Research Article

Improved Chaotic Quantum-Behaved Particle Swarm Optimization Algorithm for Fuzzy Neural Network and Its Application

Yuexi Peng¹,^{ID} Kejun Lei²,^{ID} Xi Yang,² and Jinzhang Peng³,^{ID}

¹*School of Physics and Electronics, Central South University, Changsha 410083, China*

²*College of Information Science and Engineering, Jishou University, Jishou 416000, China*

³*College of Physics and Mechatronics Engineering, Jishou University, Jishou 416000, China*

Correspondence should be addressed to Kejun Lei; 1258489458@qq.com

Received 7 February 2020; Accepted 5 March 2020; Published 28 March 2020

Guest Editor: Kehui Sun

Copyright © 2020 Yuexi Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traditional fuzzy neural network has certain drawbacks such as long computation time, slow convergence rate, and premature convergence. To overcome these disadvantages, an improved quantum-behaved particle swarm optimization algorithm is proposed as the learning algorithm. In this algorithm, a new chaotic search is introduced, and benchmark function experiments prove it outperforms the other five existing algorithms. Finally, the proposed algorithm is presented as the learning algorithm for Takagi–Sugeno fuzzy neural network to form a new neural network, and it is utilized in the water quality evaluation of Dongjiang Lake of Hunan province. Simulation results demonstrated the effectiveness of the new neural network.

1. Introduction

Artificial neural network (ANN) [1] is an effective method to deal with nonlinear problems. Because of the powerful fitting ability, the ANN can nearly simulate any complex nonlinear functional relationship without knowing the correlation between the input and the output. At present, the ANN has been widely used in practical applications [2–5]. In fact, fuzzy neural network (FNN) [6] can handle complex engineering problem as well [7–9]. It combines the advantages of fuzzy mathematics and ANN. Due to the introduction of the fuzzy logic concept, it is very suitable for the classification of nonlinear or highly uncertain information. However, the FNN also inherits the disadvantages of the ANN, such as long computation time and slow convergence rate. Therefore, how to solve these shortcomings becomes a problem to be solved urgently [10–12].

It is an effective way to improve the FNN performance by replacing the traditional learning algorithm by metaheuristic algorithms such as the particle swarm optimization (PSO) algorithm [13–15]. However, when the considered problem

is a complex high-dimensional problem, PSO algorithm has the disadvantage of premature convergence [16–19]. After studying the results of particle convergence behavior, Sun et al. [20] proposed a novel metaheuristic algorithm called quantum-behaved particle swarm optimization (QPSO) algorithm. It combines PSO algorithm with the quantum mechanic and has better global searching ability than that of the PSO algorithm [21, 22]. Since the QPSO algorithm was proposed, many researchers devoted to apply it for practical applications [23–26] or improve the algorithm itself [27–31]. For example, Mariani et al. [29] proposed a novel chaotic QPSO algorithm for the image matching, but this algorithm only introduces chaos variables into the particle position initialization. In fact, the effectiveness should be better if the chaos variables are introduced both at the beginning and the end of the algorithm search stage. Mariani et al. [29] proposed a QPSO algorithm combined with the Zaslavskii chaotic map and applied it to the optimization of shell and tube heat exchangers. Then, Turgut et al. [30] proposed another chaotic QPSO algorithm for solving nonlinear system of equations, and the benchmark function

experiments are proved that the algorithm using Logistic chaotic map has the best performance. On the basis of Ref. [30], Turgut [31] proposed a hybrid chaotic QPSO algorithm for thermal design of plate fin heat exchangers. However, this algorithm is a little complicated for setting up three different populations to search simultaneously, and the convergence rate is not fast enough.

Therefore, to accelerate the convergence rate and improve the optimization precision further, an algorithm called improved chaotic quantum-behaved particle swarm optimization (ICQPSO) algorithm is proposed. Compared with the other chaotic PSO algorithms [17, 29–31], it is simple and easy to implement. The ICQPSO algorithm is introduced as the learning algorithm in the Takagi–Sugeno fuzzy neural network (TSFNN) [32], which can get accurate output value by inputting most forms of information, to form the improved neural network called ICQPSO-TSFNN. Finally, to demonstrate the effectiveness of the proposed method, we focus on the topic of the FNN for the water quality evaluation. The ICQPSO-TSFNN is applied to evaluate the water quality of the Dongjiang Lake in the Hunan Province from 2002 to 2013.

