An Improved Quantum-behaved Particle Swarm Optimization Algorithm Based on Random Weight

Dazhi Pan, Ying Ci, Min He, Hongying He
College of Mathematic and Information, China West Normal University, Nanchong 637009, China
Email: pdzzj@126.com

***Abstract*— The standard particle swarm optimization (PSO) algorithm converges very fast, while it is very easy to fall into the local extreme point. According to waiting effect among particles with mean-optimal position(MP), the quantum-behaved particle swarm optimization (QPSO) algorithm can prevent the particle from falling into local extreme point prematurely, but its convergence speed and convergence precision are both low. In order to further improve the precision of QPSO algorithm, the evaluation method of δ trap characteristic length $L(t)$ of wave function for describing the particle's state is modified. In QPSO, a random weight to each particle in swarm is introduced, and according to the order of each particle's best position fitting value, there are three evaluation programs for $L(t)$, which are random-weight mean-optimal position(RMP), reverse-order random-weight mean-optimal position(RRMP) and same-order random-weight mean-optimal position(SRMP). Through the test of several typical functions, its result shows that the convergence accuracy of QPSO with RMP and RRMP is better than those of QPSO with MP, so the evaluation of $L(t)$ with RMP and RRMP is feasible and effective.**

***Index Terms*— Random weight, Random-weight mean-optimal position, Particle swarm optimization, Quantum-behaved particle swarm optimization**

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 bases on the social behavior metaphor [1] and is a population-based optimization technique. As an alternative tool of genetic algorithms, PSO gains lots of attention from various of optimal control system applications. PSO is a stochastic search technique with less memory requirement, meanwhile it can compute more effectively and even easier to implement compare with other evolutionary algorithms. One key advantage is that PSO has memory, i.e., every particle remembers its best solution (local best) as well as the group's best solution (global best). So PSO is well suited to tackle dynamical problems. Another advantage of PSO is that it's initial population is fixed throughout the execution of the algorithm, and there is no need to apply operators for the population, which is a both time- and memory-storage-consuming process. But it's very easy to fall into the local extreme point which is called premature.

In order to overcome the undesirable premature convergence of PSO, some improved PSO algorithms are proposed. A Guaranteed Convergence PSO (GCPSO) [3] has been discussed,using a separate velocity update formula for the best particle in the swarm. Shelokar et al. [4]proposed an improved PSO hybridized with ant colony approach which applies PSO for global optimization and uses the idea of ant colony approach to update positions of particles to attain the feasible solution space rapidly. An improved PSO(IPSO) is presented in [5], in the IPSO process, the population is divided into several subgroups, the entire population is shuffled at periodic stages in the evolution, and then points are reassigned to subgroups to ensure information sharing. Pan [6] proposed an improved particle swarm optimization algorithm based on the optimal and sub-optimal position(OSP-PSO). OSP-PSO enlarges the search space and enhances global search ability, and adopts mutation operator to keep the swarm's diversity. Zhao [7] proposed A perturbed particle swarm algorithm for numerical optimization which is based upon the concept of perturbed global best to deal with the problem of premature convergence and diversity maintenance within the swarm. Aiming at robot path planning in an environment with danger sources, Gong [8] develops an improved multi-objective particle swarm optimization algorithm, in which a self-adaptive mutation operation based on the degree of a path blocked by obstacles is designed to improve the feasibility of a new path. In order to solve complex multimodal optimizing problems, Zhai [9] proposes a Baldwin effect based on learning particle swarm optimizer (BELPSO) to improve the performance of PSO, and experimental simulations show that BELPSO has a wider search range of feasible solution space than PSO.

In 2004, J.Sun [10] proposed QPSO, 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 the characteristic length $L(t)$ of δ trap, which is determined by the mean-optimal position(MP). Because MP enhances the cooperation between

This work was supported the Scientific Research Fund of Sichuan Education(No. 11ZA040 and No. 10ZB018).

doi:10.4304/jsw.8.6.1327-1332

particles and particles' waiting with each other, QPSO can prevent particles trapping into local minima. But the convergence speed and convergence accuracy of QPSO are also slow.

