Sobol Mutated Quantum Particle Swarm Optimization

Millie Pant, Radha Thangaraj and V. P. Singh
Department of Paper Technology,
Indian Institute of Technology Roorkee, India.
millifpt@iitr.ernet.in, t.radha@ieee.org, singhfpt@iitr.ernet.in

Abstract—**This paper presents a new mutation operator called the Sobol Mutation (SOM) operator for enhancing the performance of Quantum Particle Swarm Optimization (QPSO) algorithm. The SOM operator unlike most of its contemporary mutation operators do not use the random probability distribution for perturbing the swarm population, but uses a quasi random Sobol sequence to find new solution vectors in the search domain. The proposed version is called Sobol Mutation for quantum inspired PSO (SOM-QPSO) and its comparison is made with Basic Particle Swarm Optimization (BPSO), QPSO and some other variants of QPSO. The empirical results show that SOM operator significantly improves the performance of QPSO.**

Index Terms—**Particle Swarm Optimization, Mutation, Quantum behavior, Sobol sequence.**

I. INTRODUCTION

Particle Swarm Optimization (PSO) is relatively a newer addition to a class of population based search technique for solving numerical optimization problems. Metaphorically, PSO imitates the collective and cooperative behavior of species moving in groups. Some classic examples being a swarm of birds, school of fish, cooperative behavior of ants and bees etc.

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].

Researchers have shown that although PSO finds solutions much faster than most of the contemporary search techniques like Evolutionary and Genetic Algorithms, it usually do not improve the quality of solutions as the number of iterations increase and thus becomes a victim of premature convergence resulting in a suboptimal solution. This draw back of PSO is due to the lack of diversity, which forces the swarm particles to converge to the global optimum found so far (after a certain number of iterations), which may not even be a local optimum. Thus without an effective diversity enhancing mechanism the optimization algorithm technique is not able to efficiently explore the search space. One of the methods for maintaining the diversity of the population is inclusion of the concept of mutation (a phenomenon borrowed from Evolutionary Algorithms). Most of the modern mutation operators defined in literature make use of random probability distribution (for example Gaussian mutation [2], Cauchy mutation [3] etc).

In the present article we have considered the quantum version of PSO denoted generally as QPSO. This is a recent modification in the field of PSO and is based on the uncertainty principle that the position and velocity of the particle cannot be determined simultaneously. In this article we have included the concept of mutation in the QPSO, but instead of using the random probability distribution, we have defined a Sobol Mutation operator called SOM which uses quasi random (Sobol) sequence mainly because quasi random sequences cover the search domain more evenly in comparison to the random probability distributions, thereby increasing the chances of finding a better solution. The SOM operator defined in this paper is applied to two versions of QPSO called SOM-QPSO1 and SOM-QPSO2. In SOM-QPSO1, mutation is applied to the global best (gbest) particle, where as in SOM-QPSO2, the worst particle of the swarm is mutated. The concept of sobol mutation is already used by Pant et al [4], but there they applied it to the basic PSO, whereas the present study discusses the behavior of sobol mutation in quantum based PSO.

The remaining organization of the paper is as follows: section II gives a brief review of Quasi Random Sequences (QRS) and Sobol Sequence. Section III describes the Quantum Particle Swarm Optimization. In Section IV, we give the proposed algorithms, Section V, gives the experimental settings and numerical results of some selected unconstrained benchmark problems. The paper finally concludes with Section VI.

II. Quasi Random Sequences (QRS)

QRS or low discrepancy sequences are less random than pseudorandom number sequences, but are more useful for computational methods, which depend on the generation of random numbers. Some of these tasks involve approximation of integrals in higher dimensions, simulation and global optimization. Some well known QRS are: Vander Corput, Sobol, Faure and Halton. These sequences have been applied to initialize the swarm and the numerical results show a marked improvement over the traditional PSO, which uses uniformly, distributed random numbers [5], [6]**.**

QRS are said to be better than pseudo random sequences, because of their ability to cover the search space more evenly in comparison to pseudo random sequences (see Figures 1 and 2).

