The Algorithms Optimization of Artificial Neural Network Based on Particle Swarm

Yang Xin-quan[1,2*]

[1]Computer and Information Engineering Department, HeZe University of Heze, Shandong, 274015, China; [2]Embedded systems and Internet of Things institute, HeZe University of Heze, Shandong, 274015, China

Abstract: As a new kind of swarm intelligence algorithm, particle swarm optimization (PSO) algorithm can be calculated conveniently to achieve fast convergence and good convergence performance advantages. However, it shows shortcoming of falling into local extreme point. In this paper, a harmony search algorithm was used to improve PSO. Harmony Search Algorithm, as a new optimization algorithm, presents a good global search performance. By examining four standard test functions, the accuracy of convergence speed or convergence using improved PSO harmony search algorithm was validated.

Keywords: Neural network, optimization, particle swarm optimization algorithm.

1. INTRODUCTION

With the development of artificial neural network research, theoretical researches on the neural network focus on the neural basis of theoretical utilization according to research results and explore more functions. The performance of neural network mathematical model is more superior [1-4]. The studies on network algorithms and performance such as stability, convergence, fault tolerance, and robustness have been widely conducted. Optimization of neural networks has attracted increasing attentions.

PSO algorithm, as a typical swarm intelligence algorithm, is performed by simulating behaviors of birds foraging to form swarm intelligence-based iterative evolutionary computation techniques [5-7]. This algorithm is conducted to realize the particle solution space extreme optimization. Each particle has an objective function determined by fitness value. On the basis of their speeds and own optimal solutions, the optimal solution groups were deduced to finally converge to global optimal solution. PSO algorithm is mainly utilized to perform function optimization, neural network training, engineering applications, and in bioengineering, electromagnetic, data mining and other fields. The results achieved are excellent.

2. BASIC PARTICLE SWARM ALGORITHM

Suppose there are m particles in n-dimensional search space, the weight and volume of each particle are unknown, in the search space at a certain flight speed, the flight speed and direction can be dynamically adjusted according to flight experiences of individuals and groups.

X^i denotes the current position of particle *i*; V_i is the flight speed for particle *i*; P_i is the optimal location of particle i experienced. It is best known as the location of the individual. *i* has experienced fitness value corresponding to the minimum position. For minimization problem, the smaller the objective function value, the better it adapts to the corresponding value.

Considering the minimization problem [8-12], the subject position of the target function is the optimal *f*(*x*), *i* is the particle determined by using the formula (1):

$$P_i(t+1)=\begin{cases} P_i(t), & if\ f(X_i(t+1)) \geq f(P_i(t)) \\ X_i(t+1), & if\ f(X_i(t+1)) < f(P_i(t)) \end{cases} \tag{1}$$

Based on particle swarm, all particles have experienced the optimal location for P_g (*t*), are called as the best position in the group. Then:

$$\begin{aligned} & P_g(t) \in \{P_0(t), P_1(t), \cdots, P_s(t)\} \Big| f(P_g(t)) \\ & = \min\{f(P_0(t)), f(P_1(t)), \cdots, f(P_s(t))\} \end{aligned} \tag{2}$$

The basic particle swarm optimization evolutionary is

$$\begin{aligned} & V_{ij}(t+1) \\ & = V_{ij}(t) + c_1 r_{1j}(t)[P_{ij}(t) - X_{ij}(t)] + c_2 r_{2j}(t)[P_{gj}(t) - X_{ij}(t)] \end{aligned} \tag{3}$$

The process of basic particle swarm optimization is presented as follows:

Step 1: Initializing the position and velocity of the particle swarm.

Step 2: Calculating the fitness of each particle.

Step 3: Illustrating the current positions and personal best positions of each particle respectively prior to comparing their results. If the fitness of the current position of the personal best position is better than before, most individuals are updated to excellent location.

Step 4: For all particles, their current positions and the best positions before drawing groups are compared. If the

1874-110X/14 2014 Bentham Open

individual fitness value is better than the current position of the group before conducting optimal position, the best position is updated in groups.

Step 5: Particle swarm particles are updated based on the speed and position in the formulae 3.2 and 3.3.

