

Research Article

Radial Basis Function Neural Network Based on an Improved Exponential Decreasing Inertia Weight-Particle Swarm Optimization Algorithm for AQI Prediction

Jinna Lu,^{1,2} Hongping Hu,² and Yanping Bai²

¹ School of Information and Communication Engineering, North University of China, Taiyuan, Shanxi 030051, China

² Department of Mathematics, North University of China, Taiyuan, Shanxi 030051, China

Correspondence should be addressed to Yanping Bai; byp666@163.com

Received 17 April 2014; Revised 8 July 2014; Accepted 9 July 2014; Published 17 July 2014

Academic Editor: Suohai Fan

Copyright © 2014 Jinna Lu 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.

This paper proposed a novel radial basis function (RBF) neural network model optimized by exponential decreasing inertia weight particle swarm optimization (EDIW-PSO). Based on the inertia weight decreasing strategy, we propose a new Exponential Decreasing Inertia Weight (EDIW) to improve the PSO algorithm. We use the modified EDIW-PSO algorithm to determine the centers, widths, and connection weights of RBF neural network. To assess the performance of the proposed EDIW-PSO-RBF model, we choose the daily air quality index (AQI) of Xi'an for prediction and obtain improved results.

1. Introduction

The radial basis function (RBF) neural network is a novel and effective feed-forward neural network [1], which has good performance of best approximation and global optimum. It has been broadly used in considerable applications, such as function approximate, classification, regression problems, prediction, signal processing, and other problems [2–6]. The RBF neural network architecture has three layers composed of input layer, hidden layer, and output layer. The input layer is composed of input vectors. Before the input vectors are input to the network, data processing should be done, such as normalization processing. This processing also can be done in the input layer.

The hidden layer is composed of hidden neurons, the number of which is determined by the issues described. The RBF networks are different from other types of neural networks mainly in the hidden neurons [7, 8]. Each hidden neuron has a radial basis function which is a center symmetric nonlinear function with local distribution. The radial basis function consists of a center position and a width parameter. Once the center and width are determined, the input vectors are mapped to the hidden space by the mapping $f: R^I \rightarrow R^J$.

Suppose that the input layer has I input units; the hidden layer has J radial basis functions. The output of the j th hidden neuron is expressed as

$$h_j(X) = \varphi\left(-\frac{\|X - c_j\|}{d_j}\right) \quad j = 1, 2, \dots, J, \quad (1)$$

where X is the overall input sample as $X = (X_1, X_2, \dots, X_N)^T$, and each X_n is the I -dimension input vector expressed as $X_n = (x_{n1}, x_{n2}, \dots, x_{nI})$. c_j and d_j are, respectively, the center and the width of the j th hidden neuron, and c_j is an I -dimension vector as $c_j = (c_{j1}, c_{j2}, \dots, c_{ji}, \dots, c_{jI})$. $\|\cdot\|$ is Euclidean norm usually taking 2-norm. $\varphi(\cdot)$ is the radial basis function. It can take a variable of formula expression such as B-Spline RBF [9], thin-plate Spline RBF [10], Cauchy RBF [11], and Gaussian RBF [12]. Among them the Gaussian function is the most used, as in

$$\varphi(r) = \exp\left(-\frac{r^2}{2}\right), \quad (2)$$

where r is the variable of radial basis function $\varphi(\cdot)$.

The output layer implements the mapping $f: R^J \rightarrow R^K$. K is the number of output neurons. The output function is a linear combination of the outputs of the radial basis functions through connection weights which connect the hidden layer and the output layer, which is shown as

$$y_k = \sum_{j=1}^J \omega_{jk} h_j(X) \quad k = 1, 2, \dots, K, \quad (3)$$

where ω_{jk} is the connection weight between the j th hidden layer and the k th output of the network.

RBF neural network contains three groups of parameters which are centers c_{ji} , widths d_j , and connection weights ω_{jk} . To optimize the RBF parameters, many optimization algorithms have been proposed, such as orthogonal least squares (OLS) algorithm [13], Expectation-Maximization (EM) algorithm [14], gradient descent algorithm [15], K-means clustering algorithm [16], Genetic algorithm (GA) [17], ant colony optimization (ACO) algorithm [18] and particle swarm optimization (PSO) algorithm [19, 20], and support vector machine (SVM) and extreme learning machine (ELM) [21]. Compared to other algorithms, PSO algorithm has many advantages: stable convergence, few parameters, and fast convergence speed. Many researchers have successfully applied the PSO algorithm in the learning and structure improvement of the RBF neural network for application problems.

The particle swarm optimizing (PSO) algorithm is put forward by Eberhart and Kennedy in 1995 [22], which is initially motivated by the intelligent collective behavior of birds in the foraging process. In PSO algorithm, each bird also called a particle has a position and a velocity and searches for the optimal solution by updating the position and the velocity. In the following years, many researchers introduce "inertia weight" and propose many dynamic variations of PSO based on the inertia weight [23–26]. Different inertia weight strategies imply different incremental changes in velocity per time step which means exploration of new search areas in pursuit of a better solution. In this paper, we propose an Exponential Decreasing Inertia Weight PSO (EDIW-PSO) algorithm to get the optimal parameters of RBF network.

This paper establishes a RBF network model based on EDIW-PSO algorithm. Section 2 introduces the basic PSO algorithm and several variants of inertia weight. Section 2.3 gives the improved EDIW-PSO algorithm based on an Exponential Decreasing Inertia Weight strategy. Section 3 presents the methodology of the proposed EDIW-PSO-RBF structure. Section 4 shows an experiment using this methodology comparing with other three models. The last section summarizes the conclusions of this study.

2. Dynamic Particle Swarm Optimizing Algorithm

2.1. Basic Particle Swarm Optimizing Algorithm. PSO algorithm is a parallel evolutionary computation algorithm. In PSO, each potential solution to an optimization problem is treated as a bird, which is also called a particle. The set of

particles, also known as a swarm, is flown through the D -dimensional search space of the problem. Each particle changes its own position and velocity based on the experiences of the particle itself and those of its neighbors. In the searching process, every particle is connected to and able to share information with every other particle in the swarm and the swarm communication topology is known as a global neighborhood described in [27]. This information sharing mechanism keeps the overall consistency to get the global solution for the overall swarm.

