IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 2, January 2013
ISSN (Print): 1694-0784 | ISSN (Online): 1694-0814
www.IJCSI.org

A Binary Quantum-behaved Particle Swarm Optimization Algorithm with Cooperative Approach

Jing Zhao[1,2], Jun Sun[2] and Wenbo Xu[3]

**[1] School of Digital Media, Jiangnan University
Wuxi, 214122, Jiangsu, China**

**[2] School of Information, Shandong Polytechnic University,
Jinan, 250353, Shandong, China**

**[3] School of Internet of Things Engineering, Jiangnan University
Wuxi, 214122, Jiangsu, China**

Abstract

A novel binary Quantum-behaved Particle Swarm Optimization algorithm with cooperative approach (CBQPSO) is introduced. In the proposed algorithm, the updating method of particle's previous best position and swarm's global best position are performed in each dimension of solution vector to avoid loss some components that have moved closer to the global optimal solution in the vector. Five test functions are used to test the performance of CBQPSO. The results of experiments show that the proposed technique can increase diversity of population and converge more rapidly than other binary algorithms.

Keywords: *Quantum-behaved Particle Swarm Optimization, Binary, Cooperative Approach, Test Functions.*

1. Introduction

Particle Swarm Optimization (PSO) is an evolutionary computation technique developed by Dr. Eberhart and Dr. Kennedy in 1995[1], inspired by social behavior of bird flocking or fish schooling. The optimal solution is obtained by exchanging information between individuals. However, the algorithm cannot converges to the global minimum point with probability one under suitable condition[2]. Jun Sun *et al* have proposed a global convergence-guaranteed PSO algorithm[3], Quantum-behaved Particle Swarm Optimization (QPSO) algorithm, which is inspired by quantum mechanics. It has been shown that QPSO outperforms PSO on several aspects, such as simple evolution equations, more few control parameters, fast convergence speed, simple operation and so on[4,5].

In 1997, Kennedy proposed the binary version of PSO (BPSO)[6], and Jun Sun *et al* proposed the binary version of QPSO (BQPSO) in 2007[7]. This paper will focus on developing the binary version of QPSO with cooperative method (CBQPSO). In the proposed algorithm, each dimension of particle's new solution vector replaces in turn the corresponding dimension of particle's previous best position and swarm's global best position to calculate the fitness value.

The rest structure of this paper is as follows. In section 2, a brief introduction of the BPSO is presented. The BQPSO is described in section 3. Next, the novel CBQPSO is depicted in section 4. Then the experiment results are given in section 5. Finally, the conclusion is put forward in section 6.

2. Binary Particle Swarm Optimization

In PSO, the population with M individuals, which is treated as a particle, is called a swarm X in the D-dimensional space. The position vector and velocity vector of particle i at the generation t represented as $x_i(t)=(x_{i1}(t), x_{i2}(t), \cdots, x_{iD}(t))$ and $v_i(t)=(v_{i1}(t), v_{i2}(t), \cdots, v_{iD}(t))$ .The particle moves according to the equations:

$$v_{id}(t+1)=wv_{id}(t)+c_1r_1(pbest_{id}-x_{id}(t)) + c_2r_2(gbest_d - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \quad (2)$$

Where $i=1,2,\cdots,M; d=1,2,\cdots,D$, w is the inertia weight, whose value is typically setup to vary linearly from 0.9 to 0.4. c_1 and c_2 are called the acceleration coefficients which usually are set as $c_1=c_2$. r_1 and r_2 are random number uniformly distributed in $(0,1)$ .Vector $pbest_i=(pbest_{i1}, pbest_{2i}, \cdots, pbest_{iD})$ is the best previous position of particle i with the name personal best position($pbest$), while the global best position($gbest$), $gbest=(gbest_1, gbest_2, \cdots, gbest_D)$, is the best particle position among all the particles in the population.

