Harmony Search Algorithm with Chaos for Training RBFNN

Jiaqi Di
College of information and Computer Engineering, Northeast Forestry University
College of computer science and information engineering, Harbin Normal University
Email:di_jacky@163.com

Nihong Wang
College of information and Computer Engineering, Northeast Forestry University
Email: wnh@mail.nefu.edu.cn

***Abstract*—Harmony search algorithm is a heuristic optimization method inspired from the improvisation process of musicians. A new version of harmony search algorithm with chaos is proposed. This algorithm initializes and updates the harmony memory with the chaos optimization algorithm based on secondary carrier wave, adjusts the parameters dynamically to improve the convergence rate and optimization accuracy. In this paper this new algorithm is used to train RBF neural network. The training performance and generalization capability of the new algorithm are tested and verified on function approximation and Iris classification problem. The training time, MSE and accuracies of the new algorithm are compared with standard HS, IHS, GHS algorithms. It turns out that the new algorithm has better convergence rate and accuracy than others. Finally, we applied the new algorithm into the sewage treatment water quality prediction. The relative errors between the actual values and predictive values prove that a RBFNN trained by the new algorithm has better performance than others.**

***Index Terms*—Harmony Search Algorithm, Chaos Optimization Algorithm, RBF Neural Network, Water Quality Prediction**

I. INTRODUCTION

The reason why so many scholars take high attention to neural networks for so many years is neural networks have impressive properties such as adaptability, strong learning capability and ability to generalize. NN training process is to constantly adjust the individual weights and biases between different neural. Neural network is alike a black box, using a dataset called training set include pairs of inputs and outputs to establish a nonlinear model. This learning process will continue until the errors between the outputs and the desired outputs achieve a fixed value by adjusting the network weights and biases.

As its short training time and approximate any continuous function to arbitrary precision, radial basis function neural network (RBFNN) is used widely in many areas. RBFNN has a variety of learning algorithms, such as k-means clustering algorithm, gradient descent method, orthogonal least squares and so on. But all these are based on the centers of the radial basis function are determined correctly. Bad centers will cause jumbled and huge neural network structure, even affect the performance of the network. To make up for this shortcoming, many intelligence algorithms have been used for the training of RBFNN, such as genetic algorithm (GA) [1] and [2] , particle swarm optimization (PSO) algorithm[3]and[4]. Harmony search (HS) algorithm [5] is a meta-heuristic algorithm which mimics the improvisation process of musicians. As its simple concept and easy to implement, it has been adopted for training NNs too. Sinem kulluk adopted Self-adaptive global best harmony search algorithm to train NNs for classification problems [6]. They compared the results with other harmony search algorithms and BP algorithm. The experiments presented the SGHS algorithm performed better than the other algorithms.

In this paper, a new harmony search algorithm with chaos is proposed. The new algorithm dynamically updates values of the harmony memory considering rate (HMCR), specifies Bandwidth (BW) adjustable direction. This can improve the optimization accuracy of the algorithm. It not only use the chaos algorithm [7]to initialize the harmony memory, but also use the current optimal solution to be the secondary carrier to narrow down the searching area, guiding the mechanism doing the searching towards to the optimal solution area. By this mean it will improve the convergence rate greatly. We used the new algorithm to train RBFNN, investigated the performance of the RBFNN from function approximation and iris classification problem and compared the results with those obtained by standard HS, IHS, GHS algorithms. Finally, we applied the new algorithm into the sewage treatment water quality prediction. Water quality prediction results obtained by this new algorithm are compared with the other algorithms' too.

The rest of this paper is organized as follows: in

Manuscript received January 16, 2013; revised March 1, 2013; accepted April 3, 2013.

Corresponding author: Nihong Wang

doi:10.4304/jsw.8.9.2231-2237

Section 2, review of the standard harmony search algorithm is summarized, and a short introduction on improvements for HS is offered. Section 3 introduces our method in detail. Section 4 introduces the general procedure of the new algorithm for training RBFNN. The function approximation and iris classification problem are given to verify the performance of the algorithm. In section 5, the proposed method is applied to predict the water quality of the sewage treatment. The simulation results are compared with those obtained via other algorithms. Finally, conclusion is given in the last section.

