Research on the Performance of Noisy Chaotic Neural Network based on Travelling Salesman Problem

Haibo Zhang, Xiaoxiang Wang, Hongtao Zhang

Key Laboratory of Universal Wireless Communications, Ministry of Education

Beijing University of Posts and Telecommunications

Beijing, China

***Abstract*—Since Hopfield and Tank applied their neural networks to Travelling Salesman Problem (TSP), this classical NP-hard (non-deterministic polynomial) problem has been intensively studied in the field of artificial neurocomputing. Lipo Wang et al. proposed an efficient approach named noisy chaotic neural network (NCNN), which has been proved to be a powerful tool to solve combinatorial optimization problems in their literatures. However, its exact parameters choice is exquisitely sensitive and complicated under different scenarios. In order to improve the convergence performance, the characteristics of its parameters are investigated in detail again in this paper. We focus on further researching the effects parameters have on the performance of NCNN. Through a large quantity of analyses and numerical simulations, we present the modified scheme with a new parameter set which gives 1) less steep sigmoid function, 2) stronger synaptic weights, and 3) higher initial temperature for annealing. Simulation results show that the modified scheme has much faster convergence speed with a small amount of accuracy loss, compared with the original NCNN which used a traditional parameter set.**

***Keywords*—NP-hard, TSP, NCNN, convergence speed**

I. INTRODUCTION

The TSP is a classical NP-hard combinatorial optimization problem, because the time to solve this problem on computer increases exponentially with the increasing of city number. Some heuristic methods, such as stochastic simulated annealing algorithms, genetic algorithms, combinatorial evolution strategies, especially neural networks techniques, were commonly adopted [1-9]. A model for a large network of neurons with a graded response (or sigmoid input-output relation) was studied [1]. This deterministic system had collective properties in very close correspondence with the earlier stochastic model based on McCulloch-Pitts neurons. After Hopfield and Tank's work [2], there have been extensive research interests and efforts in theory and application of Hopfield neural networks (HNN) because of its advantage over other approaches for solving optimization problems. But HNN often fails to converge to valid solutions. When it converges, the obtained solution is often far from the optimal solution.

Luonan Chen and Kazuyuki Aihara proposed a neural network model with transient chaos, or a transiently chaotic neural network (TCNN) as an approximation method for combinatorial optimization problems, by introducing transiently chaotic dynamics into neural networks in [3]. The proposed neural network has richer and more flexible dynamics, so that it can be expected to have higher ability of searching for global optimal or near-optimal solutions. Their computer simulations showed that chaotic simulated annealing (CSA) led to good solutions for the TSP much more easily compared to the Hopfield-Tank approach [2]. In [4], they have theoretically proven that TCNN had global searching ability which ensured that neural networks carried out global search to solve the NP-hard problem. However, TCNN also suffers from converging to local minima and isn't sure to find the global optimal solution.

The CSA may not find the global optimal solution no matter how slowly annealing is carried out, because chaotic dynamics are completely deterministic. In contrast, stochastic simulated annealing (SSA) tends to settle down to a global optimum if the temperature reduction is sufficiently slow [5]. Hence, Lipo Wang et al. combined the best features of both SSA and CSA, proposed a new approach for solving optimization problems, i.e., stochastic chaotic simulated annealing (SCSA), by using an NCNN [5-8], which had better performance compared with earlier HNN [1-2] and TCNN [3-4] by introducing stochastic noise into TCNN. Therefore, NCNN contributes to escape from the local minima and converge to a stable equilibrium point, and effectively overcomes the local minima problem.

However, NCNN requires subtle adjustment of parameters, such as the gain factor of neuron activation function, synaptic weights, initial temperature, etc. Because of the intrinsic characteristics of its chaotic and stochastic mechanics, the choice of parameters plays a very important role in solving the NP-hard problem. It even decides whether an NP-hard problem can be successfully solved by NCNN (or by HNN and TCNN).

In this paper, the adjustment and choice of parameters about NCNN are in detail investigated again. We try to make clear the effects of parameters on the performance of NCNN. Especially, it is significantly different in terms of selecting the gain factor of neuron activation function. By analyses and numerical simulations, the modified scheme with a new parameter set is efficient and has much faster convergence speed with a small amount of accuracy loss, compared with the original NCNN which used a traditional parameter set [5].

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

The TSP is a classic of difficult optimization problem. It may be stated as follows: given a group of cities to be visited and the distance between any two cities, find the shortest tour that visits each city only once and returns to the starting point.