The rest of this paper is organized as follows. Section 2 mainly introduces the proposed ICQPSO algorithm. The performance test of ICQPSO algorithm are shown in Section 3. The application of ICQPSO-TSFNN for water quality evaluation is presented in Section 4. Finally, we summarize the results and indicate future directions.

2. ICQPSO Algorithm

The proposed ICQPSO algorithm aims at improving the performance of dealing with high-dimensional complex problems by introducing a new chaotic search method, and it also lays the foundation for the application of TSFNN in Section 4.

2.1. QPSO Algorithm. For the QPSO algorithm [20], the particle movement is completely different from that of the PSO algorithm [33]. Newtonian principles are invalid in quantum world as the velocity and position update cannot be determined simultaneously. Hence, there is no velocity vector in the particle of QPSO algorithm. The new state of each particle is determined by the wave function $\psi(\vec{x}, t)$, and there also exists $|\psi(t)|^2$ which is the probability density function of the position of each particle. Considering a one-dimensional optimization problem, the Monte Carlo stochastic simulation is employed to obtain the position equation of the particle i :

$$X_i(t+1) = p_i(t) \pm \frac{L_i(t)}{2} \ln\left(\frac{1}{u}\right), \quad (1)$$

where $p_i(t)$ is the local attractor of the particle i ($i = 1, 2, \dots, N$, N is the total number of particle swarm), and it is defined by

$$p_i(t) = \varphi P_{bi}(t) + (1 - \varphi) P_g(t), \quad (2)$$

$$\varphi = \frac{c_1 r_1}{c_1 r_1 + c_2 r_2}, \quad (3)$$

where $P_{bi}(t)$ represents the optimal position of the particle i at the t th iteration and $P_g(t)$ represents the global optimal position in the particle swarm at the t th iteration. c_1 and c_2 are the learning factors, which provide the optimal selection function. r_1 and r_2 are the random numbers between (0, 1). $L_i(t)$ is defined as

$$L_i(t+1) = 2\beta(t)|\text{mbest} - X_i(t)|, \quad (4)$$

$$\text{mbest} = \sum_{i=1}^N \frac{P_{bi}}{N}, \quad (5)$$

where $\beta(t)$ is the contraction expansion coefficient that determines the convergence rate of the algorithm [22], and it is calculated by

$$\beta(t) = \beta_{\min} + \frac{(T-t)(\beta_{\max} - \beta_{\min})}{T}, \quad (6)$$

where t and T are current and maximum iteration number, respectively. According to Ref. [22], $\beta_{\max} = 1.0$ and $\beta_{\min} = 0.5$. The position equation of the particle i is updated by

$$X_i(t+1) = p_i(t) + \beta(t)|\text{mbest} - X_i(t)| \ln\left(\frac{1}{u}\right), \quad r_3 \geq 0.5, \quad (7)$$

$$X_i(t+1) = p_i(t) - \beta(t)|\text{mbest} - X_i(t)| \ln\left(\frac{1}{u}\right), \quad r_3 < 0.5, \quad (8)$$

where u and r_3 are generated according to a uniform probability distribution at the range (0, 1).

Assuming that the considered problem is a minimum optimization problem, the implementation of the QPSO algorithm is summarized as follows:

- (a) Initialize particles in the population with random position vectors.
- (b) Evaluate the fitness values of each particle.
- (c) Calculate the local attractor point as defined in equation (2).
- (d) Calculate *mbest* vector according to equation (5).
- (e) Compare the fitness of the $P_i(t)$ with the fitness of the P_{bi} . If the fitness value of $P_i(t)$ is smaller than that of the P_{bi} , then replace the P_{bi} by the $P_i(t)$.
- (f) Compare the fitness of the P_{bi} with the fitness of the P_g . If the fitness value of the P_{bi} is smaller than that of the P_g , then replace the P_g by the P_{bi} .
- (g) Update the position of the particles according to equation (7) or equation (8).
- (h) Repeat Step (b) to Step (f) until the termination criteria are satisfied.

Since there is no velocity limit, the QPSO algorithm can jump out of the local optimum more easily. So, it has stronger global convergence ability than that of the PSO algorithm.