In order to improve the performance of QPSO, many improved methods for QPSO have been proposed. Yang [11] proposes a hybrid quantum-behaved particle swarm optimization based on cultural algorithm and differential evolution, which strengthen the QPSO's performance. Wang [12] introduces Gaussian disturbance into QPSO, which can effectively prevent the stagnation of the particles and make them escape the local optimum easily. Su [13] integrates simulated annealing into QPSO, so the improved QPSO can avoid the default of falling into local extremum. Zhou [14] work out a revised QPSO (RQPSO) technique with a novel iterative equation, which helps to prevent the evolutionary algorithms from tending to be easily trapped into local optima and lead to a rapid decline of diversity.

In this paper, we introduce a set of random weights for each particle in swarm, and get three evaluating programs to reassess the $L(t)$ based on the order of the each particle's best position fitting value. The programs are random-weight mean-optimal position(RMP), reverse-order random-weight mean-optimal position(RRMP) and same-order random-weight mean-optimal position(SRMP). Then the feasibility and effectiveness of these programs are examined by several typical functions.

II. REVIEW OF QPSO

A. QPSO algorithm

QPSO is a complex nonlinear system, following the state superimposed principle. 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. $P_i(t)$ and $G(t)$ can be modified by following equations.

$$P_i(0) = X_i(0),$$
$$P_i(t) = \begin{cases} P_i(t-1), & if\ f(P_i(t-1)) \geq f(X_i(t)) \\ X_i(t), & if\ f(P_i(t-1)) < f(X_i(t)) \end{cases}$$
$$i = 1, \ldots, M; t = 1, \ldots, T \quad (1)$$

$$G(t) = \{P_j(t) | P_j(t) = \max_{1 \leq i \leq M} f(P_i(t))\},$$
$$t = 0, \ldots, T \quad (2)$$

Where $f(\cdot)$ is the fitness value at X position. When X is better, the value of $f(X)$ is bigger. M expresses the colony size. T is the maximum number of iteration and t is the current number of iteration.

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.(3) $\sim$ (5).

$$p_{id}(t) = \varphi(t) P_{id}(t) + (1 - \varphi(t)) G_d(t),$$
$$d = 1, \ldots, D \quad (3)$$

$$MP(t) = (\frac{1}{M} \sum_{i=1}^{M} P_{i1}(t), \frac{1}{M} \sum_{i=1}^{M} P_{i2}(t),$$
$$\ldots, \frac{1}{M} \sum_{i=1}^{M} P_{iD}(t)) \quad (4)$$

$$X_{id}(t+1) = \ p_{id}(t) \pm \alpha |MP_d(t) - X_{id}(t)|$$
$$\times \ln[\frac{1}{u_{id}(t)}], d = 1, \ldots, D \quad (5)$$

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

B. Algorithm of QPSO

Assuming that the colony size is M and the largest number of iterations is T, the QPSO algorithm is described as following:

Step 1 Initialization : generate randomly M particles' $X_i(0), i = 1, ..., M$;
Step 2 According to the (1)and(2), update each particle's $P_i(0)$ and get whole swarm's $G(0)$;
Step 3 Carry out iterative computation of T generations;
Step 3.1 Let t equal to 1;
Step 3.2 Compute $MP(t)$ based on (4);
Step 3.3 Execute the ith generation iteration;
Step 3.3.1 Let i equal to 1;
Step 3.3.2 Compute the ith particle's $p_i(t)$ based on (3), and update its $X_i(t)$ and $P_i(t)$ with (5) and (1);
Step 3.3.3 According to (2), update the whole swarm optimal position $G(t)$;
Step 3.3.4 Let $i = i + 1$, $if (i \leq M)\ then\ goto\ Step 3.3.2,\ else\ goto\ Step 3.4;\ endif$;
Step 3.4 Let $t = t + 1$, $if (t \leq T)\ then\ goto\ Step 3.2,\ else\ goto\ Step 4;\ endif$;
Step 4 Iteration is over, $G(T)$ is the solution of problem with QPSO.

