ICIC Express Letters ICIC International ©2017 ISSN 1881-803X
Volume **11**, Number **1**, January **2017** pp. **143–149**

DESIGN AND APPLICATION OF FUZZY LOGIC SYSTEM BASED ON QPSO INTELLIGENT ALGORITHM

Qiufeng Fan, Tao Wang, Yang Chen and Zhifeng Zhang

Department of Mathematics Science
Liaoning University of Technology
No. 169, Shiying Street, Guta District, Jinzhou 121001, P. R. China
lxywt@lnut.edu.cn

Received June 2016; accepted September 2016

Abstract. *Due to its wide application, fuzzy logic system has become a hot research topic in the field of academic and practical application. For parameter identification, the common method is the least square algorithm, BP algorithm, etc. Although there are a lot of methods, they are not intelligent algorithms. By combining with the TSK fuzzy logic system and the neural network, this paper designed a five layers fuzzy neural network system. The quantum behaved particle swarm optimization (QPSO) intelligent algorithm was used to tune the fuzzy logic system parameters, and the design of intelligent system is applied in the prediction of the international gold prices. By comparing the performance of QPSO and BP algorithm, the performance index and simulation results illustrated the effectiveness of the proposed design method, which obtained better effect in contrast with the BP algorithm.*
Keywords: Fuzzy logic system, FNN, QPSO algorithm, BP algorithm, International gold prices

1. **Introduction.** Fuzzy logic system is a system based on rule-base and fuzzy logic. TSK fuzzy logic system is proposed by Takagi, Sugeno and Kang [1,2]. In 1985, Takagi and Sugeno design T-S model [1]. In 1988, Sugeno and Kang extend this method by adding an unbiasedness constraint [2]. In the extended method the premise structure, consequent structure, premise parameters, and consequent parameters are tuned recursively. TSK fuzzy inference model not only can be used in fuzzy controller, but also can approximate any nonlinear system. It can be applied to general fuzzy systems, especially for local linear and piecewise control system. Fuzzy neural network is developed on the basis of neural network and fuzzy system, and the fusion of the two make up the deficiency of the neural network in fuzzy data processing problems and fuzzy logic defects in learning [3,4]. It is a system which includes linguistic computing, logic inference, distributed processing and nonlinear dynamic process.

In 1995, particle swarm optimization (PSO) algorithm is a population-based optimization algorithm proposed by Kennedy et al. [5,6]. PSO algorithm starts from the random solution, through the iteration to find the optimal solution, and through the fitness function to evaluate the quality of the solution. In order to improve its convergence, Sun et al. proposed a QPSO algorithm based on the standard PSO in 2004 [7-9]. QPSO algorithm is a new model of particle swarm optimization algorithm from the point of view of quantum mechanics. The model thinks the particles have the behavior of quanta, using the Heisenberg uncertainty principle to describe the motion state of the particle, the particle can be searched in the whole feasible solution space, thereby obtaining global optimization solution. Therefore, the QPSO algorithm is a globally convergent iterative search algorithm. In 2010∼2011, Lian et al. applied the QPSO algorithm to the radial basis function neural network and a non-singleton interval type-2 fuzzy logic system of universal image noise removal [10,11]. In 2014, Mendel proposed using the QPSO algorithm to train the fuzzy logic system [12]. At present, the design of the fuzzy logic system is mostly using the BP algorithm, the least square algorithm, the hybrid algorithm, etc., and few use intelligent algorithms to design the fuzzy logic system. By combining with the TSK fuzzy logic system and the neural network, this paper designed a five layers fuzzy neural network system. The QPSO intelligent algorithm was used to tune the fuzzy logic system parameters, and the design of intelligent system is applied in the prediction of the international gold prices. By comparing the performance of QPSO and BP algorithm, the root mean square error (*RMSE*) and simulation results illustrated the effectiveness of the proposed design method, which obtained better effect in contrast with the BP algorithm.

The rest of the paper is organized as follows. In Section 2, we present the principle of QPSO algorithm. In Section 3, the design of fuzzy logic system based on QPSO algorithm is introduced. Section 4 presents application of the QPSO-trained fuzzy logic system in the international gold prices. Finally, the paper is concluded in Section 5.