If an initial arbitrary ordering of N cities is given, a solution to the TSP may be represented as an $N \times N$ permutation matrix. Each row of this matrix corresponds to a particular city, while each column corresponds to a particular position in the tour (i.e., visiting order). $\mathbf{D}$ is an $N \times N$ distance matrix, in which each element $d_{x,y}$ represents the distance between city x and city y. The travelling distance from x th city to y th city is equal to that from y th city to x th city (i.e., $d_{x,y} = d_{y,x}$, $x \neq y$, and $d_{x,x} = 0$).

B. Problem Formulation

The objective of solving the TSP is to find the shortest route of visiting all given N cities and returning to the starting city. Because there are N cities and N stops, $N \times N$ neurons are used to solve the TSP in the NCNN formulation. Based on the analyses and definitions above, the optimization problem in this paper can be formulated as follows:

$$\text{minimize: } \sum_x \sum_i \sum_{y \neq x} d_{x,y} V_{x,i} (V_{y,i+1} + V_{y,i-1}) \tag{1}$$

$$\text{subject to: } \sum_i V_{x,i} = 1,\ i = 1,\dots,N \tag{2}$$

$$\sum_x V_{x,i} = 1,\ x = 1,\dots,N \tag{3}$$

$$\sum_x \sum_i V_{x,i} = N,\ V_{x,i} \in \{0,1\} \tag{4}$$

where x and y denote the serial numbers of cities to be visited. i is the visiting order. $\mathbf{V}$ is the travelling indicator matrix ($V_{x,i}$ and $V_{y,i}$ are its elements), where each element is an output of a neuron. If $V_{x,i} = 1$, which represents city x is visited at the order i, then the corresponding neuron is turned on, otherwise $V_{x,i} = 0$.

III. The Parameters Choice of NCNN Model For TSP

The above problem (1)-(4) is the NP-hard optimization problem, whose complexity increases exponentially with the increasing of neuron number, the global optimal solution is extremely hard to be found due to the nonlinear characteristic. In this paper, a modified scheme with a new parameter set is provided to solve the problem more effectively.

A. NCNN Model

The NCNN model is resulted from adding decaying stochastic noise into TCNN. NCNN can be described as follows [3-8]:

$$x_p(t) = \frac{1}{1 + \exp\left(-y_p(t) * u0\right)} \tag{5}$$

$$y_p(t+1) = \lambda y_p(t) + \alpha \left(\sum_{q \neq p} w_{pq}(t) x_q(t) + b_p \right) - z_p(t)\left(x_p(t) - I_0\right) + n_p(t) \tag{6}$$

$$z_p(t+1) = (1 - \beta_1) z_p(t) \tag{7}$$

$$n_p(t+1) = (1 - \beta_2) n_p(t) \tag{8}$$

where $x_p(t)$ is the output of neuron p, $y_p(t)$ is the internal state of neuron p, $u0$ is the gain factor of neuron activation function; λ is the damping factor of nerve membrane, α is the positive scaling parameter for inputs, $w_{pq}(t)$ is the connection weight from neuron q to neuron p, b_p is the input bias of neuron p, $z_p(t)$ is the additional self-feedback connection weight of neuron p, I_0 is a positive parameter; $n_p(t)$ is stochastic noise injected into neuron p in the range $[-A_m, A_m]$ with a uniform distribution, A_m is the noise amplitude; β_1 and β_2 are damping factors for the time-dependent neuronal self-coupling and the random noise respectively ($0 \le \beta_1, \beta_2 \le 1$); t is the iteration times.

B. Energy Function and Motion Equation of NCNN

1) Energy Function of NCNN

The energy function E is determined by considering the objective function and all constraints in the problem. The goal of the neural network model for solving TSP is to minimize the energy function E, which is described as follows:

$$E = \frac{A_e}{2} \sum_x \sum_i \sum_{j \neq i} V_{x,i} V_{x,j} + \frac{B_e}{2} \sum_i \sum_x \sum_{y \neq x} V_{x,i} V_{y,i} + \frac{C_e}{2} \Big(\sum_x \sum_i V_{x,i} - N\Big)^2 + \frac{D_e}{2} \sum_x \sum_i \sum_{y \neq x} d_{x,y} V_{x,i} (V_{y,i+1} + V_{y,i-1}) \tag{9}$$

where positive constants A_e, B_e, C_e and D_e are the corresponding punishing parameters.