Step 6: If the fitness value fails to meet the required standard conditions, or the number of iterations does not reach the set value, step two needs to be repeated. Otherwise ends.

In the case of the particle swarm in evolution, cognitive part is

$$V_{ij}(t+1)=V_{ij}(t)+c_1r_{1j}(t)[P_{ij}(t)-X_{ij}(t)] \tag{4}$$

Then each particle can be updated based on their experiences. If the information sharing and exchange of information among the particles are insufficient, it shows no significance in swarm intelligence.

If the particle swarm evolves only in the social part, we obtain

$$V_{ij}(t+1)=c_2r_{2j}(t)[P_{gj}(t)-X_{ij}(t)] \tag{5}$$

The groups only with PSO have no memory capacity when the information is updated on the optimal position. Although the convergence rate will be accelerated, it is likely to fall into local optima.

3. APPLICATION OF PSO ALGORITHM IN THE HARMONY SEARCH

PSO algorithm has the global search ability, is apt to fall into local optimal solution drawback. Aiming at this shortcoming, this research uses the global search performance and better harmony search algorithm for improving particle swarm algorithm. By initializing the group velocity and position of all the particles, PSO algorithm is to update the particle velocity and position of each particle according to their own best positions and global best positions. Consequently, a global search for obtaining the optimal solution can be carried out. Particle swarm of each particle represents a vector solution; while harmony memory is composed by a number of vector solutions, then one can take the whole particle swarm as a harmony memory. Each vector solution is particle swarm in each particle. In this case, harmony search algorithm and particle swarm optimization can be combined together. By integrating PSO algorithm with harmony search, the basic principle is to initialize the particles in particle swarm initialization. The initial position of each particle is

$$X^k=(X_1^k,X_2^k,\cdots X_n^k) \tag{6}$$

The equivalent of the entire particle swarm harmony memory is

$$HM=\begin{pmatrix} X_1^1 & X_2^1 & \cdots & X_n^1 \\ X_1^2 & X_2^2 & \cdots & X_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^m & X_2^m & \cdots & X_n^m \end{pmatrix} \tag{7}$$

According to previous formulae for updating the speed and position of particle swarm particle, we obtain

$$\begin{aligned} &V_{ij}(t+1) \\ &=V_{ij}(t)+c_1r_{1j}(t)[P_{ij}(t)-X_{ij}(t)]+c_2r_{2j}(t)[P_{gj}(t)-X_{ij}(t)] \end{aligned} \tag{8}$$

$$X_{ij}(t+1)=X_{ij}(t)+V_{ij}(t+1) \tag{9}$$

This research calculates the fitness of each particle to generate a random number between rand 0 and 1. If the rand value is less than the probability of harmony memory HMCR, a new particle is generated. The value of each dimension that the particles correspond to HM matrix column vector dimension randomly selected in the harmony memory. Otherwise, the new particles are generated from the solution space of any value. If the random number is less than the tone adjustment rand probability PAR, the position of the swarm particles are disturbed.

The new particle fitness value (Xp) is calculated. If the fitness is better than that of the worst fitness value in particle swarm of new particles, it will be replaced with a new generation of particles worst fitness value. To achieve a final result, we continue to determine the optimal position of the particle positions and most groups, and update the speed and position of the particle swarm particles.

4. PERFORMANCE OF PARTİCLE SWARM SİMULATİON

In order to verify the performance of integrated PSO algorithm and harmony search, this paper performed four standard test functions experiment.

4.1. Swarm Function

$$y=\sum_{i=1}^{m}(x_1^2-10\cos(2\pi x_1)+10) \tag{10}$$

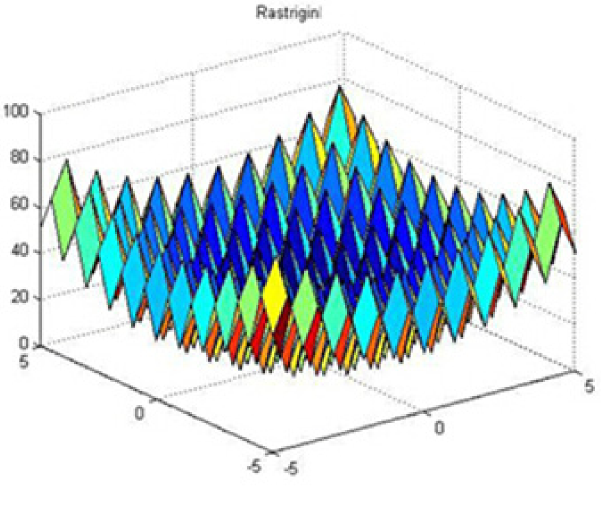


Fig. (1). Restringing function.