II. Harmony Search Algorithm

Harmony search algorithm is a meta-heuristic algorithm proposed in recent years. The concept comes from the improvisation process of musicians. Like the musicians repeatedly adjust the tone of each instrument in the band to eventually achieve the most pleasant music evaluated by the audience's esthetics, the harmony search algorithm is to find the optimal solution which most fit the objective function. Each musical instrument is analogous to each decision variable; musical instrument's pitch range is analogous to decision variable's value range; musician's improvisations are analogous to local and global search schemes in optimization; audience's esthetics are analogous to objective function; the most beautiful harmony is analogous to the optimal solution vector.

However, HS algorithm uses fixed values of pitch adjusting rate (PAR) and BW which two parameters control the convergence rate and the ability for fine tuning. Mahdavi proposed a new version of the HS, called the improved harmony search (IHS) [8]. IHS algorithm dynamically updates values of PAR and BW. Wang and Huang proposed a different version of the HS to set the parameters, called self-adaptive harmony search algorithm [9].They used the harmony memory to automatically adjust parameter values. Thus this mechanism could progressively make finer tuning. They also used the low-discrepancy sequences to initialize the harmony memory rather than the pseudo-random number generator. Benefit from the thought of the particle swarm algorithm, Omran and Mahdavi proposed the global best HS algorithm (GHS) to enhance the performance of the HS [10]. GHS algorithm modifies the pitch adjustment rate rule of the standard HS in such a way that when improvising a new harmony should consider the best harmony in the HM. Bilal Alatas proposes a new HS algorithm that use chaotic maps for parameter adaptation in order to improve the convergence characteristics and to prevent the HS to get stuck on local solutions [11].

A. Standard Harmony Search Algorithm

Standard HS algorithm can be described into three important steps: initial, improvise a new harmony and update harmony memory. An initial population of harmonies is randomly generated and stored in a memory called harmony memory (HM). The number of the harmony vectors in the HM is called harmony memory size(HMS).At each iteration a new candidate harmony is improvised using three rules: 'memory consideration rule', 'pitch adjustment rule' and 'random selection'. This newly generated harmony is compared with the worst harmony in the current HM. If the new harmony is better than the worst harmony, then the worst harmony vector is replaced by the new candidate harmony vector. HM is updated. This process is repeated until the number of iterations (NI) is reached. Summarize the general procedure of standard HS algorithm as follow.

Setp1: Construct the object function; initial the parameters and number of iteration.

Standard HS algorithm has five basic parameters, which are HMS, HMCR, PAR, BW and NI. Except HMS and NI, all the other parameters take value between 0 and 1.

Step2: Initial the HM.

HM is constituted by HMS harmony vectors. These harmony vectors are randomly generated, the generating function is like this:

$$x^i = L + (U - L) \times r \qquad (1)$$
$$\mathrm{i} = 1,2,...\mathrm{HMS}\ , \text{where } \mathrm{r} \in (0,1)$$

U and L are the upper and lower limits of the harmony vector. HMS harmony vectors generate the following HM matrix. M is the number of dimensions of the harmony vector.

$$HM = \left[\begin{array}{cccc|c} x_1^1 & x_2^1 & \cdots & x_m^1 & f\left(x^1\right) \\ x_1^2 & x_2^2 & \cdots & x_m^2 & f\left(x^2\right) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_m^{HMS} & f\left(x^{HMS}\right) \end{array}\right]$$

Step3: Improvise a new harmony.