The first term on the right of (9) is row constraint in (2), because each city is visited once and only once in the tour. Hence, one element must be equal to 1 in each row of matrix $\mathbf{V}$, whereas the other elements in each row must be equal to 0. The value is minimum (converges to 0) only when at least one of them ($V_{x,i} V_{x,j}$) is equal to 0, which can efficiently avoid visiting the same city again.

Similarly, the second term represents whether more than one city are visited at the same order or not in (3). If more than one city are visited simultaneously, then there are several 1 in a certain column of $\mathbf{V}$. Its value will reach zero only when at least one of them ($V_{x,i} V_{y,i}$) is equal to 0, which can efficiently avoid visiting multi-city simultaneously.

The third term represents the constraint of total number of cities and the visiting times to all cities in (4), and its value will converge to zero gradually with NCNN running. The last term represents the optimization objective function in (1), which will reach a positive constant with the least value after system convergence. Consequently, it will minimize the total distance of a tour.

According to NCNN, if the suitable parameters are set, the energy function E will gradually decrease. Moreover, E is bounded (is greater or equal to a certain positive constant in the TSP), so the iteration of the algorithm will lead to a stable state that does not further change with iteration. When NCNN converges to an optimal solution, the fourth term of (9) will reach a positive constant with the least value (and will not decrease even if NCNN keeps on running), and the other terms are all equal to zero.

2) Motion Equation of NCNN

The motion equation for the xi th neuron is derived to seek the state of satisfying E as a positive constant by the gradient descent method. The motion equation of NCNN can be deduced as follows:

$$\frac{dU_{x,i}}{dt} = -\frac{\partial E}{\partial V_{x,i}} = -A_e \sum_{j \neq i} V_{x,j} - B_e \sum_{y \neq x} V_{y,i} - C_e (\sum_x \sum_i V_{x,i} - N) - D_e \sum_{y \neq x} d_{x,y} (V_{y,i+1} + V_{y,i-1}) \quad (10)$$

where $U_{x,i}$ and $V_{x,i}$ are the internal state and the output of the xi th neuron, and the relation of $V_{x,i}(t)$ and $U_{x,i}(t)$ is defined as that of $x_p(t)$ and $y_p(t)$ in (5):

$$V_{x,i}(t) = \frac{1}{1 + \exp\left(-U_{x,i}(t) * u0\right)} \quad (11)$$

The following procedure describes the proposed parallel algorithm, where Euler's method is first adopted to build a discrete time model of (10), and then we add the negative self-feedback to generate transient chaotic dynamics. Finally, we add a damping stochastic noise to obtain the discrete time model of NCNN [5-8]:

$$\begin{aligned} U_{x,i}(t+1) = {} & \lambda U_{x,i}(t) - \alpha A_e \sum_{j \neq i} V_{x,j}(t) - \alpha B_e \sum_{y \neq x} V_{y,i}(t) \\ & - \alpha C_e \left(\sum_x \sum_i V_{x,i}(t) - N \right) \\ & - \alpha D_e \sum_{y \neq x} d_{x,y} \left(V_{y,i+1}(t) + V_{y,i-1}(t) \right) \\ & - z_{x,i}(t) \left(V_{x,i}(t) - I_0 \right) + n_{x,i}(t) \end{aligned} \quad (12)$$

$$z_{x,i}(t+1) = (1 - \beta_1) z_{x,i}(t) \quad (13)$$

$$n_{x,i}(t+1) = (1 - \beta_2) n_{x,i}(t) \quad (14)$$

where the symbols and variables are the same as those in the aforementioned equations.

C. The Parameters Setting of NCNN

Despite that there've been great improvement in terms of the performance of NCNN over the past several years, this model still has some problems which need to be solved when applied to the TSP. Next, we will discuss in detail how to set the various parameters more efficiently.

1) Sigmoid Function

The relationship between the output and the input of the general sigmoid activation function is defined as (11). Fig 1 shows the change curves of $V(U)$ with different $u0$, which controls the steepness (or slope) of a sigmoid activation function. When $u0 \to \infty$, (11) is known as the threshold activation function. When $u0 \to 0^+$, (11) is named the ramp (linear) activation function. Hence, to obtain enough search space, $u0$ should be set as small as possible. If so, the asymptotically stable performance, one of the most important characteristics in sigmoid activation functions, can be utilized fully. Otherwise, the change of $V(t)$ will be very small after $U(t)$ goes beyond a certain range, or the first-order derivative of $V(t)$, i.e., $\partial V(t) / \partial U(t)$, becomes close to infinity. For example, as shown in Fig 1, when $u0 = 250$ (generally used in the previous literatures [3-8]), $V(t) \to 0$ or $V(t) \to 1$ at about $U(t) = \pm 0.02$, which will result in another phenomenon that the outputs of neurons are close to 0 or 1 quickly in a discrete manner when $|U(t)| > 0.02$. As a result, the chaotic dynamic searching space will be shrunk greatly so that the searching probability for optimal solution decreases. Hence, in order to improve searching ability, $u0$ should be decreased to a suitable value.