In BPSO[6,8], Eq. (3) replaces Eq. (2).

if $(rand() < S(V_{id}))$ then $X_{id} = 1$ else $X_{id} = 0$ (3)

Where $S(v)$ is a sigmoid limiting transformation function($s(v) = \frac{1}{(1+e^{-v})}$), and $rand()$ is a random number selected from a uniform distribution in $(0,1)$.

3. Binary Quantum-behaved Particle Swarm Optimization

3.1 Quantum-behaved Particle Swarm Optimization

In PSO algorithm, the state of particle is depicted by its position vector and velocity vector, which determine the trajectory of the particle. The particle moves along a determined trajectory in Newtonian mechanics, but this is not the case in quantum mechanics. In quantum world, the term *trajectory* is meaningless, because position and velocity of a particle cannot be determined simultaneously according to *uncertainty principle*. Therefore, if individual particles in a PSO system have quantum behavior, the PSO algorithm is bound to work in a different fashion.

In quantum time-space framework, Jun Sun *et al.* introduce QPSO algorithm. The equations are as follows:

$$mbest = \frac{1}{M}\sum_{i=1}^{M} pbest_i = (\frac{1}{M}\sum_{i=1}^{M} pbest_{i1}, \frac{1}{M}\sum_{i=1}^{M} pbest_{i2}, \cdots, \frac{1}{M}\sum_{i=1}^{M} pbest_{iD}) \quad (4)$$

$$p_{id} = \varphi \times pbest_{id} + (1-\varphi) \times gbest_d \quad (5)$$

$$x_{id}(t+1) = p_{id} \pm \beta |mbest_d - x_{id}(t)| * \ln(\frac{1}{u}) \quad (6)$$

where φ is a random number uniformly distributed in $(0,1)$. $mbest$ is mean best position of the population. Parameter β is called the Contraction-Expansion coefficient, which can be tuned to control the convergence speed of the algorithm. From the results of stochastic simulation[9], it can be concluded that in QPSO, when $\beta < 1.782$, the particles will converge. In the process of iteration, $\pm$ is decided by the random number, when it is bigger than 0.5, minus sign (－) is proposed, others plus sign (＋) is proposed.

3.2 Binary Quantum-behaved Particle Swarm Optimization

In this section, a discrete binary version of QPSO (BQPSO) is proposed. Because the iteration equations of QPSO are far different from those of PSO, the methodology of BPSO does not apply to QPSO. In QPSO, there are no velocities and trajectories concepts but position and distance. In BQPSO, the position of the particle is represented as a binary string. The distance is defined as the Hamming distance between two binary strings. That is

$$|X - Y| = d_H(X, Y) \quad (7)$$

Where X and Y are two binary strings and represent two positions. The function $d_H()$ is to get the Hamming distance between X and Y. The Hamming distance is the count of bits different in the two strings.

The *j*th bit of the $mbest$ is determined by the states of the *j*th bits of all particles' $pbest$ in BQPSO. If more particles take on 1 at the *j*th bit of their own $pbest$, the *j*th bits of $mbest$ will be 1; otherwise the bit will be 0. However, if half of the particles take on 1 at the *j*th bit of their $pbest$, the *j*th bit of $mbest$ will be set randomly to be 1 or 0, with probability 0.5 for either state.

In BQPSO, the point p_i is obtained by crossover operation on $pbest_i$ and $gbest$. Firstly make one-point or multi-point crossover operation on $pbest_i$ and $gbest$ to generate two offspring. Then randomly select one of the offspring and output it as the point p_i.

Consider iterative Eq. (6) and transform it as

$$b = d_H(x_i, p_i) = \beta \times d_H(x_i, mbest) \times \ln(\frac{1}{u}) \quad (8)$$

We can obtain the new string x_i by the transformation in which each bit in p_i is mutated with the probability computed by

$$c_d = \begin{cases} \frac{b}{l} \\ 1 \quad if\ \frac{b}{l} > 1 \end{cases} \quad (9)$$

Where l is the length of the dth dimension of particle i. In the process of iteration, if $rand() < c_d$, the corresponding bit in the position of particle i will be reversed, otherwise remains it.

