A New Quantum Behaved Particle Swarm Optimization

Millie Pant
Department of Paper Technology
IIT Roorkee, Saharanpur
India
millifpt@iitr.ernet.in

Radha Thangaraj
Department of Paper Technology
IIT Roorkee, Saharanpur
India
t.radha@ieee.org

Ajith Abraham
Q2S, Norwegian Center of Excellence
NTNU, Trondheim
Norway
ajith.abraham@ieee.org

ABSTRACT

This paper presents a variant of Quantum behaved Particle Swarm Optimization (QPSO) named Q-QPSO for solving global optimization problems. The Q-QPSO algorithm is based on the characteristics of QPSO, and uses interpolation based recombination operator for generating a new solution vector in the search space. The performance of Q-QPSO is compared with Basic Particle Swarm Optimization (BPSO), QPSO and two other variants of QPSO taken from literature on six standard unconstrained, scalable benchmark problems. The experimental results show that the proposed algorithm outperforms the other algorithms quite significantly.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: Language Contructs and Features – *abstract data types, polymorphism*

General Terms

Algorithms, Performance, Reliability, Experimentation

Keywords

Particle swarm optimization, Interpolation, Global optimization, Quantum behavior.

1. INTRODUCTION

Particle Swarm Optimization (PSO) is relatively a newer addition to a class of population based search technique for solving numerical optimization problems. The particles or members of the swarm fly through a multidimensional search space looking for a potential solution.

In classical (or original PSO), developed by Kennedy and Eberhart in 1995 [1], each particle adjusts its position in the search space from time to time according to the flying experience of its own and of its neighbors (or colleagues).

For a D-dimensional search space the position of the i^{th} particle is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. Each particle maintains a memory of its previous best position $P_{besti} = (p_{i1}, p_{i2} \ldots p_{iD})$.

The best one among all the particles in the population is represented as $P_{gbest} = (p_{g1}, p_{g2} \ldots p_{gD})$. The velocity of each particle is represented as $V_i = (v_{i1}, v_{i2}, \ldots v_{iD})$. In each iteration, the P vector of the particle with best fitness in the local neighborhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity. The two basic equations which govern the working of PSO are that of velocity vector and position vector given by:

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

The first part of equation (1) represents the inertia of the previous velocity, the second part is the cognition part and it tells us about the personal experience of the particle, the third part represents the cooperation among particles and is therefore named as the social component. Acceleration constants c_1, c_2 and inertia weight w are the predefined by the user and r_1, r_2 are the uniformly generated random numbers in the range of [0, 1].

PSO has undergone a plethora of changes since its development. One of the recent developments in PSO is the application of Quantum laws of mechanics to observe the behavior of PSO. Such PSO's are called Quantum PSO (QPSO). Some variants of QPSO include mutation based PSO [2], [3], where mutation is applied to the mbest (mean best) and gbest (global best) positions of the particle, also in one of the variants of QPSO a perturbation constant called Lyapunov constant is added. However to the best of our knowledge no has used the concept of recombination operator in QPSO.

This paper presents a QPSO called Q-QPSO which uses the quantum theory of mechanics to govern the movement of swarm particles along with an interpolation (quadratic interpolation) based recombination operator.

The concept of quadratic interpolation as a recombination operator was introduced by us [4], [5], for improving the performance of classical PSO, where it gave significantly good results. This motivated us to apply this concept for QPSO also to improve its performance.

The remaining of the paper is organized as follows: Section 2 briefly describes the Quantum Particle Swarm Optimization. Section 3, explains the proposed Q-QPSO, Section 4, gives the experimental settings and numerical results of some selected unconstrained benchmark problems. The paper finally concludes with Section 5.

2. QUANTUM PARTICLE SWARM OPTIMIZATION

The development in the field of quantum mechanics is mainly due to the findings of Bohr, de Broglie, Schrödinger, Heisenberg and Bohn in the early twentieth century. Their studies forced the scientists to rethink the applicability of classical mechanics and the traditional understanding of the nature of motions of microscopic objects [6].

As per classical PSO, a particle is stated by its position vector x_i and velocity vector v_i, which determine the *trajectory* of the particle. The particle moves along a determined trajectory following Newtonian mechanics. However if we consider quantum mechanics, then the term trajectory is meaningless, because x_i and v_i of a particle cannot be determined simultaneously according to *uncertainty principle*.