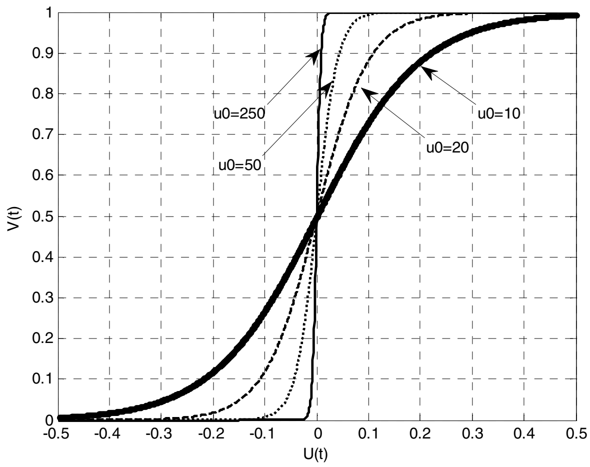

Fig. 1 The sigmoid activation function with different $u0$

In addition, α represents the influence of the energy function to the neurodynamics, or the balance between the self-feedback and the gradient term $\partial E / \partial V_{x,i}$. If the scaling parameter α is extremely small so that the energy function cannot sufficiently affect the neurodynamics, the network converges to a wrong solution. The obtained solutions in these cases are either infeasible solutions or local minima. On the other hand, if α is too large, the influence of the energy function becomes too strong to generate the transient chaos.

2) Chaotic Dynamic and Stochastic Noise

In order to accelerate the convergence process and escape from the local minima, i.e., converge to the global optimal solution as quickly as possible, the parameters of chaos and noise should be adjusted. That is to say, we do not wait for the network to fully converge. Rather, we continuously check the validity of the solution and stop iteration as soon as a valid solution is found even though the chaos or noise does not disappear yet. Hence, the noise amplitude A_m and the initial value $z(0)$ of the additional self-feedback connection weight (i.e., the temperature for annealing) must be large enough so as to find the optimal solution through the rich neurodynamics (stochastic noise and flexible chaos) of NCNN. Correspondingly, the damping factors β_1 and β_2 should be increased in order to speed up the convergence process.

Based on the analysis above, we propose a new parameter set as follows:

$$u0 = 10,\ \alpha = 0.05,\ z(0) = 0.85,\ A_m = 2.$$

It is worth noting that their value are distinctly different from those in [3-8], such as $u0 = 250$, $\alpha = 0.02$, $z(0) = 0.1$, $A_m = 0.02$ were adopted in [5] (named algorithm 1 in this paper). Of course, fine tuning may be necessary when solving different optimization problems or different instances in the same optimization problem.

In addition, the selection of punishing parameters is based on the rule that all terms in the energy function should be comparable in magnitude, so that none of them dominates [6]. Along with the growing problem, the values of them will be increased or decreased slightly.

D. Discretization of the Output of Neuron

The neuron outputs take a continuous value between 0 and 1. To accelerate convergence speed, the neuron output rule is updated by using the average value of neuron output matrix as the threshold to fire the neuron, i.e., to discrete rapidly the final output into 0 or 1 as follows [6]:

$$V_{x,i} = \begin{cases} 1,\ if\ V_{x,i} \geq \dfrac{1}{N^2}\sum_{x=1}^{N}\sum_{i=1}^{N} V_{x,i} \\ 0, \qquad others \end{cases} \tag{15}$$

IV. SIMULATION RESULTS

In this section, the performance evaluation is provided by MATLAB based on the TSP. Although the parameters of NCNN are mainly chosen experimentally in some previous literatures [3-8], in this paper, some different parameters are adopted: $u0 = 10$, $\alpha = 0.05$, $z0 = 0.85$, $A_m = 2$. In addition, others are similar to those used in other optimization problems [3-8]: $I_0 = 0.65$, $\lambda = 0.95$, $\beta_1 = 0.02$, $\beta_2 = 0.02$. The four punishing parameters in (9) are set in this paper as follows: $A_e = 3.5$, $B_e = 3.5$, $C_e = 2.4$, $D_e = 4.3$.