III. RANDOM WEIGHT MEAN OPTIMAL POSITION

In QPSO algorithm, the waiting effect among particles with MP can prevent it to trap into local minima prematurely, but its convergence speed and convergence accuracy are both slow. In order to further improve the convergence accuracy of QPSO, We introduce a set of random weights to particles and construct three new evaluating methods with each particle's current optimal position for $L(t)$.

A. Random weight

In order to get the random-weight mean-optimal position to evaluate $L(t)$, we need a set of weights which are produced by random function. The process generating random weight is as following.

First, using the random function to generate M random numbers in $[0, 1]$, and constituting a M-dimensional vector $R_1(t)$ at the tth generation during the iterative process according to eq.(6).

$$R_1(t) = (r_{11}(t), r_{12}(t), \ldots, r_{1M}(t)) \\ r_{1i}(t) \sim U(0, 1), i = 1, \ldots, M \tag{6}$$

Second, normalizing $R_1(t)$ and getting random weight vector $R_2(t)$ based on the following equation.

$$R_2(t) = (r_{21}(t), r_{22}(t), \ldots, r_{2M}(t)), r_{2i}(t) \\ = \frac{r_{1i}(t)}{\sum_{i=1}^{M} r_{1i}(t)}, i = 1, \ldots, M \tag{7}$$

B. Evaluating program for $L(t)$

Based on random weight vector $R_2(t)$ and each particle's current optimal position $P_i(t)$, there are three evaluating programs for $L(t)$.

Evaluating program 1: According to eq.(8) with $R_2(t)$ and $P_i(t)$, constructing the point to evaluate $L(t)$, which is called random-weight mean-optimal position(RMP). $X_i(t)$ is updated with eq.(9), and the QPSO that replaces MP with RMP is denoted as RQPSO.

$$RMP(t) = R_2(t) \times P(t) = (\sum_{i=1}^{M} R_{2i}(t)P_{i1}(t), \\ \sum_{i=1}^{M} R_{2i}(t)P_{i2}(t), \ldots, \sum_{i=1}^{M} R_{2i}(t)P_{iD}(t)) \tag{8}$$

$$X_{id}(t+1) = p_{id}(t) \pm \alpha|RMP_d(t) - X_{id}(t)| \\ \times \ln[\frac{1}{u_{id}(t)}], d = 1, \ldots, D \tag{9}$$

where $P(t) = (P_1(t), P_2(t), \ldots, P_M(t))^\top$.

Evaluating program 2: First, each component in $R_2(t)$ on behalf of random weight is sorted by descending order and random descending weight $R_3(t)$ is generated with eq.(10); Second, each particle's $P_i(t)$ is sorted based on its fitness value by ascending order and the ordered sequence $RP(t)$ is generated with eq.(11) and (12); Finally, With $R_3(t)$ and $RP(t)$, constructing the point for re-evaluating $L(t)$ based on eq.(13), which is called reverse-order random-weight mean-optimal position(RRMP). $X_i(t)$ is updated with eq.(14), and the QPSO that replaces MP with RRMP is denoted as RRQPSO.