With the above definition and modifications of iterative equations, the BQPSO algorithm is described as the following procedure:

Step 1 Initialize an array of binary bits for all particles, particle's personal best positions *pbest* and swarm's global best position *gbest* ;

Step 2 For each particle, determine the *mbest* and get a stochastic position p_i by exerting crossover operation on $pbest_i$ and *gbest* ;

Step 3 For each dimension, compute the mutation probability c_d and then update the particle's new position x_i by c_d ;

Step 4 Evaluate the objective function value of the particle, and compare it with the objective function value of *pbest* and *gbest* . If the current objective function value is better than that of *pbest* and *gbest* , then update *pbest* and *gbest* ;

Step 5 Repeat step 2~4 until the stopping criterion is satisfied or reaches the given maximal iteration.

4. Binary Quantum-behaved Particle Swarm Optimization with Cooperative Approach

As BPSO and BQPSO described, each particle represents a complete solution vector for the objective function $f(X)=f[(X_1,X_2,\cdots,X_N)]$. Each update step is also performed on a full *D*-dimensional vector. Then it may be appear the possibility that some dimension in the solution vector have moved closer to the global optimum, while others moved away from the global optimum. Whereas the objective function value of the solution vector is worse than the former value. BPSO and BQPSO take the new solution vector for a complete vector and neglect the deteriorated components during the iterations. As long as the current objective function value is better than the former value, then update *pbest* and *gbest* . Therefore, the current solution vector can be give up in next iteration and the valuable information of the solution vector is lost unknowingly. In order to make full use of the beneficial information, the cooperative method[10,11] is introduced to BQPSO. In the proposed method, we expect that the operation can avoid the undesirable behavior, which is a case of taking two steps forward (some dimension improved), and one step back (some dimension deteriorated).

4.1 Cooperative Approach

We expect that once for every time a component in the vector has been updated, resulting in much quicker feedback. Thus, a cooperative method for doing just this is presented. In the new method each dimension of the new solution vector replaces in turn the corresponding dimension of *pbest* and *gbest* , and then compare the new objective function value to decide whether to update *pbest* and *gbest* .

The process is as follows:

Step 1 For each particle i , initialize $cgbest = gbest$, $cpbest_i = pbest_i$;

Step 2 For each dimension of particle i , replace the dimension of *cpbest* and *cgbest* by the corresponding dimension of the particle;

Step 3 Evaluate the new objective function value of *cpbest* and *cgbest* , and compare them with the objective function value of *pbest* and *gbest* . If the current objective function value is better than that of *pbest* and *gbest* , then update *pbest* and *gbest* ;

Step 4 Repeat step 2~3 until all the dimension of the particle is compared.

4.2 CBQPSO

With above modifications, the iteration process of CBQPSO is described step-by-step below.

Step 1 Initialize an array of binary bits for all particles, particle's personal best positions *pbest* and swarm's global best position *gbest* ;

Step 2 Update the particle's new position x_i by BQPSO;

Step 3 Evaluate the objective function value of the particle, and compare them with the objective function value of *pbest* and *gbest* . If the current objective function value is better than that of *pbest* and *gbest* , then update *pbest* and *gbest* ;

Step 4 Use cooperative strategy to update *pbest* and *gbest* ;

Step 5 Repeat step 2~4 until the stopping criterion is satisfied or reaches the given maximal iteration.

The proposed algorithm tries to improve convergence precision by comparing each dimension of solution vector. It must extend the search space and then increase the time consumption. Two adaptive control methods are proposed. Firstly, the cooperative strategy is adopted in a certain interval. In our method, it set to 5. Then the cooperative strategy is performed when the bit of the particle is different from the corresponding bit of *pbest* and *gbest* .