2. **QPSO Algorithm.** In the PSO with M individuals, each individual is treated as an infinitesimal particle in the D-dimensional space, with the current position vector and velocity vector of particle i at the nth iteration represented as $X_{i,n} = \left(X_{i,n}^1, X_{i,n}^2, \cdots, X_{i,n}^D\right)$ and $V_{i,n} = \left(V_{i,n}^1, V_{i,n}^2, \cdots, V_{i,n}^D\right)$ [5,6]. The particle moves according to the following equations:

$$V_{i,n+1}^j = \omega V_{i,n}^j + c_1 r_{i,n}^j \left(P_{i,n}^j - X_{i,n}^j\right) + c_2 R_{i,n}^j \left(G_{i,n}^j - X_{i,n}^j\right) \tag{1}$$

$$X_{i,n+1}^j = X_{i,n}^j + V_{i,n+1}^j \tag{2}$$

for $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, D$, where c_1 and c_2 are the acceleration coefficients. The parameter ω is the inertia weight. The parameters $r_{i,n}^j$ and $R_{i,n}^j$ are random numbers, $r_{i,n}^j$, $R_{i,n}^j \sim U(0,1)$, $V_{i,n}^j \in [-V_{\max}, V_{\max}]$. $P_{i,n} = \left(P_{i,n}^1, P_{i,n}^2, \cdots, P_{i,n}^D\right)$ is the best previous position of particle i, and $G_n = (G_n^1, G_n^2, \cdots, G_n^D)$ is the global best position.

Accordingly, $P_{i,n}$ can be updated by

$$P_{i,n} = \begin{cases} X_{i,n} & \text{if } f(X_{i,n}) < f(P_{i,n-1}) \\ P_{i,n-1} & \text{if } f(X_{i,n}) \geq f(P_{i,n-1}) \end{cases} \tag{3}$$

and G_n can be found by $G_n = P_{g,n}$, where $g = \arg\max_{1 \leq i \leq M}[f(P_{i,n})]$.

The QPSO algorithm is improved based on the PSO algorithm, and it differs from the PSO algorithm mainly in two aspects.

1) Using (4) instead of (3), the best personal position of each particle is updated by the following:

$$P_{i,n}^j = \varphi_{i,n}^j \cdot P_{i,n}^j + \left(1 - \varphi_{i,n}^j\right) \cdot G_n^j \tag{4}$$

where $\varphi_{i,n}^j$ is a sequence of uniformly distributed random numbers over $(0,1)$.

2) Using (5) instead of (2), the position of the particle is updated by the following:

$$X_{i,n+1}^j = P_{i,n}^j \pm \alpha \cdot \left|X_{i,n}^j - C_{i,n}^j\right| \cdot \ln\left(\frac{1}{u_{i,n+1}^j}\right) \tag{5}$$

where $C_n = \left(C_n^1, C_n^2, \cdots, C_n^D\right)$ is mean best position defined by the average of the personal best positions among all the particles, and the parameter α is contraction-expansion (CE) coefficient. Namely,

$$C_n^j = \left(\frac{1}{M}\right) \sum_{i=1}^{M} P_{i,n}^j \quad (1 \leq j \leq D) \tag{6}$$

3. **Design of Fuzzy Logic System Based on QPSO Algorithm.**

3.1. **Fuzzy neural network.** In this paper, the TSK fuzzy logic system has n inputs and one output, we assume there are M rules, and the lth rule can be expressed as:

$$\begin{aligned} R^l\ &: \text{IF } x_1 \text{ is } F_1^l \text{ and } x_2 \text{ is } F_2^l \text{ and } \cdots \text{ and } x_n \text{ is } F_n^l, \\ &\quad \text{THEN } G^l \text{ is } p_0^l + p_1^l x_1 + p_2^l x_2 + \cdots + p_n^l x_n \qquad l = 1, \cdots, M \end{aligned}$$

where $F_1^l, F_2^l, \cdots, F_n^l$ are the membership functions of the antecedent, G^l is the consequent of the lth rule, and $p_0^l, p_1^l, \cdots, p_n^l$ are the consequent parameters.

The fuzzy logic system is integrated into the neural network to get the five layers fuzzy neural network system as follows.

Layer 1: Input layer: $X = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$.