Therefore, if individual particles in a PSO system have quantum behavior, the performance of PSO will be far from that of classical PSO [7].

In the quantum model of a PSO, the state of a particle is depicted by wavefunction $\Psi(x,t)$, instead of position and velocity. The dynamic behavior of the particle is widely divergent from that of the particle in traditional PSO systems. In this context, the probability of the particle's appearing in position x_i from probability density function $|\Psi(x,t)|^2$, the form of which depends on the potential field the particle lies in [2].

The particles move according to the following iterative equations [8], [9]:

$$x(t+1) = p + \beta * |mbest - x(t)| * \ln(1/u) \;\; if \;\; k \geq 0.5$$
$$x(t+1) = p - \beta * |mbest - x(t)| * \ln(1/u) \;\; if \;\; k < 0.5 \qquad (3)$$

where

$$p = (c_1 p_{id} + c_2 P_{gd})/(c_1 + c_2) \qquad (4)$$

$$mbest = \frac{1}{M}\sum_{i-1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \ldots, \frac{1}{M}\sum_{i=1}^{n} P_{id}\right) \qquad (5)$$

Mean best (mbest) of the population is defined as the mean of the best positions of all particles, u, k, c1 and c2 are uniformly distributed random numbers in the interval [0, 1]. The parameter β is called contraction-expansion coefficient. The flow chart of QPSO algorithm is shown in Figure 1.

```
Initialize the Swarm
Do
Calculate mbest by equation (5)
Update particles position using equation (3)
Update Pbest
Update Pgbest.
While maximum iteration is reached.
```

Figure 1 Flow of QPSO Algorithm

3. PROPOSED Q-QPSO ALGORITHM

The proposed Q-QPSO algorithm is a simple and modified version of QPSO in which we have introduced the concept of recombination. The Q-QPSO algorithm starts like the usual QPSO using equations (3), (4) and (5). At the end of each iteration, the quadratic interpolation recombination operator is invoked to generate a new swarm particle. The new particle is accepted in the swarm only if it is better than the worst particle (i.e. the particle having maximum fitness) present in the swarm. This process is repeated iteratively until a better solution is obtained.

The quadratic crossover operator is a nonlinear operator which produces a new solution vector lying at the point of minima of the quadratic curve passing through the three selected swarm particles.

The selection of particles, say {a, b, c} is done as follows:

a = X_{min}, (the swarm particle having minimum (or best) fitness function value)

{b, c} = {randomly chosen particles from the remaining members of the swarm.

(a, b and c should be three different particles)

The idea behind the inclusion of an interpolation operator is to facilitate the Q-QPSO with a recombination operator which will help in finding a new solution point in the search space. Since, we are always holding the particle having the best fitness function value to take part in recombination process, the probability of generating a new trial vector with fitness function which is better than at least one of the existing solution vectors in the swarm increases. Thus the new particle is accepted in the swarm only if it is better than the worst particle.

Mathematically, the new particle $\tilde{x}^i = (\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^n)$, is given as:

$$\tilde{x}^i = \frac{1}{2}\frac{({b^i}^2 - {c^i}^2)*f(a) + ({c^i}^2 - {a^i}^2)*f(b) + ({a^i}^2 - {b^i}^2)*f(c)}{(b^i - c^i)*f(a) + (c^i - a^i)*f(b) + (a^i - b^i)*f(c)} \qquad (6)$$

The computational steps of Q-QPSO algorithm are given by:

Step 1: Initialize the swarm with uniformly distributed random numbers.

Step 2: Calculate mbest using equation (5)

Step 3: Update particles position using equation (3)

Step 4: Evaluate the fitness value of each particle

Step 5: If the current fitness value is better than the best fitness value (Pbest) in history

Then Update Pbest by the current fitness value.

Step 6: Update Pgbest (global best)

Step 7: ***Find a new particle using equation (6)***

Step 8: ***If the new particle is better than the worst particle in the swarm***

Then replace the worst particle by the new particle.

Step 9: Go to step 2 until maximum iterations reached.