The position and velocity of the i th particle in D -dimensional solution space are denoted as $L_i = (l_{i1}, l_{i2}, \dots, l_{iD})$ and $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, respectively, where $i = 1, 2, \dots, m$ is the number of the swarm, $l_{id} \in [l_d, u_d]$, $d = 1, 2, \dots, D$, and l_d and u_d are the lower and upper bounds of the d th dimension.

According to a preset fitness function, we obtain the personal best position (also named as the local best fitness) of the i th particle denoted as $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ and the global best position (also named as the global best fitness) found so far of all particles of the swarm denoted as $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$. At each iterative, the i th particle updates its position and velocity as follows:

$$\begin{aligned} V_i^{t+1} &= \hat{\omega} V_i^t + c_1 r_1 (p_i^t - L_i^t) + c_2 r_2 (p_g^t - L_i^t), \\ L_i^{t+1} &= L_i^t + V_i^{t+1}, \end{aligned} \quad (4)$$

where c_1 and c_2 are acceleration factors and positive constants, r_1 and r_2 are random numbers ranging from 0 to 1, and $\hat{\omega}$ is the inertia weight on the interval $[0, 1]$ keeping the memory of the old velocity vector of the same particle. When $\hat{\omega}$ is a constant [22], it can lead to a static PSO, and when $\hat{\omega}$ is varying iteratively, it leads to a dynamic PSO.

2.2. Several Variants of Inertia Weight. The inertia weight determines the proportion of the current particle velocity. Large inertia weight can lead to large speed and strong searching ability of particles, but the global optimal solution may be missed. In contrast, small inertia weight makes the particle have strong development capability, but it needs long search time for fine tuning the local optimal solution.

By changing the inertia weight dynamically, the search capability is dynamically adjusted. Many researchers have proposed several variants of PSO according to the impact of $\hat{\omega}$. Most of the PSO variants use time-varying inertia weight strategies in which the value of the inertia weight varies with the iteration numbers. These methods can be either linear or nonlinear. In 1998, Shi and Eberhart introduced a Linearly Decreasing Inertia Weight (LDIW) strategy to get better inertia weight $\hat{\omega}$ as the following formula [23]:

$$\hat{\omega}(t) = \hat{\omega}_{\min} + (\hat{\omega}_{\max} - \hat{\omega}_{\min}) \frac{t}{T}, \quad (5)$$

where t is the number of current iterative steps, T is the maximum number of iterative steps the PSO is allowed to continue, and $\hat{\omega}_{\max}$ is the initial inertia weight and $\hat{\omega}_{\min}$ is the final inertia weight.

TABLE I: Values of inertia weight $\hat{\omega}$ by the proposed adaption when $t = T$, for varying sets of $\{c, \hat{\omega}_{\max}, \hat{\omega}_{\min}\}$.

| c | $\hat{\omega}_{\max}$ 0.7 | $\hat{\omega}_{\min}$ 0.2 | $\hat{\omega}_{\max}$ 0.9 | $\hat{\omega}_{\min}$ 0.2 | $\hat{\omega}_{\max}$ 0.7 | $\hat{\omega}_{\min}$ 0.3 | $\hat{\omega}_{\max}$ 0.9 | $\hat{\omega}_{\min}$ 0.3 | $\hat{\omega}_{\max}$ 0.7 | $\hat{\omega}_{\min}$ 0.4 | $\hat{\omega}_{\max}$ 0.9 | $\hat{\omega}_{\min}$ 0.4 |
|-----|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|
| 1 | | 0.3839 | | 0.4575 | | 0.4472 | | 0.5207 | | 0.5104 | | 0.5839 |
| 2 | | 0.2677 | | 0.2947 | | 0.3541 | | 0.3812 | | 0.4406 | | 0.4677 |
| 3 | | 0.2249 | | 0.2349 | | 0.3199 | | 0.3299 | | 0.4149 | | 0.4249 |
| 4 | | 0.2092 | | 0.2128 | | 0.3073 | | 0.3110 | | 0.4055 | | 0.4092 |
| 5 | | 0.2034 | | 0.2047 | | 0.3027 | | 0.3040 | | 0.4020 | | 0.4034 |
| 6 | | 0.2012 | | 0.2017 | | 0.3010 | | 0.3015 | | 0.4007 | | 0.4012 |
| 7 | | 0.2005 | | 0.2006 | | 0.3004 | | 0.3005 | | 0.4003 | | 0.4005 |
| 8 | | 0.2002 | | 0.2002 | | 0.3001 | | 0.3002 | | 0.4001 | | 0.4002 |
| 9 | | 0.2001 | | 0.2001 | | 0.3000 | | 0.3001 | | 0.4000 | | 0.4001 |
| 10 | | 0.2000 | | 0.2000 | | 0.3000 | | 0.3000 | | 0.4000 | | 0.4000 |

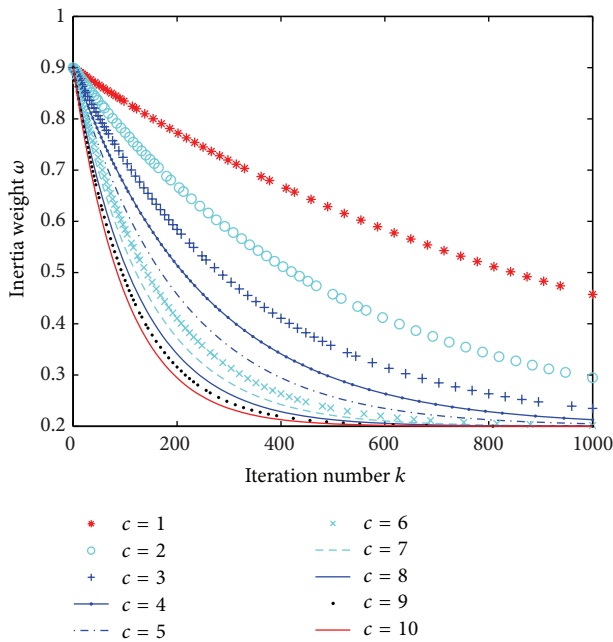


FIGURE 1: The decreasing curves of inertia weight $\hat{\omega}$ attained by the proposed EDIW strategy for varying c .