The new harmony vector has HMCR probability comes from the harmony memory, 1-HMCR probability generates randomly, as in (2).

$$x_j^{new} \leftarrow \begin{cases} x_j^{new} \in \left\{x_j^1, x_j^2, \ldots, x_j^{HMS}\right\} \\ \text{with probability HMCR} \\ x_j^{new} \in X_j \\ \text{with probability 1-HMCR} \end{cases} \qquad (2)$$

If the new harmony vector comes from the harmony memory, there is PAR probability further adjustment is required according to BW parameter, as in (3). BW is the step size of fine tuning.

$$x_j^{new} \leftarrow \begin{cases} x_j^{new} \pm \mathrm{r} \times BW \ \text{with probability PAR} \\ x_j^{new} \quad \text{with probability 1-PAR} \end{cases} \qquad (3)$$

Step4: Update the HM.

If the new harmony is better than the worst harmony, then the worst harmony vector is replaced by the new candidate harmony vector, HM is updated.

Step5: Repeat Step3, Step4 until NI is reached. The

best harmony vector in the HM is the optimal solution.

B. Improved Harmony Search Algorithm

Mahdavi proposed the improved harmony search for improving the performance of HS algorithm. IHS algorithm dynamically updates values replacing the fixed values of PAR and BW, as in (4) and (5).Other steps of the IHS algorithm are the same as the standard HS algorithm:

$$PAR(gn) = \frac{PAR_{\min} + (PAR_{\max} - PAR_{\min})}{NI} \times gn \quad (4)$$

$$BW(\mathrm{gn}) = BW\max \times \mathrm{e}^{\gamma}$$
$$\gamma = \ln(\frac{BW\min}{BW\max}) \times gn \div NI \quad (5)$$

The gn means current iteration.

C. Global-best Harmony Search Algorithm

Benefit from the thought of the particle swarm algorithm, Omran and Mahdavi proposed the global best HS algorithm to enhance the performance of the HS. GHS algorithm modifies the pitch adjustment rate rule of the HS in such a way that the new harmony consider the best harmony in the HM and remove the parameter BW. Since the optimal solution vector most fit the objective function, the harmony vectors which fit the objective function well should have a higher possibility to be selected. Then the (3) is replaced by $x_j^{new} = x_j^{best}$ .Other steps remain the same as the HS algorithm.

III. Harmony Search Algorithm With Chaos

In this section, we introduced our method include the dynamic HMCR, BW adjustable direction and chaos optimization algorithm for initialing and searching. The method of setting PAR and BW are adopted Mahdavi's idea as in (4) and (5).

A. Dynamic HMCR

Inspired by the (4) and reference [错误!未定义书签。], we use a dynamically updating Equation of HMCR as in (6).

$$HMCR(gn) = \frac{H_{\max} - (H_{\max} - H_{\min})}{NI} \times gn \quad (6)$$

HMCR changes from small to large. In the early stage of the searching, larger values conducive to find local optimal solution. In the late stage of the searching, smaller value can increase the diversity of solutions to escape from the local optima.

B. BW Adjustable Direction

Adopting the idea of reference [12], we specify the BW adjustable direction. If the fitness value of the new harmony is less than the average value of all harmonies in the HM, then $x_j^{\mathrm{new'}} = x_j^{\mathrm{new}} + BW$. On the contrary, $x_j^{\mathrm{new'}} = x_j^{\mathrm{new}} - BW$.

C. Chaos Optimization Algorithm

Chaotic motion seems a messy process. In fact, there are some fine inherent laws in it. It has strong randomness, ergodicity, regularity and can reach all state in a specific area without repeating. No doubt using chaotic algorithm to search for the optimal solution is superior to search randomly .In this paper, we use the typical logistic chaos mapping as in (7).

$$y_{n+1} = \mu y_n(1 - y_n)$$
$$\mathrm{n} = 0,1,...,0 \le \mu \le 4 \quad (7)$$

Where μ is the chaotic attractor, n is the number of iterations. When μ is 4, the system is in completely chaotic state.

