Int. J. Fuzzy Syst. (2018) 20(3):835–846
https://doi.org/10.1007/s40815-017-0357-3

Design and Application of Interval Type-2 TSK Fuzzy Logic System Based on QPSO Algorithm

Qiu-feng Fan[1] · Tao Wang[1] · Yang Chen[1] · Zhi-feng Zhang[1]

Received: 2 March 2017 / Revised: 22 May 2017 / Accepted: 13 July 2017 / Published online: 31 July 2017

Abstract Studies on interval type-2 TSK fuzzy logic system is a hot topic in the current academic area. Parameter identification is very important for system design. Commonly used parameter identification methods are the least-square algorithm, BP algorithm, etc., few scholars use QPSO algorithm for parameter identification. In this paper, we propose a design of interval type-2 TSK fuzzy logic system based on quantum behaved particle swarm optimization (QPSO) intelligent algorithm. Firstly, by combining the A1–C1, A2–C0, A2–C1 interval type-2 TSK fuzzy logic system with neural network, the fuzzy neural network system is designed. Then, the QPSO intelligent algorithm was used to tune the fuzzy neural network system parameters, and the designed system model is applied to predict the Nasdaq Composite Index and International Gold Prices. Both QPSO algorithm and BP algorithm have been used to train the system model. By comparing QPSO algorithm and BP algorithm, the performance index and simulation results illustrated the proposed model is effective and feasible, which can achieve a better performance. Finally, compare the performance of the four fuzzy logic systems, it can be seen the effect of A2–C1 fuzzy logic system is better than that of the other three fuzzy logic systems.

Keywords Fuzzy logic system · Neural network · QPSO algorithm · BP algorithm · Nasdaq Composite Index · International Gold Prices

✉ Tao Wang
lxywt@lnut.edu.cn

[1] College of Science, Liaoning University of Technology, Jinzhou 121001, China

1 Introduction

In 1975, Zadeh proposed the definition and operation of type-2 fuzzy sets [1]. 1976–2002, M. Mizumoto and K. Tanaka, R. I. John, Karnik and J. M. Mendel studied the properties of type-2 fuzzy sets [2–5]. In 1985, Takagi and Sugeno design T–S model [6]. In the complex, nonlinear, multivariable system modeling and control, the superiority of the application is gradually shown. In 1988, Sugeno and Kang extend this method by adding an unbiasedness constraint, get the TSK model [7]. At this stage, fuzzy theory has been widely used in many fields, such as chemical plant, subway, robot and so on [8–10]. In 1999, J. M. Mendel proposed type-2 fuzzy logic and fuzzy logic system, forming the theoretical framework of the type-2 fuzzy logic system [11, 12]. In 2007, O. Castillo and P. Melin have done a lot of work on the fuzzy theory and the application of interval type-2 fuzzy logic system to control problems. At the same time, the use of computer software technology (genetic algorithm, neural network, etc.), as well as software systems, greatly improve the convenience and practicality of the application [13, 14]. The application of type-2 fuzzy logic can better deal with the uncertainty in the language, thus reducing the uncertainty in the system. The type-2 TSK fuzzy system is extended from the corresponding type-1 TSK fuzzy system. The type-2 TSK fuzzy system has the ability of universal approximation and good robustness and has the ability to model complex input–output relations. In recent years, it has been successfully applied in many fields, such as control, prediction [15–18], communication and so on [19–29]. Recent theoretical and practical studies confirm that IT2FLSs handle uncertainties more appropriately than their type-1 counterparts [30]. Due to the complexity and difficulty of the ordinary type-2 TSK fuzzy system, the application of interval type-2 TSK fuzzy system can greatly reduce the computational complexity.

In 1995, particle swarm optimization (PSO) algorithm is a population-based optimization algorithm proposed by Kennedy and Eberhart [31, 32]. Because the PSO algorithm is easy to fall into the local optimal solution, Sun et al. [33–35] proposed a QPSO algorithm based on the standard PSO in 2004. QPSO algorithm is a new model of particle swarm optimization algorithm from the point of view of quantum mechanics. The model think the particle 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, Lian et al. [36] applied the QPSO algorithm to the radial basis function neural network. In 2011, Zhai et al. [37] applied the QPSO algorithm to a non-singleton interval type-2 fuzzy logic system of universal image noise removal. In 2013, Pan et al. [38] applied the QPSO algorithm to the hybrid fuzzy neural network. In 2015, Wei Wang used adaptive QPSO algorithm to train radial basis function neural network, and applied it to traffic flow prediction [39]. In 2016, Zhu et al. [40] combine QPSO algorithm and radial basis function neural network and apply it to short-term power load forecasting. They found that through the optimization of the QPSO algorithm, the model has high prediction accuracy and stability. In 2016, Liu et al. [41] used QPSO algorithm to optimize the fuzzy neural network, and applied to the digital simulation. In the same year, Li et al. [42] used QPSO algorithm to optimize RBF neural network and applied to text classification. It is found that the QPSO algorithm improves the learning speed, accuracy and robustness. In 2014, Mendel et al. [43] proposed using the QPSO algorithm to train the fuzzy logic system. At present, the design of the fuzzy logic system is mostly using the BP algorithm, the least-square algorithm and the hybrid algorithm, etc. The least-squares method only used to tune the consequent parameters. The BP algorithm needs to compute the partial derivative, which is more complex, and cannot guarantee the global convergence. The convergence speed of genetic algorithm is slow when dealing with complex optimization problems. The PSO algorithm is easy to fall into the local optimal solution. The QPSO intelligent algorithm does not need to adjust the velocity vector, and the parameters need to be adjusted are few, the form is simple, the precision is high, the convergence speed is fast, and it has strong global convergence. Although the QPSO intelligent algorithm in dealing with complex optimization problems, the stability is relatively poor, but it has a strong global convergence and can find the global optimal solution of the optimization problem with fewer iterations and less time consuming to have the same solution precision and other algorithms. At present, scholars have used intelligent algorithm to design the system [44, 45], but few scholars use QPSO intelligent algorithm to design fuzzy logic system and apply it to prediction. By combining with the TSK fuzzy logic system and the neural network, this paper designed six 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 Nasdaq Composite Index and International Gold Prices with the impact of uncertainty. 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 Sect. 2, we present the principle of QPSO algorithm. In Sect. 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 Nasdaq Composite Index and International Gold Prices. Finally, the paper is concluded in Sect. 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 $X_{i,n} = (X_{i,n}^1, X_{i,n}^2, \ldots, X_{i,n}^D)$ and $V_{i,n} = (V_{i,n}^1, V_{i,n}^2, \ldots, V_{i,n}^D)$ [31, 32]. The particle moves according to the following equations:

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

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

where $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, D$, 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, $r_{i,n}^j, R_{i,n}^j \sim U(0,1)$, $V_{i,n}^j \in [-V_{\max}, V_{\max}]$. $P_{i,n} = (P_{i,n}^1, P_{i,n}^2, \ldots, P_{i,n}^D)$ is the best previous position of particle i, $G_n = (G_n^1, G_n^2, \ldots, G_n^D)$ is the global best position.

In order to overcome the shortcoming that PSO algorithm is easy to fall into local optimal solution, QPSO algorithm is proposed [33]. In the QPSO algorithm, the position and velocity vector of each particle is replaced by the wave function. The population size of the particle is M, and the position of the particle is updated according to the following equations:

$$X_{i,n+1}^{j} = P_{i,n}^{j} \pm \alpha \cdot \left|X_{i,n}^{j} - C_{i,n}^{j}\right| \cdot \ln\left(\frac{1}{u_{i,n+1}^{j}}\right) \quad (3)$$

$$C_{n}^{j} = \left(\frac{1}{M}\right)\sum_{i=1}^{M} p_{i,n}^{j} \quad (1 \le j \le D) \quad (4)$$

$$P_{i,n}^{j} = \varphi_{i,n}^{j} \cdot P_{i,n}^{j} + (1 - \varphi_{i,n}^{j}) \cdot G_{n}^{j} \quad (5)$$

where $C_n = (C_n^1, C_n^2, \ldots, C_n^D)$ is mean best position defined by the average of the personal best positions among all the particles, $\varphi_{i,n}^{j}$ and $u_{i,n+1}^{j}$ are random numbers uniformly distributed between (0, 1). The parameter α is contraction–expansion (CE) coefficient, and the convergence speed is controlled.