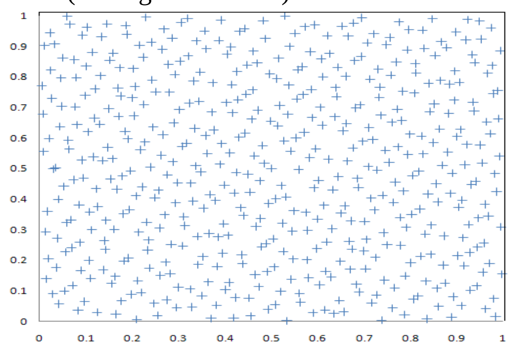


Fig 1 Sample points generated using a Sobol sequence

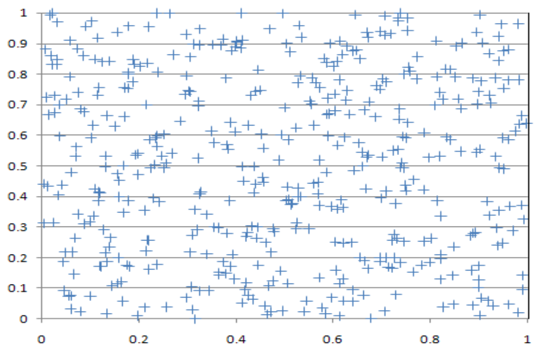


Fig 2 Sample points generated using a pseudo random sequence

A. Discrepancy of a Sequence

Mathematically, discrepancy of a sequence is the measure of its uniformity. It is computed by comparing the actual number of sample points in a given volume of a multi-dimensional space with the number of sample points that should be there assuming a uniform distribution defined.

For a given set of points $x^1, x^2, \ldots, x^N \in I^S$ and a subset $G \subset I^S$, define a counting function $S_N(G)$ as the number of points $x^i \in G$. For each $x = (x_1, x_2, \ldots x_S) \in I^S$, let G_x be the rectangular S dimensional region, such that $G_x = [0,x_1) \times [0,x_2) \ldots \times [0,x_S)$, with volume $x_1 x_2 \ldots x_N$. Then the discrepancy of points is given by $D^*_N(x^1, x^2, x^3 \ldots x^N) = \text{Sup} \, |S_N(G_x) - N x_1 x_2 \ldots x_S|, \, x \in I^S$.

Sobol Sequence

The construction of the Sobol sequence [7] uses linear recurrence relations over the finite field, F2, where F2 = {0, 1}. Let the binary expansion of the non-negative integer n be given by $n = n_1 2^0 + n_2 2^1 + \ldots.. + n_w 2^{w-1}$. Then the nth element of the jth dimension of the Sobol Sequence, $X_n^{(j)}$, can be generated by:

$$X_n^{(j)} = n_1 v_1^{(j)} \oplus n_2 v_2^{(j)} \oplus \ldots\ldots \oplus n_w v_w^{(j)} \qquad (3)$$

where $v_i^{(j)}$ is a binary fraction called the ith direction number in the jth dimension.

These direction numbers are generated by the following q-term recurrence relation:

$$v_i^{(j)} = a_1 v_{i-1}^{(j)} \oplus a_2 v_{i-2}^{(j)} \oplus \ldots \oplus a_q v_{i-q+1}^{(j)} \oplus v_{i-q}^{(j)} \oplus (v_{i-q}^{(j)} / 2^q) \qquad (4)$$

We have i > q, and the bit a_i, comes from the coefficients of a degree-q primitive polynomial over F2.

III. 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 [8].

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 [9].

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 [10].

The particles move according to the following iterative equations [11], [12]:

$$x(t+1) = p + \beta * |mbest - x(t)| * \ln(1/u) \ \ if \ \ k \geq 0.5$$
$$x(t+1) = p - \beta * |mbest - x(t)| * \ln(1/u) \ \ if \ \ k < 0.5 \quad (5)$$

where

$$p = (c_1 p_{id} + c_2 P_{gd})/(c_1 + c_2) \quad (6)$$

$$mbest = \frac{1}{M}\sum_{i-1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}, ..., \frac{1}{M}\sum_{i=1}^{n} P_{id}\right) \quad (7)$$

Mean best (mbest) of the population is defined as the mean of the best positions of all particles, u, k, c_1 and c_2 are uniformly distributed random numbers in the interval [0, 1]. The parameter β is called contraction-expansion coefficient.

IV. Proposed Algorithm

The proposed algorithm is an extension to the Basic Particle Swarm Optimization, by including the component of mutation in it. The mutation operator defined in the present work uses quasi random Sobol sequence and is called a Sobol mutation (SOM) operator. We have proposed two versions using SOM, called SOM-QPSO1 and SOM-QPSO2. The two versions differ from each other in the sense that in SOM-QPSO1, the best particle of the swarm is mutated, whereas in SOM-QPSO2, the worst particle of the swarm is mutated.

The SOM operator is defined as

SOM = R_1 + (R_2 / ln R_1),

Where R1 and R2 are random numbers in a Sobol sequence.

The idea behind applying the mutation to the worst particle is to push the swarm from the back. The quasi random numbers used in the SOM operator allows the worst particle to move forward systemically.