TABLE I: PARAMETERS CHOICE OF THE MODIFIED SCHEME

Case	A_m	$z(0)$	β_1	β_2	Average iteration	Optimal rate %
1	1.8	0.85	0.02	0.02	68	93.6
2	1.9	0.85	0.02	0.02	68	94.9
3	2.0	0.85	0.02	0.02	69	96.4
4	2.1	0.85	0.02	0.02	70	95.2
5	2.0	0.6	0.02	0.02	66	93.4
6	2.0	0.7	0.02	0.02	67	94.3
7	2.0	0.8	0.02	0.02	68	94.9
8	2.0	0.9	0.02	0.02	71	95.4
9	2.0	0.85	0.01	0.01	78	96.9
10	2.0	0.85	0.015	0.015	76	92.1
11	2.0	0.85	0.025	0.025	67	90.7
12	2.0	0.85	0.03	0.03	66	89.1

In order to set these parameters efficiently, the simulation is carried out 1000 times with maximum iterations steps 300 per time to solve the 10-city problem based on the randomly generated initial neuron states $U_{x,i}(0)$ between [-1, 1]. The statistical results obtained by the modified scheme are shown in Table I. As expected, the suitable initial self feedback $z(0)$ can produce enough chaos and implement efficient search to improve the optimal rate. The sufficient initial amplitude A_m of random noise can help the system escape from the local optimal solutions and reduce the probability of suboptimal or infeasible solutions. The iteration steps can also be controlled through damping factor β_1 and β_2. The larger β_1 and β_2, the faster the NCNN converges, but the lower the optimal rate, and vice versa.

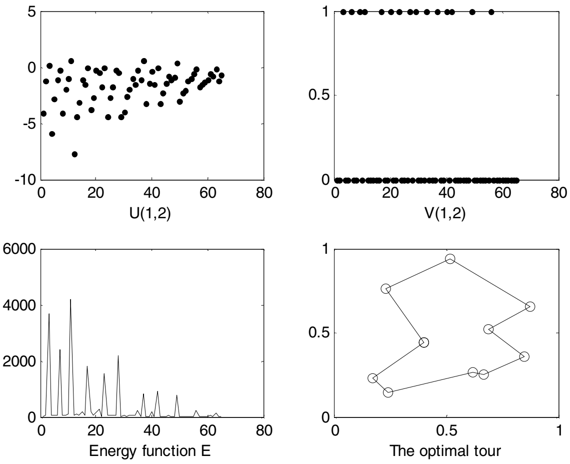

Fig. 2 The optimal tour of 10 cities TSP found by modified scheme

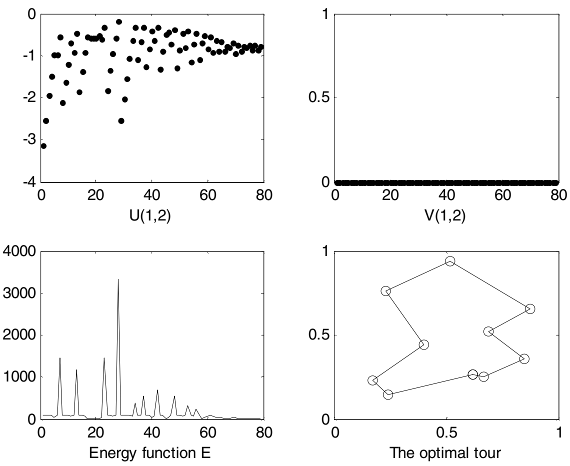

Fig. 3 The optimal tour of 10 cities TSP found by modified algorithm 1

To test our algorithm, the well-trained NCNN model is applied to the Hopfield-Tank 10 cities TSP introduced in section III in [2]. Fig. 2 represents the optimal tour found by modified scheme. Because of the change of the several important parameters, the number of average convergence steps (69 steps) is much lower than that of algorithm 1 in [5]

(124 steps). To compare them further, algorithm 1 is carried out again by the discretization of the output of neuron as (15). We plot the internal state and the neuron output in Fig 3, the No. (1, 2) neuron is selected randomly, i.e., $U_{1,2}$, $V_{1,2}$. The results indicate that the convergence speed is improved evidently (79 steps). Moreover, the energy functions of two modified schemes do not regularly gradually decrease to a fixed point as algorithm 1 in [5]. This is because that each neuron output is forced to be 0 or 1 after system convergence. Note that, although we adopt the scheme of (15) in algorithm 1, the value of $V_{1,2}$ in Fig 3 is always 0 due to the effect of sigmoid function (see Fig 1, $u0 = 250$) whereas 0 and 1 appear alternately because of the successive bifurcations from strange attractors in Fig 2 (see Fig 1, $u0 = 10$). Especially, before the chaos and stochastic noise disappear, two systems have reached a stable point in Fig 2 and Fig 3.