$$R_3(t) = sort(R_2(t),' descend') \tag{10}$$

$$[B(t), I(t)] = sort(f(P(t))), f(P(t)) = \\ (f(P_1(t)), f(P_2(t)), \ldots, f(P_M(t))) \tag{11}$$

$$RP(t) = (RP_1(t), RP_2(t), \ldots, RP_M(t))^\top \\ = (P_{I_1(t)}(t), \ldots, P_{I_2(t)}(t), \ldots, P_{I_M(t)}(t))^\top \tag{12}$$

$$RRMP(t) = R_3(t) \times RP(t) = \\ (\sum_{i=1}^{M} R_{3i}(t)RP_{i1}(t), \sum_{i=1}^{M} R_{3i}(t)RP_{i2}(t), \\ \ldots, \sum_{i=1}^{M} R_{3i}(t)RP_{iD}(t)) \tag{13}$$

$$X_{id}(t+1) = p_{id}(t) \pm \alpha|RRMP_d(t) - X_{id}(t)| \\ \times \ln[\frac{1}{u_{id}(t)}], d = 1, \ldots, D \tag{14}$$

where $sort(\cdot,' descend')$ realizes sorting in descending order, $sort(\cdot)$ realizes sorting in ascending order and returns sorting results $B(t)$ and the index list $I(t) = (I_1(t), I_2(t), ..., I_M(t))$ which expresses index number in the original sequence of the element in sorting sequence.

Evaluating program 3: First, each component in $R_2(t)$ on behalf of random weight is sorted by ascending order and random ascending weight $R_4(t)$ is generated with eq.(15); Second, each particle's $P_i(t)$ is sorted based on its fitness value by ascending order and the ordered sequence $RP(t)$ is generated with eq.(11) and (12); Finally, With $R_4(t)$ and $RP(t)$, constructing the point for re-evaluating $L(t)$ based on eq.(16), which is called same-order random-weight mean-optimal position(SRMP). $X_i(t)$ is updated with eq.(17), and the QPSO that replaces MP with SRMP is denoted as SRQPSO.

$$R_4(t) = sort(R_2(t)) \tag{15}$$

$$SRMP(t) = R_4(t) \times RP(t) = \\ (\sum_{i=1}^{M} R_{4i}(t)RP_{i1}(t), \sum_{i=1}^{M} R_{4i}(t)RP_{i2}(t), \\ \ldots, \sum_{i=1}^{M} R_{4i}(t)RP_{iD}(t)) \tag{16}$$

$$X_{id}(t+1) = p_{id}(t) \pm \alpha|SRMP_d(t) - X_{id}(t)| \\ \times \ln[\frac{1}{u_{id}(t)}], d = 1, \ldots, D \tag{17}$$

C. Complexity Analysis of RQPSO,RRQPSO and SRQPSO

Compared with QPSO, these three new algorithms only add the sorting operation of the particle swarm fitness and random weight in every generation. The time complexity of the three new algorithms is $O(T \times M \times D)$, which is is not change.

IV. ALGORITHM TESTING

In order to compare the feasibility and performance of RQPSO, RRQPSO and SRQPSO with those of QPSO in this section, there are four nonlinear benchmark testing functions that are commonly used in [5], [6], [15]. These functions, the admissible range of the variable and the optimum are summarized in following.

1) Rastrigin function

$$F_1(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$$

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

2) Rosenbrock function

$$F_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

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

3) Griewark function

$$F_3(x) = \frac{1}{400}\sum_{i=1}^{n} x_1^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$$

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

4) Schaffer function

$$F_4(x) = 0.5 + \frac{sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$$

The admissible range of the Schaffer's variable is $[-100, 100]^2$,and the Minimum is 0.

In the testing process, these algorithms are used to solve the Minimum of the four testing functions, so the fitness function $f(X)$ can be designed as following.

$$f(X) = \begin{cases} C_{max} - F(X), & if\ F(X) \le C_{max} \\ 0, & if\ F(X) > C_{max} \end{cases}$$

Where C_{max} is a greater constant positive number.

In the testing process, the values have twelve sets for colony size M, function dimension D and iterative times T, which are shown in Table I.

The compressing-expansive factor α is recommended from Sun [16] with a linearly decreasing way based on following equation.