Chaos theory is used in these aspects: First one, Initial. Using a random harmony vector as the initial value to generate a chaos sequence, taking a few of vectors to put into HM, competing the fitness of all the harmony vectors until there are HMS harmony vectors and one vector be the best harmony for several times. Moreover all the harmony vectors in the HM are unique. Second one, produce a new chaos sequence. Using the best harmony with fine tuning according to BW as the initial value to be the secondary carrier and generate a new chaos sequence. Last one, according to HS algorithm with the new parameter setting methods to improvise a new harmony. When there is 1-HMCR probability to generate a new harmony outside the HM, we can directly take a vector from the current chaos sequence rather than generate randomly.

IV. The New Algorithm Trains RBFNN

A. A Brief Overview of RBFNN

RBFNN is a typical three-layer feed forward neural network including input layer, hidden layer and output layer. The input layer passes the input signals to the hidden layer. The hidden layer maps the input signals to the nonlinear space by the radial basis function. The output layer linear classifies the nonlinear pattern. A simple RBFNN structure is shown in figure 1.

In RBFNN, the most commonly used radial basis function is Gaussian function as in (8).

$$H_i(x) = \exp[-\|x - c_i\|^2 / (2\sigma_i^2)], i = 1,2,..s \quad (8)$$

Where s is the number of the hidden layer nodes. H, c, σ is the output, center and bandwidth of a hidden layer node. The σ limits the scope of the function. It indicates the degree of sensitivity of the neurons to external stimuli. The smaller the value is, the more sensitive it is. Overall, the hidden layer uses radial basis function for a nonlinear variation by mapping the m-dimensional inputs into s-dimensional space to extraction important features.

The output layer gathers the information from the

hidden layer linearly combination with different weight. As the excitation function is a pure linear function, so the output is as in (9).

$$f_j(x) = \sum_{i=0}^{s} \lambda_{ji} H_i(x) \tag{9}$$

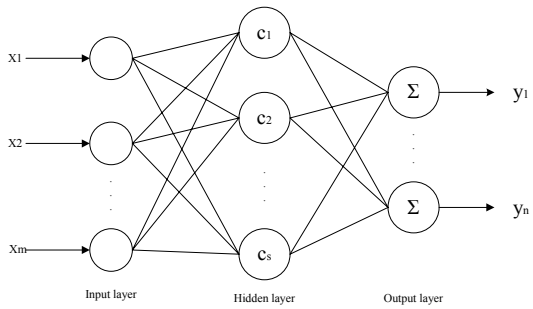

Figure 1. RBFNN structure

B. The New Algorithm Trains RBFNN

The concept of using the proposed algorithm to train RBFNN is regard the network learning process as searching for the optimal solution in the solution space. Compose all the key parameters in the RBFNN which are centers c, widths δ and weights ω between the hidden layer and the output layer together to be an m-dimensional vector. Using k-means cluster algorithm to roughly get the number of the hidden nodes. Based on this value, give an appropriate range to search so that we can obtain the optimal network structure as soon as possible.

The procedure of the new algorithm training RBFNN is described as follow. Figure 2 is the flow chart of the process that the new HS algorithm trains RBFNN.

Step1: Initial all the parameters. According to the training set and number of the hidden nodes got by the k-means cluster algorithm, construct a possible network structure. Initial the parameters in the HS, such as HMS, HMCR, PAR, BW and NI.

Step2: Initial or update the HM. There are HMS harmony vectors in the HM. Each harmony vector has m variables. M is the number of all the centers, widths and weights. First, generating a harmony vector randomly or getting the current best harmony vector. Then produce a new chaos sequence based on this harmony vector with adjustment about BW via the secondary carrier wave. At last, taking a few of vectors to put into the HM, competing the fitness of all the harmony vectors until there are HMS harmony vectors in the HM and the best harmony stays the same for several times. Moreover all the harmony vectors in the HM are unique.