2.2. Chaotic Search. Chaos has inherent randomness and ergodicity [34, 35]. It allows the chaotic search to be programmed and traverses every state in a certain search region, while every state is visited only once. Therefore, introducing chaos sequence generated by the chaotic system into the QPSO algorithm can improve the algorithm performance [28–31]. Here, the logistic map is considered, and it is defined by

$$x(n+1) = \mu x(n)[1 - x(n)], \quad (9)$$

where μ is the control parameter. $x(n)$ is the system variable, and $x(n) \in (0, 1)$. A different μ value leads to different dynamical behavior. If $\mu = 4$, the logistic map is chaotic, and the system variable trajectory is dense over the whole search space.

Based on the logistic map, the new chaotic search method is designed. First, the chaotic initialization is presented, and it can make the distribution of all the particles more uniform with respect to the random distribution of the QPSO algorithm. The particle position is initialized as

$$X(n+1) = X_{\min} + (X_{\max} - X_{\min})x(n), \quad n = 1, 2, \dots, \quad (10)$$

where $x(n)$ is the system variable of the logistic map. $X_{\min}$ and $X_{\max}$ are the minimum and maximum boundary of particle position, respectively.

When the search stage of the QPSO algorithm is over, the chaotic search starts to apply and makes the particles jump out of the local optimal state quickly. The search range is determined by the current global optimal position, and the position of particle i in the chaotic search is updated by

$$X_i(n) = [P_g(t) - aP_g(t)] + \{[P_g(t) + aP_g(t)] - [P_g(t) - aP_g(t)]\}x(n), \quad (11)$$

where $P_g(t)$ is the global optimal position at the t th iteration, T is the maximum iteration number of QPSO algorithm, $x(n)$ is the chaotic variable at the n th ($n = 1, 2, \dots, n_{\max}$) iteration, and $n_{\max}$ is the maximum iteration number of chaotic search. Chaotic search and QPSO algorithm are two independent processes. a ($0 < a < 1$) is a fixed value which is set according to the requirement, and a small value a is helpful for the fast convergence of the algorithm. When the algorithm falls into the local optimal value, it is necessary to set a larger a to expand the range of chaotic search for jumping out of the local optimum. In this paper, chaotic search is defined as Algorithm 1.

2.3. Implementation of the ICQPSO Algorithm. The ICQPSO flowchart is presented, as shown in Figure 1, and the implementation steps are listed as follows:

- (a) Chaotic initialize particle positions, and set the QPSO algorithm parameters.
- (b) Evaluate the fitness value of each particle.
- (c) Calculate local attractor point as defined by equation (2).

- (d) Calculate the mbest vector according to equation (5).
- (e) Compare the fitness of the $P_i(t)$ with the fitness of the P_{bi} . If the fitness of the $P_i(t)$ is smaller than that of the P_{bi} , then replace it.
- (f) Compare the fitness of the P_{bi} with the fitness of the P_g . If the fitness of the P_{bi} is smaller than that of the P_g , then replace it.
- (g) Update particle positions according to equation (7) or (8).
- (h) Execute the chaotic search.
- (i) Repeat Step (b) to Step (h) until the termination criteria are met.

3. Performance Tests of ICQPSO Algorithm

To assess the performance of the proposed ICQPSO algorithm, six benchmark functions are examined, and statistical results are compared with PSO [13], SINPSO [17], APSO [18], QPSO [21], and HCQPSO [31, 32]. QPSO and PSO are the basic traditional algorithms. SINPSO is a hybrid algorithm which combines PSO algorithm with chaotic ant colony algorithm. APSO is a variant PSO algorithm which has the adaptive inertia weight. HCQPSO is an improved QPSO algorithm which is introduced the chaotic sequence.

Here, Ackley, Rastrigin, and Griewank functions are high-dimensional multimodal functions, while Sphere, Rosenbrock, and Schwefel's 1.2 functions are high-dimensional unimodal functions. Formulations and properties of these unimodal and multimodal benchmark functions are listed in Table 1. The dimension of all data is set as 30, and the parameter settings for each algorithm are given in Table 2. Algorithms are developed in Matlab 2014b and run on Intel@Core TM with 2.80 GHz CPU and 4.0 GB RAM. Over 50 consecutive algorithm runs are performed due to the stochastic nature, and 1000 fitness values are calculated for each algorithm.