A. Pseudo code of SOM-QPSO Algorithms

The Pseudo code of SOM-QPSO1 is described as follows:

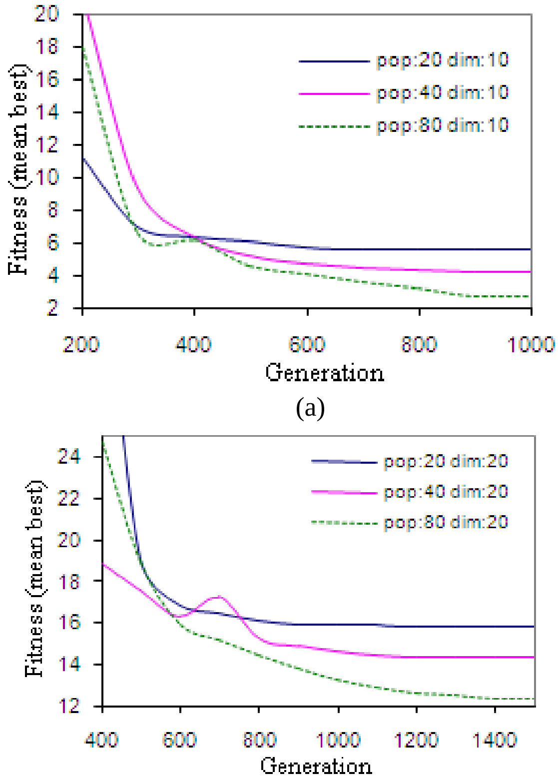
```
Initialize the population
Do
    β linearly decreases from 1.0  to 0.5
    For i=1 to population size M
        For d=1 to dimension D
    x_id=p+ β*|mbest-x_id|*ln(1/u)   if k ≥ 0.5
    x_id=p+ β*|mbest-x_id|*ln(1/u)   if k < 0.5
        End for
    If  ( f(Xi) < f(Pi) )    Pi = Xi.
    If  ( f(Pi) < f(Pg) )    Pg = Pi
    End if
    End if
          For d=1 to dimension D
       R1 = sobolrand();  R2 = sobolrand();
       temp_d = R1 +  ( R2 / ln R1)
          End for
       If ( f(temp) < f(Pg) )   Pg = temp
       End if
     End for
Until stopping criteria is reached
```

The function sobolrand() returns a random number distributed by sobol sequence. The computational steps of SOM-QPSO2 are same as that of SOM-QPSO1, except for the fact that the worst particle in the swarm is mutated instead of the best particle.

V. Benchmark Problems and Results

In the present study we have taken 3 benchmark problems (Table I), which are considered to be starting point for checking the credibility of any optimization algorithm.

All the test problems are highly multimodal and scalable in nature. The real optimum of all the test problems is zero. Each function is tested with a swarm size of 20, 40 and 80 for dimension 10, 20, 30. The maximum number of generations is set as 1000, 1500 and 2000 corresponding to the dimensions 10, 20 and 30 respectively. A total of 30 runs for each experimental setting are conducted and the average fitness of the best solutions throughout the run is recorded.



(b)

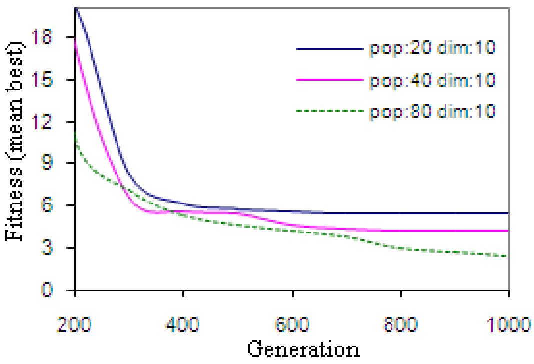
Fig 3 Performance for Rosenbrock function of SOM-QPSO1: (a) Dimension 10 (b) Dimension 20

The mean best fitness value for the functions $f_1 - f_3$ are given in Tables II – IV, respectively, in which Pop represents the swarm population, Dim represents the dimension and Gne represents the maximum number of permissible generations. Figures 3 and 4 show the mean best fitness curves for the Rosenbrock function corresponding to the algorithms SOM-QPSO1 and SOM-QPSO2 respectively. Figures 5 and 6 show the performance curves of Rastringin and Rosenbrock function respectively.