Layer 2: Membership functions layer: which adopts a Gaussian shape. The membership grade of each node is described as

$$\mu_{F_i^l}(x_i) = \exp\left\{-\frac{1}{2}\left(\frac{x_i - m_{F_i^l}}{\sigma_{F_i^l}}\right)^2\right\} \quad (i = 1, 2, \cdots, n,\ l = 1, 2, \cdots, M)$$

Layer 3: The rules layer: F^l are rule firing strengths, defined as:

$$F^l = \prod_{i=1}^{n} \mu_{F_i^l}(x_i) \quad (i = 1, 2, \cdots, n,\ l = 1, 2, \cdots, M)$$

Layer 4: The weight layer: ω^l are rules' weights, defined as:

$$\omega^l = \frac{F^l}{\sum\limits_{i=1}^{M} F^l} \quad (l = 1, 2, \cdots, M)$$

Layer 5: Output layer: y is the output of the system.

$$y = f_s(X) = \sum_{l=1}^{M} \omega^l G^l$$

G^l is a linear function of $x_1, x_2, \cdots, x_n$, and $G^l = p_0^l + p_1^l x_1 + p_2^l x_2 + \cdots + p_n^l x_n$.

3.2. **Parameters tuning of fuzzy logic system based on QPSO algorithm.** This section presents how to employ the QPSO algorithm to tune the parameters of the fuzzy logic system. The fuzzy logic system has two types of parameters which need to be tuned, that is, the premise parameters and the consequent parameters. In the design of fuzzy neural network system, the premise parameters are $\left\{m_{F_i^l}, \sigma_{F_i^l}\right\}$, and the consequent parameters are $\left\{p_0^l, p_1^l, \cdots, p_n^l\right\}$. The premise parameters and the consequent parameters are identified by the QPSO algorithm. Thus, the position of each particle in the QPSO represents a set of premise parameters and consequent parameters. The fitness function is defined as root mean squared error between actual output and desired output, which can be expressed by

$$RMSE = \sqrt{\frac{1}{N-K}\sum\nolimits_{j=k-n+1}^{N} \left(f_s(X^j) - Y^j\right)^2}$$

The QPSO algorithm is used to tune the parameters of the fuzzy logic system as follows.

Step 1: Initialize the swarm of particles such that the position of each particle is uniformly distributed within the search scope, and set the maximum iteration $N_{\max}$.

Step 2: Set the position of each particle $X_{i,n}$ as the parameters of the fuzzy logic system $\left\{m_{F_i^l}, \sigma_{F_i^l}, p_0^l, p_1^l, \cdots, p_n^l\right\}$. Then calculate fitness value of each particle, and set each particle's personal best position as $P_{i,0} = X_{i,0}$.

Step 3: Find out the mean value of all particles' personal best position C_n by using (6).

Step 4: For each particle in the population, execute from Step 5 to Step 8.

Step 5: Calculate each particle's fitness value, $RMSE(X_{i,n})$, and then compare it with the fitness of its personal best position, $RMSE(P_{i,n-1})$. If $RMSE(X_{i,n}) < RMSE(P_{i,n-1})$, then $P_{i,n} = X_{i,n}$; otherwise, $P_{i,n} = P_{i,n-1}$.

Step 6: Compare the fitness value of each particle's personal best position, $RMSE(P_{i,n})$ with that of the global best position, $RMSE(G_{n-1})$. If $RMSE(P_{i,n}) < RMSE(G_{n-1})$, then $G_n = P_{i,n}$; otherwise, $G_n = G_{n-1}$.

Step 7: Properly select the value of α.

Step 8: Update the position of each particle $X_{i,n}$ by using (4) and (5).

Step 9: If the termination condition is met, output the results; otherwise, go to Step 2, and set the iteration number $n = n + 1$.

4. **Application and Simulation Results.** This section presents the experiment on the application of a QPSO-trained fuzzy logic system in the international gold prices. Gold is a special article with both commodity attribute and currency attribute, so the international gold prices will affect the global gold market. Due to the impact of the global economy, the international gold prices fluctuate every day, including the opening price, the middle price and closing price. In this example, the international gold prices were selected from January 1, 2015 to November 30, 2015, a total of 237 data.

In this paper, we use the first three data to predict the next data, that is, three inputs and one output. In the fuzzy logic system, the rule's antecedent is the Gaussian membership function. The fuzzy logic system is designed by using singleton fuzzifier, product inference. Set the number of rules $M = 8$, and the lth rule can be expressed as:

$$\begin{aligned} R^l &: \text{IF } x_1 \text{ is } F_1^l \text{ and } x_2 \text{ is } F_2^l \text{ and } x_3 \text{ is } F_3^l, \\ &\quad \text{THEN } G^l \text{ is } p_0^l + p_1^l x_1 + p_2^l x_2 + p_3^l x_3 \qquad l = 1, \cdots, 8 \end{aligned}$$

Using Matlab to program the system, the first 129 data used to do training, design type-1 fuzzy logic system estimator, after 108 data used to test the system.