The statistical results of all function tests are shown in Table 3 and Figure 2. The results reveal that the ICQPSO algorithm is superior over other algorithms in terms of finding the minimum object. In Ackley, Sphere, and Schwefel's 1.2 function tests, only the ICQPSO finds the optimal value 0. In Rastrigin and Griewank function tests, both of the ICQPSO algorithm and HCQPSO algorithm find the optimal value 0, but the ICQPSO converges much faster than that of the HCQPSO. The Rosenbrock function is the most difficult to optimize that all algorithms cannot have the high precision solution; the ICQPSO is the second most efficient algorithm amongst others, just slightly worse than the HCQPSO algorithm. Although the precision of ICQPSO algorithm is lower than that of the HCQPSO algorithm in the Rosenbrock function, its convergence rate is always higher than that of the HCQPSO algorithm. Therefore, it can be concluded that the ICQPSO algorithm is the most efficient among the six algorithms. It lays the foundation for the following water quality evaluation experiment.

```

If  $P_g(t) > 1$ 
  For  $n = 1 : n_{\max}$ 
     $X(n) = [P_g(t) - 0.5P_g(t)] + \{[P_g(t) + 0.5P_g(t)] - [P_g(t) - 0.5P_g(t)]\}x(n)$ 
    Update  $x(n)$  with equation (11)
    Calculate each fitness of  $X(n)$ , if it is smaller than  $P_g(t)$ , then replace the  $P_g(t)$ 
  Else
    For  $n = 1 : n_{\max}$ 
       $X(n) = [P_g(t) - 0.01P_g(t)] + \{[P_g(t) + 0.01P_g(t)] - [P_g(t) - 0.01P_g(t)]\}x(n)$ 
      Update  $x(n)$  with equation (11)
      Calculate each fitness of  $X(n)$ , if it is smaller than  $P_g(t)$ , then replace the  $P_g(t)$ 
    End

```

ALGORITHM 1: Chaotic search in the ICQPSO algorithm.

FIGURE 1: Flowchart of the proposed ICQPSO algorithm.

4. The Application of ICQPSO-TSFNN for Water Quality Evaluation

4.1. ICQPSO-TSFNN. To obtain better evaluation precision, the ICQPSO algorithm is used to replace the error correction learning (ECL) algorithm for TSFNN weights correction. In this paper, there is only one node in the output layer, which is the evaluation value of water quality. The node number of the input layer and hidden layer are n and m , respectively, so the architecture of the TSFNN is $n - m - 1$. The process of the proposed ICQPSO-TSFNN is described as follows:

- Set the initial parameters of the TSFNN and ICQPSO algorithm.
- Divide data into training sets and testing sets, all the data are normalized to ensure the quality of evaluation results.
- The ICQPSO algorithm is applied for the correction of TSFNN weights. Initialize particle positions in the ICQPSO algorithm, and the positions represent the center value, width value, and fuzzy system parameters, respectively.
- Input the training data and use TSFNN to calculate the fitness of each particle. Here, the fitness value is the mean square error (MSE) between the output values and the target ones.
- Calculate the fitness value and adjust the weights of TSFNN until the termination criteria are met.
- Use the trained TSFNN for the testing stage.
- If the testing results are satisfactory, output this TSFNN.

The training and testing flowchart of the ICQPSO-TSFNN are shown in Figure 3.

4.2. The Evaluation Criteria of Water Quality. The purpose of water quality evaluation is to determine the water quality rank, which is based on the collected water samples by a certain mathematical model. After considering the water

TABLE 1: The six benchmark functions.

Function name	Definition formula	Range	Optimal value
Ackley	$f(x) = -20 \exp(-0.2 \sqrt{(1/n) \sum_{i=1}^n x_i^2}) - \exp((1/n) \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	0
Rastrigin	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
Griewank	$f(x) = \sum_{i=1}^n x_i^2 / 4000 - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1$	$[-600, 600]$	0
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0
Rosenbrock	$f(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	0
Schwefel's 1.2	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]$	0

TABLE 2: Algorithm parameter settings.

Algorithm	Parameter settings
PSO	$N = 25, c_1 = c_2 = 1.4962, V_{\max} = 1, V_{\min} = -1, \omega = 0.7298$
APSO	$N = 25, c_1 = c_2 = 2, \alpha = 0.1$
SINPSO	$N = 25, c_1 = c_2 = 1.4962, V_{\max} = 1, V_{\min} = -1, \omega = 0.7298$ $r(i, d) = 0.5 + (0.005)\text{rand}, C_i(d) = 0.999, \psi_d = 10.42, M_i = 0.5$
QPSO	$N = 25, c_1 = c_2 = 2, \beta_{\max} = 1.0, \beta_{\min} = 0.5$
HCQPSO	$N = 25, c_1 = c_2 = 2, \beta_{\max} = 1.0, \beta_{\min} = 0.5$
ICQPSO	$N = 25, c_1 = c_2 = 2, \beta_{\max} = 1.0, \beta_{\min} = 0.5, n_{\max} = 100$

TABLE 3: Statistical results of different test function.