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

TABLE I.
THE TWELVE KINDS OF TEST PARAMETER SETS FOR RQPSO,RRQPSO AND SRQPSO

	$s1$	$s2$	$s3$	$s4$	$s5$	$s6$
M	20	20	20	40	40	40
T	1000	1500	2000	1000	1500	2000
D	10(2)	20(2)	30(2)	10(2)	20(2)	30(2)
	$s7$	$s8$	$s9$	$s10$	$s11$	$s12$
M	60	60	60	80	80	80
T	1000	1500	2000	1000	1500	2000
D	10(2)	20(2)	30(2)	10(2)	20(2)	30(2)

Where α_1 and α_2 are the upper and lower bounds of the compressing-expansive factor. According to literature [16], their values are 1.0 and 0.5.

All four Algorithms tests are carried out in the same testing environment. The experimental hardware condition of equipment is that CPU is Intel(R) Core(TM)2 Duo CPU T6500 @2.10GH and memory capacity is 2.0GB, the software condition is that OS is Microsoft Window XP Service Pack3 and the application soft is Mathlab7.0.

To evaluate the performance of RQPSO, RRQPSO and SRQPSO, the QPSO is used to optimize those given testing functions with parameter sets in Table[I]. RQPSO, RRQPSO and SRQPSO algorithms for each testing function are operated 60 times independently, taking the average values as the optimal results. The optimization results for testing functions are listed in Table II, Table III, Table IV and Table V, in which the optimal value and the worst value are bold and italic represent respectively for each function at each set of parameters.

TABLE II.
THE OPTIMAL RESULTS OF THE QPSO RQPSO, RRQPSO AND SRQPSO FOR RASTRIGIN

$Method$	QPSO	RQPSO	RRQPSO	SRQPSO
$s1$	*7.1471*	4.5781	**4.4865**	4.8421
$s2$	*23.7961*	**15.3303**	16.9518	18.4896
$s3$	*41.3902*	30.2092	**30.0433**	32.585
$s4$	3.7151	**2.9689**	3.2831	*3.8472*
$s5$	11.6908	11.9571	**11.1076**	*13.1998*
$s6$	23.1825	24.7247	**22.6522**	*25.0066*
$s7$	2.6866	**2.5549**	2.6640	*2.8356*
$s8$	10.9282	10.7296	**10.4814**	*11.1601*
$s9$	*21.7564*	20.3801	**19.6836**	21.6404
$s10$	2.3220	**2.2557**	2.2765	*2.4542*
$s11$	*9.9339*	9.7177	**8.9587**	9.7673
$s12$	19.3354	19.1205	**18.8231**	*19.6836*

From Table II, for Rastrigin function, RRQPSO gets eight optimal values and RQPSO gets four optimal value with twelve sets of parameters, but QPSO gets five worst values and SRQPSO gets seven worst values.

In Table III, for Rosenbrock function, RRQPSO gets three optimal values and one worst value, RQPSO obtains tree optimal values, QPSO receives seven worst values and SRQPSO gains six optimal values and four worst values. Although SRQPSO obtains optimum more often, it obtains worst results with four sets of parameter,indicating that SRQPSO is unstable.

In Table IV, for Griewark function, RRQPSO gets eight optimal values and RQPSO gets three optimal value,

TABLE III.
THE OPTIMAL RESULTS OF THE QPSO RQPSO, RRQPSO AND SRQPSO FOR ROSENBROCK

$Method$	QPSO	RQPSO	RRQPSO	SRQPSO
$s1$	6.6852	6.3199	**2.6587**	*15.0079*
$s2$	32.8543	26.7549	**17.0545**	*43.4751*
$s3$	60.0337	45.9647	**38.7437**	*83.7691*
$s4$	*3.0344*	**1.3989**	1.5369	1.658
$s5$	8.4057	**6.0187**	7.8124	*12.8878*
$s6$	*24.1959*	**19.116**	20.5876	22.052
$s7$	*1.6726*	1.5196	1.6629	**0.9969**
$s8$	6.3794	8.2414	*9.7088*	**4.9328**
$s9$	*25.7472*	14.848	24.2464	**14.563**
$s10$	*1.5647*	1.3278	1.5456	**0.7286**
$s11$	*15.2379*	9.4222	11.9055	**3.5037**
$s12$	*24.9615*	20.7495	24.0163	**14.6451**