3 Design of Interval Type-2 TSK Fuzzy Logic System

3.1 System Model

3.1.1 A1–C1 Interval Type-2 Fuzzy Logic System

The A1–C1 interval type-2 fuzzy logic system having n inputs and one output, we assume there are M rules, the kth rule can be expressed as:

$$R^k\text{: IF} \quad x_1 \text{ is } F_1^k \quad \text{and} \quad x_2 \text{ is } F_2^k \quad \text{and} \cdots \text{and} \quad x_n \text{ is } F_n^k,$$
$$\text{THEN } Y^k \text{ is } C_0^k + C_1^k x_1 + C_2^k x_2 + \cdots + C_n^k x_n \qquad k = 1, \ldots, M$$

where $F_1^k, F_2^k, \ldots, F_n^k$ are the membership functions of the antecedent, the type-1 fuzzy set. Y^k is the consequent of the kth rule, $C_0^k, C_1^k, \ldots, C_n^k$ are type-1 fuzzy set of the consequent parameters. $C_i^k = [c_i^k - s_i^k, c_i^k + s_i^k]$ $(i = 0, 1\ldots, n)$, c_i^k is the center of C_i^k, s_i^k is the span of C_i^k.

The fuzzy logic system is integrated into the neural network to get the six layers fuzzy neural network system, the system structure is as follows:

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

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

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

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

$$F^k = \prod_{i=1}^{n} \mu_{F_i^k}(x_i)$$

Layer 4: The weight layer: calculate the weight of rules:

$$\omega^k = \frac{F^k}{\sum_{k=1}^{M} F^k}$$

Layer 5: The rules' consequent layer:

$$Y^k = \left[y_l^k, y_r^k\right]$$

where $y_l^k = \sum_{i=1}^{n} c_i^k x_i + c_0^k - \sum_{i=1}^{n} |x_i| s_i^k - s_0^k$, $y_r^k = \sum_{i=1}^{n} c_i^k x_i + c_0^k + \sum_{i=1}^{n} |x_i| s_i^k + s_0^k$

Layer 6: Output layer: y is the output of the system

$$y = f_s(X) = \frac{y_l + y_r}{2} = \frac{\sum_{k=1}^{M} \omega^k \left(\sum_{i=1}^{n} c_i^k x_i + c_0^k\right)}{\sum_{k=1}^{M} \omega^k}.$$

3.1.2 A2–C0 Interval Type-2 Fuzzy Logic System

The A2–C0 interval type-2 fuzzy logic system having n inputs and one output, we assume there are M rules, the kth rule can be expressed as:

$$R^k\text{: IF} \quad x_1 \text{ is } \tilde{F}_1^k \quad \text{and} \quad x_2 \text{ is } \tilde{F}_2^k \quad \text{and} \cdots \text{and} \quad x_n \text{ is } \tilde{F}_n^k,$$
$$\text{THEN } Y^k \text{ is } p_0^k + p_1^k x_1 + p_2^k x_2 + \cdots + p_n^k x_n \qquad k = 1, \ldots, M$$

where $\tilde{F}_1^k, \tilde{F}_2^k, \ldots, \tilde{F}_n^k$ are the membership functions of the antecedent, the interval type-2 fuzzy set. Y^k is the consequent of the kth rule, $p_0^k, p_1^k, \ldots, p_n^k$ are the consequent parameters.

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

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

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

$$\tilde{\mu}_{\tilde{F}_i^k}(x_i) = \left[\underline{\mu}_{\tilde{F}_i^k}, \overline{\mu}_{\tilde{F}_i^k}\right] \quad i = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, M$$

where $\underline{\mu}_{\tilde{F}_i^k}(x_i) = \exp\left\{-\frac{1}{2}\left(\frac{x_i - \underline{m}_{\tilde{F}_i^k}}{\underline{\sigma}_{\tilde{F}_i^k}}\right)^2\right\}$, $\overline{\mu}_{\tilde{F}_i^k}(x_i) = \exp\left\{-\frac{1}{2}\left(\frac{x_i - \overline{m}_{\tilde{F}_i^k}}{\overline{\sigma}_{\tilde{F}_i^k}}\right)^2\right\}$

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

$$F^k = \left[\underline{F}^k, \overline{F}^k\right]$$

where $\underline{F}^k = \prod_{i=1}^{n} \underline{\mu}_{\tilde{F}_i^k}(x_i)$, $\overline{F}^k = \prod_{i=1}^{n} \overline{\mu}_{\tilde{F}_i^k}(x_i)$

Layer 4: The weight layer: calculate the weight of rules:

$$\underline{f}^k = \frac{\underline{F}^k}{\sum_{k=1}^{M} \underline{F}^k}, \quad \overline{f}^k = \frac{\overline{F}^k}{\sum_{k=1}^{M} \overline{F}^k}$$

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

$$y = f_{\mathrm{s}}(X) = \sum_{k=1}^{M} f^k Y^k.$$

Y^k is a linear function of $x_1, x_2, \ldots, x_n$, $Y^k = p_0^k + p_1^k x_1 + p_2^k x_2 + \cdots + p_n^k x_n$

3.1.3 A2–C1 Interval Type-2 Fuzzy Logic System

The A2–C1 interval type-2 fuzzy logic system having n inputs and one output, we assume there are M rules, the kth rule can be expressed as:

$$R^k\text{: IF } x_1 \text{ is } \tilde{F}_1^k \text{ and } x_2 \text{ is } \tilde{F}_2^k \text{ and} \cdots \text{and } x_n \text{ is } \tilde{F}_n^k,$$
$$\text{THEN } Y^k \text{ is } C_0^k + C_1^k x_1 + C_2^k x_2 + \cdots + C_n^k x_n \qquad k = 1, \ldots, M$$

where $\tilde{F}_1^k, \tilde{F}_2^k, \ldots, \tilde{F}_n^k$ are the membership functions of the antecedent, the interval type-2 fuzzy set. Y^k is the consequent of the kth rule, C_0^k, C_1^k, …, C_n^k are type-1 fuzzy set of the consequent parameters. $C_i^k = [c_i^k - s_i^k,\ c_i^k + s_i^k]$ $(i = 0, 1\ldots, n)$, c_i^k is the center of C_i^k, s_i^k is the span of C_i^k.

The fuzzy logic system is integrated into the neural network to get the six layers fuzzy neural network system, the system structure is as follows:

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

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

$$\tilde{\mu}_{\tilde{F}_i^k}(x_i) = \left[\underline{\mu}_{\tilde{F}_i^k}, \overline{\mu}_{\tilde{F}_i^k}\right] \quad i = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, M$$

where $\underline{\mu}_{\tilde{F}_i^k}(x_i) = \exp\left\{-\frac{1}{2}\left(\frac{x_i - \underline{m}_{\tilde{F}_i^k}}{\underline{\sigma}_{\tilde{F}_i^k}}\right)^2\right\}$, $\overline{\mu}_{\tilde{F}_i^k}(x_i) = \exp\left\{-\frac{1}{2}\left(\frac{x_i - \overline{m}_{\tilde{F}_i^k}}{\overline{\sigma}_{\tilde{F}_i^k}}\right)^2\right\}$

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

$$F^k = \left[\underline{F}^k, \overline{F}^k\right]$$

where $\underline{F}^k = \prod_{i=1}^{n} \underline{\mu}_{\tilde{F}_i^k}(x_i)$, $\overline{F}^k = \prod_{i=1}^{n} \overline{\mu}_{\tilde{F}_i^k}(x_i)$

Layer 4: The weight layer: calculate the weight of rules:

$$\underline{f}^k = \frac{\underline{F}^k}{\sum_{k=1}^{M} \underline{F}^k}, \quad \overline{f}^k = \frac{\overline{F}^k}{\sum_{k=1}^{M} \overline{F}^k}$$