Chatterjee and Siarry propose a nonlinear decreasing inertia weight (NDIW) strategy to modulate inertia weight adaptation with time for improved performance of PSO algorithm [24]. The proposed adaption of $\hat{\omega}(t)$ is given as

$$\hat{\omega}(t) = \hat{\omega}_{\min} + (\hat{\omega}_{\max} - \hat{\omega}_{\min}) \left\{ \frac{(t - T)^n}{T^n} \right\}, \quad (6)$$

where n is the nonlinear modulation index. With $n = 1$, the system becomes a special case of linearly adaptive inertia weight with time, as proposed by Shi and Eberhart [23]. After implementing the performance of this strategy for the famous Sphere function, they draw a conclusion that $n = 1.2$ was a typical value and could be suitably determined for each case individually. When $n = 1.2$, during the early iterations a higher value of $\hat{\omega}$ facilitates taking larger steps in the solution space, and during the later iterations $\hat{\omega}$ is

decreased more rapidly than the linear case, which is very suitable to determine the optimal region among the already discovered suboptimal regions.

In [25], Arumugam and Rao propose a global-local best inertia weight (GLbestIW) method in which the inertia weight is neither set to a constant value nor set as linearly decreasing time-varying function. The inertia weight is determined by the ratio of the global best fitness and the local best fitness in each iteration. The GLbestIW is given by the following equation:

$$\hat{\omega}_i(t) = 1.1 - \frac{p_g^t}{p_i^t}, \quad (7)$$

where p_g^t is the global best fitness at t th iterative and p_i^t is the local best fitness of the i th particle at t th iterative.

2.3. The Proposed Inertia Weight Variant. Larger $\hat{\omega}$ is conducive to find the global best solution as soon as possible in the early iterative steps but may lead to miss the global best solution easily in later iterative steps. However, smaller $\hat{\omega}$ means longer time to provide slower updating for fine tuning a local exploration. Hence in the early iterative steps, larger $\hat{\omega}$ is needed for coarse global exploration, but in later iterations $\hat{\omega}$ should decrease for fine tuning the local exploration. Appropriate inertia weight can help find the best solution with the least number of iterative steps.

A larger inertia weight facilitates global exploration and a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area. In order to balance the global and local exploration, we present a new Exponential Decreasing Inertia Weight (EDIW) strategy. In this strategy, the inertia weight is exponential decreasing with the increase of iterative step t . The proposed adaption of $\hat{\omega}(t)$ is given as

$$\hat{\omega}(t) = \hat{\omega}_{\min} + (\hat{\omega}_{\max} - \hat{\omega}_{\min}) * \exp\left(-\frac{ct}{T}\right), \quad (8)$$

where c is controlling parameter to control the convergence rate of the inertia weight, $c > 0$. When $t = 0$, $\hat{\omega}(t) = \hat{\omega}_{\max}$. When $t = T$, $\hat{\omega}(t)$ can be expressed by the following equation:

$$\hat{\omega}(t) = \hat{\omega}_{\min} + (\hat{\omega}_{\max} - \hat{\omega}_{\min}) * \exp(-c). \quad (9)$$

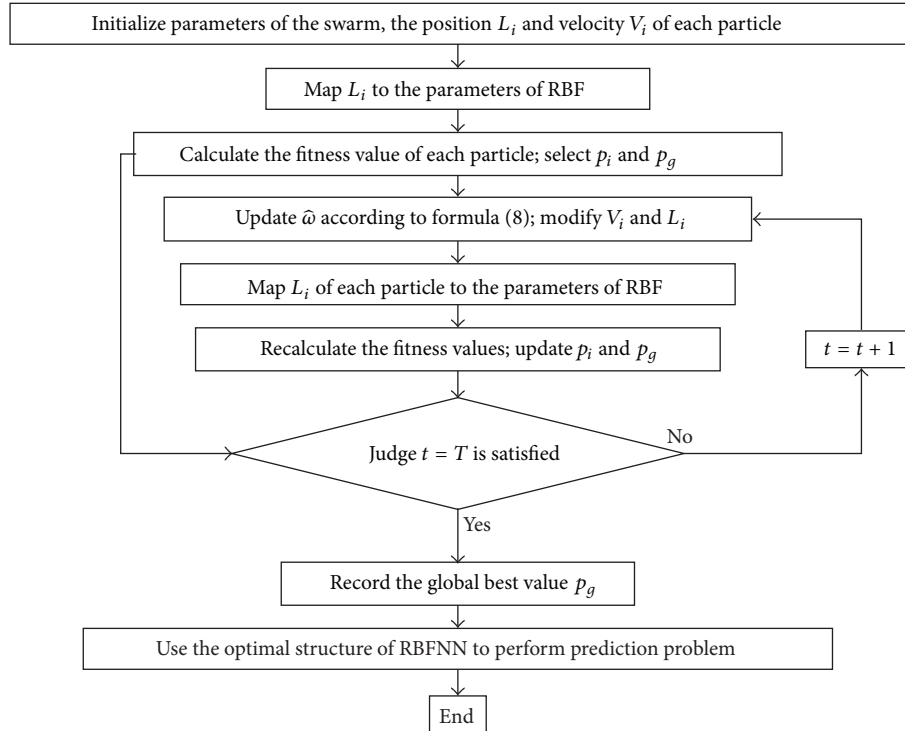


FIGURE 2: The flowchart of EDIW-PSO-RBF learning algorithm.