TABLE IV.
THE OPTIMAL RESULTS OF THE QPSO RQPSO, RRQPSO AND SRQPSO FOR GRIEWARK

$Method$	QPSO	RQPSO	RRQPSO	SRQPSO
$s1$	0.1433	0.1076	**0.0930**	*0.1537*
$s2$	0.0377	0.0340	**0.0338**	*0.0650*
$s3$	0.0882	0.0450	**0.0352**	*0.1286*
$s4$	*0.1029*	0.0848	**0.0816**	0.0820
$s5$	*0.0300*	0.0183	**0.0165**	0.0193
$s6$	0.0160	**0.0111**	0.0126	*0.0247*
$s7$	*0.0792*	0.0767	**0.0677**	0.0726
$s8$	*0.0194*	**0.0146**	0.0157	0.0170
$s9$	0.0102	0.0080	**0.0071**	*0.0140*
$s10$	*0.0798*	0.0698	0.0704	**0.0687**
$s11$	*0.0218*	0.0170	**0.01628**	0.01632
$s12$	*0.0141*	**0.0084**	0.0090	0.0092

SRQPSO gets one optimal value and five worst values, QPSO gets seven worst values.

TABLE V.
THE OPTIMAL RESULTS OF THE QPSO RQPSO, RRQPSO AND SRQPSO FOR SCHAFFER

$Method$	QPSO	RQPSO	RRQPSO	SRQPSO
$s1$	*6.23E-03*	2.28E-03	**2.27E-03**	3.73E-03
$s2$	*4.66E-03*	2.91E-03	**1.62E-03**	4.21E-03
$s3$	*4.28E-03*	1.46E-03	**1.00E-03**	3.08E-03
$s4$	*1.75E-03*	1.13E-03	**4.86E-04**	1.46E-03
$s5$	*2.53E-03*	4.88E-04	**1.63E-04**	1.62E-03
$s6$	*2.12E-03*	4.86E-04	**1.62E-04**	1.46E-03
$s7$	3.268E-04	1.62E-04	**5.99E-07**	*6.50E-04*
$s8$	3.24E-04	2.86E-04	**5.97E-07**	*8.10E-04*
$s9$	4.86E-04	*5.48E-04*	**8.94E-07**	1.62E-04
$s10$	1.64E-04	1.62E-04	**1.92E-07**	*4.86E-04*
$s11$	1.619E-04	1.619E-04	**4.75E-08**	*1.62E-04*
$s12$	*1.62E-04*	1.619E-04	**3.88E-08**	1.619E-04

In Table V, For Schaffer function, the results of RRQPSO are all of optimal value, but there are one worst result is in RQPSO, four worst results are in SRQPSO and seven worst results are in QPSO.

Based on these optimizing results of four testing functions with four methods from Table II $\sim$ Table V, among the 48 optimizing results of four testing functions, RRQPSO obtains 31 times of optimal values and RQPSO obtains 10 times, while QPSO gets 26 times of the worst values and SRQPSO gets 20 times. In the vast majority of parameter combinations, RRQPSO and RQPSO are better than QPSO, so QPSO can be improved by $RMP(t)$

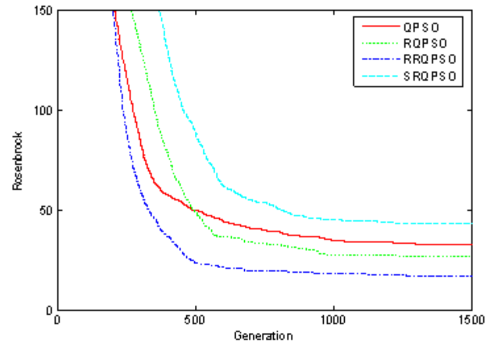

Figure 1. Evolution of average Minimum of Rosenbrock with 40 particles and 1500 iterations