5. Experiments

In this section, the performance of CBQPSO algorithm is tested on the following five different standard functions[7] to be maximized. Then the results are compared with BPSO and BQPSO.

$$f_1(X) = 78.6 - \sum_{i=1}^{3} x_i^2 \qquad (-5.12 \le x_i \le 5.12)$$

$$f_2(X) = 3905.93 - (100(x_1^2 - x_2)^2 - (1 - x_1)^2) \quad (-2.048 \le x_i \le 2.048)$$

$$f_3(X) = 25 - (x_1 + x_2 + x_3 + x_4 + x_5) \quad x_i \in Z, \quad (-5.12 \le x_i \le 5.12)$$

$$f_4(X) = 1248.2 - \sum_{i=1}^{30} x_i^4 \quad (-1.28 \le x_i \le 1.28)$$

$$f_5(X) = 500 - 1 \Bigg/ \left(0.002 + \sum_{j=1}^{25} \frac{1}{j + 1 + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)$$

$$a = \begin{pmatrix} 32.0 & 16.0 & 0 & 16.0 & 32.0 \\ -32.0 & -16.0 & 0 & 16.0 & 32.0 \end{pmatrix}$$

$$(-65.536 \le x_i \le 65.536)$$

In the numerical experiments, the algorithms parameters settings are described as follow: for BPSO, the acceleration coefficients are set to $c_1 = c_2 = 2$ and the inertia weight w is decreasing linearly from 0.9 to 0.4. In experiments for BQPSO and CBQPSO, the value of β is 1.4[12] . All experiments are run 50 independent times respectively with a population of 20, 40 and 80 particles on an Intel(R) Xeon(R) E5504 @2.00GHz 2.00GHz , 1GB RAM computer with the software environment of MATLAB2009a. All the algorithms terminate when the number of iterations succeeds 200.

The best fitness value (BFV), maximum value and minimum value are recorded after the algorithm terminates at each run. The performance of all the algorithms is evaluated by average BFV (Avg. BFV) and Standard Deviation (St. Dev.). All the measurements are listed on Table 1. Fig.1 illustrates the convergence process of average BFV of three algorithms over 50 runs with 40 particles on five test functions.

The optima of function $f1$, whose fitness value is 78.6, can be find out by BPSO, BQPSO and CBQPSO. As can be seen from Table 1, the average BFV and St. Dev. of CBQPSO is best. And BQPSO outperforms BPSO. As of solution quality, CBQPSO and BQPSO with 20 particles make 12 successful searches out of 50 trial runs, whereas BPSO find out the optima for 7 times. And the corresponding times is 14, 13 and 2 respectively with 40 particles. When the population number is 80, the optima are found out for 29, 20 and 4 times corresponding with CBQPSO, BQPSO and BPSO.

On the function $f2$, all the algorithms can be found the optimum fitness value 3905.93. However CBQPSO generates best average BFV and St. Dev.. And BQPSO takes second place. As can be seen from Table 1, BQPSO has the worst performance than other two algorithms with 40 particles. Note that the St. Dev. of BQPSO with 40 particles is better than that of BPSO.

The third function $f3$ is a simple integer function with an optimum of 55. CBQPSO, BQPSO and BPSO with 80 particles hit the optima for 50 times out of 50 runs. CBQPSO and BQPSO have better quality of solution than BPSO with 20 and 40 particles.

In order to measure the average fitness value over the entire population, Gaussian noise is introduced into f_4 function. In this function, the average BFV of BQPSO is inferior to CBQPSO but superior to BPSO. However the St. Dev. of BQPSO is the best results.

The last function $f5$ has an optimum 500. All the algorithms can be found out the best value 499.26991. CBQPSO with 40 and 80 partilces is able to hit the optimum beyond 47 times out of 50 runs. The number of successful searches of BPSO is better than BQPSO. However the average BFV and St. Dev. of BPSO is inferior to BQPSO.