The steps of tuning fuzzy logical system parameters based on Section 3.2 QPSO algorithm, set the maximum iteration $N_{\max} = 100$, $\alpha = 0.7$, the number of particles being 80. Given initial particles are as follows:

$$\begin{aligned} &m_{F_i^l} = 538 + 0.35 \times rands(1), \quad \sigma_{F_i^l} = 200 + 0.35 \times rands(1) \\ &p_0^l = 0.3 \times ones(1,8), \cdots, p_3^l = 0.3 \times ones(1,8) \end{aligned}$$

Using the previously designed QPSO algorithm to tune these initial particles, the QPSO algorithm training output and the actual output of the system are shown in Figure 1. Figure 1 shows the tracking effect is very good. After 108 data are used to test the performance of the designed system, tracking results are shown in Figure 3.

Using the same system structure, Gaussian membership function, singleton fuzzifier, product inference and the same number of rules, BP algorithm is used for training and testing. Comparison of BP and QPSO algorithms, Figure 1 and Figure 3 are the training and tracking performance of the two algorithms, respectively. Figure 2 is the diagram of local zoom of Figure 1. The convergence processes of the two algorithms are shown in Figure 4.

After 100 iterations, the *RMSE* of the two algorithms are obtained. The training errors and testing errors of QPSO and BP algorithms are shown in Table 1.

It can be seen from Figure 1, Figure 2, Figure 3, Figure 4 and Table 1 that it is better to use the QPSO algorithm than the BP algorithm to tune the parameters of the fuzzy logic system, and the *RMSE* of QPSO algorithm is relatively small. It shows that the design of fuzzy logic system based on QPSO algorithm is effective and feasible.