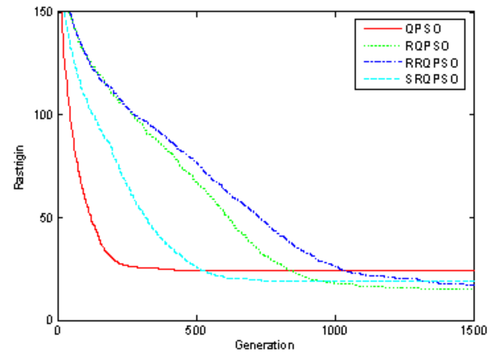

Figure 2. Evolution of average Minimum of Ratrigin with 40 particles and 1500 iterations

and $RRMP(t)$ to evaluate $L(t)$. Though SRQPSO gets 7 times of optimal values, it gets 20 times of the worst values. So its performance is just about right to QPSO, and it can't improve the performance of QPSO with $SRMP(t)$ to assess $L(t)$.

In order to better understand the iterative process of each algorithm on each testing function, Fig.1$\sim$ Fig.4 shows the four algorithms' iterating procedures for four testing functions during 1500 iterations with 40 particles. In Fig.1, RRQPSO has the fastest convergence speed and the highest convergence accuracy, and those of SRQPSO are the worst. Based on Fig.2, QPSO owns the fastest convergence speed, but it fall into local extremum point prematurely. In Fig.3, RQPSO and RRQPSO have good convergence precision. In Fig.4, RRQPSOhas the best convergence accuracy. According the iterations of four algorithms of four testing functions, RRQPSO and RQPSO are better than QPSO, and SRQPSO slightly inferior to QPSO.

V. CONCLUSIONS

In order to improve the convergence rate and convergence precision of QPSO, this paper proposed three evaluating programs to re-evaluate δ trap characteristic length $L(t)$ of wave function. Based on random weights and particles' $P_i(t)$, there are three evaluating positions: RMP, RRMP and SRMP, which are used to re-evaluate

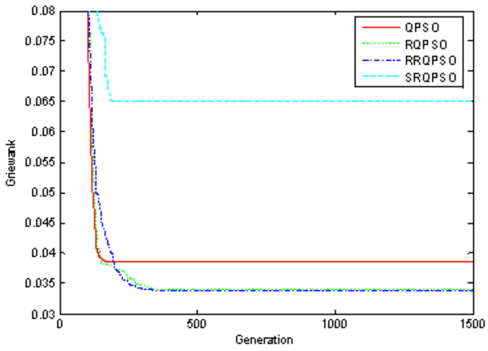

Figure 3. Evolution of average Minimum of Griewark with 40 particles and 1500 iterations

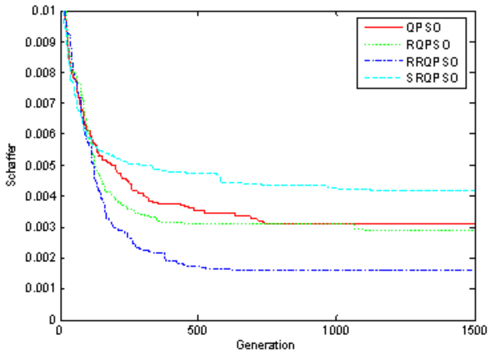

Figure 4. Evolution of average Minimum of Schaffer with 40 particles and 1500 iterations

$L(t)$ and whose corresponding algorithms are called as RQPSO, RRQPSO and SRQPSO. In order to assess the performance of the proposed algorithms, four nonlinear benchmark testing functions are resolved by these proposed algorithms. From the Experimental results, RQPSO and RRQPSO are far superior to QPSO and SRQPSO. Therefore, QPSO can be improved by using RMP or RRMP to evaluate $L(t)$.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

