Delta_{h1}(t), \Delta_{h2}(t),..., \Delta_{hD}(t)), h = 1,..., H\}$ according to eq.(7).

$$\Delta_{hd}(t) = C_{\alpha d} \times r_{hd}(t), d = 1,..., D \quad (7)$$

Step 2.4 Around $X_i^0(t)$, constitute $\{X_i^h(t), h = 1,..., H\}$ based on eq.(8).

$$X_i^h(t) = X_i^0(t) + \Delta_h(t), h = 1,..., H \quad (8)$$

Step 2.5 Renew the number of the *i*th particle's positions: $N_i=1+H$.

Step 3Evaluate each position in $\{X_i^h(t), h = 0,..., N_i - 1\}$ with fitness function, and determine the *i*th particle's iterative position based on following equation.

$$\{X_i(t) = X_i^s(t) \mid f(X_i^s(t)) = \max_{0 \le h \le N_i - 1} f(X_i^h(t))\} \quad (9)$$

3.2 Mutation operator

With the iteration process of CX-QPSO, the diversity of particle groups gradually loses. In order to prevent the swarm to premature converge prematurely to the local extreme points, mutation operator of the genetic algorithm is embedded into the CX-QPSO, which can enhance the diversity of the population, and improve the algorithm global convergence.

In CX-QPSO, each particle's position is looked directly as a chromosome which adopts real coding, and the mutation operator acts on only one dimension of particle's position which is selected randomly. Let P_m be the probability of mutation, the mutation operator is as following.

Mutation Operator:

Step 1 Generate r_m randomly between zero and one;

Step 2 if $r_m < P_m$ then

Step 2.1 Generate r_d randomly between zero and one, and constitute the variation dimension based on eq.(10).

$$d = \lceil r_d \times D \rceil \quad (10)$$

Step 2.2 Generate r_v randomly between zero and one, and act mutation to the *d*th dimension of the *i*th particle's position according to eq.(11).

$$x_{id}(t) = r_v \times (U_d - L_d) \quad (11)$$

Where U_d and L_d are the upper and lower of the *d*th dimension.

Step 2.3 Update the *i*th particle's $P_i(t)$.

3.3 Algorithm of CX-QPSO

Assuming that the colony size is M and the largest number of iterations is *T*, CX-QPSO algorithm is described as following.

Step 1Initialization : randomly generate *M* particles' $X_i(0), i=1,...,M$;

Step 2 Update each particle's optimal position $P_i(0)$ and get whole swarm's optimal position *G(0)*;

Step] Carry out iterative computation of *T* generations;

Step 3.1 Let *t* equal to 1;

Step 3.2 Compute *MP(t)* based on eq.(2);

Step 3.3 Execute the *t*th generation iteration;

Step 3.3.1 Let *i* equal to 1;

Step 3.3.2 Compute the *i*th particle's $p_i(t)$ based on eq.(1);

Step 3.3.3 generate the *i*th particle's initial iterative position $X_i^0(t)$ with eq.(3);

Step 3.3.4 Carry out **Chaotic Operator**;

step 3.3.5 Carry out **Mutation Operator**;

Step 3.3.6 Update each particle's optimal position $P_i(t)$ and get whole swarm's optimal position *G(t)*;

Step 3.3.7 Let $i=i+1$, if $i \le M$ then goto *Step3.3.2*, else goto *Step3.4* endif;

Step 3.4 Let $t=t+1$, if $t \le T$ then goto *Step3.2*, else goto *Step4* endif;

Step 4 Iteration is over, *G(T)* is solution for problem with CX-QPSO.

IV. DETERMINING THE CHAOTIC PROBABILITY AND MUTATION PROBABILITY

In CX-QPSO, the values of chaotic probability P_C and mutation probability P_m will product a great impaction on the performance of CX-QPSO, the following experimental scheme is designed to determine their values.

4.1 Experimental scheme

To determine the appropriate parameter values, the following function is selected for analysis by many groups of test parameters. The selected function is Rosenbrock as following.

$$f_1(x)=\sum_{i=1}^{n-1}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2)$$

The admissible range of the Rosenbrock's variable is $[-30,30]^n$, and the optimum is 0.