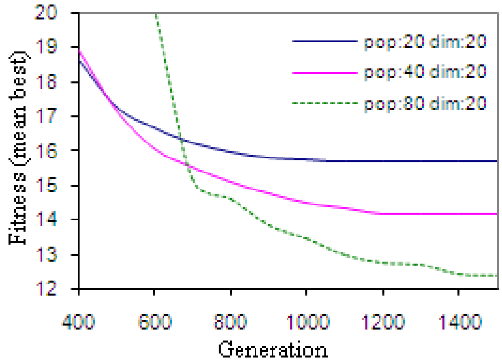
Comparison of the proposed algorithms is done with BPSO, QPSO and two more variants of QPSO. The numerical results show that in all the test cases except two cases in griewank function the proposed algorithms perform much better than the other algorithms. If we compare the performance of SOM-QPSO1 and SOM-QPSO2 with each other then from the numerical results we can see that SOM-QPSO2 in which the worst particle of the swarm is mutated gave better results than SOM-QPSO1 in 18 test cases out of the total 27 cases tried.

TABLE I
NUMERICAL BENCHMARK PROBLEMS

Function	Dim (n)	Range
Rastringin $f_1(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	10 20 30	[2.56,5.12]
Griewank $f_2(x)=\frac{1}{4000}\sum_{i=0}^{n-1}x_i^2-\prod_{i=0}^{n-1}\cos(\frac{x_i}{\sqrt{i+1}})+1$	10 20 30	[300,600]
Rosenbrock $f_3(x)=\sum_{i=0}^{n-1}100(x_{i+1}-x_i^2)^2+(x_i-1)^2$	10 20 30	[15,30]



(a)



(b)

Fig 4 Performance for Rosenbrock function of SOM-QPSO2: (a) Dimension 10 (b) Dimension 20

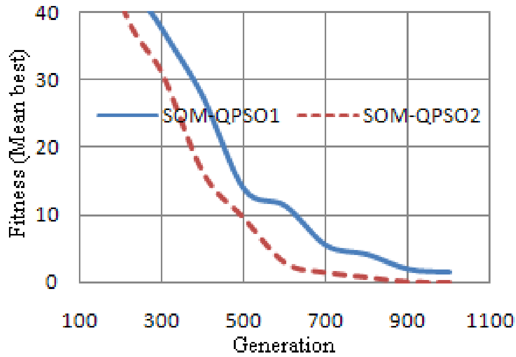


Fig 5 Performance curves of Rastringin function for SOM-QPSO1 and SOM-QPSO2

TABLE II RESULTS OF RASTRINGIN FUNCTION (MEAN BEST)

Pop	Dim	Gne	SOM-QPSO1 (gbest)	SOM-QPSO2 (gworst)	BPSO [10]	QPSO [10]	Mutation gbest [13]	Mutation gbest [9]
20	10	1000	1.46912	**1.216721**	5.5382	5.2543	5.2216	4.3976
	20	1500	9.242158	**4.625661**	23.1544	16.2673	16.1562	14.1678
	30	2000	11.405516	**10.827897**	47.4168	31.4576	26.2507	25.6415
40	10	1000	**0.980288**	1.153735	3.5778	3.5685	3.3361	3.2046
	20	1500	**3.13032**	3.813158	16.4337	11.1351	10.9072	9.5793
	30	2000	**6.381452**	9.119192	37.2896	22.9594	19.6360	20.5479
80	10	1000	1.494764	**0.095692**	2.5646	2.1245	2.0185	1.7166
	20	1500	6.089801	**3.409459**	13.3826	10.2759	7.7928	7.2041
	30	2000	6.006929	**5.15692**	28.6293	16.7768	14.9055	15.0393

TABLE III RESULTS OF GRIEWANK FUNCTION (MEAN BEST)

Pop	Dim	Gne	SOM-QPSO1 (gbest)	SOM-QPSO2 (gworst)	BPSO [10]	QPSO [10]	Mutation gbest [13]	Mutation gbest [9]
20	10	1000	0.07226	**0.057871**	0.09217	0.08331	0.0627	0.0780
	20	1500	0.022172	**0.014902**	0.03002	0.02033	0.0209	0.0235
	30	2000	0.012498	0.012113	0.01811	0.01119	0.0110	**0.0099**
40	10	1000	**0.040943**	0.045446	0.08496	0.06912	0.0539	0.0641
	20	1500	**0.014368**	0.017868	0.02719	0.01666	0.0238	0.0191
	30	2000	0.012306	**0.01002**	0.01267	0.01161	0.0119	0.0098
80	10	1000	**0.028259**	0.039229	0.07484	0.03508	0.0419	0.0460
	20	1500	0.01835	**0.013274**	0.02854	0.01460	0.0136	0.0186
	30	2000	**0.010335**	0.012353	0.01258	0.01136	0.0120	**0.0069**