REFERENCES

[1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro machine and Human Science, Nagoya, Japan*, 1995, pp. 39–43.
[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ*, 1995, pp. 1942–1948.
[3] F. Van den Bergh and A. Engelbrecht, "A new locally convergent particle swarm optimizer," in *Proceeding of IEEE Conf. on Syst., Man and Cybernetics*, 2002, pp. 45–49.
[4] P. Shelokar, P. Siarry, and V. Jayaraman, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Appl. Math. and Comput.*, vol. 188, no. 1, pp. 129–142, 2007.
[5] Y. Jiang, T. Hu, and C. Huang, "An improved particle swarm optimization algorithm," *Appl. Math. and Comput.*, vol. 193, no. 1, pp. 231–239, 2007.
[6] D. Pan and Z. Liu, "An improved particle swarm optimization algorithm based on the optimal and sub-optimal positions," *Journal of Comput. Infor. Syst.*, vol. 8, no. 5, pp. 1987–1994, 2011.
[7] X. Zhao, "A perturbed particle swarm algorithm for numerical optimization," *Appl. Soft. Comput.*, vol. 10, no. 1, pp. 119–124, 2010.
[8] D. Gong, J. Zhang, and Y. Zhang, "Multi-objective particle swarm optimization for robot path planning in environment with danger sources," *Journal of Computers*, vol. 6, no. 8, pp. 1554–1561, 2011.
[9] J. Zhai and K. Wang, "Baldwin effect based particle swarm optimizer for multimodal optimization," *Journal of Computers*, vol. 7, no. 9, pp. 2114–2119, 2012.
[10] J. Sun and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *proceedings of IEEE Congr. Evol. Comput., Portland,USA*, 2004, pp. 325–331.
[11] K. Yang, K. Maginu, and H. Nomura, "Cultural algorithm-based quantum-behaved particle swarm optimization," *J. comput. Math.*, vol. 87, no. 10, pp. 2143–2157, 2010.
[12] X. Wang, H. Long, and J. Sun, "Quantum-behaved particle swarm optimization based on gaussian disturbance," *Appl. Res. Comput.*, vol. 27, no. 9, pp. 2093–2096, 2010.
[13] H. Su, "Simulated annealing based quantum-behaved particle swarm optimization," *J. Comput. Inf. Syst.*, vol. 6, no. 6, pp. 1795–1800, 2010.
[14] D. Zhou, J. Sun, C. Lai, and etc., "An improved quantum-behaved particle swarm optimization and its application to medical image registration," *J. comput. Math.*, vol. 88, no. 6, pp. 1208–12 239, 2011.
[15] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the IEEE Congr. Evol. Comput., Piscataway, NJ*, 1999, pp. 1945–1950.
[16] J. Sun, "Particle swarm optimization with particles having quantum behavior," in *Jiangnan University*, 2009.

Dazhi Pan received his B.S. degree in applied mathematics from China West Normal University, Nanchong, China in June 1996 and his M.S. degree in computer applications from Kunming University of Science and Technology, Kunming, China in June 2003 and Ph.D. degree in petroleum engineering computation, Southwest Petroleum University, Chengdu, China in January 2012. He is a professor at the school of Mathematic and Information, China West Normal University, China. His current research interest includes swarm intelligence and algorithm analysis and design.

Ying Ci received the B.S. degree in information and computing science from China West Normal University, Nanchong, China, in 2003, and M.S. degree in computer applications from Southwest Jiaotong University, Chengdu, China in 2006. She is a lecturer at the school of Mathematic and Information, China West Normal University, China. Her research interests include intelligent computing and information security.

Min He is currently a college student and working towards her B.S. degree at China West Normal University, China. Her current research interest includes intelligent computing and algorithm analysis and design.