4. EXPERIMENTAL SETTINGS AND BENCHMARK PROBLEMS

In Q-QPSO algorithm a linearly decreasing contraction-expansion coefficient (β) is used which starts at 1.0 and ends at 0.5. the acceleration constants are taken as 2.0. In order to check the compatibility of the proposed Q-QPSO algorithm we have tested it on 10 benchmark problems (unconstrained) given in Table 1. All the test problems are highly multimodal and scalable in nature and are considered to be starting point for checking the credibility of any optimization algorithm.

There is not much literature available in which QPSO is used extensively for solving a variety of test problems. Therefore, for the present study, we have considered 10 test problems out of which the first three problems are the ones that have been tested with some variants of QPSO. The remaining 6 problems we have solved with our version and with QPSO and BPSO. As in [3], for functions f1, f2 and f3, three different dimension sizes are tested. They are 10, 20 and 30. The maximum number of generations is set as 1000, 1500 and 2000 corresponding to the dimensions 10, 20 and 30 respectively. Different population sizes are used for each function with different dimensions. The population sizes are 20, 40 and 80. We have tested the functions f4 – f10 with dimensions 10, 30 and 50. A total of 30 runs for each experimental setting are conducted and the average fitness of the best solutions throughout the run is recorded.

Tables 2, 3 and 4 show the mean best fitness of Q-QPSO, BPSO, QPSO and its two variants in literature for functions f1, f2 and f3 respectively. Table 5 shows the mean best fitness values of Q-QPSO, BPSO and QPSO for the functions f4 – f10. Tables 6, 7 and 8 shows the T-test values for the benchmark problems f1, f2 and f3 in comparison with the other algorithms. Figures 2, 3 and 4 depict the performance with a focus on mean best fitness for some selected functions. In all the Tables, 'Pop' represents population, 'Dim' represents dimension and 'Gen' represents Generation.

Table 1. Numerical benchmark problems

Function	Range	Optimum
$f_1(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	[2.56,5.12]	0
$f_2(x)=\frac{1}{4000}\sum_{i=0}^{n-1}x_i^2-\prod_{i=0}^{n-1}\cos(\frac{x_i}{\sqrt{i+1}})+1$	[300,600]	0
$f_3(x)=\sum_{i=0}^{n-1}100(x_{i+1}-x_i^2)^2+(x_i-1)^2$	[15,30]	0
$f_4(x)=-\sum_{i=1}^{n}x_i\sin(\sqrt{\lvert x_i\rvert})$	[-500,500]	-418.9829*n
$f_5(x)=(\sum_{i=0}^{n-1}(i+1)x_i^4)+rand[0,1]$	[-1.28,1.28]	0
$f_6(x)=20+e-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))$	[-32,32]	0
$f_7(x)=\max\lvert x_i\rvert,\quad 0\le i<n$	[-100,100]	0
$f_8(x)=\sum_{i=0}^{n-1}\lfloor x_i+1/2\rfloor^2$	[-100,100]	0
$f_9(x)=\sum_{i=0}^{n-1}\lvert x_i\rvert+\prod_{i=0}^{n-1}x_i$	[-10,10]	0
$f_{10}(x)=\sum_{i=0}^{n-1}(\sum_{j=0}^{i}x_i)^2$	[-100,100]	0

Table 2. Comparison results of function f1 (Mean best)

Pop	Dim	Gen	Q-QPSO	BPSO [2]	QPSO [2]	QPSO Mutation [3]		QPSO Mutation [2]	
						Gbest	mbest	gbest	mbest
20	10	1000	**4.915e-19**	5.5382	5.2543	5.2216	4.4788	4.3976	4.7332
	20	1500	**7.806e-19**	23.1544	16.2673	16.1562	15.6535	14.1678	13.6202
	30	2000	**6.071e-19**	47.4168	31.4576	26.2507	27.8098	25.6415	27.7975
40	10	1000	**6.550e-13**	3.5778	3.5685	3.3361	3.3837	3.2046	2.8160
	20	1500	**8.673e-19**	16.4337	11.1351	10.9072	11.0128	9.5793	9.9143
	30	2000	**5.493e-19**	37.2896	22.9594	19.6360	21.0179	20.5479	19.8991
80	10	1000	**0.86794**	2.5646	2.1245	2.0185	2.1833	1.7166	1.8923
	20	1500	**0.97712**	13.3826	10.2759	7.7928	8.0755	7.2041	7.8625
	30	2000	**5.493e-19**	28.6293	16.7768	14.9055	14.9965	15.0393	15.4082