Restringing function is a multi-peak continuous function, with a large number of local extreme points of inflection caused by the cosine and each independent variable. Owing to Restringing function has a large number of extreme points, optimization algorithms in search global optimal solution is likely to fall into local extreme point (Fig. **1**). Therefore, the function optimization algorithm is relatively complex. Its extreme value problems are difficult to be solved.

4.2. Gırewank Function

$$y=\sum_{i=1}^{m}\frac{x_i^2}{4000}-\prod_{i=1}^{m}\cos(\frac{x_i}{\sqrt{i}})+1 \qquad (11)$$

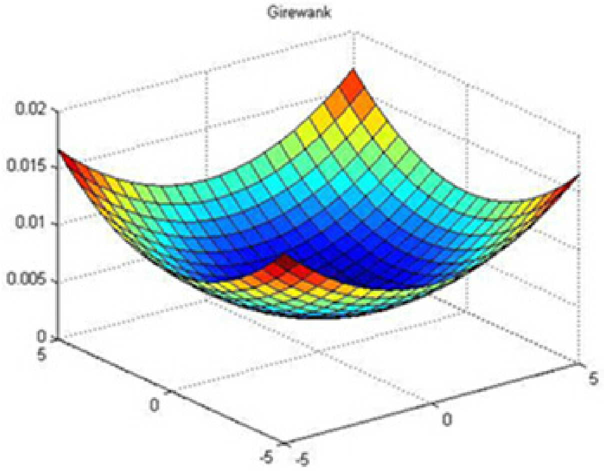


Fig. (2). Girewank function.

Girewank function refers to that the complex multi-peak function among various variables is interrelated. Because Girewank function has a large number of local minima and tall obstructions, so the optimization algorithm is also more likely to fall into local extreme point (Fig. **2**). The function optimization algorithm is also considered to be relatively complex. It extreme value problems are more difficult to be deals with.

4.3. Ackley Functıon

$$y=20\exp(-0.2\sqrt{\frac{1}{m}\sum_{i=1}^{m}x_i^2})-\exp(\frac{1}{m}\sum_{i=1}^{m}\cos(2\pi x_1))+20+e \qquad (12)$$

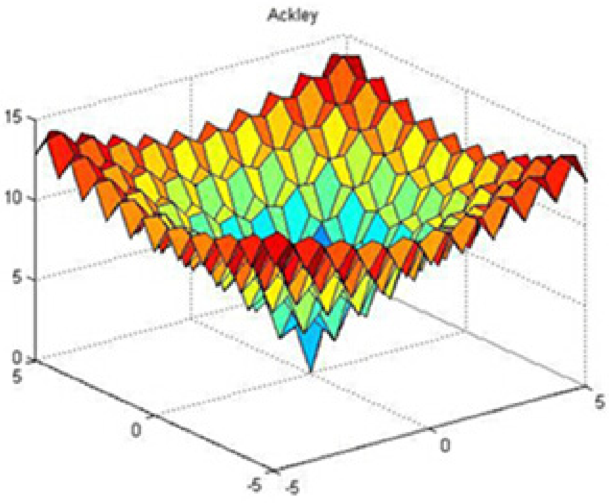


Fig. (3). Ackley function.

Ackley function indicates that a function of the cosine function modulation index generate a multi-peak continuous function. Due to the superposition of adjustment cosine function, Ackley function surface has a lot of local extreme points (Fig. **3**). In comparison with the first two functions, global extreme points of Ackley function show some optimization extremes in optimization algorithm when function is simple.