Step3: Translate the harmony vector into the parameters of the neural network and train it with the training set. Calculate the standard error on the training set and the fitness value of the harmony vectors. The purpose of using the new algorithm to train RBFNN is to meet certain accuracy requirements in a simple network topology, make the approximation error accuracy and network complexity indicators minimum. The standard error is the square root of the sum of squares of the difference of the desired outputs and the actual outputs. Network complexity is determined by the number of the hidden nodes. The fitness function needs to reflect these two aspects of performance. Therefore the fitness function is as (10).

$$E = \sqrt{\frac{n \times \sum_{i=1}^{R} (P-T)^2}{R}} \tag{10}$$

Where P is the desired outputs; T is the actual outputs; R is the number of training samples; n is the number of the hidden nodes. Obviously, the smaller the standard and m are, the smaller the fitness function value is.

Step4: Improvise a new harmony and compare the fitness with other harmony vectors in the HM. This step is the same as the step 3 and step 4 in the HS algorithm except three points. The first one, the HMCR is dynamically update as (6). The second one, the BW adjusting direction is just as what we said in Ⅲ.B. The last one, when the random constant is larger than the HMCR, get harmony vector directly from the current chaos sequence rather than generating randomly outside the HM.

Step5: Repeat step 2-4 until reach the termination condition. There are two situations to reach the termination condition. One is the iteration reaches the specified maximum iterations. The other is the fitness value less than a certain constant ε. At this moment, the network approximation accuracy and network complexity all have achieved the most optimal state.

Step6: Translate the optimal harmony vector into the RBFNN parameters and save all data.

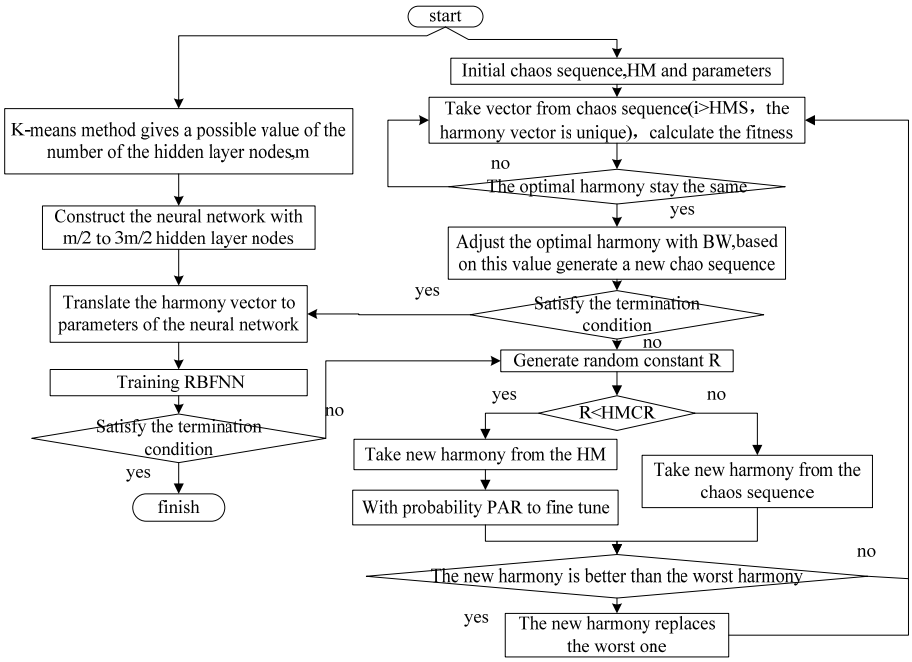

Figure 2. Flow chart of the new HS algorithm training RBFNN

C. Test Experiment

RBFNN has been used for function approximation and classification problems usually. So we use function $y = \sin x \times e^{-x}$ and Iris classification problem to do this test experiment. All the training set and testing set are trained and tested for 10 times, we get the average for comparing. The main Parameters of HS, IHS, GHS and the new algorithm we proposed are set as in table Ⅰ. The maximum iteration is 500.

The training set is composed by 100 samples generated randomly in [0,π] and their corresponding function value. The testing set is composed by 60 values generated randomly in [0,2π] with π/100 increment and their corresponding function value. The input and output layer all have one neural only. The possible number of hidden nodes derived by the k-means cluster algorithm is 9.