Layer 5: The rules' consequent layer:

$$Y^k = \left[y_l^k, y_r^k\right]$$

where $y_l^k = \sum_{i=1}^{n} c_i^k x_i + c_0^k - \sum_{i=1}^{n} |x_i| s_i^k - s_0^k$, $y_r^k = \sum_{i=1}^{n} c_i^k x_i + c_0^k + \sum_{i=1}^{n} |x_i| s_i^k + s_0^k$

Layer 6: Output layer: y is the output of the system.

$$y = f_{\mathrm{s}}(X) = \frac{y_l + y_r}{2}$$

where

$$y_l = \frac{\sum_{k=1}^{M} \underline{f}^k y_l^k}{\sum_{k=1}^{M} \underline{f}^k} = \frac{\sum_{k=1}^{L} \overline{f}^k y_l^k + \sum_{k=L+1}^{M} \underline{f}^k y_l^k}{\sum_{k=1}^{L} \overline{f}^k + \sum_{k=L+1}^{M} \underline{f}^k}$$

$$y_r = \frac{\sum_{k=1}^{M} \overline{f}^k y_r^k}{\sum_{k=1}^{M} \overline{f}^k} = \frac{\sum_{k=1}^{R} \underline{f}^k y_r^k + \sum_{k=R+1}^{M} \overline{f}^k y_r^k}{\sum_{k=1}^{R} \underline{f}^k + \sum_{k=R+1}^{M} \overline{f}^k}$$

R, L are found by using KM algorithm [46], and then get y_l, y_r.

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 logic system, the three premise parameters are $\left\{m_{F_i^k}, \sigma_{F_i^k}\right\}$, $\left\{\underline{m}_{\tilde{F}_i^k}, \overline{m}_{\tilde{F}_i^k}, \underline{\sigma}_{\tilde{F}_i^k}, \overline{\sigma}_{\tilde{F}_i^k}\right\}$, $\left\{\underline{m}_{\tilde{F}_i^k}, \overline{m}_{\tilde{F}_i^k}, \underline{\sigma}_{\tilde{F}_i^k}, \overline{\sigma}_{\tilde{F}_i^k}\right\}$, respectively. The consequent parameters are $\{c_0^k, c_1^k, \ldots, c_n^k, s_0^k, s_1^k, \ldots, s_n^k\}$, $\{p_0^k, p_1^k, \ldots, p_n^k\}$, $\{c_0^k, c_1^k, \ldots, c_n^k, s_0^k, s_1^k, \ldots, s_n^k\}$, respectively. 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 error between actual output and desired output, which can be expressed by

$$E_j = \frac{1}{2} * [f_{\mathrm{s}}(X^j) - Y^j]^2 \tag{6}$$

And the performance index of the system is root-mean-square error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{K} \sum_{j=1}^{K} (f_s(X^j) - Y^j)^2} \tag{7}$$

where Y^j is the desired output, and $f_{\mathrm{s}}(X^j)$ is the output of the system.

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 are uniformly distributed within the search scope, 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. 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 (4).

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

Step 5: Calculate each particle's fitness value, $E(X_{i,n})$, and then compare it with the fitness of its personal best position, $E(P_{i,n-1})$. If $E(X_{i,n}) < E(P_{i,n-1})$, 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, $E(P_{i,n})$ with that of the global best position, $E(G_{n-1})$. If $E(P_{i,n}) < E(G_{n-1})$, 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 (3) 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

4.1 The Application of System Model in Prediction of the Nasdaq Composite Index

The Nasdaq Composite Index is the average stock price index that reflects the changes in the Nasdaq stock market. The index is one of the most representative of the US stock market index, is a representative of the various sectors of the market value of the barometer of change. Currently, the index includes more than 5 thousand companies, more than any other single securities market. It has become one of the most influential indexes of the stock market because of its wide base. In this example, the Nasdaq Composite Index were selected from July 1, 2015 to April 30, 2016, a total of 210 data. Using Matlab to draw the 210 data of the noiseless sequence diagram is shown in Fig. 1.

In this paper, we use the first four data to predict the next data, that is, four inputs $X = (x_1, x_2, x_3, x_4)^{\mathrm{T}}$ and one output $y \in Y$. 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 $M = 16$, the kth rules of the three fuzzy logic systems are as follows:

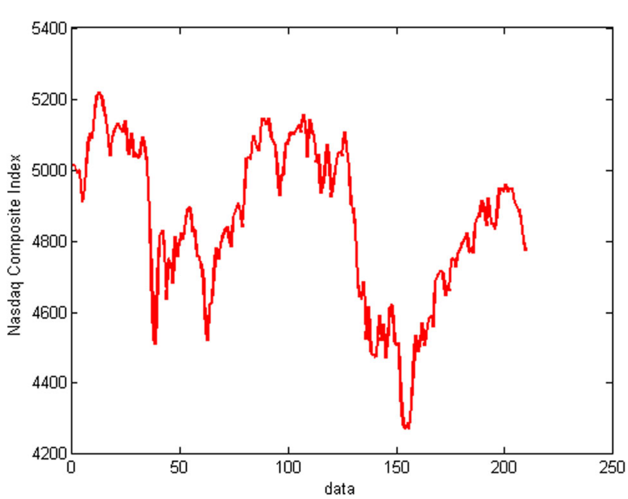

Fig. 1 The diagram of Nasdaq Composite Index

$$\begin{aligned} &R^k\text{: IF}\quad x_1 \text{ is } F_1^k \quad \text{and} \quad x_2 \text{ is } F_2^k \quad \text{and} \cdots \text{and} \quad x_4 \text{ is } F_4^k,\\ &\text{THEN } Y^k \text{ is } C_0^k + C_1^k x_1 + C_2^k x_2 + \cdots + C_4^k x_4 \qquad k = 1, \ldots, 16 \end{aligned}$$

where $F_1^k, F_2^k, F_3^k, F_4^k$ are the membership functions of the antecedent, the type-1 fuzzy set. Y^k is the consequent of the kth rule, $C_0^k, C_1^k, \ldots, C_4^k$ are type-1 fuzzy set of the consequent parameters. $C_i^k = [c_i^k - s_i^k, c_i^k + s_i^k]$ $(i = 0, 1\ldots, 4)$, c_i^k is the center of C_i^k, s_i^k is the span of C_i^k.

$$\begin{aligned} &R^k\text{: IF}\quad x_1 \text{ is } \tilde{F}_1^k \quad \text{and} \quad x_2 \text{ is } \tilde{F}_2^k \quad \text{and} \cdots \text{and} \quad x_4 \text{ is } \tilde{F}_4^k,\\ &\text{THEN } Y^k \text{ is } p_0^k + p_1^k x_1 + p_2^k x_2 + \cdots + p_4^k x_4 \qquad k = 1, \ldots, 16 \end{aligned}$$

where $\tilde{F}_1^k, \tilde{F}_2^k, \tilde{F}_3^k, \tilde{F}_4^k$ are the membership functions of the antecedent, the interval type-2 fuzzy set. Y^k is the consequent of the kth rule, $p_0^k, p_1^k, \ldots, p_4^k$ are the consequent parameters.

$$\begin{aligned} &R^k\text{: IF}\quad x_1 \text{ is } \tilde{F}_1^k \quad \text{and} \quad x_2 \text{ is } \tilde{F}_2^k \quad \text{and} \cdots \text{and} \quad x_4 \text{ is } \tilde{F}_4^k,\\ &\text{THEN } Y^k \text{ is } C_0^k + C_1^k x_1 + C_2^k x_2 + \cdots + C_4^k x_4 \qquad k = 1, \ldots, 16 \end{aligned}$$

where $\tilde{F}_1^k, \tilde{F}_2^k, \tilde{F}_3^k, \tilde{F}_4^k$ are the membership functions of the antecedent, the interval type-2 fuzzy set. Y^k is the consequent of the kth rule, $C_0^k, C_1^k, \ldots, C_4^k$ are type-1 fuzzy set of the consequent parameters.