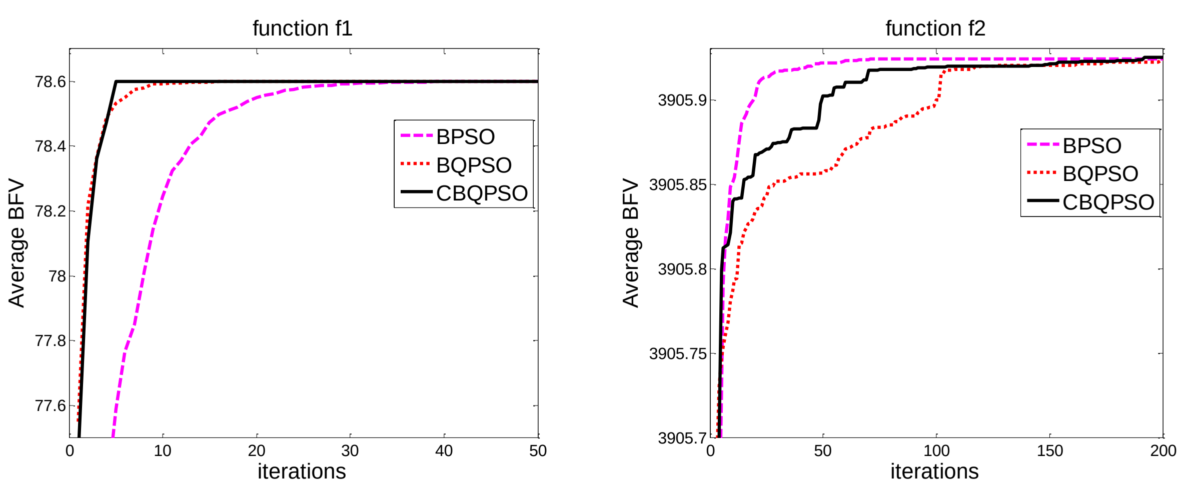
As is illustrated in Fig.1, we can see that the effectiveness of the proposed CBQPSO. CBQPSO can converge to the optimum more rapidly than BQPSO and BPSO on three functions except $f2$ and $f5$. On $f2$, BPSO converges more quickly but generates worse solution than CBQPSO. On $f5$, BPSO converges rapidly than other two

IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 2, January 2013
ISSN (Print): 1694-0784 | ISSN (Online): 1694-0814
www.IJCSI.org

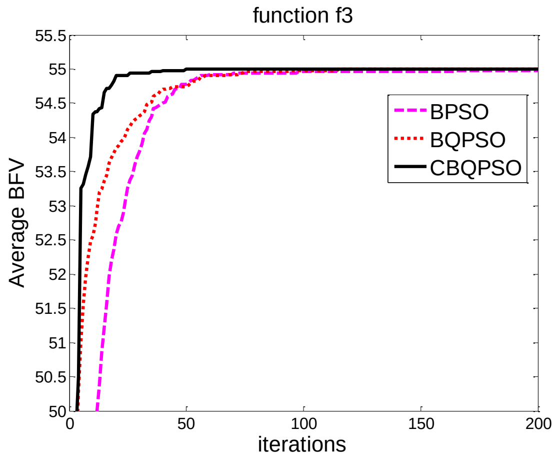
algorithms at the early stage of running, but CBQPSO exceeds BPSO soon and generates a slightly better solution. Compared with BPSO and BQPSO, experimental results show the effectiveness of the proposed CBQPSO.