TABLE I.
PARAMETER SETTING OF HS, IHS, GHS, NEW HS ALGORITHM

Algorithm	HMS	L	U	HMCR	PAR	BW
HS	8	-1	1	0.9	0.3	0.01
IHS	8	-1	1	0.95	Min:0.01 Max:0.99	Min:0.001 Max:0.1
GHS	8	-1	1	0.9	0.3	-
New HS	8	-1	1	Min:0.01 Max:0.99	Min:0.01 Max:0.99	Min:0.001 Max:0.1

The results in Table Ⅱ show that GHS and the new HS we proposed have better performance than others, and the new HS is better than GHS in training time, average MSE and SSE. Finally, we get the optimal number of hidden nodes is 7.

TABLE II.
TRAINING TIME AND MSE IN THE FUNCTION APPROXIMATION PROBLEM

Algorithm	Training time(s)	Mean Squared Error, MSE			
		Average MSE	Median MSE	Standard error	Best MSE
HS	56	6.02e-02	7.83e-02	2.53e-03	3.75e-03
IHS	33	4.87e-02	6.83e-02	3.22e-03	3.04e-03
GHS	26	3.49e-02	4.72e-02	1.83e-03	1.62e-03
New HS	17	7.83e-03	6.03e-02	3.42e-03	1.74e-03

Iris data set is a well-known benchmark used to test the classifier performance. It has 150 samples divided into three kinds: Setosa, Versicolor and Virginica. Each of them has 50 samples. All the data have four features: sepal length, sepal width, petal length, and petal width.40 samples of each of them for network training, rest 10 samples of each of them for testing. The network contains four input nodes which stand for four properties of the flower, three output nodes stand for three categories of flowers. The possible number of hidden nodes derived by the k-means cluster algorithm is 13.

In the table Ⅲ, the training times of four algorithms are short. The new algorithm we proposed still is the first one to complete the train process and find the global optimum. GHS also has good testing accuracy and short training time. IHS has more high accuracy than HS, but the training time is the same as the HS. The optimal number of hidden nodes is 12.

TABLE III.
MSE, TRAINING AND TESTING ACCURACY COMPARISONS IN THE IRIS CLASSIFICATION PROBLEM

Algorithm	Training time(s)	MSE	Training accuracy	Testing accuracy
HS	8	5.43e-02	96.838	96.303
IHS	8	2.68e-02	97.464	97.322
GHS	6	1.75e-02	97.502	97.743
New HS	5	1.90e-02	97.568	97.689

All in all, HS has the longest training time. The training and testing accuracy need improving. Although IHS has better accuracy than HS, it is still not good enough, and the training time is still too long. GHS performances quite well, the results are very close to the new HS. But the new HS has fast convergence speed, can find the global optimal solution quickly. So hybrid the chaos algorithm with the harmony search algorithm together is feasible and effective.

V. APPLICATION IN WATER QUALITY PREDICTION

Sewage treatment is a highly complex, nonlinear biochemical reaction process. Artificial neural network is widely used in the sewage treatment field of intelligent modeling. According to the input water qualities and the process parameters, we can predict the output water quality. This can give some decision supports to the workers to control the sewage treatment process avoiding the output water quality overpass integrated wastewater reactor (SBBR) activated sludge process. The input player is composed by 5 nodes. They are influent flow, temperature of the sewage, input chemical oxygen demand (COD), total nitrogen (TN) and total phosphorus (TP). The output layer includes aeration time, output COD, TN and TP. The data set comes from one sewage treatment plant adopted SBBR activated sludge process in Chongqing. After data pretreatment, we get 196 samples.166 samples are for training, the rest 30 samples are for testing. The possible number of hidden nodes derived by the k-means cluster algorithm is 15.HS, IHS, GHS and the new HS algorithm train the RBFNN of water quality prediction separately. The parameter setting is the same as the table Ⅰ.They share the same network architecture and objective function, too. Figure 3 Shows the testing results of aeration time, output COD, TN and TP of RBFNN trained by HS, IHS, GHS and New HS algorithms.