The Nasdaq Composite Index consists of 210 data, which constitute the input–output data pairs of the 206 groups. The first 126 groups are used to design the intelligent system, and after 80 groups are used to test the system. The steps of tuning fuzzy logical system parameters based on the Sect. 3.2 QPSO algorithm, set the maximum iteration $N_{\max} = 200$, α decreased linearly from 2 to 1.5, the number of particles are 416. The initial particles are given as follows (Table 1).

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 Figs. 2, 3 and 4. It shows that the tracking effect is very good. The after 80 groups are used to test the performance of the designed system, tracking results are shown in Figs. 5, 6 and 7.

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 algorithm, Figs. 2, 3, 4, 5, 6 and 7 are the training and tracking performance of the two algorithms, respectively.

Table 1 Initial particles of three fuzzy logic systems for forecasting the Nasdaq Composite Index

A1–C1	A2–C0	A2–C1
$m = 200 + \text{rands}\ (16, 4)$	$m = 199 + \text{rands}\ (16, 4)$	$m = 199 + \text{rands}\ (16, 4)$
$\sigma = 538 + \text{rands}\ (16, 4)$	$\sigma = 538 + \text{rands}\ (16, 4)$	$\sigma = 538 + \text{rands}\ (16, 4)$
$c_0 = 0.3 * \text{rands}\ (16, 1)$	$\overline{m} = 200 + \text{rands}\,(16, 4)$	$\overline{m} = 200 + \text{rands}\,(16, 4)$
$c_1 = 0.3 * \text{rands}\ (16, 1)$	$\overline{\sigma} = 540 + \text{rands}\,(16, 4)$	$\overline{\sigma} = 540 + \text{rands}\,(16, 4)$
$\vdots$	$p_0 = 0.3 * \text{rands}\ (16, 1)$	$c_0 = 0.3 * \text{rands}\ (16, 1)$
$c_4 = 0.3 * \text{rands}\ (16, 1)$	$p_1 = 0.3 * \text{rands}\ (16, 1)$	$\vdots$
$s_0 = 0.3 * \text{rands}\ (16, 1)$	$\vdots$	$c_4 = 0.3 * \text{rands}\ (16, 1)$
$s_1 = 0.3 * \text{rands}\ (16, 1)$	$p_4 = 0.3 * \text{rands}\ (16, 1)$	$s_0 = 0.3 * \text{rands}\ (16, 1)$
$\vdots$		$\vdots$
$s_4 = 0.3 * \text{rands}\ (16, 1)$		$s_4 = 0.3 * \text{rands}\ (16, 1)$

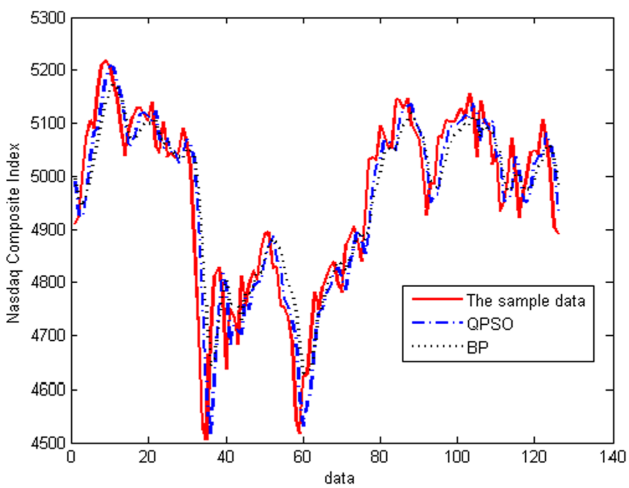

Fig. 2 Results of training data for A1–C1

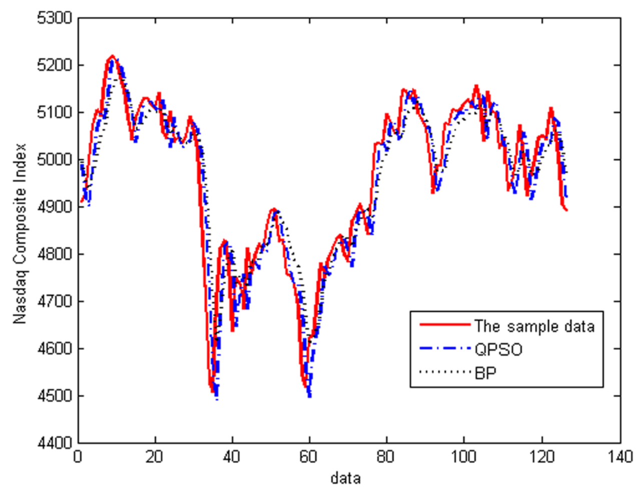

Fig. 4 Results of training data for A2–C1

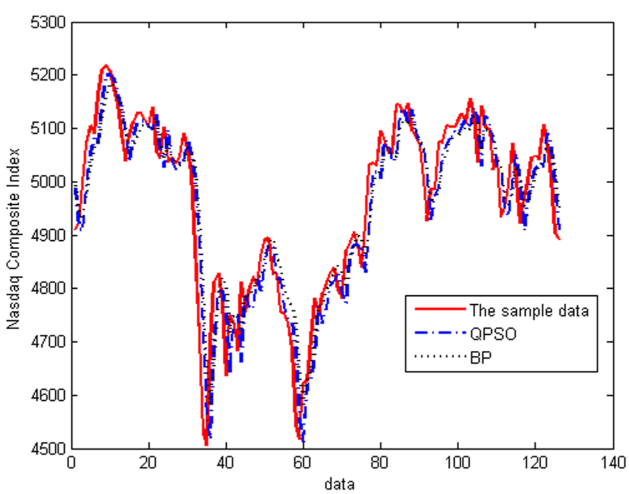

Fig. 3 Results of training data for A2–C0

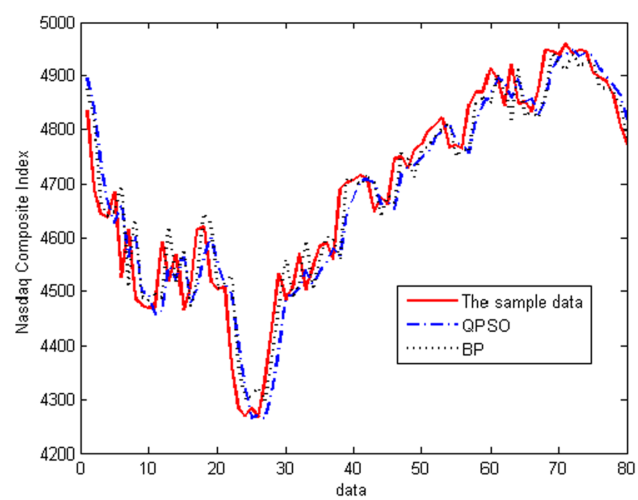

Fig. 5 Results of testing data for A1–C1

After 200 iterations, the *RMSE* of the two algorithms are obtained. The training errors and testing errors of QPSO and BP algorithms are shown in Tables 2 and 3.