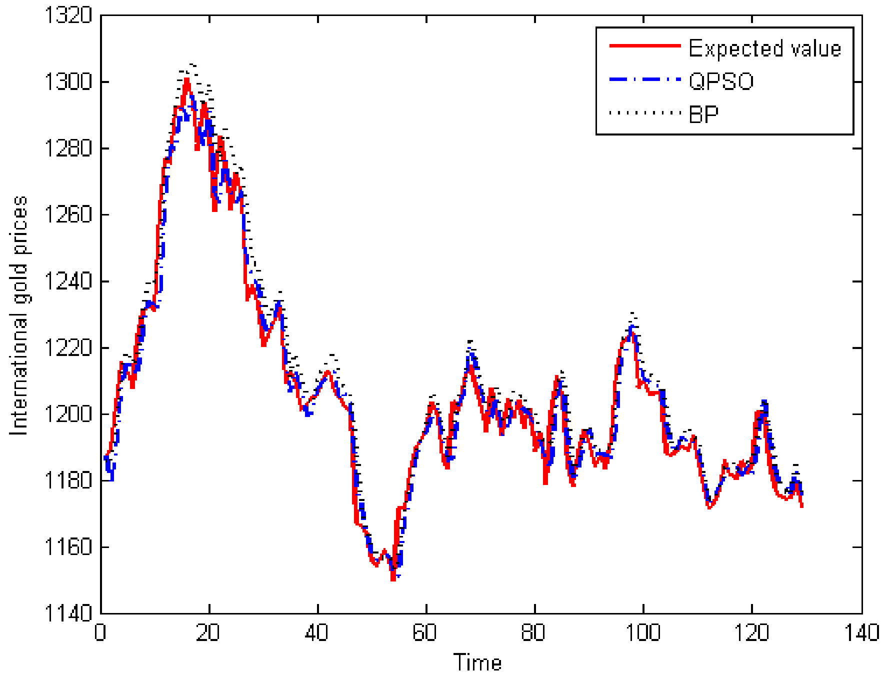

Figure 1. Results on training data

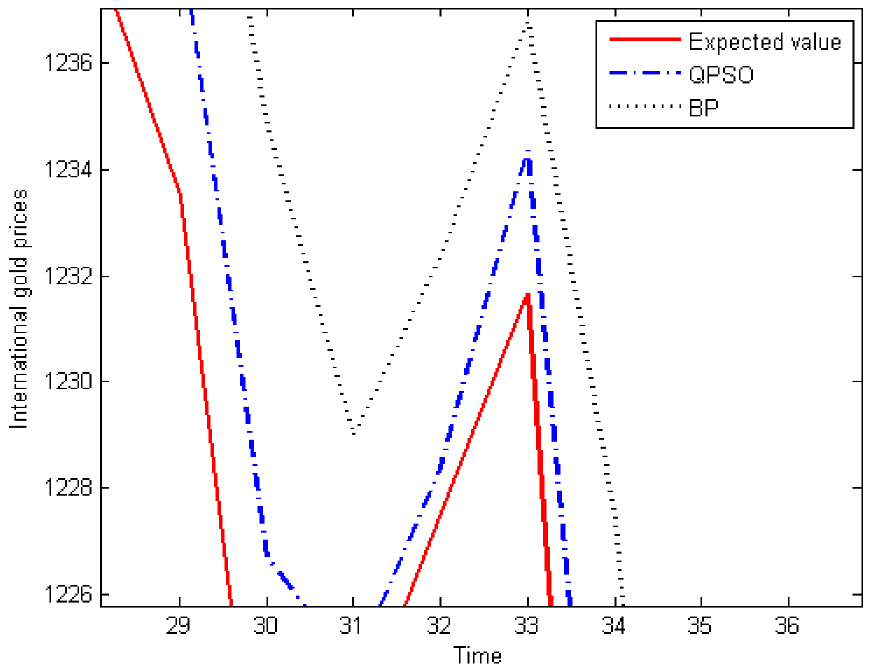

Figure 2. The diagram of local zoom

Table 1. The *RMSE* of QPSO and BP algorithms

	trnRMSE	*testRMSE*
QPSO	6.9691	6.2373
BP	8.0238	7.4435

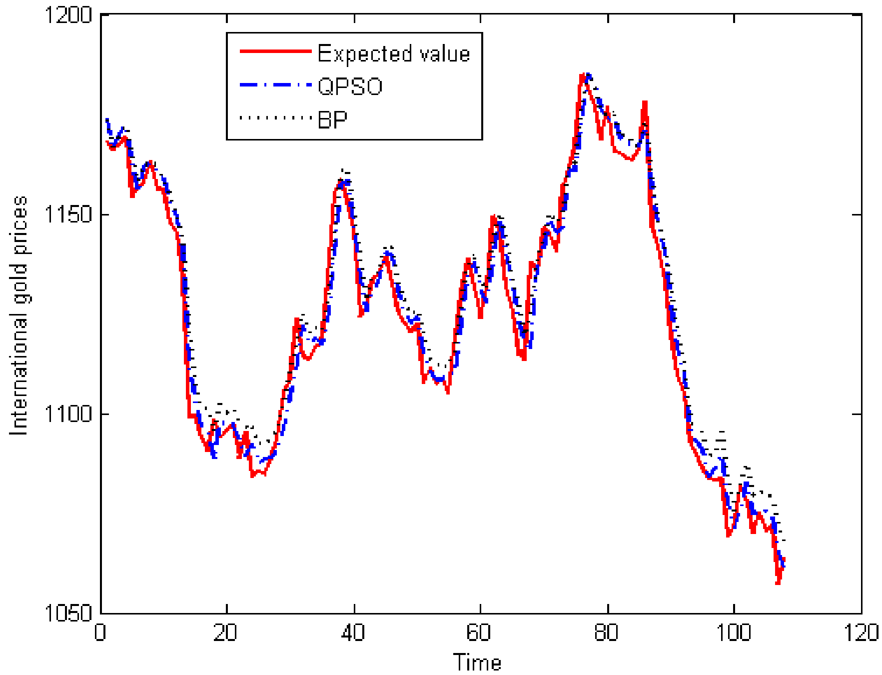

FIGURE 3. Results on testing data

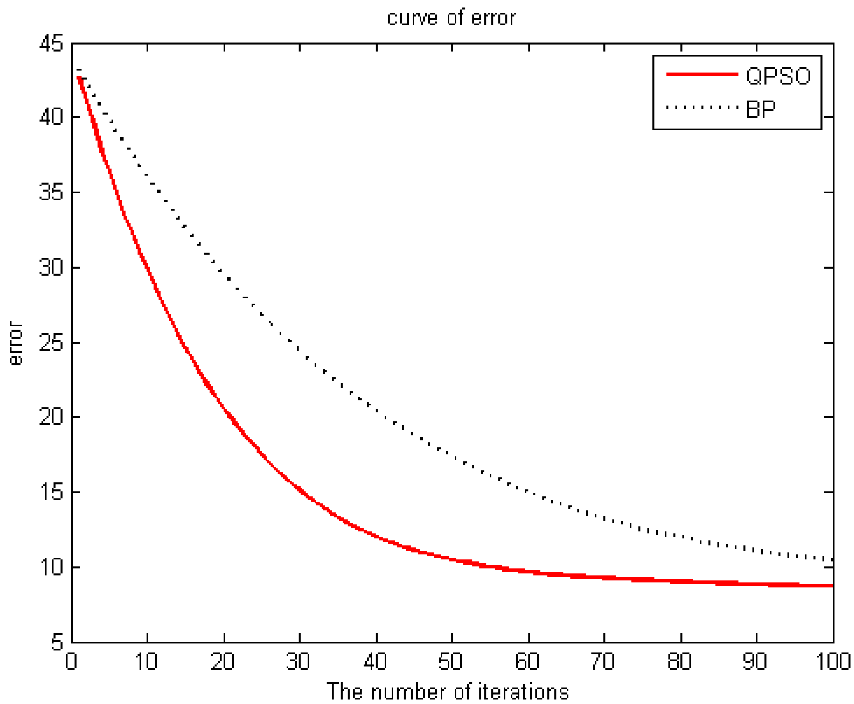

FIGURE 4. The convergence processes of QPSO and BP over 100 iterations

5. **Conclusions.** Fuzzy logic systems are widely used in practice. In order to better apply the fuzzy logic system, training algorithm is the key to determine the system parameters. To overcome the complexity of the previous algorithms, this paper proposes a fuzzy logic system based on QPSO intelligent algorithm to tune the system parameters. The design of intelligent system is applied in the prediction of the international gold prices, and the simulation is given. By comparing the performance of QPSO and BP algorithm, the

performance index and the simulation results show that the QPSO algorithm is better than the BP algorithm to tune the parameters of the fuzzy logic system. Future research will be concentrated on IT2 FLSs design, algorithms that are based on this paper.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61374113), and by Liaoning Province College Basic Scientific Research Business Funding Project (JW201615421).

REFERENCES

[1] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. System, Man, and Cybernetics*, vol.15, no.1, pp.387-403, 1985.
[2] M. Sugeno and G. T. Kang, Structure identification of fuzzy model, *Fuzzy Sets and Systems*, vol.28, no.1, pp.15-33, 1988.