In the testing process, there are nine sets for colony size *M*, function dimension *D* and iterative times *T*, which are shown in Table 1.

Table 1. The nine kinds of test parameter sets for CX-QPSO

	s1	s2	s3	s4	s5	s6	s7	s8	s9
M	20	20	20	40	40	40	60	60	60
T	1000	1500	2000	1000	1500	2000	1000	1500	2000
D	10	20	30	10	20	30	10	20	30

The compressing-expansive factor α is recommended from Sun[8] with a linearly decreasing way based on following equation.

$$\alpha(t)=\frac{T-t}{T}(\alpha_1-\alpha_2)+\alpha_2 \tag{12}$$

Where α_1 and α_2 are the upper and lower bounds of the compressing-expansive factor. According to literature[9], their values are 1.0 and 0.5.

In chaotic operator, there are double parameters: *H* and C_α, which need to be fixed. In CX-QPSO, we let *H* be equal to 4 and C_α be equal to $0.00005\times(U_d-L_d)$.

In the process of testing CX-QPSO for optimizing the Rosenbrock function, through the analysis of preliminary experimental data, there are twelve sets of parameter values for chaotic probability and mutation probability, which are listed in Table 2.

Table 2. Twelve sets of parameter values for P_C and P_m

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
P_m	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4
P_C	0.2	0.3	0.4	0.5	0.2	0.3	0.4	0.5	0.2	0.3	0.4	0.5

4.2 Analyzing CX-QPSO performance with the experimental scheme

According to Table 1 and Table 2 parameter sets, for each set of parameters, we execute CX-QPSO in 60 dependent runs, and take the mean results as the optimal results, the results are listed in Table 3.

Table 3. The optimal results of CX-QPSO with *C1- C12* for Rosenbrock

	s1	s2	s3	s4	s5	s6	s7	s8	s9
C1	0.6804	9.8793	26.1921	0.4810	2.6538	8.8743	0.9671	6.4975	17.2280
C2	0.5192	11.2186	28.4347	0.6791	3.4549	13.5775	0.9689	5.7558	8.1765
C3	0.7937	7.6274	29.7926	0.8374	2.4708	9.0879	0.9734	6.7084	13.2653
C4	0.8258	10.4184	21.7804	1.2793	2.2014	11.4071	1.0322	4.2412	11.1890
C5	2.2226	8.8002	24.8689	0.8443	2.3211	11.6877	0.9001	4.6745	11.0871
C6	0.4232	6.6761	20.8133	0.5640	2.7287	9.5709	0.7960	3.7020	8.6876
C7	0.3227	6.5968	30.3435	1.9188	3.3879	10.5506	0.8550	6.0181	9.7695
C8	0.6255	9.5737	24.3873	0.5584	2.2847	10.7289	0.7347	5.4774	15.5790
C9	0.6317	7.6931	21.6881	0.6603	2.3779	8.3043	0.9305	3.8305	17.7253
C10	0.6316	7.5281	22.2506	0.6005	2.9128	7.7188	0.9968	4.9390	12.8217
C11	0.6583	6.4127	21.0082	0.5549	2.9868	11.4760	0.8682	4.1207	12.9143
C12	1.9256	12.6698	20.0616	0.6209	2.3545	10.5419	0.8746	4.4135	12.3929
QPSO	7.3711	34.3630	59.6210	2.2314	6.5903	23.6340	1.7677	9.2751	21.3420

In order to analyze the performance of the CX-QPSO algorithm in each parameter set of P_C and P_m, we compare one result to other fifteen results with same parameter values of M and T and D. The evaluating processes are as following.

Firstly, computing the maximum and minimum values of each group results by CX-QPSO at different P_C and P_m based on eq.(13).

$$M_j = \max_{1\le i\le 12} d_{ij}, m_j = \min_{1\le i\le 12} d_{ij}, j = 1,...,9 \quad (13)$$

Where d_{ij} is the optimal result by CX-QPSO at the paramer set of C_i and s_j.