5. OPTIMIZATION BP NETWORK OF PARTICLE SWARM

BP network is one of the most common and widely used neural networks. In the section two, the artificial neural network and BP algorithm are demonstrated in detail. The BP network connection weights are mainly determined using traditional BP algorithm gradient optimization method. However, this method is not only inefficient, but also apt to fall into local optima. The particle swarm optimization algorithm is likely to achieve rapid calculation. Besides, it is able to act same role as the genetic algorithms evolutionary algorithm in optimization of the neural network.

5.1. Design Ideas of the Algorithm

Step 1: Determine the network structure, as well as all the connection weights in the network including the all thresholds. This figure shows that the solution vector of particle swarm particle dimension. The solution vector of each dimension corresponds to a connection weight. In this case, each particle group of a particle represents a vector consisting of the weights of different particles. This is different with a collection of weights.

Step 2: Initialization of particle swarm. The particle swarm particle size, evolutionary times, reconciliation space particle velocity range, the initial velocity and position of each particle are determined. Meanwhile, the individuals and groups initialized present most optimal position locations.

Step 3: Training of neural network. The particle swarm particle which represents each solution vector is mapped to the weight in the network to construct the neural network. In order to ensure the generalization ability of neural network, the samples collected are divided into training samples and test samples. The input value is introduced into the training sample in the neural network. Afterwards, the actual output value is calculated. Then compared with the output values of the samples, the error values are obtained to calculate each mean square error of the network, so as to adapt the particle values.

Step 4: Updating particle based on PSO algorithm. The fitness value of each particle is used to determine the optimal location and the optimal position of each particle populations. Then, according to the evolution equation for the particle position, velocity of the position and each particle are updated. The particle position is changed. The solution vectors of each particle show changes. By repeating step three, a new fitness value of each particle is calculated again to have loop meeting the condition requirement.

Step 5: Test of neural network. Convergence of the optimal solution space is mapped to the location of the particle weights in the neural network to construct the final neural network. Then test samples are input into the final version of the neural network to calculate the error value and the desired output of the actual output, as an evaluation based on neural networks.

5.2. Evaluation And Analyses of Algorithms

Evaluating capabilities of particle swarm optimization algorithm in neural networks can generally be made based on the performance of the trained neural network. BP network is used to carry out a specific object class or function fitting. There are four indicators in particle swarm optimization BP networks.

(1) Classifying error rate of training samples

The total number of training sample is set by classifying M and BP network. A total of m samples show classification errors in the training. The error rate classified of training sample is:

$$\varepsilon_{Train} = \frac{m}{M} \times 100\% \tag{13}$$

(2) Classification for the error rate of test samples

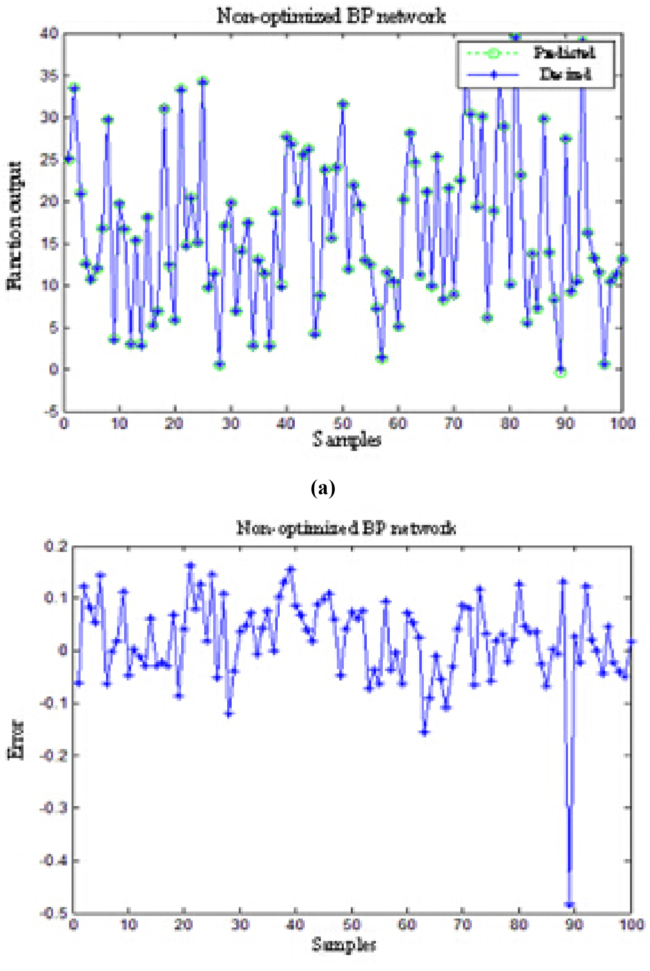
The total number of test sample is set based on N and BP network. A total of n samples present the classification error in training. The classifying error rate of training samples:

$$\varepsilon_{Test} = \frac{n}{N} \times 100\% \tag{14}$$

(3) Mean square errors of training samples

$$MSE_{Train} = \frac{1}{M}\sum_{i=1}^{M}(Y - Y')^2 \tag{15}$$

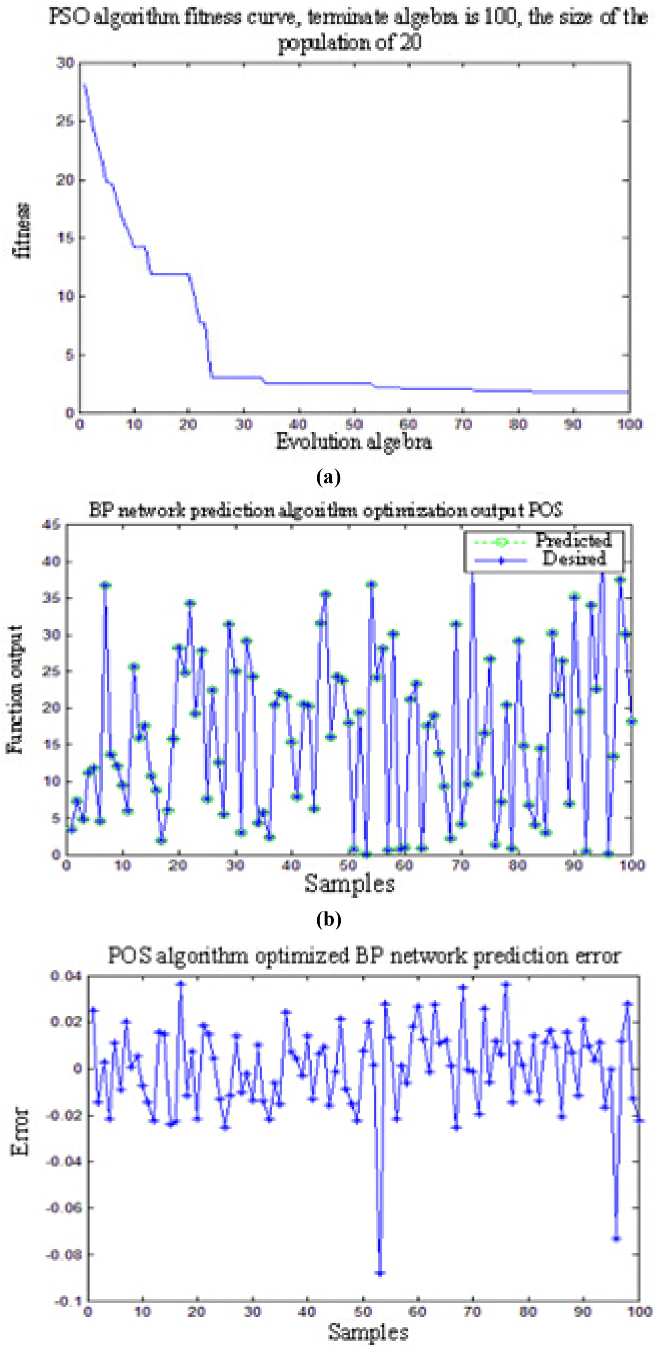
When the BP network is used to solve classification problems, the four performances of tested neural network can be referred.



(a)

(b)

Fig. (4). Non-optimized BP network prediction.