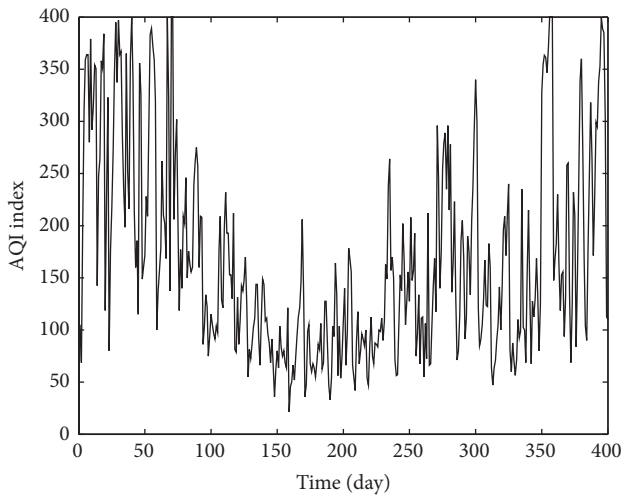


FIGURE 3: The actual daily air quality index (AQI) of Xi'an from January 2 in 2013 to February 5 in 2014.

When $t = T$, we obtain different values of inertia weight $\hat{\omega}$ by the proposed adaption by setting varying sets of $\{c, \hat{\omega}_{\max}, \hat{\omega}_{\min}\}$. The results are shown in Table 1. According to Table 1, when $t = T$, for same $\hat{\omega}_{\max}$ and $\hat{\omega}_{\min}$, small c makes $\hat{\omega}$ be still far from the final inertia weight $\hat{\omega}_{\min}$, and the value of $\hat{\omega}$ is more and more close to $\hat{\omega}_{\min}$ with the increase of c . When $c \geq 6$, $\hat{\omega}$ is nearly equal to $\hat{\omega}_{\min}$. So the condition of $c \geq 6$ can ensure that $\hat{\omega}(t)$ varies fully in the given range of $[\hat{\omega}_{\min}, \hat{\omega}_{\max}]$ when t ranges from 1 to T .

When c takes different values, different decreasing effects will be got. For convenience to compare the different decreasing effects, we set $\hat{\omega}_{\max} = 0.9$, $\hat{\omega}_{\min} = 0.2$, and $T = 1000$. The decreasing curves of inertia weight $\hat{\omega}$ attained by the proposed EDIW strategy for varying c are shown in Figure 1.

According to Figure 1, for the same c , the rate of descent of $\hat{\omega}$ gradually declines as the the increase of iterative step and begins to flatten in the later iterative steps. For different c , the rate of descent of $\hat{\omega}$ gradually increases as the increase of c in the early iterative steps. Smaller c can ensure that $\hat{\omega}$ does not decrease so fast in the early iterations but will be still far from $\hat{\omega}_{\min}$ in the final iteration. For example, when $c = 1$ and $t = 1000$, the inertia weight $\hat{\omega}$ is 0.4575 which is far from $\hat{\omega}_{\min}$ (0.2). Larger c is conducive to making $\hat{\omega}$ decrease fast to discover the promising suboptimal regions but may lead to the algorithm prematurely to begin the local search. For example, when $c = 10$ and $T = 1000$, the inertia weight $\hat{\omega}$ begins to flatten at the iterative step $t = 400$. Parameter c should be appropriately selected in EDIW-PSO algorithm. Based on the analysis above, we can choose the value of $c \in (6, 8)$. During the early iterations $\hat{\omega}$ decreases more rapidly than the linear case, which is very suitable for the algorithm to discover promising suboptimal regions. During the later iterations $\hat{\omega}$ is fine-tuned, which is conducive to determining the optimal region among the already discovered optimal regions.

3. The Proposed RBF Model by EDIW-PSO Algorithm

In this section we use the proposed EDIW-PSO algorithm to determine the optimal structure of the RBFNN and establish

TABLE 2: The parameters of RBFNN by EDIW-PSO: centers c_{ji} , widths d_j , and connection weights ω_{jk} .

| j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| c_{j1} | 1.0000 | -0.9849 | -0.9569 | 0.7735 | 0.5874 | 1.0000 | -0.4653 | -1.0000 | 0.7688 | -0.3863 |
| c_{j2} | 0.5752 | -0.1652 | -0.9023 | -0.9643 | 0.6898 | 0.4439 | 0.4378 | 0.4779 | 0.0860 | -0.8635 |
| c_{j3} | 0.4994 | 0.2723 | -1.0000 | -0.1271 | 0.7500 | 0.5629 | 0.2198 | -0.9292 | 0.7286 | -0.9566 |
| c_{j4} | 0.5250 | -0.2811 | 0.7298 | 0.9254 | -0.1321 | 0.3404 | -0.3397 | -0.1867 | 0.5373 | -0.6491 |
| c_{j5} | -0.0776 | -1.0000 | 0.1190 | 0.4846 | 0.2787 | 0.1364 | -0.8673 | 0.9034 | 1.0000 | 0.0959 |
| c_{j6} | -0.4532 | 0.9372 | 1.0000 | 0.5370 | 0.2975 | 0.9214 | 0.4095 | -1.0000 | -0.0813 | -0.2635 |
| c_{j7} | -0.0124 | 1.0000 | -0.4612 | 0.0190 | -0.1061 | -0.2451 | 0.7454 | 0.1273 | 0.1925 | 0.0609 |
| d_j | -0.0156 | -0.9724 | 0.0728 | 0.0588 | -0.5133 | 0.8992 | 1.0000 | -0.0857 | 0.7017 | -0.9959 |
| ω_j | -0.0289 | -0.7936 | -0.5968 | 0.8891 | -0.5265 | -0.0812 | 0.3979 | 0.2548 | -0.9623 | 0.8871 |

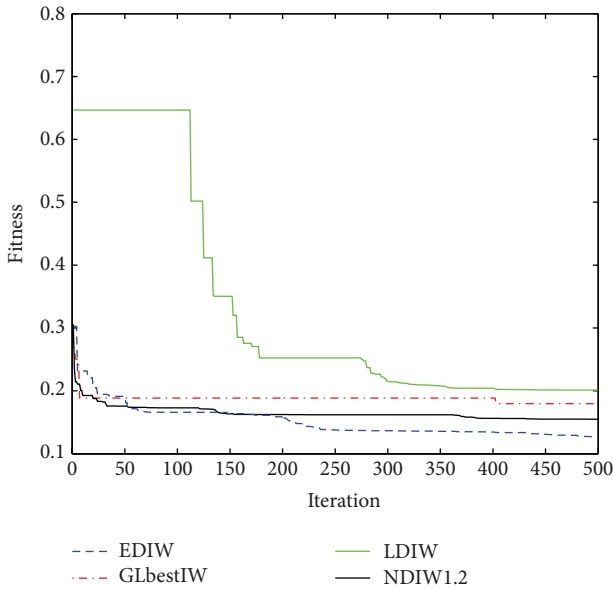


FIGURE 4: The best global fitness value for the AQI of Xi'an. Blue short dash line shows the best global fitness value by the improved EDIW-PSO-RBF model; red dashdotted line shows the best global fitness value by the GLbestIW-PSO-RBF model; green solid line shows the best global fitness value by the LDIW-PSO-RBF model; black solid line shows the best global fitness value by the NDIW-PSO-RBF model.