	Best	Mean	Worst
Ackley			
PSO	$2.0861e+00$	$3.5337e+00$	$5.0738e+00$
APSO	$2.6173e-01$	$1.5059e+00$	$2.5767e+00$
SINPSO	$7.5123e-02$	$1.5673e+00$	$2.2246e+00$
QPSO	$4.8376e-08$	$3.1851e+00$	$5.5752e+00$
HCQPSO	$8.8888e-16$	$8.8888e-16$	$8.8888e-16$
ICQPSO	0	3.3159e-15	3.5527e-15
Rastrigin			
PSO	$4.4773e+01$	$5.2945e+01$	$6.3585e+01$
APSO	$3.7384e-01$	$1.6870e+01$	$2.2759e+01$
SINPSO	$3.3364e-04$	$4.1799e-04$	$4.5238e-03$
QPSO	$1.5946e+00$	$2.5262e+00$	$3.4837e+00$
HCQPSO	0	0	0
ICQPSO	0	0	0
Griewank			
PSO	$1.1751e+00$	$3.7878e+00$	$7.9665e+00$
APSO	$4.8418e-03$	$1.0163e-03$	$1.1055e-01$
SINPSO	$8.9135e-01$	$9.8711e-01$	$9.9887e-01$
QPSO	0	$2.4653e-04$	$7.3144e-03$
HCQPSO	0	0	0
ICQPSO	0	0	0
Sphere			
PSO	$7.9163e-03$	$2.2771e-02$	$4.9712e-02$
APSO	$2.2724e-14$	$6.9874e-12$	$6.5521e-11$
SINPSO	$1.2264e-09$	$3.2379e-10$	$4.2347e-10$
QPSO	$4.3948e-19$	$3.2996e-15$	$8.1298e-14$
HCQPSO	$4.3454e-39$	$1.1921e-34$	$8.4116e-33$
ICQPSO	7.9196e-91	5.3151e-82	6.0556e-81
Rosenbrock			
PSO	$2.4057e+01$	$5.4534e+01$	$1.0105e+02$
APSO	$1.1128e+01$	$1.4055e+01$	$3.0779e+01$
SINPSO	$8.5902e+00$	$1.0317e+01$	$1.1547e+01$
QPSO	$1.2473e+01$	$4.0305e+01$	$1.2017e+02$
HCQPSO	$3.5643e+00$	$4.6541e+00$	$8.3287e+00$
ICQPSO	4.0444e+00	5.2713e+00	6.5344e+00

TABLE 3: Continued.

	Best	Mean	Worst
Schwefel's 1.2			
PSO	$5.3828e+01$	$1.4982\text{ }ee+02$	$3.2404e+02$
APSO	$2.1917e+00$	$9.0975e+00$	$2.1640e+01$
SINPSO	$3.5402e+00$	$8.1004e+00$	$1.2023e+01$
QPSO	$1.2287e+00$	$5.5542e+00$	$7.7122e+00$
HCQPSO	$1.6231e-14$	$1.0412e+00$	$3.2824e+00$
ICQPSO	$1.3000e-24$	$1.0612e-01$	$8.0364e+00$

FIGURE 2: Continued.

FIGURE 2: Convergence properties of algorithms: (a) Ackley, (b) Rastrigin, (c) Griewank, (d) Sphere, (e) Rosenbrock, and (f) Schwefel's 1.2.

FIGURE 3: Training and testing flowchart of the proposed ICQPSO-TSFNN.

quality monitoring data analysis, we mainly choose the following six indicators as evaluation parameters: ammonia nitrogen, dissolved oxygen, chemical oxygen demand, permanganate index, total phosphorus, and total nitrogen. The water environmental quality index is listed in Table 4. We judged the water quality rank by the TSFNN output values. The classification criteria of water quality rank are shown in Table 5.

4.3. Training and Testing Samples. The experimental data are derived from the environmental hydrological data of the Dongjiang Lake watershed in Zixing, Hunan Province, from 2002 to 2013. 400 datasets from 2002 to 2007 are selected for the training and testing stages. 350 datasets are chosen as the training samples, and the other 50 datasets are taken as the

testing samples. Finally, we also select 72 datasets from the Chukou, Dongping, and Bailang hydrological stations from 2008 to 2013. These data are evaluated by the well-trained TSFNN to demonstrate its practicality. The 10 datasets in one of the stations are shown in Table 6.