Table 1: Results of BPSO, BQPSO and CBQPSO on five testing functions

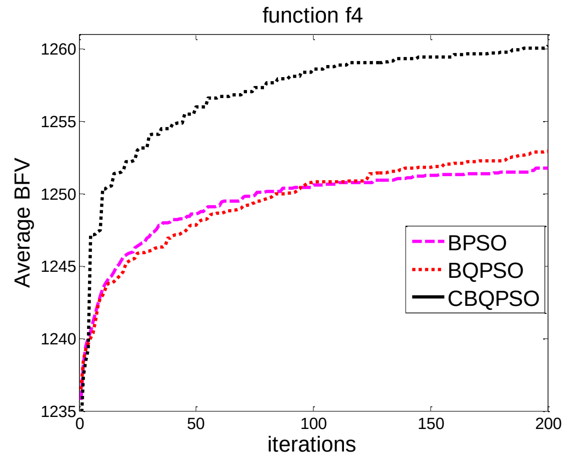
Function	***Particles***	***BPSO***		***BQPSO***		***CBQPSO***	
		Mean (St.Dev.)	***MAX (MIN)***	***Mean (St.Dev.)***	***MAX (MIN)***	***Mean (St.Dev.)***	***MAX (MIN)***
$f1$	20	78.59986 0.000086	78.6 78.5997	78.59987 0.000099	78.6 78.5997	78.59988 0.000087	78.6 78.5997
	40	78.59985 0.000076	78.6 78.5997	78.59989 0.000086	78.6 78.5997	78.59991 0.000070	78.6 78.5998
	80	78.59984 0.000086	78.6 78.5997	78.59992 0.000082	78.6 78.5997	78.59995 0.000058	78.6 78.5998
$f2$	20	3905.9002 0.110331	3905.93 3905.1536	3905.9102 0.037457	3905.93 3905.7815	3905.9144 0.038555	3905.93 3905.6873
	40	3905.9242 0.016954	3905.93 3905.8418	3905.9235 0.016072	3905.93 3905.8312	3905.9252 0.013599	3905.93 3905.8728
	80	3905.9292 0.001747	3905.93 3905.9188	3905.9292 0.001608	3905.93 3905.9214	3905.9296 0.000863	3905.93 3905.9246
$f3$	20	54.86 0.350510	55 54	54.96 0.197949	55 54	55 0.141421	55 54
	40	54.98 0.141421	55 54	55 0	55 55	55 0	55 55
	80	55 0	55 55	55 0	55 55	55 0	55 55
$f4$	20	1250.7889 3.918132	1258.2255 1240.5389	1253.5857 3.305561	1261.7592 1247.7543	1259.3092 3.602731	1266.8519 1252.1388
	40	1251.7949 4.256897	1263.1885 1241.2841	1252.9749 3.470581	1262.1260 1245.6837	1260.2212 4.843062	1274.1392 1250.1568
	80	1251.8510 3.902359	1264.5341 1243.9959	1254.2206 3.011134	1260.8333 1245.5395	1262.8852 4.664300	1270.5571 1252.4547
$f5$	20	498.71163 0.498272	499.2699 497.76306	498.75278 0.483852	499.2699 497.81977	499.25442 0.038667	499.2699 499.15955
	40	498.95986 0.366094	499.2699 498.10809	498.97292 0.422411	499.2699 497.62203	499.2699 0.000038	499.2699 499.26975
	80	499.03857 0.352775	499.2699 498.10906	499.13661 0.203900	499.2699 498.51791	499.2699 0.000022	499.2699 499.26975



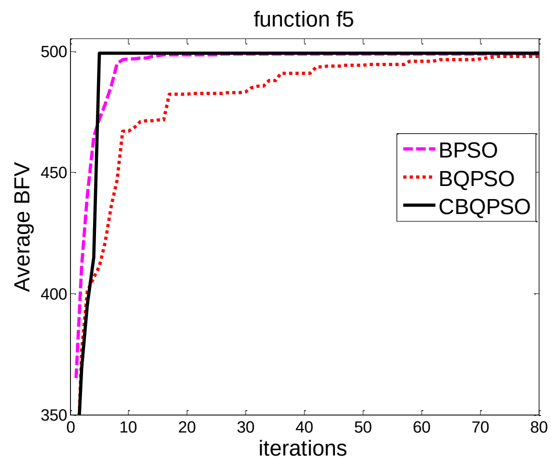
(a) $f1$ (b) $f2$



(c) $f3$



(d) $f4$



(e) $f5$

Fig. 1 The convergence process of three algorithms with 40 particles.