the EDIW-PSO-RBF network model. The position vector L of each particle is needed to be optimized, which represents RBF centers c_{ji} , widths d_j , and connection weights ω_{jk} , where $i = 1, 2, \dots, I$, $j = 1, 2, \dots, J$, and $k = 1, 2, \dots, K$. Therefore the dimension of L of each particle in EDIW-PSO algorithm is $D = I \cdot J + J + J \cdot K$. We map each L to the RBFNN and obtain the prediction output. The fitness function of EDIW-PSO algorithm is defined in terms of the relative mean square error (RMSE) between the prediction output values and the actual values in the network training process. Thereby, we can minimize the fitness value of the network by the powerful

search performance of EDIW-PSO algorithm. The fitness function is denoted by formula

$$f = e = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \left(\frac{y_{nk} - \hat{y}_{nk}}{\hat{y}_{nk}} \right)^2, \quad (10)$$

where \hat{y}_{nk} and y_{nk} are the actual value and the prediction value of the k th output neuron in the n th sample, and N is the number of training samples, while K is the number of output neurons.

The iteration process of the improved EDIW-PSO-RBF learning algorithm can be described clearly as follows.

Step 1. Initialize the relative parameters, including the size of swarm M , the boundary of position L_{\max} and velocity V_{\max} , the acceleration factors c_1 and c_2 , and the maximum iterative steps T . Initialize $t = 1$; for each particle, select two D -dimensional vectors randomly to initialize the position and velocity of this particle, respectively.

Step 2. Map the position vector L_i of each particle to the parameters of RBFNN.

Step 3. Calculate the fitness value of each particle according to formula (10). Set the current position of each particle as the personal best fitness p_i . Then find the minimum fitness value as the global best fitness p_g of the whole swarm.

Step 4. Update the inertia weight $\hat{\omega}$ according to formula (8). Modify the particle velocity V_i and position L_i according to formula (4).

Step 5. Map the new position vector L_i of each particle to the parameters of RBFNN, input training data, and train RBFNN.

Step 6. Recalculate the fitness values of the new particles and modify p_i^t and p_g^t . For each particle, if the current fitness value is better than the previous local best, then set the current fitness value to be the local best; or keep the previous local best. For the global swarm, if the best value of all current local best is better than the previous global best, then update the value of the global best; or keep the previous global best.

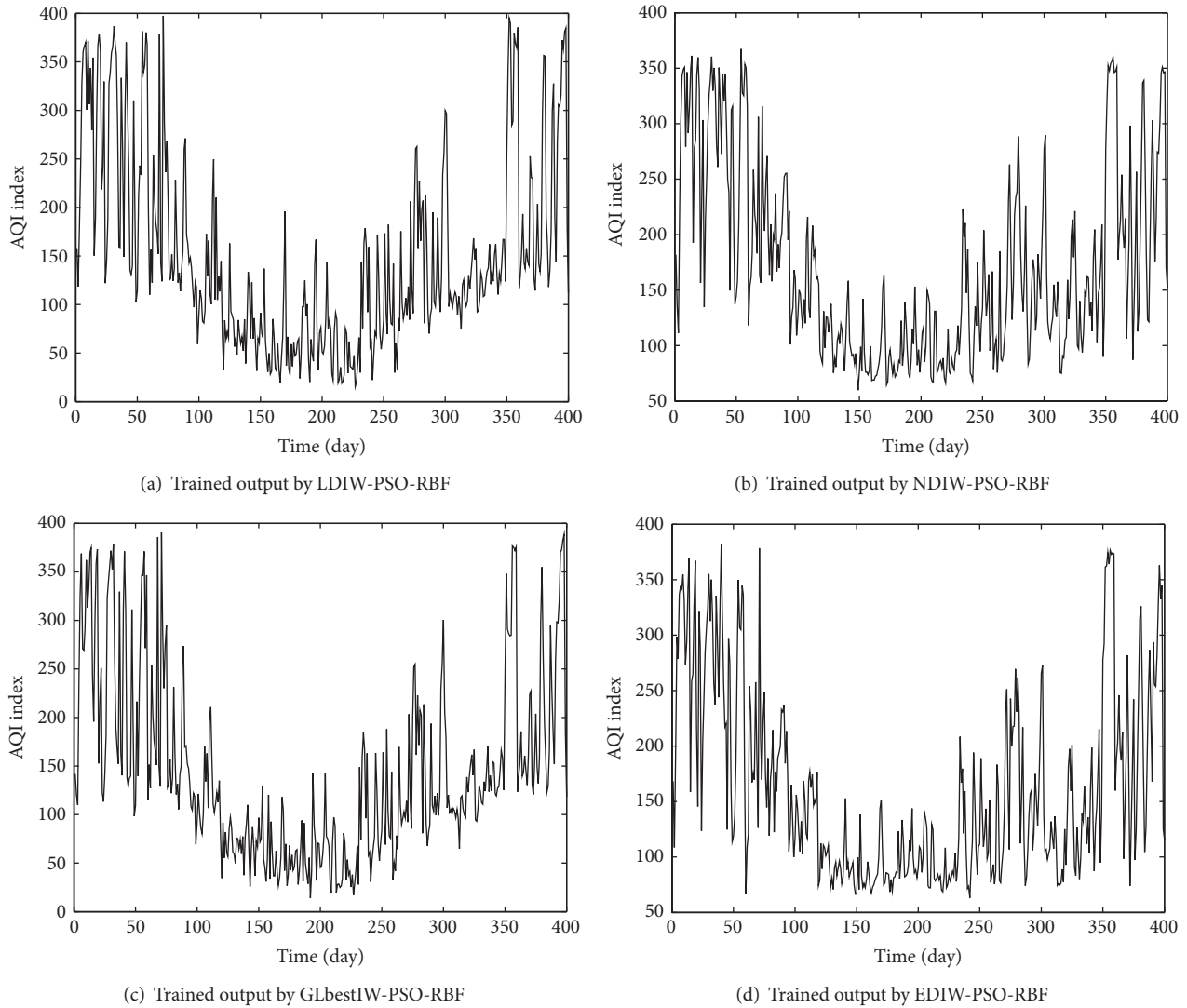


FIGURE 5: The trained output curve by four models for the daily air quality index (AQI) of Xi'an.