The water quality evaluation is carried out by the PSO-BP neural network [36] and three TSFNNs which are based on the ECL algorithm, QPSO algorithm, and ICQPSO algorithm, respectively. To verify the effectiveness of the proposed method, the same training and testing samples are used for these neural networks. All the TSFNN architecture is 6-12-1, and BP's architecture is 6-5-1. Set the algorithm population size $N = 25$. Through experiments, we noticed that when the training error is less than $1e-04$, there is almost no error in the water quality rank of the neural network prediction and the actual water quality rank. Therefore, we set the termination criteria as when the training error is less than $1e-04$, or the maximum number of iterations $T = 100$.

4.4. Result Analysis. The results of training stage are shown in Figure 4. Obviously, the convergence rate of the ICQPSO-TSFNN is quicker than that of the other three neural networks. The testing results are given in Table 7. As it is shown, the ICQPSO-TSFNN only needs 19 iterations to reach the termination criteria, but the other three networks still cannot reach it until the maximum iteration is arrived. So, the ICQPSO-TSFNN has much less training runtime than that of the PSO-BP and QPSO-TSFNN, except the ECL-TSFNN. However, the precision of the ICQPSO-TSFNN is much higher than that of the ECL-TSFNN. Finally, from Figure 5, it is shown that the ICQPSO-TSFNN still has the lowest MSE at the testing stage. Therefore, it can be concluded that the ICQPSO-TSFNN is the most efficient among the four neural networks.

TABLE 4: Chinese surface water environmental quality index (unit: mg/L).

Items	Rank I	Rank II	Rank III	Rank IV	Rank V
Ammonia nitrogen $\leq$	0.150	0.500	1.000	1.500	2.000
Dissolved oxygen $\geq$	7.500	6.000	5.000	3.000	2.000
Chemical oxygen demand $\leq$	15.000	15.000	20.000	30.000	40.000
Permanganate index $\leq$	2.000	4.000	6.000	10.000	15.000
Total phosphorus $\leq$	0.020	0.100	0.200	0.300	0.400
Total nitrogen $\leq$	0.200	0.500	1.000	1.000	2.000
Applicable places	Source of water, national nature reserve	Life drinking water, surface water source protection area, rare aquatic habitat, fish and shrimp production field, etc	Life drinking water, surface water secondary protected areas, fish and shrimp wintering areas, etc	General industrial water and nondirect contact with the human body recreational water	Agricultural water areas and general landscape water

TABLE 5: Classification criteria of the water quality rank.

Items	Rank I	Rank II	Rank III	Rank IV	Rank V
Output value	(0, 1.5)	[1.5, 2.5)	[2.5, 3.5)	[3.5, 4.5)	≤ 4.5

TABLE 6: Data samples (unit: mg/L).

Items	1	2	3	4	5	6	7	8	9	10
Ammonia nitrogen	0.104	0.082	0.182	0.097	0.186	0.186	0.185	1.990	0.502	0.335
Dissolved oxygen	7.190	8.260	8.130	7.860	7.230	7.230	7.350	7.020	6.000	6.700
Chemical oxygen demand	6.600	8.320	10.000	11.26	9.210	9.210	15.000	10.810	9.570	9.150
Permanganate index	3.180	2.550	2.130	1.980	3.260	3.260	2.200	1.900	1.120	1.060
Total phosphorus	0.038	0.014	0.010	0.024	0.046	0.046	0.012	0.044	0.031	0.010
Total nitrogen	0.242	0.132	0.254	0.342	0.156	0.175	0.185	0.196	0.161	0.165

FIGURE 4: Fitness value at the training stage.

TABLE 7: Testing results of different models.

Networks	Training MSE	Training runtime (s)	Iterations	Testing MSE
ECL-TSFNN	$7.6224e-03$	22.1163	100	$5.4745e-02$
PSO-BP	$1.9243e-03$	39.2283	100	$9.6671e-03$
QPSO-TSFNN	$6.0541e-04$	57.4561	100	$4.3438e-03$
ICQPSO-TSFNN	$9.6106e-05$	24.2267	19	$8.5183e-05$

FIGURE 5: Testing stage error.