Table 3. Comparison results of function f2 (Mean best)

Pop	Dim	Gen	Q-QPSO	BPSO[2]	QPSO[2]	QPSO Mutation [3]		QPSO Mutation[2]	
						gbest	mbest	gbest	mbest
20	10	1000	**0.062657**	0.09217	0.08331	0.0627	0.0732	0.0780	0.0932
	20	1500	**0.005091**	0.03002	0.02033	0.0209	0.0189	0.0235	0.0193
	30	2000	0.015442	0.01811	0.01119	0.0110	0.0103	**0.0099**	0.0114
40	10	1000	0.057393	0.08496	0.06912	0.0539	**0.0520**	0.0641	0.0560
	20	1500	**0.005827**	0.02719	0.01666	0.0238	0.0247	0.0191	0.0171
	30	2000	**0.007874**	0.01267	0.01161	0.0119	0.0105	0.0098	0.0092
80	10	1000	**0.031527**	0.07484	0.03508	0.0419	0.0542	0.0460	0.0554
	20	1500	**0.006633**	0.02854	0.01460	0.0136	0.0194	0.0186	0.0123
	30	2000	**0.006648**	0.01258	0.01136	0.0120	0.0082	0.0069	0.0111

Table 4. Comparison results of function f3 (Mean best)

Pop	Dim	Gen	Q-QPSO	BPSO [2]	QPSO [2]	QPSO Mutation [3]		QPSO Mutation [2]	
						Gbest	mbest	gbest	mbest
20	10	1000	**5.544203**	94.1276	59.4764	27.4620	22.1870	21.2081	15.3939
	20	1500	**15.538104**	204.336	110.664	49.1176	68.4096	61.9268	67.6978
	30	2000	**25.687072**	313.734	147.609	97.5952	113.3080	86.1195	76.1894
40	10	1000	**4.200496**	71.0239	10.4238	7.8741	7.9850	8.1828	9.5005
	20	1500	**14.158022**	179.291	46.5957	28.4435	52.9333	40.0749	55.4853
	30	2000	**24.126324**	289.593	59.0291	62.3854	64.1942	65.2891	68.0551
80	10	1000	**2.893087**	37.3747	8.63638	6.7098	5.7159	7.3686	6.4841
	20	1500	**12.033052**	83.6931	35.8947	31.0929	24.4566	30.1607	38.3067
	30	2000	**22.426013**	202.672	51.5479	43.7622	45.2270	38.3036	52.4678

Table 5. Comparison results of functions f4 – f10 (Mean best)

Function	Dim	Gen	BPSO	QPSO	Q-QPSO
f4	10	1000	-2389.365	-3871.03	**-3898.67**
	30	2000	-6466.188	-8967.29	**-9998**
	50	3000	-10473.09	-13105.9	**-14783.6**
f5	10	1000	0.502671	0.452975	**0.376021**
	30	2000	0.617222	0.501799	**0.497801**
	50	3000	0.788322	0.598823	**0.537782**
f6	10	1000	6.965e-12	4.407e-015	**1.210e-015**
	30	2000	3.618e-05	7.568e-012	**5.828e-015**
	50	3000	3.43866	1.018e-007	**6.267e-014**
f7	10	1000	3.79373e-006	1.09816e-09	**7.93766e-013**
	30	2000	7.836	4.11969	**0.00571485**
	50	3000	27.8394	21.1619	**0.2213**
f8	10	1000	0.00000	0.00000	**0.00000**
	30	2000	0.05	0.00000	**0.00000**
	50	3000	1.7	0.1	**0.00000**
f9	10	1000	2.4148e-14	2.02058e-32	**3.32734e-34**
	30	2000	2.04745e-07	3.2466e-13	**7.94348e-16**
	50	3000	7.73949	1.67575e-08	**8.66751e-11**
f10	10	1000	2.51528e-22	5.51618e-54	**1.00925e-62**
	30	2000	1.34381e-07	1.44342e-21	**2.9990e-27**
	50	3000	0.00176764	1.06795e-10	**3.78622e-17**