Step 7. Judge whether the particle satisfies the conditions: $t = T$. If the condition is met, go to Step 8; otherwise, let $t = t + 1$, and go back to Step 4.

Step 8. Record the global best value p_g ; exit the iteration.

Step 9. Use the optimal structure of RBFNN to perform prediction problem.

Apply the above 9 steps until the terminal conditions hold. The flow chart is as in Figure 2.

4. Experiment

In order to ensure the prediction accuracy of the proposed EDIW-PSO-RBF model, we choose the daily air quality index (AQI) of Xi'an [28] for the time series prediction. In recent days, the air pollution affects peoples' travel and life. Daily AQI is a dimensionless index quantitatively describing air

quality, which is calculated by the following six indicators: sulfur dioxide (SO_2), nitrogen dioxide (NO_2), particulate matter (PM_{10} : particle size is less than or equal to 10 microns), particulate matter ($\text{PM}_{2.5}$: particle size is less than or equal to 2.5 microns), carbon monoxide (CO), and ozone (O_3). Among them, SO_2 , NO_2 , and CO are all the 24-hour average density; O_3 is the 8-hour moving average density. We choose 400 sets of data from 2013.1.1 to 2014.2.5 as train data and 5 sets of data from 2014.2.6 to 2014.2.10 as test data. Normalization processing is done with all the sets of data before it is used in the model. We adopt mapminmax function to normalize the data to a range of $[-1, 1]$ as the following formula:

$$y = (y_{\max} - y_{\min}) \frac{x - x_{\min}}{x_{\max} - x_{\min}} + y_{\min}, \quad (11)$$

where x is the original data before normalization, x_{\min} and x_{\max} are the minimum value and the maximum value before normalization, respectively, y is the data after normalization,

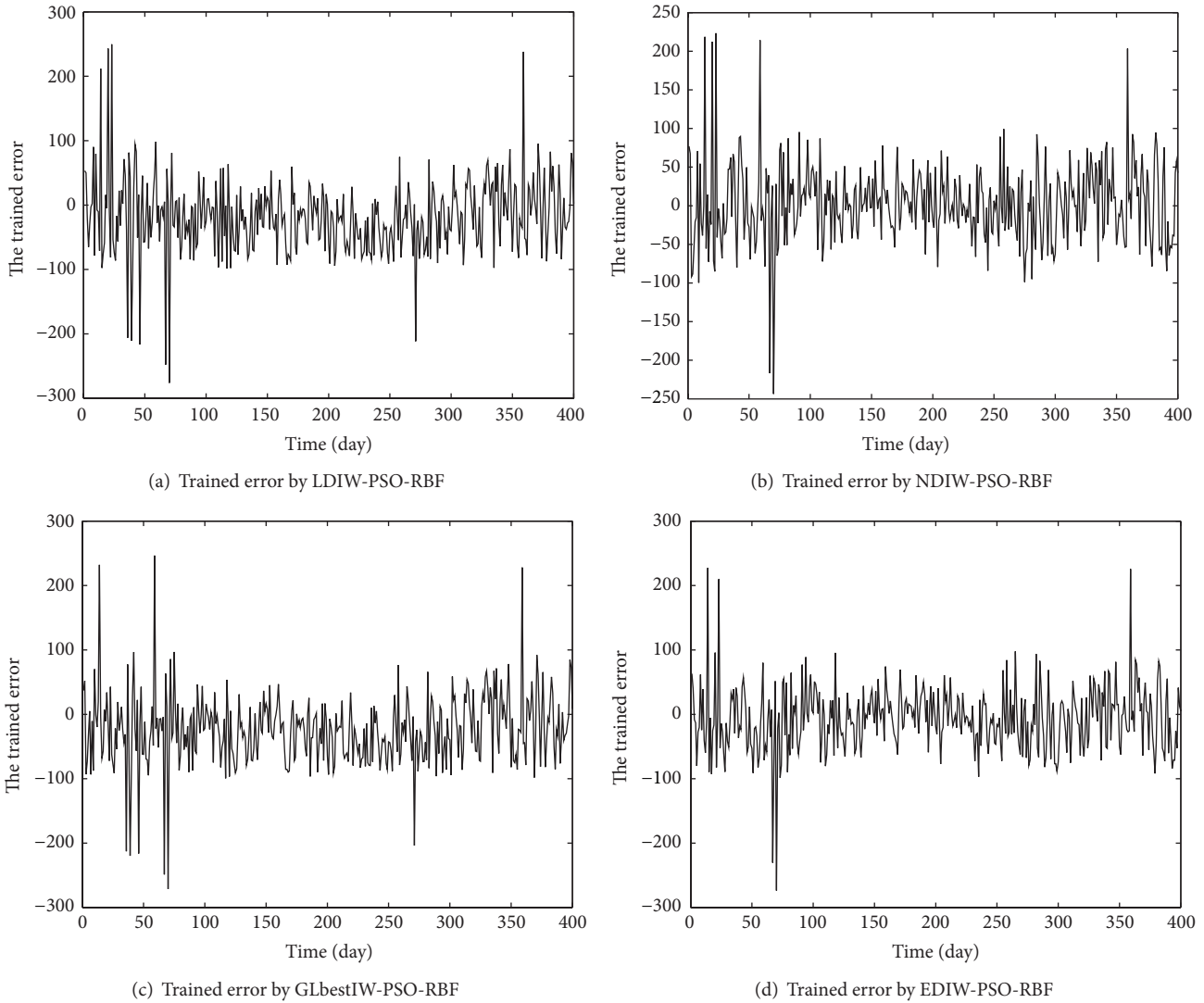


FIGURE 6: Plots of trained absolute errors between the trained outputs and the actual values by four models for the daily air quality index (AQI) of Xi’an.

and y_{\min} and y_{\max} are the minimum value and the maximum value after normalization, and, respectively, they are -1 and 1 .