Secondly, calculating the performance factor of all of optimal results based on the maximum and minimum values of each group results with eq.(14).

$$e_{ij} = \frac{d_{ij} - m_j}{M_j - m_j}, i = 1,...,12; j = 1,...,9 \quad (14)$$

Thirdly, summary of performance factor under each parameter P_C and P_m, and get the optimal parameter set of P_C and P_m with eq.(15).

$$E_i = \sum_{j=1}^{9} e_{ij}, i = 1,...,12; \{optimal\ paramter\ set\ C_k \mid E_k = \min_{1\le i\le 12} E_i\} \quad (15)$$

Table 4. The performance factor of CX-QPSO with *C1- C12*

	s1	s2	s3	s4	s5	s6	s7	s8	s9	E_i
C1	0.1883	0.5540	0.5962	0.0000	0.3609	0.1972	0.7811	0.9299	0.9479	4.5556
C2	0.1034	0.7681	0.8144	0.1378	1.0000	1.0000	0.7872	0.6831	0.0000	5.2940
C3	0.2479	0.1941	0.9464	0.2479	0.2149	0.2337	0.8024	1.0000	0.5329	4.4202
C4	0.2648	0.6402	0.1672	0.5553	0.0000	0.6295	1.0000	0.1793	0.3155	3.7518
C5	1.0000	0.3816	0.4676	0.2527	0.0955	0.6774	0.5559	0.3235	0.3048	4.0589
C6	0.0529	0.0421	0.0731	0.0578	0.4206	0.3161	0.2059	0.0000	0.0535	1.2220
C7	0.0000	0.0294	1.0000	1.0000	0.9466	0.4833	0.4042	0.7704	0.1668	4.8008
C8	0.1594	0.5052	0.4207	0.0539	0.0665	0.5138	0.0000	0.5905	0.7752	3.0852
C9	0.1626	0.2046	0.1582	0.1247	0.1408	0.0999	0.6582	0.0427	1.0000	2.5918
C10	0.1626	0.1783	0.2129	0.0831	0.5676	0.0000	0.8810	0.4115	0.4865	2.9834
C11	0.1766	0.0000	0.0921	0.0514	0.6266	0.6413	0.4485	0.1393	0.4962	2.6719
C12	0.8437	1.0000	0.0000	0.0973	0.1222	0.4819	0.4700	0.2367	0.4416	3.6932

The values of e_{ij} and E_i are computed and are listed in Table 4. From Table 4, the optimal results of CX-QPSO for Rosenbrock are much better than those of QPSO under all 12 kinds of value combinations of the chaotic and mutation probability. According to the evaluation criteria with eq.(13-15), the performance factors of CX-QPSO with C6 is less than two, and its value is close to 1. Because E_6 is smallest among twelve performance factors, the value combinations of the chaos and mutation probability can take *C6* while CX-QPSO is used to resolve optimal problems. In the following algorithm testing process, the chaotic probability P_C and mutation probability P_m are both 0.3 for CX-QPSO.

V. ALGORITHM TESTING

In order to further compare the feasibility and performance of CX-QPSO at parameter set *C6* to those of QPSO, in this section, we will adopt following three nonlinear benchmark testing functions, which are commonly used in [10,11,12], to examine CX-QPSO's feasibility and performance. These functions, the admissible range of the variable and the optimum are summarized in following.

A. Rastrigin function

$$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$$

The admissible range of the Rastrigin's variable is $[-5.15,5.15]^n$, and the mini-optimum is 0.

B. Griewark function

$$f_3(x) = \frac{1}{400}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$$

The admissible range of the Griewark's variable is $[-600,600]^n$, and the mini-optimum is 0.