Table 6. T-test* value for the function f1: comparison of Q-QPSO with other algorithms

Pop	Dim	Gen	BPSO [2]	QPSO [2]	QPSO Mutation [3]		QPSO Mutation [2]	
					gbest	mbest	gbest	mbest
20	10	1000	9.9530	9.9402	13.5224	10.3788	9.5110	9.8434
	20	1500	12.1083	14.9068	10.8334	8.9853	15.7493	13.3406
	30	2000	15.1352	22.4110	16.9158	7.6914	21.0956	20.911
40	10	1000	9.1640	9.4523	7.7164	6.8459	5.7384	8.1806
	20	1500	16.4220	16.9198	11.2975	12.5773	18.6672	16.8778
	30	2000	14.2989	17.3561	25.1958	7.4558	22.4234	24.0754
80	10	1000	2.5229	1.9482	1.69199	1.8635	1.2992	1.5181
	20	1500	7.2362	6.5989	6.6325	9.2566	7.2967	7.7037
	30	2000	15.1607	20.4847	14.3885	19.2611	19.7066	18.6037

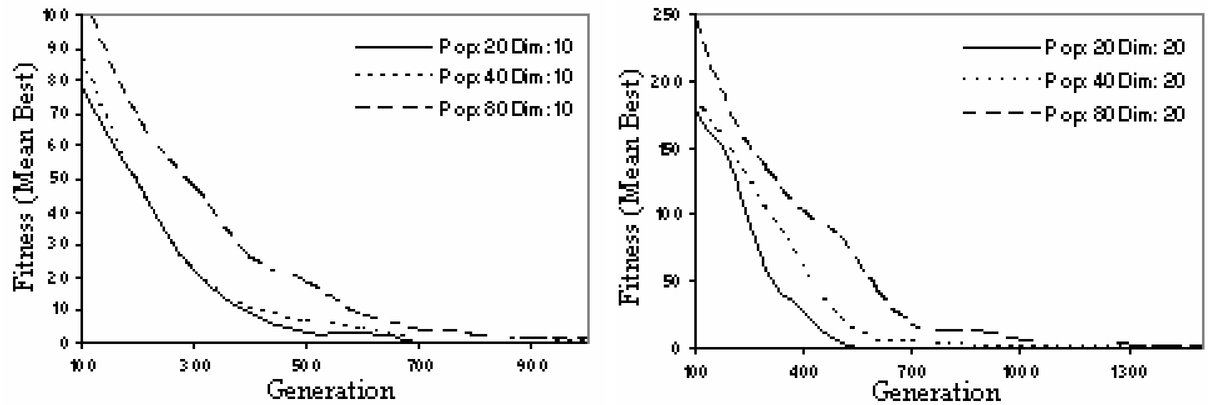
Table 7. T-test* value for the function f2: comparison of Q-QPSO with other algorithms

Pop	Dim	Gen	BPSO [2]	QPSO [2]	QPSO Mutation[3]		QPSO Mutation [2]	
					gbest	mbest	gbest	mbest
20	10	1000	0.6204	0.4417	0.00093	0.2277	0.3304	0.6398
	20	1500	4.2078	3.6522	4.2408	4.2339	4.6468	5.1432
	30	2000	0.2525	-0.4289	-0.4429	-0.523	-0.568	-0.4077
40	10	1000	0.8184	0.3627	-0.1081	-0.1695	0.2101	-0.0435
	20	1500	4.5453	3.2313	4.8420	2.2757	4.2572	3.3365
	30	2000	1.6119	1.4395	1.2345	1.0282	0.6970	0.5324
80	10	1000	1.9956	0.1995	0.5749	1.1770	0.8125	1.2151
	20	1500	4.2989	2.9130	2.2932	3.2294	3.6094	2.0360
	30	2000	2.1889	2.0719	1.7473	0.5976	0.1293	1.7143

Table 8. T-test* value for the function f3: comparison of Q-QPSO with other algorithms