In this section, we assess the effectiveness of the proposed EDIW-PSO-RBF model comparing with other three inertia weight variants, which are LDIW-PSO-RBF model [23], NDIW-PSO-RBF model [24], and GLbestIW-PSO-RBF model [25]. Firstly, we build a RBF network consisting of 7 input neurons and 1 output neuron. The 7 input neurons consist of the six indicators and the air quality index, and the 1 output neuron is the next day’s air quality index. The number of neurons in hidden layer is determined to be 10. Therefore, the dimension of each particle in the modified PSO algorithm is $D = 7 \times 10 + 10 + 10 \times 1 = 90$.

Secondly, several parameters in the PSO simulation must be specified. In the proposed EDIW-PSO-RBF model, the value of c is set to be 8. In the NDIW-PSO-RBF model, the nonlinear modulation index n is set to be 1.2. In all the four models, the acceleration factors c_1 and c_2 are fixed to be 2.

TABLE 3: The values of MSE, RMSE, and MAPE (%) of trained output by four models for the daily air quality index (AQI) of Xi’an.

| Error | LDIW-PSO | NDIW-PSO | GLbestIW-PSO | EDIW-PSO |
|----------------|----------|----------|--------------|----------|
| MSE ($*e^2$) | 3.6565 | 2.6560 | 3.5800 | 2.5389 |
| RMSE | 0.2239 | 0.1846 | 0.1935 | 0.1350 |
| MAPE (%) | 3.2143 | 2.9898 | 3.1684 | 2.8644 |

The minimum velocity V_{\min} and minimum position L_{\min} of every particle both are set to be -1 . Meanwhile, the maximum velocity V_{\max} and maximum position L_{\max} of every particle both are set to be 1. The maximum number of iterations is set to be 1000. The population size is set as 50.

To assess the performance of the four different models, mean square error (MSE), relative mean square error

TABLE 4: The results of predicted output by four models for the daily air quality index (AQI) of Xi'an of the following five days (2014.2.6–2014.2.10).

| Day | Actual | LDIW- PSO | NDIW- PSO | GLbestIW- PSO | EDIW- PSO |
|-----|----------|--------------|--------------|------------------|--------------|
| 1 | 231.2500 | 224.3392 | 222.8994 | 223.3160 | 234.1575 |
| 2 | 222.0000 | 247.3886 | 245.6876 | 243.5894 | 236.7476 |
| 3 | 280.0000 | 240.9040 | 250.5690 | 249.1804 | 241.5187 |
| 4 | 215.0000 | 246.8951 | 248.3747 | 254.2518 | 241.8096 |
| 5 | 247.5000 | 249.7290 | 246.5483 | 243.1647 | 238.9121 |

TABLE 5: The values of MSE, RMSE, and MAPE (%) of predicted output by four models for the daily air quality index (AQI) of Xi'an.

| Error | LDIW-PSO | NDIW-PSO | GLbestIW-PSO | EDIW-PSO |
|----------------|----------|----------|--------------|----------|
| MSE ($*e^2$) | 6.4862 | 5.2236 | 6.0768 | 4.9985 |
| RMSE | 0.0111 | 0.0096 | 0.0113 | 0.0080 |
| MAPE (%) | 8.8246 | 8.1400 | 8.8342 | 7.5166 |

(RMSE), and mean absolute percentage error (MAPE) are used as criteria, defined as

$$\begin{aligned} \text{MSE} &= \frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2, \\ \text{RMSE} &= \frac{1}{N} \sum_{n=1}^N \left(\frac{y_n - \hat{y}_n}{\hat{y}_n} \right)^2, \\ \text{MAPE} &= \frac{1}{N} \sum_{n=1}^N \left| \frac{y_n - \hat{y}_n}{\hat{y}_n} \right| \times 100\%, \end{aligned} \quad (12)$$

where \hat{y}_n and y_n denote the actual value and the network output value at the n th day, respectively.

The actual daily air quality index (AQI) of Xi'an from January 2 in 2013 to February 5 in 2014 are shown in Figure 3. In this experiment, the proposed EDIW-PSO method, the LDIW-PSO method, the NDIW-PSO method, and the GLbestIW-PSO method are adopted to train RBFNNs. The global best fitness values of the four methods in the training process are given in Figure 4, which shows that the proposed EDIW-PSO method has the fastest convergence rate and the lowest fitness value in the four PSO methods.

By the EDIW-PSO, when the fitness reaches its minimum, the values of the optimal parameters of RBFNN: centers c_{ji} , widths d_j , and connection weights ω_{jk} are shown as in Table 2. Figure 5 shows the trained output curve by four methods for the daily air quality index (AQI) of Xi'an. Figure 6 shows the plots of trained absolute errors between the trained outputs and the actual values by four methods. And the values of MSE, RMSE, and MAPE (%) of trained output by four methods for the daily air quality index (AQI) of Xi'an are shown in Table 3. From Table 3, the three errors MSE, RMSE, and MAPE (%) of trained output by EDIW-PSO are, respectively, $2.5389 * e^2$, 0.1350, and 2.8644, which are all the smallest among the four methods.

Based, respectively, on the trained optimal parameters of RBFNN by the four methods, we predict the daily air quality

index (AQI) of Xi'an of following five days from 2014.2.6 to 2014.2.10. Table 4 shows the predicted outputs by using the LDIW-PSO-RBFNN model, the NDIW-PSO-RBFNN model, the GLbestIW-PSO-RBFNN model, and the EDIW-PSO-RBFNN model. Table 5 shows the values of MSE, RMSE, and MAPE (%) of predicted output by the four models according to formula (12). By EDIW-PSO-RBF model, the three errors MSE, RMSE, and MAPE (%) are, respectively, $4.9985 * e^2$, 0.0080, and 7.5166, which are all the smallest among the four models.