From figure 3 we can see that GHS and the new HS algorithms performance much better than HS and IHS algorithms. The relative errors of all the testing results are shown in table Ⅳ.

It is obvious in the table Ⅳ that the new HS algorithm we proposed has the least relative error. GHS also performances very well that the relative errors are much less than IHS and HS. In these four outputs, TN and TP are very close to the actual values. But the testing result of aeration time and COD are not so good.

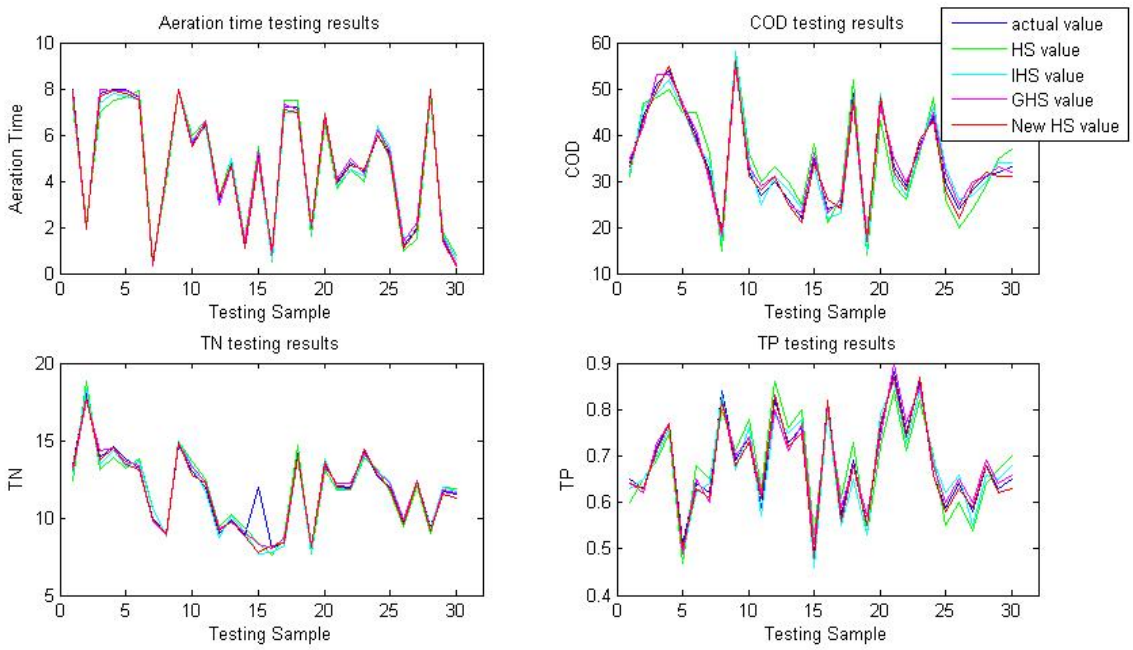

Figure 3. Testing result of aeration time, output COD, TN and TP

TABLE IV.
RELATIVE ERROR OF THE TESTING RESULT

	HS			IHS			GHS			New HS		
Water quality (relative error)	Min (%)	Max (%)	Avr (%)	Min (%)	Max (%)	Avr (%)	Min (%)	Max (%)	Avr (%)	Min (%)	Max (%)	Avr (%)
aeration time	2.4	25	13.3	1.3	16.7	8.7	1.3	10.7	4.3	0.9	8.3	3.8
COD	3.6	21	10.5	2.1	11.2	5.9	0.7	7.4	3.7	0.6	6.7	3.3
TN	1.7	7.5	4.7	0.8	10.8	2.7	0.5	2.2	1.2	0.4	1.2	0.6
TP	2.5	8.2	5.4	1.3	6.1	3.4	0.4	5.1	2.2	0.3	3.4	1.3