It can be seen from Figs. 2, 3, 4, 5, 6 and 7 and Tables 2, 3, it is better to use the QPSO algorithm than the BP algorithm to tune the parameters of the fuzzy logic

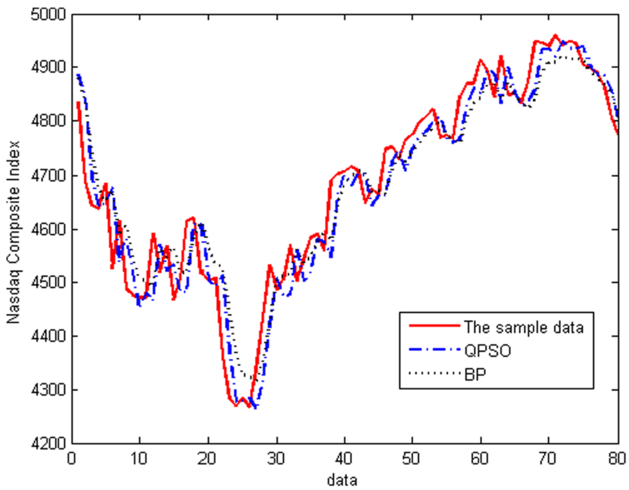

Fig. 6 Results of testing data for A2–C0

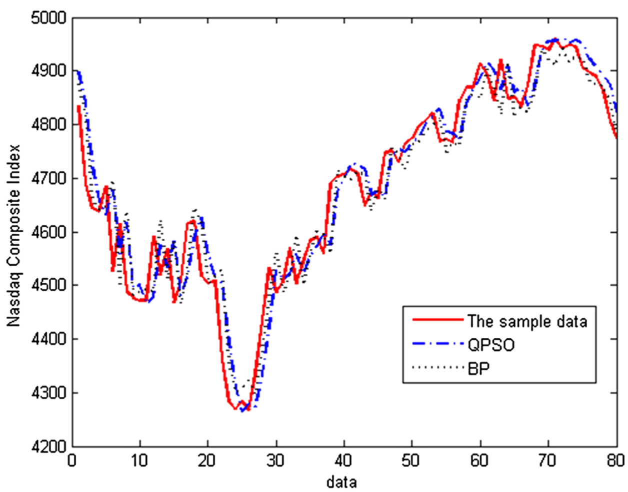

Fig. 7 Results of testing data for A2–C1

Table 2 The training errors of QPSO and BP algorithms

	A1–C1	A2–C0	A2–C1
QPSO	68.2833	66.5269	64.7974
BP	78.3707	74.3280	73.1950

Table 3 The testing errors of QPSO and BP algorithms

	A1–C1	A2–C0	A2–C1
QPSO	60.2879	59.9996	59.2354
BP	61.8134	60.6652	60.3306

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.

4.2 The Application of System Model in Prediction of 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. Using Matlab to draw the 237 data of the noiseless sequence diagram is shown in Fig. 8.

In this paper, we use the first three data to predict the next data, that is, three inputs $X = (x_1, x_2, x_3)^{\mathrm{T}}$ and one output $y \in Y$. 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 $M = 8$, the kth rules of the three fuzzy logic systems are as follows:

$$R^k:\quad \text{IF}\quad x_1 \text{ is } F_1^k \quad\text{and}\quad x_2 \text{ is } F_2^k \quad\text{and}\quad x_3 \text{ is } F_3^k,$$
$$\text{THEN } Y^k \text{ is } C_0^k + C_1^k x_1 + C_2^k x_2 + C_3^k x_3 \quad k = 1, \ldots, 8$$

where F_1^k, F_2^k, F_3^k are the membership functions of the antecedent, the type-1 fuzzy set. Y^k is the consequent of the kth rule, C_0^k, C_1^k, …, C_3^k are type-1 fuzzy set of the consequent parameters. $C_i^k = [c_i^k - s_i^k, c_i^k + s_i^k]$ $(i = 0, 1\ldots, 3)$, c_i^k is the center of C_i^k, s_i^k is the span of C_i^k

$$R^k:\quad \text{IF}\quad x_1 \text{ is } \tilde{F}_1^k \quad\text{and}\quad x_2 \text{ is } \tilde{F}_2^k \quad\text{and}\quad x_3 \text{ is } \tilde{F}_3^k,$$
$$\text{THEN } Y^k \text{ is } p_0^k + p_1^k x_1 + p_2^k x_2 + p_3^k x_3 \quad k = 1, \ldots, 8$$

where $\tilde{F}_1^k, \tilde{F}_2^k, \tilde{F}_3^k$ are the membership functions of the antecedent, the interval type-2 fuzzy set. Y^k is the consequent of the kth rule, p_0^k, p_1^k, …, p_3^k are the consequent parameters

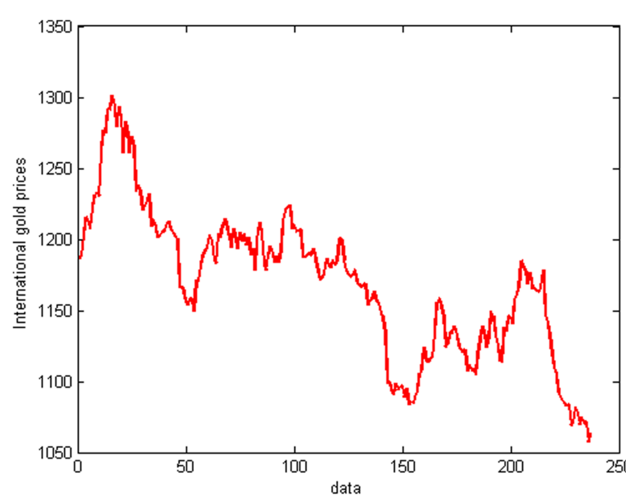

Fig. 8 The diagram of International Gold Prices

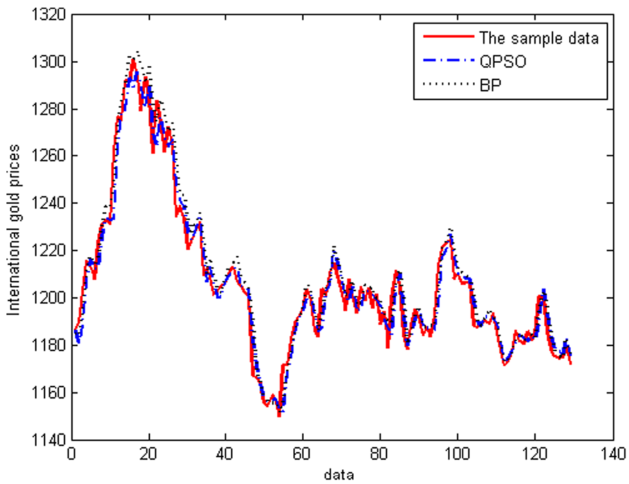

Fig. 9 Results of training data for A1–C1

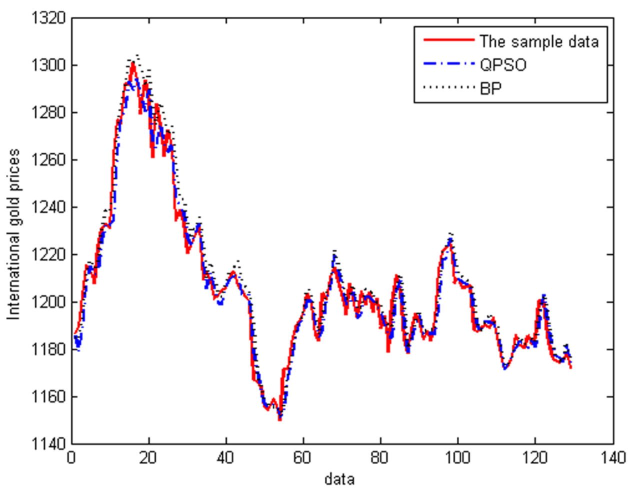

Fig. 10 Results of training data for A2–C0

$$R^k:\quad \text{IF}\quad x_1 \text{ is } \tilde{F}_1^k \quad \text{and} \quad x_2 \text{ is } \tilde{F}_2^k \quad \text{and} \quad x_3 \text{ is } \tilde{F}_3^k,$$
$$\text{THEN } Y^k \text{ is } C_0^k + C_1^k x_1 + C_2^k x_2 + C_3^k x_3 \quad k = 1, \ldots, 8$$

where $\tilde{F}_1^k, \tilde{F}_2^k, \tilde{F}_3^k$ are the membership functions of the antecedent, the interval type-2 fuzzy set. Y^k is the consequent of the kth rule, C_0^k, C_1^k, …, C_3^k are type-1 fuzzy set of the consequent parameters.