C. Schaffer function

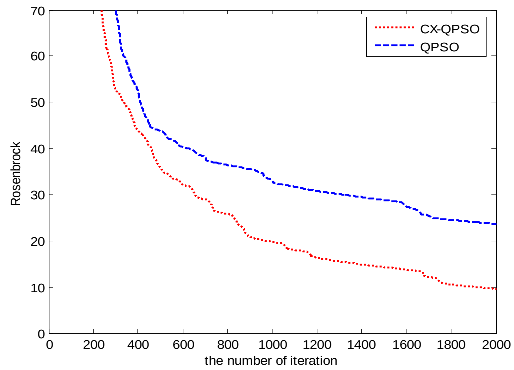
$$f_4(x) = 0.5 + \frac{\sin^2 \sqrt{\sum_{i=1}^{n} x_i^2} - 0.5}{[1.0 + 0.001\sum_{i=1}^{n} x_i^2]^2}$$

The admissible range of the Schaffer's variable is $[-100,100]^n$, and the min-optimum is 0. To evaluate the performance of CX-QPSO, QPSO is looked as the reference algorithm. CX-QPSO and QPSO will be used to optimize those given testing functions with parameter sets in Table 1, and the chaotic probability P_C and mutation probability P_m of CX-QPSO are both 0.3. CX-QPSO and QPSO for each testing function are operated 60 times independently, and the average values are used as the optimal results. The optimization results for testing functions are listed in Table 5.

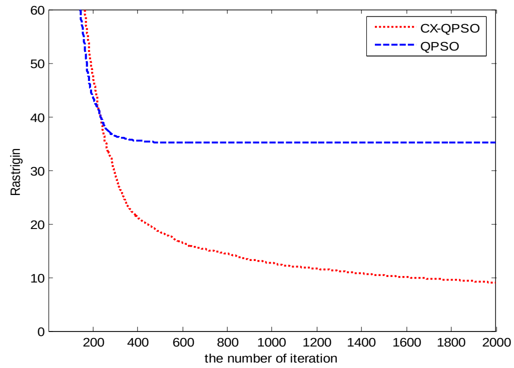
Table 5. The optimal results of CX-QPSO and QPSO for four testing functions

Function	f1		f2		f3		f4	
method	QPSO	CX-QPSO	QPSO	CX-QPSO	QPSO	CX-QPSO	QPSO	CX-QPSO
s1	7.3711	0.4232	7.2035	0.2997	0.1218	0.0892	0.0382	0.0292
s2	34.3630	6.6761	22.6054	2.8452	0.0566	0.0352	0.0969	0.0820
s3	59.6210	20.8133	42.4250	7.7982	0.0955	0.0238	0.1741	0.1243
s4	2.2314	0.5640	5.0146	0.0955	0.0871	0.0693	0.0289	0.0175
s5	6.5903	2.7287	16.5163	3.0742	0.0151	0.0150	0.0646	0.0475
s6	23.6340	9.5709	35.2215	9.0103	0.0180	0.0141	0.1086	0.0908
s7	1.7677	0.7960	3.8804	0.1997	0.1027	0.0693	0.0218	0.0157
s8	9.2751	3.7020	14.6060	2.6735	0.0173	0.0171	0.0515	0.0461
s9	21.3420	8.6876	30.3064	9.3525	0.0123	0.0106	0.0908	0.0727

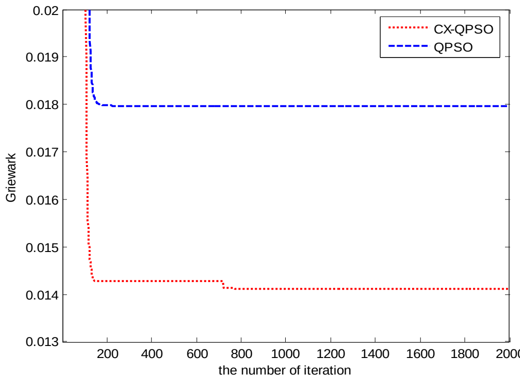
To further compare the performance of QPSO and CX-QPSO, the iterative processes of four functions with QPSO and CX-QPSO are compared at *s6* and *C6* parameter set, and the comparison is shown in Figure 1.