After the training and testing, the four networks are used to evaluate the water quality rank of Chukou, Dongping and Bailang hydrological stations from 2008 to 2013. The results are given in Table 8, and it shows that the ICQPSO-TSFNN only has 1 error, while the other three

networks have 31, 21, and 16 errors, respectively. Therefore, the proposed ICQPSO-TSFNN has remarkable improvement for water quality evaluation, and it is more suitable for processing the daily hydrological information.

TABLE 8: Evaluation results for different TSFNN.

Monitoring stations	ECL-TSFNN	PSO-BP	QPSO-TSFNN	ICQPSO-TSFNN
Chukou	11	5	7	0
Dongping	11	10	5	1
Bailang	9	7	4	0
Total errors	31	22	16	1

5. Conclusion

In this paper, we focus on the improvement of FNN learning algorithm and its application performance. An improved algorithm called ICQPSO is proposed. It is a hybrid algorithm which combined the QPSO algorithm with a new chaotic search. Six benchmark function experiments are performed in six existing metaheuristic algorithms. The results show that the ICQPSO algorithm has the best performance. Finally, the new algorithm is introduced in TSFNN to form a new neural network called ICQPSO-TSFNN, and it is used to evaluate the water quality of the Dongjiang Lake. The results demonstrate that the new neural network is more efficient than that of three other neural networks. Therefore, it is concluded that the ICQPSO-TSFNN is feasible and effective. The future work is to apply the proposed algorithm to solve other complex nonlinear engineering problems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61861019 and 61362018), the Natural Science Foundation of Hunan Province of China (Grant nos. 2019JJ50483 and 2019JJ50484), the project of Hunan Provincial Department of Education (Grant nos. 18B316 and 18C0557), and the Key Innovation Project of Graduate Student of Central South University (Grant no. 2018ZZTS009).

References

- [1] S. G. Rao and A. R. Rao, *Artificial Neural Networks in Hydrology*, Springer Press, Amsterdam, Netherlands, 2000.
- [2] L. Kalin, S. Isik, J. E. Schoonover, and B. G. Lockaby, "Predicting water quality in unmonitored watersheds using artificial neural networks," *Journal of Environmental Quality*, vol. 39, no. 4, pp. 1429–1440, 2010.
- [3] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, no. 6325, pp. 602–606, 2017.
- [4] G. W. Kim and K. Y. Lee, "Artificial neural networks-based machine learning for wireless networks: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.
- [5] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, 2019.
- [6] H. K. Kwan and Y. Cai, "A fuzzy neural network and its application to pattern recognition," *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 3, pp. 185–193, 1994.
- [7] E. Yucel, M. Syed Ali, N. Gunasekaran, and S. Arik, "Sampled-data filtering of Takagi-Sugeno fuzzy neural networks with interval time-varying delays," *Fuzzy Sets and Systems*, vol. 316, pp. 69–81, 2017.
- [8] D. Krleza and K. Fertalj, "Graph matching using hierarchical fuzzy graph neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 892–904, 2017.
- [9] P. Shi, Y. Zhang, M. Chadli, and R. K. Agarwal, "Mixed H-infinity and passive filtering for discrete fuzzy neural networks with stochastic jumps and time delays," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 903–909, 2016.
- [10] Y. Hou, L. Zhao, and H. Lu, "Fuzzy neural network optimization and network traffic forecasting based on improved differential evolution," *Future Generation Computer Systems*, vol. 81, pp. 425–432, 2018.
- [11] B. Pizzileo, K. Kang Li, G. W. Irwin, and W. Zhao, "Improved structure optimization for fuzzy-neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, pp. 1076–1089, 2012.
- [12] H. Han, L. Zhang, X. Wu, and J. Qiao, "An efficient second-order algorithm for self-organizing fuzzy neural networks," *IEEE Transactions on Cybernetics*, vol. 49, no. 13, pp. 1–13, 2017.
- [13] H. Peng, F. Liu, and Z. R. Xu, "Variable universe fuzzy control for vehicle semi-active suspension system with MR damper combining fuzzy neural network and particle swarm optimization," *Neurocomputing*, vol. 306, pp. 130–140, 2018.
- [14] C.-C. Peng and C.-H. Chen, "Compensatory neural fuzzy network with symbiotic particle swarm optimization for temperature control," *Applied Mathematical Modelling*, vol. 39, no. 1, pp. 383–395, 2015.
- [15] R. Cheng and Y. Bai, "A novel approach to fuzzy wavelet neural network modeling and optimization," *International Journal of Electrical Power & Energy Systems*, vol. 64, pp. 671–678, 2015.
- [16] W. D. Annicchiarico and M. Cerrolaza, "Improved dynamical particle swarm optimization method for structural dynamics," *Mathematical Problems in Engineering*, vol. 2019, Article ID 8250185, 11 pages, 2019.
- [17] X. B. Xu, K. F. Zheng, L. I. Dan et al., "New chaos-particle swarm optimization algorithm," *Journal of Communications*, vol. 33, no. 1, pp. 24–16, 2012.
- [18] A. Adeli and A. Broumandnia, "Image steganalysis using improved particle swarm optimization based feature selection," *Applied Intelligence*, vol. 48, no. 6, pp. 1609–1622, 2018.
- [19] M. K. Marichelvam, M. Geetha, and Ö. Tosun, "An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human