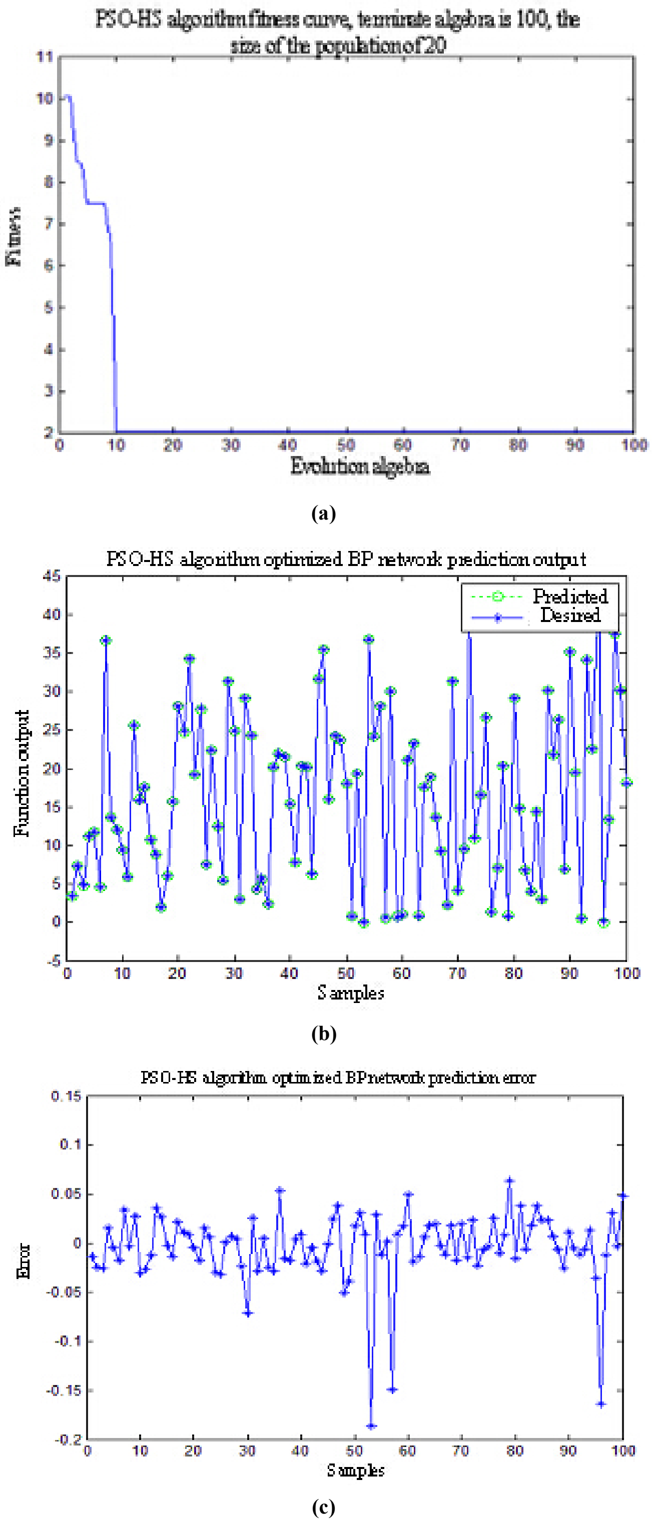


(a)

(b)

(c)

Fig. (5). BP network prediction algorithm optimization using PSO.



(a)

(b)

(c)

Fig. (6). BP network prediction results with PSO-HS algorithm optimization.

6. EXPERIMENT AND SIMULATION

In practical engineering, there are usually a number of complex nonlinear systems with complicated equations. Those systems are difficult to be modeled mathematically in accurate way. In this case, BP neural network can be utilized to present these nonlinear systems. This approach is known as a black box system. The first BP network training system is endowed with input and output data, so that BP network can fit the unknown system. Then, one can use the trained BP neural network to predict the output of the system.

In experiments, BP network is not optimized. Basic PSO algorithm is used to optimize BP network; while PSO-HS algorithm is applied to optimize the simulation results of BP network MATLAB. 100 sets of experiments are conducted. Among which, each case is tested for 100 times. The experimental results are shown in Figs. (**4-6**).