VI. CONCLUSION

In this paper, a new variant of harmony search algorithm with chaos is proposed and it is used for training RBFNN. The training performance and generalization capability of the new algorithm are tested and verified on function approximation and Iris classification problem. Training time, few kinds of MSE, training accuracy and testing accuracy of the new HS algorithm are compared with HS, HIS and GHS algorithms. It turns out that the new HS algorithm can train RBFNN with a short training time and high accuracies. So we predict the water quality with RBFNN trained by the new HS algorithm. These four outputs which are aeration time, COD, TN and TP are close to the actual values, especially TN and TP. We also compared the testing results with other harmony search algorithms. The new HS algorithm still has the best prediction accuracy. So we can say it is quite effective and useful. The RBFNN trained by this new HS algorithm have better performance in function approximation and classification problem. But how to find the number of the hidden nodes accurately is our next problem need to be solved.

ACKNOWLEDGMENT

This paper sponsored by Monitoring based on Internet of things technology and application in forestry research. National public welfare industry special funds for scientific research projects. NO.201104037. Sponsored by Forest management decision support sustainable production technologies. The 12th five-year rural areas of national science and technology plan project. NO.2012AA102003-2

REFERENCES

[1] Khairy Elsayed, Chris Lacor, "Modeling and Pareto optimization of gas cyclone separator performance using RBF type artificial neural networks and genetic algorithms," *Powder Technology*, vol.217, pp. 84–99, February 2012.

[2] Ludmila I. Kuncheva, "Initializing of an RBF network by a genetic algorithm," *Neurocomputing*, vol.14, pp. 273–288, February 1997.

[3] George E. Tsekourasa, John Tsimikas, "On training RBF neural networks using input–output fuzzy clustering and particle swarm optimization," *Fuzzy Sets and Systems*, in press.

[4] Iman Poultangaria, Reza Shahnazib, Mansour Sheikhan, "RBF neural network based PI pitch controller for a class of 5-MW wind turbines using particle swarm optimization algorithm," *ISA Transactions*, vol.51, pp. 641–648, September 2012.

[5] Z W Geem, Kim JH, Loganathan GV, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, pp. 60– 68, 2001.

[6] Sinem Kulluka, Lale Ozbakira, Adil Baykasoglu, "Self-adaptive global best harmony search algorithm for training neural networks," *Procedia Computer Science*, vol. 3, pp. 282–286, 2011.

[7] Li Bing, Jiang Weisun, "Chaos optimization method and its application," *Control Theory and Applications*, vol. 14, pp. 613–615, August 1997, In Chinese.

[8] Mahdavi M, Fesanghary M, Danman gir E, "An improved harmony search algorithm for solving optimization problem," *Applied Mathematics and Computation*, vol. 188, pp. 1567–1597, May 2007.

[9] Wang Chia-ming, Huang Yin-fu, "Self-adaptive harmony search algorithm for optimization," *Expert Systems with Application*, vol. 37, pp. 2826–2837, April 2010.

[10] Mahamed G.H.Omran, Mehrdad Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, pp. 643–656, May 2008.

[11] Bial Alatas, "Chaotic harmony search algorithms," *Applied Mathematics and Computation*, vol. 216, pp. 2687–2699, July 2010.

[12] Chen Ying-zhen, Gao Yue- lin, "An improved adaptive harmony search algorithm," *Journal of Tai Yuan University Technology*, vol. 42, pp. 141– 144,154, March 2011, In Chinese.

[13] Jinxue Sui, Li Yang, Zhilin Zhu, "Mine ventilation optimization analysis and airflow control based on harmony annealing search," *Journal of Computers*, vol. 6, pp. 1270–1277, June 2011.

Jiaqi Di was bron in Hei Longjiang, China, in February 1981. A lecturer of College of computer science and information engineering, Harbin Normal University, now studying in College of information and Computer Engineering, Northeast Forestry University for Doctor's degree. He research interests include Computer intelligent algorithm, data base.

Nihong Wang was a professor of college of information and computer engineering, Northeast Forestry University. She research interests digital forestry.