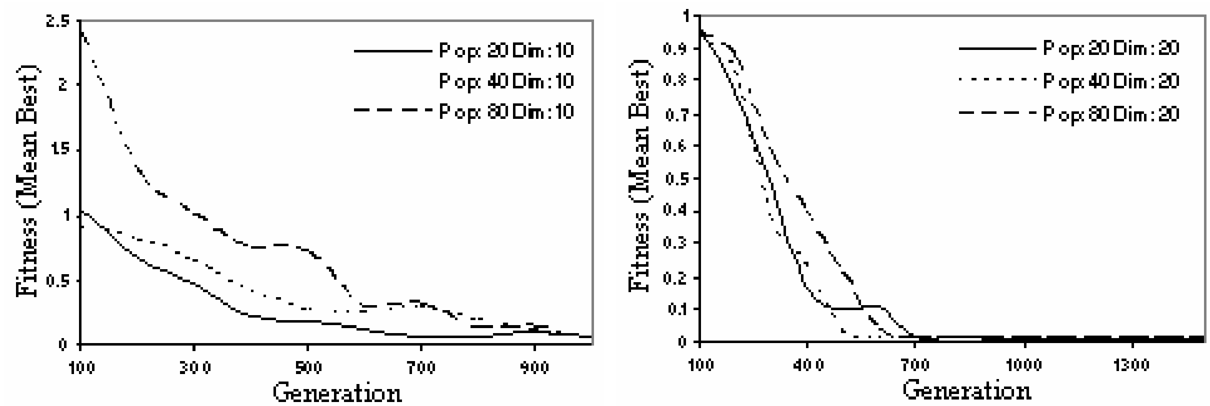
Pop	Dim	Gen	BPSO [2]	QPSO [2]	QPSO Mutation [3]		QPSO Mutation [2]	
					gbest	mbest	gbest	mbest
20	10	1000	2.4962	1.9296	2.3366	1.6658	1.4285	1.5366
	20	1500	3.5238	3.4839	3.7043	2.9247	2.7336	2.5909
	30	2000	2.8828	3.1750	2.7207	3.1501	2.5931	2.3971
40	10	1000	2.1021	2.3534	1.9774	2.3377	2.6069	3.0377
	20	1500	2.3963	4.4938	2.6616	3.4119	2.0751	3.5739
	30	2000	3.0378	3.0108	3.8951	4.1699	2.8380	4.4003
80	10	1000	3.2847	1.8772	2.5347	2.4382	2.8315	3.2502
	20	1500	2.8594	3.5828	3.2790	2.5388	2.9890	3.7068
	30	2000	3.4046	3.9045	3.6288	3.8771	3.1658	4.1947

* The t value of 29 degrees of freedom is significant at a level a 0.05 level of significance by a two-tailed t-test