- factors—a case study,” *Computers & Operations Research*, vol. 114, Article ID 104812, 2020.
- [20] J. Sun, B. Feng, and W. B. Xu, “Particle swarm optimization with particles having quantum behavior,” in *Proceedings of the IEEE Conference on Evolutionary Computation*, pp. 325–331, Portland, OR, USA, June 2004.
  - [21] W. Fang, J. Sun, Y. Ding, X. Wu, and W. Xu, “A review of quantum-behaved particle swarm optimization,” *IETE Technical Review*, vol. 27, no. 4, pp. 336–348, 2010.
  - [22] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, “Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection,” *Evolutionary Computation*, vol. 20, no. 3, pp. 349–393, 2012.
  - [23] Y. Li, L. Jiao, R. Shang, and R. Stolkin, “Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation,” *Information Sciences*, vol. 294, pp. 408–422, 2015.
  - [24] M. Xu, L. Zhang, B. Du, L. Zhang, Y. Fan, and D. Song, “A mutation operator accelerated quantum-behaved particle swarm optimization algorithm for hyperspectral endmember extraction,” *Remote Sensing*, vol. 9, no. 3, p. 197, 2017.
  - [25] C. Huang, D. Zhang, and G. Song, “A novel mapping algorithm for three-dimensional network on chip based on quantum-behaved particle swarm optimization,” *Frontiers of Computer Science*, vol. 11, no. 4, pp. 622–631, 2017.
  - [26] C. T. Cheng, W. J. Niu, Z. K. Feng et al., “Daily reservoir runoff forecasting method using artificial neural network based on quantum-behaved particle swarm optimization,” *Water*, vol. 7, no. 8, pp. 4232–4246, 2015.
  - [27] J. Yang and J. Xie, “An improved quantum-behaved particle swarm optimization algorithm,” *Applied Intelligence*, vol. 40, no. 3, pp. 479–496, 2014.
  - [28] F. Liu, H. Duan, and Y. Deng, “A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching,” *Optik*, vol. 123, no. 21, pp. 1955–1960, 2012.
  - [29] V. C. Mariani, A. R. K. Duck, F. A. Guerra, L. D. S. Coelho, and R. V. Rao, “A chaotic quantum-behaved particle swarm approach applied to optimization of heat exchangers,” *Applied Thermal Engineering*, vol. 42, no. 4, pp. 119–128, 2012.
  - [30] O. E. Turgut, M. S. Turgut, and M. T. Coban, “Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations,” *Computers & Mathematics with Applications*, vol. 68, no. 4, pp. 508–530, 2014.
  - [31] O. E. Turgut, “Hybrid chaotic quantum behaved particle swarm optimization algorithm for thermal design of plate fin heat exchangers,” *Applied Mathematical Modelling*, vol. 40, no. 1, pp. 50–69, 2016.
  - [32] Y. Wang, C. Chien, and C. Teng, “Takagi-Sugeno recurrent fuzzy neural networks for identification and control of dynamic systems,” *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, 2001.
  - [33] J. Kennedy and R. C. Eberhart, “Particle swarms optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, WA, Australia, December 1995.
  - [34] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, “Dynamic analysis of digital chaotic maps via state-mapping networks,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 6, pp. 2322–2335, 2019.
  - [35] X. Zhao, J. Liu, H. Liu, and F. Zhang, “Dynamic analysis of a one-parameter chaotic system in complex field,” *IEEE Access*, vol. 8, no. 1, pp. 28774–28781, 2020.
  - [36] C. Liu, W. Ding, Z. Li et al., “Prediction of high-speed grinding temperature of titanium matrix composites using BP neural network based on PSO algorithm,” *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 5–8, pp. 2277–2285, 2017.