6. Conclusions

In BQPSO, an improvement in two components will overrule a potentially good value for a single component. In this paper, a discrete binary version of Quantum-behaved Particle Swarm Optimization algorithm with cooperative method (CBQPSO) is introduced to improve the undesirable behavior by decomposing the solution vector. In the proposed algorithm, each dimension update of particle can feed back to personal best positions and swarm best position. The results of experiment have showed that the CBQPSO algorithm performs better than other algorithm on global convergence and has stronger ability to escape from the local optimal solution during the search process. However it can be extend the search space with the increasing complexity of the problem, time consumption is the main deficiency of CBQPSO.

References

[1] Kennedy J., and Eberhart R., "Particle Swarm Optimization", in IEEE International Conference on Neural Networks, 1995, Vol.4, pp.1942-1948.
[2] Frans Van Den Bergh, "An Analysis of Particle Swarm Optimizers", Ph.D. thesis, University of Pretoria, South Africa, 2002.
[3] Jun Sun, Bin Feng, and Wenbo Xu, "Particle Swarm Optimization with Particles Having Quantum Behavior", in IEEE Congress on Evolutionary Computation, 2004, Vol.1, pp.325-331.
[4] Jun Sun, Wei Fang, Xiaojun Wu, Vasile Palade, and Wenbo Xu, "Quantum-behaved Particle Swarm Optimization: Analysis of Individual Particle Behavior and Parameter Selection", Evolutionary Computation, Vol.20, No. 3, 2012, pp. 349-393.
[5] Wei Fang, Jun Sun, Yanrui Ding, Xiaojun Wu, and Wenbo Xu, "A Review of Quantum-behaved Particle Swarm Optimization", IETE Technical Review, Vol.27, No.4, 2010, pp.336-348.

[6] Kennedy, J., and Eberhar, R. C, "A Discrete Binary Version of the Particles Swarm Algorithm". In IEEE International Conference on Systems, Man and Cybernetics, 1997, Vol.5, pp. 4104-4108.
[7] Jun Sun , Wenbo Xu , Wei Fang, and Zhilei Chai, "Quantum-behaved Particle Swarm Optimization with Binary Encoding", in International Conference on Adaptive and Natural Computing Algorithms, 2007, Vol.1, pp. 376-385.
[8] Seyyed Ali Hashemi,and Behrouz Nowrouzian, "A Novel Discrete Particle Swarm Optimization for FRM FIR Digital Filters", Journal of Computers, Vol.7, No.6, 2012, pp. 1289-1296.
[9] Jun Sun, Xiaojun Wu, Vasile Palade, Wei Fang, Choi-Hong Lai, and Wenbo Xu, "Convergence Analysis and Improvements of Quantum-behaved Particle Swarm Optimization", Journal of Information Science, Vol.193, 2012, pp.81-103.
[10] Van den Bergh F., and Engelbrecht A. P., "A Cooperative Approach to Particle Swarm Optimization", IEEE Transactions on Evolutionary Computation, Vol.8, No.3, 2004, pp.225-239.
[11] Mehdi Neshat, Shima F. Yazdi, Daneyal Yazdani, and Mehdi Sargolzaei, "A New Cooperative Algorithm Based on PSO and K-Means for Data Clustering", Journal of Computers Science, Vol.8, No.2, 2012, pp. 188-194.
[12] Jun Sun, "Particle Swarm Optimization with Particles Having Quantum", Ph.D. thesis, Jiangnan University, Wuxi, China, 2009.