5. Conclusion

In this paper, we present and discuss an improved EDIW-PSO-RBF model to solve prediction problem. Based on the EDIW-PSO algorithm, we optimize the centers, widths, and connection weights of radial basis function (RBF) neural network. Furthermore, EDIW-PSO-RBF model is applied to the daily air quality index (AQI) prediction comparing with LDIW-PSO-RBF model, NDIW-PSO-RBF model, and GLbestIW-PSO-RBF model. In the processing of optimizing RBF, the value of fitness obtained by the proposed EDIW-PSO method is smaller than the values obtained by other three methods. And for the train data and the prediction test data, the MSE, RMSE, and MAPE (%) obtained by the improved EDIW-PSO-RBF model are all smaller than the values of which by the other three models. From the simulation results it can be observed that the PSO with the proposed inertia weight, EDIW, provides better result compared to the other PSO methods with LDIW and NDIW as well as the GLbestIW. The proposed EDIW-PSO-RBF model can be satisfactorily used in other prediction problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is fully supported by the National Natural Science Foundation Grant no. 61275120 of China.

References

- [1] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Royal Signals and Radar Establishment Malvern (UNITED KINGDOM), 1988.
- [2] G. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 57–67, 2005.
- [3] S. Albrecht, J. Busch, M. Kloppenburg, F. Metzke, and P. Tavan, "Generalized radial basis function networks for classification and novelty detection: self-organization of optimal Bayesian decision," *Neural Networks*, vol. 13, no. 10, pp. 1075–1093, 2000.

- [4] H. Han, Q. Chen, and J. Qiao, "An efficient self-organizing RBF neural network for water quality prediction," *Neural Networks*, vol. 24, no. 7, pp. 717–725, 2011.
- [5] M. B. Li, G. B. Huang, P. Saratchandran, and N. Sundararajan, "Performance evaluation of GAP-RBF network in channel equalization," *Neural Processing Letters*, vol. 22, no. 2, pp. 223–233, 2005.
- [6] A. Zhang, C. Chen, and H. R. Karimi, "A new adaptive LSSVR with on-line multikernel RBF tuning to evaluate analog circuit performance," *Abstract and Applied Analysis*, vol. 2013, Article ID 231735, 7 pages, 2013.
- [7] B. Walczak and D. L. Massart, "Local modelling with radial basis function networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 2, pp. 179–198, 2000.
- [8] L. J. Herrera, H. Pomares, I. Rojas, A. Guillén, G. Rubio, and J. Urquiza, "Global and local modelling in RBF networks," *Neurocomputing*, vol. 74, no. 16, pp. 2594–2602, 2011.
- [9] A. Saranlı and B. Baykal, "Complexity reduction in radial basis function (RBF) networks by using radial B-spline functions," *Neurocomputing*, vol. 18, no. 1–3, pp. 183–194, 1998.
- [10] A. Punzi, A. Sommariva, and M. Vianello, "Meshless cubature over the disk using thin-plate splines," *Journal of Computational and Applied Mathematics*, vol. 221, no. 2, pp. 430–436, 2008.
- [11] J. Z. H. Zhang and H. Y. Li, "A reconstruction approach to CT with cauchy RBFs network," *Advances in Neural Networks*, vol. 3174, pp. 531–536, 2004.
- [12] M. A. T. Figueiredo, "On gaussian radial basis function approximations: interpretation, extensions, and learning strategies," in *Proceedings of the IEEE 15th International Conference on Pattern Recognition*, vol. 2, pp. 618–621, Barcelona, Spain, 2000.
- [13] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 3, pp. 302–309, 1991.
- [14] R. Langari, L. Wang, and J. Yen, "Radial basis function networks, regression weights, and the expectation: maximization algorithm," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 27, no. 5, pp. 613–623, 1997.
- [15] N. B. Karayiannis, "Reformulated radial basis neural networks trained by gradient descent," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 657–671, 1999.
- [16] P. S. Grabusts, "A study of clustering algorithm application in RBF neural networks," *Scientific Proceedings of Riga Technical University*, pp. 50–57, 2001.
- [17] S. A. Billings and G. L. Zheng, "Radial basis function network configuration using genetic algorithms," *Neural Networks*, vol. 8, no. 6, pp. 877–890, 1995.
- [18] C. Man, X. Li, and L. Zhang, "Radial basis function neural network based on ant colony optimization," in *Proceedings of the International Conference on Computational Intelligence and Security Workshops (CIS '07)*, pp. 59–62, Harbin, China, December 2007.
- [19] S. Chen, X. Hong, B. L. Luk, and C. J. Harris, "Non-linear system identification using particle swarm optimisation tuned radial basis function models," *International Journal of Bio-Inspired Computation*, vol. 1, no. 4, pp. 246–258, 2009.
- [20] D. Wu, K. Warwick, Z. Ma et al., "Prediction of parkinson's disease tremor onset using a radial basis function neural network based on particle swarm optimization," *International Journal of Neural Systems*, vol. 20, no. 2, pp. 109–116, 2010.
- [21] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.
- [22] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
- [23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.
- [24] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," *Computers and Operations Research*, vol. 33, no. 3, pp. 859–871, 2006.
- [25] M. S. Arumugam and M. V. C. Rao, "On the performance of the particle swarm optimization algorithm with various inertia weight variants for computing optimal control of a class of hybrid systems," *Discrete Dynamics in Nature and Society*, vol. 2006, Article ID 79295, 17 pages, 2006.
- [26] L. Kaiyou, Q. Yuhui, and H. Yi, "A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization," in *Proceedings of the 1st International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA '06)*, pp. 977–980, January 2006.
- [27] R. Chen and C. Wang, "Project scheduling heuristics-based standard PSO for task-resource assignment in heterogeneous grid," *Abstract and Applied Analysis*, vol. 2011, Article ID 589862, 20 pages, 2011.
- [28] <http://www.xianemc.gov.cn/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