The International Gold Prices consists of 237 data, which constitute the input–output data pairs of the 234 groups. The first 129 groups are used to design the intelligent system, and after 105 groups are used to test the system. The steps of tuning fuzzy logical system parameters based on the Sect. 3.2 QPSO algorithm, set the maximum iteration $N_{\max} = 200$, α decreased linearly from 2 to 1.7, the number of particles are 160. The initial particles are given as follows:

Using the previously designed QPSO algorithm to tune these initial particles, the QPSO algorithm training output

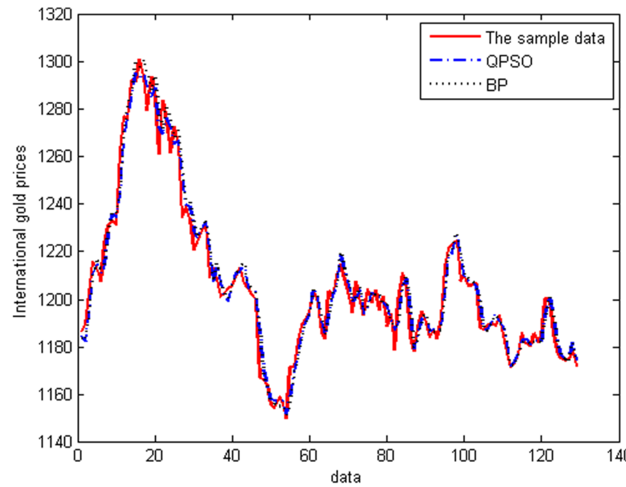

Fig. 11 Results of training data for A2–C1

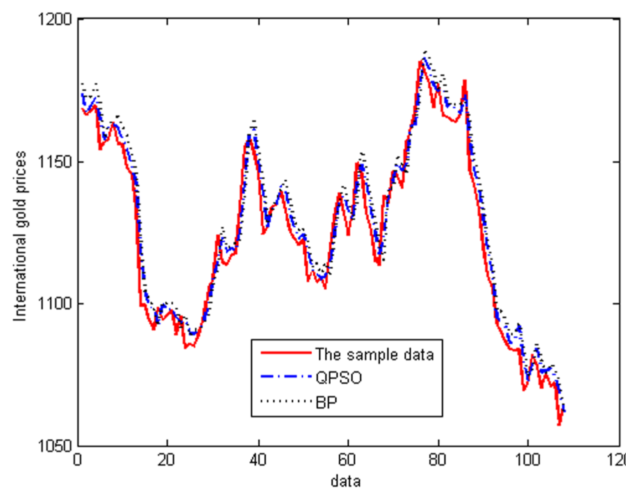

Fig. 12 Results of testing data for A1–C1

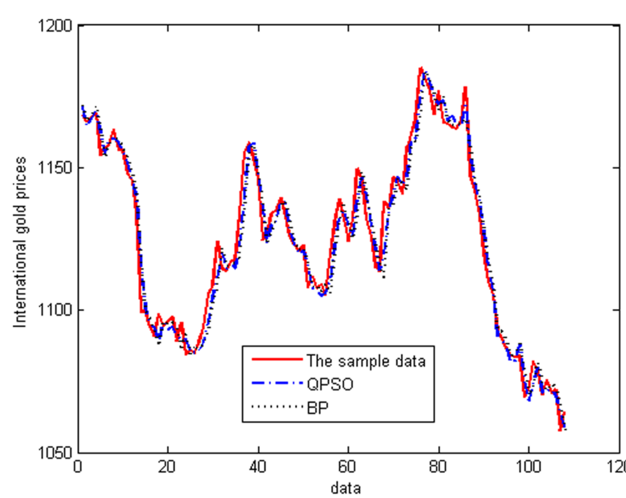

Fig. 13 Results of testing data for A2–C0

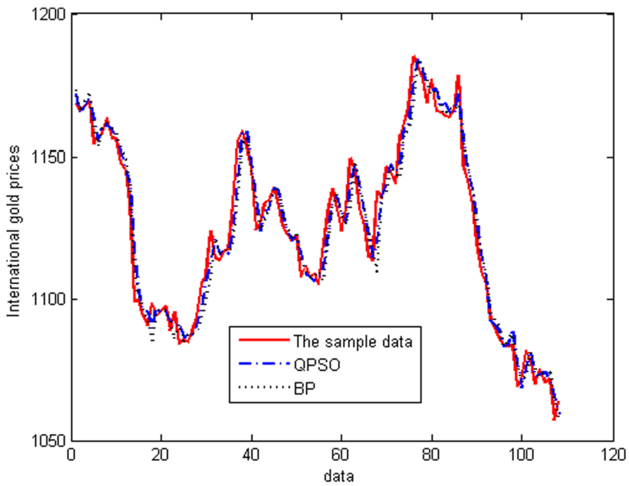

Fig. 14 Results of testing data for A2–C1

and the actual output of the system are shown in Figs. 9, 10 and 11. It shows that the tracking effect is very good. The after 105 groups are used to test the performance of the designed system, tracking results are shown in Figs. 12, 13 and 14.

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 algorithm, Figs. 9, 10, 11, 12, 13 and 14 are the training and tracking performance of the two algorithms, respectively (Table 4).

After 200 iterations, the *RMSE* of the two algorithms are obtained. The training errors and testing errors of QPSO and BP algorithms are shown in Tables 5 and 6.

It can be seen from Figs. 9, 10, 11, 12, 13 and 14 and Tables 5, 6, 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.

Table 5 The training errors of QPSO and BP algorithms

	A1–C1	A2–C0	A2–C1
QPSO	7.0231	6.5526	5.8036
BP	7.4403	6.9235	6.2673

Table 6 The testing errors of QPSO and BP algorithms

	A1–C1	A2–C0	A2–C1
QPSO	6.3309	5.7393	5.6824
BP	8.5044	7.5304	7.1341

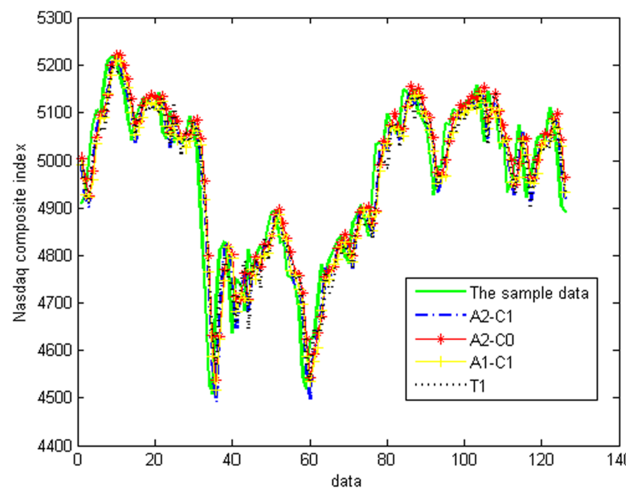

Fig. 15 The contrast result of training in Nasdaq Composite Index

4.3 Performance Comparison of Fuzzy Logic Systems

Design the Intelligent system based on the 4.1 and 4.2. Using the same system structure, Gaussian membership function, singleton fuzzifier, product inference and the same number of rules. The design of Type-1, A1–C1, A2–C0 and A2–C1 intelligent system is applied in the

Table 4 Initial particles of three fuzzy logic systems for forecasting the International Gold Prices

A1–C1	A2–C0	A2–C1
$m = 201 + 0.35 * \text{rands}\ (8,\ 3)$	$m = 199 + 0.35 * \text{rands}\ (8,\ 3)$	$m = 199 + 0.35 * \text{rands}\ (8,\ 3)$
$\sigma = 538 + 0.35 * \text{rands}\ (8,\ 3)$	$\sigma = 538 + 0.35 * \text{rands}\ (8,\ 3)$	$\sigma = 538 + 0.35 * \text{rands}\ (8,\ 3)$
$c_0 = 0.3 * \text{rands}\ (8,\ 1)$	$\overline{m} = 201 + 0.35 * \text{rands}\,(8, 3)$	$\overline{m} = 201 + 0.35 * \text{rands}\,(8, 3)$
$c_1 = 0.3 * \text{rands}\ (8,\ 1)$	$\overline{\sigma} = 540 + 0.35 * \text{rands}\,(8, 3)$	$\overline{\sigma} = 540 + 0.35 * \text{rands}\,(8, 3)$
$\vdots$	$p_0 = 0.3 * \text{rands}\ (8,\ 1)$	$c_0 = 0.3 * \text{rands}\ (8,\ 1)$
$c_4 = 0.3 * \text{rands}\ (8,\ 1)$	$p_1 = 0.3 * \text{rands}\ (8,\ 1)$	$\vdots$
$s_0 = 0.3 * \text{rands}\ (8,\ 1)$	$\vdots$	$c_4 = 0.3 * \text{rands}\ (8,\ 1)$
$s_1 = 0.3 * \text{rands}\ (8,\ 1)$	$p_4 = 0.3 * \text{rands}\ (8,\ 1)$	$s_0 = 0.3 * \text{rands}\ (8,\ 1)$
$\vdots$		$\vdots$
$s_4 = 0.3 * \text{rands}\ (8,\ 1)$	$\vdots$	$s_4 = 0.3 * \text{rands}\ (8,\ 1)$