TABLE IV RESULTS OF ROSENBROCK FUNCTION (MEAN BEST)

Pop	Dim	Gne	SOM-QPSO1 (gbest)	SOM-QPSO2 (gworst)	BPSO[10]	QPSO [10]	Mutation gbest [13]	Mutation gbest [9]
20	10	1000	5.673753	**5.507213**	94.1276	59.4764	27.4620	21.2081
	20	1500	15.890886	**15.721613**	204.336	110.664	49.1176	61.9268
	30	2000	26.041815	**25.956007**	313.734	147.609	97.5952	86.1195
40	10	1000	4.26871	**4.210739**	71.0239	10.4238	7.8741	8.1828
	20	1500	14.367628	**14.156106**	179.291	46.5957	28.4435	40.0749
	30	2000	24.617573	**24.397981**	289.593	59.0291	62.3854	65.2891
80	10	1000	2.745147	**2.520018**	37.3747	8.63638	6.7098	7.3686
	20	1500	**12.342766**	12.42195	83.6931	35.8947	31.0929	30.1607
	30	2000	22.852736	**22.742935**	202.672	51.5479	43.7622	38.3036

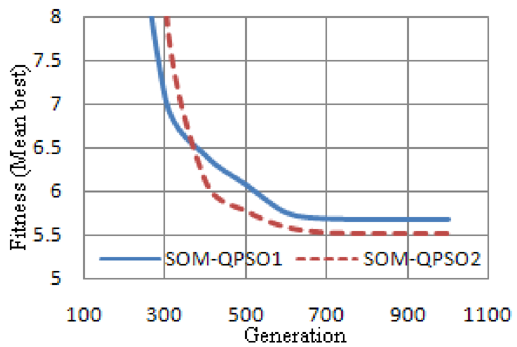


Fig 6 Performance curves of Rosenbrock function for SOM-QPSO1 and SOM-QPSO2

VI CONCLUSION

In this paper, we proposed a new mutation operator called SOM mutation operator for improving the performance of a QPSO. Two versions of algorithms called SOM-QPSO1 and SOM-QPSO2 are defined. The empirical studies show that the proposed versions are better than the BPSO and QPSO quite significantly. However, we would like to add that we have tested the algorithms on a narrow platform of benchmark problems. Thus making any concrete judgment about the proposed algorithms is not justified. In future, we will test the proposed algorithms on a wider platform consisting of more complex, multidimensional problems having larger number of variables. In addition, it will be interesting to do some theoretical research on the better performance of PSO when the worst particle is mutated.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", IEEE Int. Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995, pp. 1942-1948.
[2] H. Higashi and H. Iba, "Particle Swarm Optimization with Gaussian Mutation", In Proc. of the IEEE swarm Intelligence Symposium, 2003, pp. 72 – 79.
[3] A. Stacey, M. Jancic and I. Grundy, "Particle Swarm Optimization with Mutation", in Proc. of the IEEE Congress on Evolutionary Computation, 2003, pp. 1425 – 1430.
[4] M. Pant, R. Thangraj, V. P. Singh and A. Abraham, "Particle Swarm Optimization Using Sobol Mutation", In Proc. of Int. Conf. on Emerging Trends in Engineering and Technology, India, 2008, pp. 367 – 372.
[5] M. Pant, R. Thangaraj and A. Abraham, "Improved Particle Swarm Optimization with Low-Discrepancy Sequences", in Proc. of the IEEE Congress on Evolutionary Computation, 2008, pp. 3016 – 3023.
[6] Nguyen X. H., Nguyen Q. Uy., R. I. Mckay and P. M. Tuan, "Initializing PSO with Randomized Low-Discrepancy Sequences: The Comparative Results", In Proc. of IEEE Congress on Evolutionary Algorithms, 2007, pp. 1985 – 1992.
[7] H. M. Chi, P. Beerli, D. W. Evans, and M. Mascagni, "On the Scrambled Sobol Sequence", In Proc. of Workshop on Parellel Monte Carlo Algorithms for Diverse Applications in a Distributed Setting, LNCS 3516, Springer Verlog, 1999, pp. 775 – 782.
[8] Pang XF, "Quantum mechanics in nonlinear systems", River Edge (NJ, USA): World Scientific Publishing Company, 2005.
[9] 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.
[10] Liu J, Sun J, Xu W, "Quantum-Behaved Particle Swarm Optimization with Adaptive Mutation Operator", ICNC 2006, Part I, Springer-Verlag: 959 – 967, 2006.
[11] 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.
[12] 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.
[13] 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.