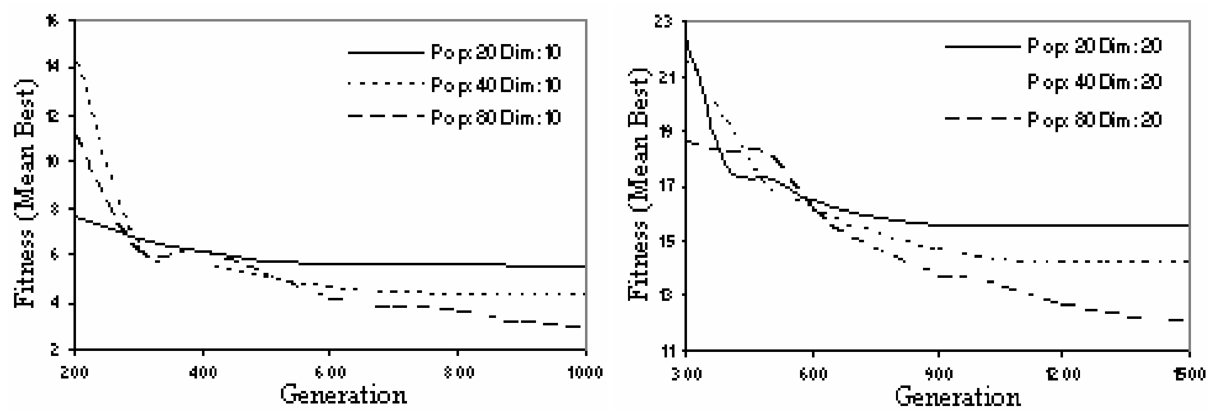
(a) **(b)**

Figure 2. Performance for Rastringin function (a) dimension 10 (b) dimension 20



(a) **(b)**

Figure 3. Performance for Griewank function (a) dimension 10 (b) dimension 20



(a) **(b)**

Figure 4. Performance for Rosenbrock function (a) dimension 10 (b) dimension 20

5. CONCLUSIONS

This paper presents a variant of Quantum PSO called Q-QPSO, incorporating the concept of recombination operator. The proposed Q-QPSO is tested on three standard unconstrained benchmark test problems and the results are compared with some of the existing QPSO (containing mutation operator) and standard PSO.

For the first test problem, which is, Rastringin's function (a highly multimodal function), the proposed Q-QPSO algorithm outperformed all the given algorithms, quite significantly and gave the near optimum solution (which is 0) in all the test cases. Similarly for the second test problem (Griewank function), Q-QPSO gave better results than the other algorithms in seven out of the nine test cases tried. For the test functions f3 – f10 also Q-QPSO gave superior results in all the cases. The dominance of the proposed Q-QPSO algorithm is also apparent from the two tailed t-test given in Tables 5, 6 and 7.

However, we would like to add that the collection of test problems taken in this paper is not exhaustive and therefore making any concrete conclusion on the performance of Q-QPSO do not sound justified. Moreover, it is very much possible that the inclusion of a crossover operator may have an adverse effect on the diversity of the algorithm which in turn may deteriorate its performance for solving problems having large number of variables. Also, it would be interesting to do a theoretical analysis on quadratic interpolation operator and its application to optimization problems.

However, on the positive side it is quite evident from the numerical results that for the highly multimodal problems having variables up to 50, Q-QPSO is definitely a better choice over the contemporary optimization algorithms.

For the future work, we shall be apply Q-QPSO for solving more complex unconstrained and constrained optimization problems. Also we shall be studying the combined effect of mutation and recombination on a Quantum behaved PSO.

6. REFERENCES

[1] Kennedy, J. and Eberhart, R. Particle Swarm Optimization. IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948, 1995.

[2] Liu J, Sun J, Xu W, Quantum-Behaved Particle Swarm Optimization with Adaptive Mutation Operator. ICNC 2006, Part I, Springer-Verlag: 959 – 967, 2006.

[3] Liu J, Xu W, Sun J. Quantum-Behaved Particle Swarm Optimization with Mutation Operator. In Proc. of the 17th IEEE Int. Conf. on Tools with Artificial Intelligence, Hong Kong (China), 2005.

[4] Millie Pant, Radha Thangaraj and Ajith Abraham, A New PSO Algorithm with Crossover Operator for Global Optimization Problems, Second International Symposium on Hybrid Artificial Intelligent Systems (HAIS'07), Advances in Softcomputing Series, Springer Verlag, Germany, E. Corchado et al. (Eds.): Innovations in Hybrid Intelligent Systems, Vol. 44, pp. 215-222, 2007.

[5] Millie Pant, Radha Thangaraj and Ajith Abraham, A New Particle Swarm Optimization Algorithm Incorporating Reproduction Operator for Solving Global Optimization Problems, 7th International Conference on Hybrid Intelligent Systems, Kaiserslautern, Germany, IEEE Computer Society press, USA, ISBN 07695-2662-4, pp. 144-149, 2007.

[6] Pang XF, Quantum mechanics in nonlinear systems. River Edge (NJ, USA): World Scientific Publishing Company, 2005.

[7] Bin Feng, Wenbo Xu, Adaptive Particle Swarm Optimization Based on Quantum Oscillator Model. In Proc. of the 2004 IEEE Conf. on Cybernetics and Intelligent Systems, Singapore: 291 – 294, 2004.

[8] Sun J, Feng B, Xu W, Particle Swarm Optimization with particles having Quantum Behavior. In Proc. of Congress on Evolutionary Computation, Portland (OR, USA), 325 – 331, 2004.

[9] Sun J, Xu W, Feng B, A Global Search Strategy of Quantum-Behaved Particle Swarm Optimization. In Proc. of the 2004 IEEE Conf. on Cybernetics and Intelligent Systems, Singapore: 291 – 294, 2004.