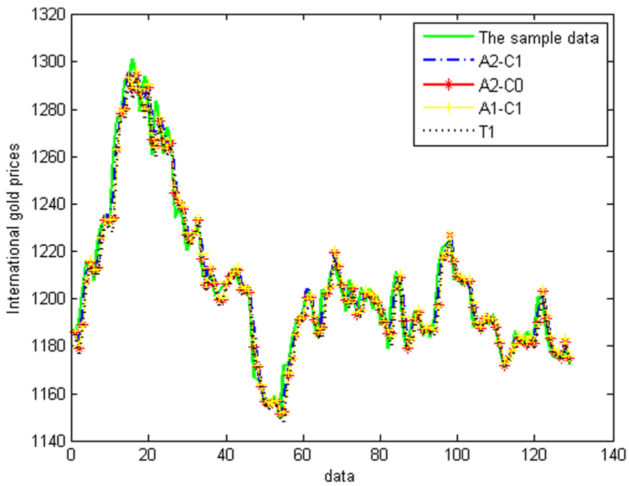

Fig. 16 The contrast result of training in International Gold Prices

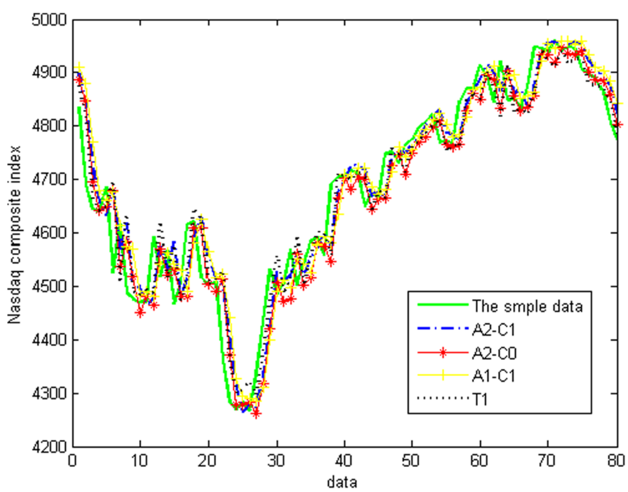

Fig. 17 The contrast result of testing in Nasdaq Composite Index

prediction of the Nasdaq Composite Index and the International Gold Prices, the training output and the actual output of the system are shown in Figs. 15 and 16. The testing output and the actual output of the system are shown in Figs. 17 and 18.

After 200 iterations, the *RMSE* of the four fuzzy logic systems are obtained. The training errors and testing errors of QPSO algorithms are shown in Tables 7 and 8.

It can be seen from Figs. 15, 16, 17 and 18 and Tables 7, 8, the effect of A2–C1 fuzzy logic system is better than that of the other three fuzzy logic systems, and the *RMSE* of QPSO algorithm is relatively small. It shows that in the interval type-2 TSK fuzzy logic system, the more parameters are adjustable, the more accurate the test results.

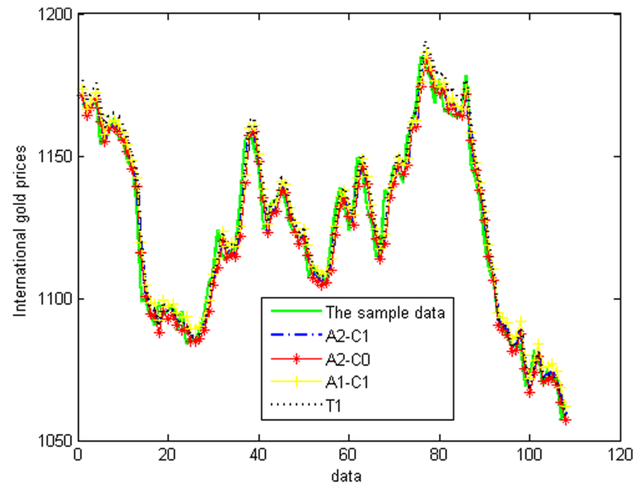

Fig. 18 The contrast result of testing in International Gold Prices

Table 7 *RMSE* of the International Gold Prices

	Training error	Testing error
T1	7.8642	6.7648
A1–C1	7.0231	6.3309
A2–C0	6.5526	5.7393
A2–C1	5.8036	5.6824

Table 8 *RMSE* of the Nasdaq Composite Index

	Training error	Testing error
T1	72.6564	61.4371
A1–C1	68.2833	60.2879
A2–C0	66.5269	59.9996
A2–C1	64.7974	59.2354

5 Conclusions

Because of the wide application of fuzzy logic system, it is very important to study the optimization algorithm of fuzzy logic system. The type-2 fuzzy logic can deal with the uncertainty in the language and reduce the uncertainty in the system, so this paper is proposed interval type-2 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 Nasdaq Composite Index and the International Gold Prices, 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. By comparing the performance of four fuzzy logic systems, it shows that the effect of A2–C1 fuzzy logic system is better than that of the other three fuzzy logic systems.

So many interesting works still lie ahead, including the researching on designing Mamdani interval type-2 fuzzy logic systems, general type-2 fuzzy logic systems, and neural network optimized by other global searching algorithms. Future studies will be concentrated on hybrid systems design, algorithms, and applications.

Acknowledgements This work is supported by the National Natural Science Foundation of China (61374113), and by Liaoning Province College Basic Scientific Research Business Funding Project (JL201615410).