The results are presented as follows:

Based on the optimization process evolving from BP network to the particle swarm optimization harmony search, the convergence rate of the robustness has significantly improved in particle. However, the convergence of the particles is lower than the original PSO algorithm.

For non-optimized BP network, the mean squared error and variance in test samples are 5.843 and 1.6592 respectively. Data distribution is more concentrated. After optimizing BP network using PSO algorithm, the mean squared error of the test sample is 3.7621. The ratio of non-optimized variance (8.7608) decreases apparently. Besides, data distribute more dispersedly. By optimizing BP network using PSO-HS algorithm, the mean squared error and the smallest variance are 1.3556 and 0.4626 respectively. All the data range between [0, 3] and distribute in more concentrated way.

Optimizing BP network using PSO HS algorithm is significantly superior to non-optimized BP network and the BP network optimizing by PSO algorithm.

CONCLUSION

This research integrates particle swarm optimization and harmony search algorithm to optimize the neural network of particle swarm algorithm based on neural network optimization theory, design ideas and performance evaluation. Through conducting modeling, parametric design, programming, and simulation and referring non-optimized BP network, basic PSO algorithm is used to optimize BP network using PSO-HS algorithm. Optimization neural network are conducted 100 experiments. The results verified that optimizing neural network using PSO-HS algorithm has a significant effect.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] E. Kse, "Determination of color changes of inks on the uncoated paper with the offset printing during drying using artificial neural networks", *Neural Computing and Applications,* vol. 25, no. 5, pp. 1185-1192, 2014.

[2] F. Rezaei, P. Karimi, and S.H. Tavassoli, "Effect of self-absorption correction on LIBS measurements by calibration curve and artificial neural network", *Applied Physics B: Lasers and Optics*, vol. 114, no. 4, pp. 591-600, 2014.

[3] K. Amarasinghe, D. Wijayasekara, and M. Manic, “EEG based brain activity monitoring using Artificial Neural Networks”, In: *Proceedings-of 7th International Conference on Human System Interactions HSI*, 2014, pp. 61-66.

[4] K.P. Waidyarathne, and S. Samarasinghe, “Artificial neural networks to identify naturally existing disease severity status”, *Neural Computing and Applications,* vol. 25, no. 5, pp. 1031-1041, 2014.

[5] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization”, *Swarm Intelligence,* vol. 1, no. 1, pp. 33-57, 2007.

[6] G. Venter, and J. Sobieszczanski-Sobieski, “Particle swarm optimization”, *American Institute of Aeronautics and Astronautics Journal,* vol. 41, no. 8, pp. 1583-1589, 2003.

[7] R.C. Eberhart, and Y. Shi, “Particle swarm optimization: developments, applications and resources”, In: *Evolutionary Computation, Proceedings of the 2001 Congress on (IEEE)*, vol. 1, pp. 81-86, 2001.

[8] E.B. Azardoost, and A. Jamshidi, “Providing a genetic algorithm optimized based on regression equations for solving flexible job-shop scheduling problem (With objective function minimization for makespan)”, In: *IIE Annual Conference and Expo 2010 Proceedings,* 2010.

[9] O. Bokanowski, and H. Zidani, “Minimal time problems with moving targets and obstacles”, In: *IFAC Proceedings Volumes (IFAC-Papers Online)*, vol. 18, (PART 1), pp. 2589-2593, 2011.

[10] R.Y. Simanchev, and I.V. Urazova, “An integer-valued model for the problem of minimizing the total servicing time of unit claims with parallel devices with precedences”, *Automation and Remote Control,* vol. 71, no. 10, pp. 2102-2108, 2010.

[11] Y. Ohtsubo, “Minimizing risk models in stochastic shortest path problems”, *Mathematical Methods of Operations Research,* vol. 57, no. 1, pp. 79-88, 2003.

[12] L. Guo, and M. Garland, “The use of entropy minimization for the solution of blind source separation problems in image analysis”, *Pattern Recognition,* vol. 39, no. 6, pp. 1066-1073, 2006.

Received: September 22, 2014 Revised: November 30, 2014 Accepted: December 02, 2014