[3] Y.-C. Chen and C.-C. Teng, A model reference control structure using a fuzzy neural network, *Fuzzy Sets and Systems*, vol.73, no.3, pp.291-321, 1995.
[4] J. L. Zhang, *Fuzzy Neural Network Control Theory and Engineering Applications*, Harbin Institute of Technology Press, Harbin, 2004.
[5] J. Kennedy and R. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, pp.1942-1948, 1995.
[6] M. Clerc and J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Trans. Evolutionary Computation*, vol.6, no.1, pp.58-73, 2002.
[7] J. Sun, B. Feng and W. Xu, Particle swarm optimization with particles having quantum behavior, *IEEE Congress on Evolutionary Computation*, vol.70, no.3, pp.1571-1580, 2004.
[8] J. Sun and W. Liu and B. Feng, A global search strategy of quantum-behaved particle swarm optimization, *IEEE Conference on Cybernetics and Intelligent Systems*, vol.1, no.11, pp.111-116, 2005.
[9] W. Fang, J. Sun, Z. Xie and W. Xu, Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter, *Acta Physica Sinica*, vol.59, no.6, pp.3686-3694, 2010.
[10] G. Y. Lian, K. L. Huang, J. H. Chen and F. Q. Gao, Training algorithm for radial basis function neural network based on quantum-behaved particle swarm optimization, *International Journal of Computer Mathematics*, vol.87, no.3, pp.629-641, 2010.
[11] D. Zhai, M. Hao and J. M. Mendel, A non-singleton interval type-2 fuzzy logic system for universal image noise removal using quantum-behaved particle swarm optimization, *IEEE International Conference on Fuzzy Systems*, pp.957-964, 2011.
[12] J. M. Mendel, General type-2 fuzzy logic systems made simple: A tutorial, *IEEE Trans. Fuzzy Systems*, vol.22, no.5, pp.1162-1182, 2014.