References

1. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning-II. Inf. Sci. **8**(75), 301–357 (1975)
2. Mizumoto, M., Tanaka, K.: Some properties of fuzzy sets of type-2. Inf. Control **31**(4), 312–340 (1976)
3. Mizumoto, M., Tanaka, K.: Fuzzy sets of type-2 under algebraic product and algebraic sum. Fuzzy Sets Syst. **5**(3), 277–290 (1981)
4. Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets. Fuzzy Sets Syst. **122**(2), 327–348 (2001)
5. Mendel, J.M., John, R.I.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
6. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **15**(1), 387–403 (1985)
7. Sugeno, M., Kang, G.T.: Structure identification of fuzzy model. Fuzzy Sets Syst. **28**(1), 15–33 (1988)
8. Hagras, H.: A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Trans. Fuzzy Syst. **12**(4), 524–539 (2004)
9. Zeng, J., Liu, Z.Q.: Type-2 fuzzy hidden markov models and their application to speech recognition. IEEE Trans. Fuzzy Syst. **14**(3), 454–467 (2006)
10. Karnik, N.N., Mendel, J.M.: Applications of type-2 fuzzy logic systems to forecasting of time-series. Inf. Sci. **120**(1–4), 89–111 (2007)
11. Karnik, N.N., Mendel, J.M.: Introduction to type-2 fuzzy logic systems. IEEE Word Congr. IEEE Int. Conf. Fuzzy Syst. **2**(2), 915–920 (1998)
12. Karnik, N.N., Mendel, J.M., Liang, Q.: Type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **7**(6), 643–658 (1999)
13. Castillo, O., Melin, P.: A new approach for plant monitoring using type-2 fuzzy logic and fractal theory. Int. J. Gen. Syst. **33**(2), 305–319 (2015)
14. Castillo, O., Melin, P.: Intelligent systems with interval type-2 fuzzy logic. Int. J. Innov. Comput. Inf. Control **4**(4), 771–783 (2008)
15. Khosravi, A., Nahavandi, S.: Load forecasting using interval type-2 fuzzy logic systems: optimal type reduction. IEEE Trans. Ind. Inf. **10**(2), 1055–1063 (2014)
16. Chen, Y., Wang, D.Z., Tong, S.C.: Forecasting studies by designing Mamdani interval type-2 fuzzy logic systems: With the combination of BP algorithms and KM algorithms. Neurocomputing **174**(part b), 1133–1146 (2016)
17. Zarandi, M.H.F., Rezaee, B., Turksen, I.B.: A type-2 fuzzy rule-based expert system for stock price analysis. Expert Syst. Appl. **36**(1), 139–154 (2009)
18. Hassan, S., Khosravi, A., Jaafar, J., et al.: A systematic design of interval type-2 fuzzy logic system using extreme learning machine for electricity load demand forecasting. Int. J. Electr. Power Energy Syst. **82**, 1–10 (2016)
19. Banakar, A., Azeem, M.F.: Parameter identification of TSK neuro-fuzzy models. Fuzzy Sets Syst. **179**(1), 62–82 (2011)
20. Ahn, C.K.: Some new results on stability of Takagi–Sugeno fuzzy hopfield neural networks. Fuzzy Sets Syst. **179**(1), 100–111 (2011)
21. Berrios, R., Nunez, F., Cipriano, A.: Fault tolerant measurement system based on Takagi–Sugeno fuzzy models for a gas turbine in a combined cycle power plant. Fuzzy Sets Syst. **174**(1), 114–130 (2011)
22. Gao, Q., Feng, G., Wang, Y., Qiu, J.: Universal fuzzy controllers based on generalized T–S fuzzy models. Fuzzy Sets Syst. **201**(3), 55–70 (2012)
23. Wang, T., Han, C.Y., Jiang, C.J.: Design and simulation of main steam temperature controller based on interval type-2 fuzzy logic systems. ICIC Express Lett. B: Appl. Int. J. Res. Surv. **4**(5), 1359–1366 (2013)
24. Wang, T., Han, C.Y., Lan, J.: Application of interval type-2 fuzzy logic system in stock forecasting. ICIC Express Lett. Int. J. Res. Surv. **8**(11), 3015–3020 (2014)
25. Wang, T., Han, C.Y., Jin, X.L.: Design of interval type-2 fuzzy neural network system and application. ICIC Express Lett. B: Appl. Int. J. Res. Surv. **6**(4), 1041–1048 (2015)
26. Sabahi, K., Ghaemi, S., Pezeshki, S.: Gain scheduling technique using MIMO type-2 fuzzy logic system for LFC in restructure power system. Int. J. Fuzzy Syst., 1–15 (2016). doi:10.1007/s40815-016-0240-7
27. Wang, L., Liu, Z., Zhang, Y., Chen, C.L.P., Chen, X.: Type-2 fuzzy logic controller using SRUKF-based state estimations for biped walking robots. Int. J. Fuzzy Syst. **15**(4), 423–434 (2013)
28. Lee, C.H., Hsueh, H.Y.: Observer-based adaptive control for a class of nonlinear non-affine systems using recurrent-type fuzzy logic systems. Int. J. Fuzzy Syst. **15**(1), 55–65 (2013)
29. Ahammed, A.K.I.: Profoundly robust controlling strategy for uncertain nonlinear mimo system using T–S fuzzy system. Int. J. Fuzzy Syst., 1–14 (2016). doi:10.1007/s40815-016-0225-6
30. Khosravi, A., Nahavandi, S., Creighton, D., Srinivasan, D.: Interval type-2 fuzzy logic systems for load forecasting: a comparative study. IEEE Trans. Power Syst. **27**(3), 1274–1282 (2012)
31. Kennedy, J., Eberhart, R.: Particle swarm optimization. IEEE Int. Conf. Neural Netw., 1942–1948 (1995)
32. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
33. Sun, J., Feng, B., Xu, W.: Particle swarm optimization with particles having quantum behavior. IEEE Congr. Evolut. Comput. **70**(3), 1571–1580 (2004)
34. Sun, J., Liu, W.B.: A global search strategy of quantum-behaved particle swarm optimization. IEEE Conf. Cybern. Intell. Syst. **1**(11), 111–116 (2005)
35. Wei, F., Jun, S., Ping, X.Z., Xu, W.B.: Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter. Acta Phys. Sin. **59**(6), 3686–3694 (2010)
36. Lian, G.Y., Huang, K.L., Chen, J.H., Gao, F.Q.: Training algorithm for radial basis function neural network based on quantum-behaved particle swarm optimization. Int. J. Comput. Math. **87**(3), 629–641 (2010)
37. Zhai, D.Y., Hao, M.S., Mendel, J.M.: A non-singleton interval type-2 fuzzy logic system for universal image noise removal using quantum-behaved particle swarm optimization. IEEE Int. Conf. Fuzzy Syst., 957–964 (2011)
38. Min, P.Y., Hong, D.Y., Zhu, Z.Q.: Chaotic time series prediction based on QPSO–FNN. Comput. Appl. Softw. **30**(8), 91–94 (2013)

39. Wei, W.: A QPSO–RBF neural network based traffic flow prediction method. J. Taiyuan Norm. Univ. (Nat. Sci. Ed.) **14**(2), 28–32 (2015)
40. Shu, Z.Z., Ming, B.Y., Long, W.P., Peng, Z.G., Liang, Z.J.: Short-term power load forecasting model based on QPSO–RBFNN. J. Nanjing Univ. Sci. Technol. **40**(1), 97–101 (2016)
41. Liu, M., Zhang, J.Y., Wang, Y.Z.: Research on a new hybrid optimization algorithm based on QPSO and FNN. Int. J. Smart Home **10**(6), 175–186 (2016)
42. Xu, L.B., Hong, Y.J.: Document classification based on improved QPSO and RBF neural networks. Comput. Syst. Appl. **25**(7), 264–267 (2016)
43. Mendel, J.M.: General type-2 fuzzy logic systems made simple: a tutorial. IEEE Trans. Fuzzy Syst. **22**(5), 1162–1182 (2014)
44. Wong, C.C., Wang, H.Y., Li, S.A.: PSO-based motion fuzzy controller design for mobile robots. Int. J. Fuzzy Syst. **10**(1), 24–32 (2008)
45. Sambariya, D.K., Prasad, R.: Optimal tuning of fuzzy logic power system stabilizer using harmony search algorithm. Int. J. Fuzzy Syst. **17**(3), 457–470 (2015)
46. Mendel, J.M.: Type-2 fuzzy sets and systems: an overview. IEEE Comput. Intell. Mag. **2**(1), 20–29 (2007)

Qiu-feng Fan received the B.S. degree in Information and Computing Science and the M.S. degree in applied mathematics from Liaoning University of Technology, Jinzhou, China, in 2014 and 2017, respectively. Her current research interests include fuzzy reasoning and fuzzy control, type-2 fuzzy logic systems, fuzzy neural networks and system identification.

Tao Wang received the B.S. degree from Northeastern Normal University, Changchun, China, and the M.S. degree from Liaoning Normal University, Dalian, China, in 1988 and 2000, respectively. Currently, she is a professor in College of Science, Liaoning University of Technology. Her current research interests include fuzzy reasoning and fuzzy control, type-2 fuzzy logic systems, fuzzy neural networks and system identification.

Yang Chen received the B.S. degree in Information and Computing Science and the M.S. degree in applied mathematics from Liaoning University of Technology, Jinzhou, China, in 2004 and 2007, respectively. He is pursuing a Ph.D. degree in College of Information Science and Engineering, Northeastern University. He is currently a lecturer in College of Science, Liaoning University of Technology. His current research interests include fuzzy reasoning and fuzzy control, type-2 fuzzy logic systems, system identification and permanent magnetic drive.

Zhi-feng Zhang received the B.S. degree in Information and Computing Science from Liaoning University of Technology, Jinzhou, China, in 2015. Currently, he is pursuing a M.S. degree in applied mathematics in College of Science, Liaoning University of Technology. His current research interests include fuzzy reasoning and fuzzy control, type-2 fuzzy logic systems, system identification and fuzzy neural networks.