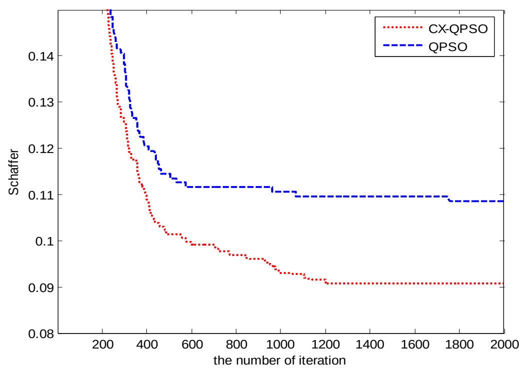
A) Rosenbrock



B) Rastrigin



C) Griewark



D) Schaffer

Fig1. Evolution of average fitness of four functions by QPSO and CX-QPSO with s6 and C6

Based on Figure 1, in the early stages of the iterative process, the iterative speed of QPSO is faster than CX-QPSO for Rastrigin, but its convergence prematurely stop, thus the convergence accuracy of the CX-QPSO is ultimately much better than that of QPSO. For the other three testing functions, the iterative speed and accuracy of CX-QPSO are far superior to QPSO. For four nonlinear testing functions, the optimal results of CX-QPSO are far superior to those of QPSO from Table 5. So CX-QPSO is feasible and the performance of CX-QPSO is better than QPSO.

VI. CONCLUSIONS

In order to improve the convergence rate and convergence precision of QPSO, this paper proposes an improved QPSO algorithm(CX-QPSO), which integrates Chaos theory into QPSO and let logistic map act on every particle's position. In order to prevent the swarm to premature converge prematurely to the local extreme points, mutation operator is embedded into CX-QPSO. Through designing testing scheme and testing CX-QPSO algorithm for Rosenbrock, we attained the appropriate chaos and mutation probability which are both 0.3. From the experimental results, CX-QPSO is far superior to QPSO, so CX-QPSO is effective and feasible.

VII. ACKNOWLEDGEMENTS

This work was supported by the Scientific Research Fund of Sichuan Education(11ZA040, 14ZA0127) and Doctor Foundation of China West Normal University(12B022).

REFERENCES

[1] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro machine and Human Science, Nagoya, Japan, 1995, pp.39-43.

[2] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995,pp. 1942-1948.

[3] J.Sun,W.B.Xu, Particle swarm optimization with particles having quantum behavior, proceedings of IEEE Congr. Evol. Comput., Portland,USA, 2004, pp. 325-331.

[4] Kaiqiao Yang, Kenjiro Maginu, Hirosato Nomura, Cultural algorithm-based quantum-behaved particle swarm optimization, J. comput. Math.2010, 87, pp. 2143-2157.

[5] X.G.Wang, H.X.Long, J.Sun,Quantum-behaved particle swarm optimization based on Gaussian disturbance, Appl. Res. Comput. ,2010, 27, pp. 2093-2096.

[6] H.S.Su, Simulated Annealing Based Quantum-Behaved Particle Swarm Optimization, J. Comput. Inf. Syst. ,2010, 8, pp. 1795-1800.

[7] H.X.Long, W.B.Xu,X.G.Wang etc.,Using selection to improve quantum-behaved particle swarm optimization, Control Decis. , 2010, 25, pp. 1499-1506.

[8] D.Zhou, J.Sun, C.H.Lai,etc., An improved quantum-behaved particle swarm optimization and its application to medical image registration, J. comput. Math. ,2010, 88, pp.1208-1223.

[9] J.Sun, Particle swarm optimization with particles having quantum behavior[D],Jiangnan University,2009.

[10] D.Z. Pan, Z.B. Liu. A second-order and double-subgroup particle swarm optimization algorithm based on crossover operation, J. Inf. Comput. Sci. ,2012, 9, pp. 1155-1166.

[11] Y.Jiang, T.S. Hu, C.C. Huang etc., An improved particle swarm optimization algorithm, Appl. Math. and Comput. ,2007, 193, pp. 231-239.

[12] X.G.Wang, H.X.Long, J.Sun,Quantum-behaved particle swarm optimization based on Gaussian disturbance, Appl. Res. Comput. , 2010, 27,pp. 2093-2096.