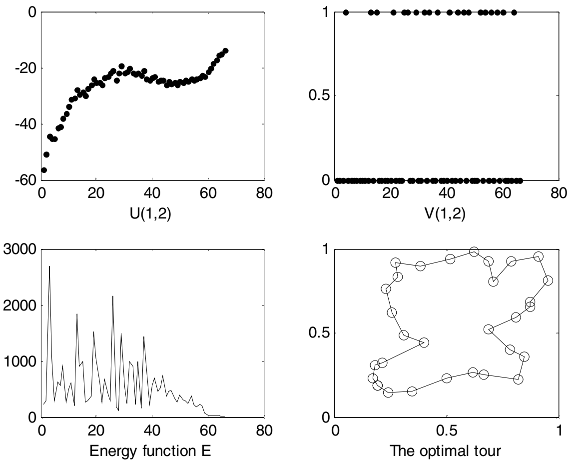

Fig. 4 The optimal tour of 30 cities TSP found by modified scheme

To test our algorithm further, we add 20 cities randomly to the Hopfield-Tank TSP model. In Fig 4, simulation results show that the modified scheme is still effective. To evaluate the efficiency of the modified scheme, we compared it with algorithm 1 (by (15)) under different scenarios, the results are summarized in Table II. According to previous theoretical analyses, with the city number increasing, the convergence performance of the modified scheme is still better than that of algorithm 1. However, the algorithm 1 explores a wider searching space because of the more lasting chaos and additional noise, therefore, it is more capable of jumping out of local minima and providing a little bit better solutions.

TABLE II: COMPARISON OF RESULTS BY THE DIFFERENT TSP

The number of cities	Algorithm 1/ Modified scheme			
	Average iteration	Global minima rate %	Local minima rate %	Infeasible rate %
30	87/74	97.9/96.3	1.8/3.3	0.3/0.4
40	95/79	97.5/96.1	2.1/3.5	0.4/0.4
50	105/83	96.9/95.7	2.7/3.8	0.4/0.5
60	112/89	96.7/95.1	2.8/4.3	0.5/0.6
70	123/98	96.1/94.3	3.3/5.0	0.6/0.7

V. CONCLUSIONS

In this paper, a large number of analyses and numerical simulations illustrate that the noisy chaotic neural network has many uncertainties, and fine tuning of various parameters should be necessary. Whether an NP-hard problem can be solved successfully by NCNN in a large extent depends on the choice of parameters. A reasonable setting method of these parameters is provided through analyzing in detail the impact each parameter has on the final results. The simulation results show that the modified scheme with a new parameter set has much faster convergence speed than the original NCNN. Also, it keeps the balance between the convergence speed and the quality of solution very well.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (60972076, 61072052), and Important National Science and Technology Specific Projects (2010ZX03003-004-03).

REFERENCES

[1] J. J. Hopfield, “Neurons with graded response have collective computational properties like those of two-state neurons”, in Proc. Nat. Acad. Sci. USA, vol. 81, pp. 3088–3092, May 1984.

[2] J. J. Hopfield and D. W. Tank, “Neural computation of decisions in optimization problems”, Biol. Cybern., vol. 52, no. 3, pp. 141–152, 1985.

[3] L. Chen and K. Aihara, “Chaotic simulated annealing by a neural network model with transient chaos”, Neural Netw., vol. 8, no. 6, pp. 915–930, 1995.

[4] L. Chen and K. Aihara, “Global searching ability of chaotic neural networks”, IEEE Trans. Circuits Syst. I, vol. 46, pp. 974–993, Aug. 1999.

[5] L. Wang, S. Li, F. Tian, and X. Fu, “A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing”, IEEE Trans. Syst., Man, Cybern., Part B: Cybern., vol. 34, no. 5, pp. 2119–2125, Oct. 2004.

[6] Lipo Wang, Wen Liu, and Haixiang Shi, “Noisy Chaotic Neural Networks With Variable Thresholds for the Frequency Assignment Problem in Satellite Communications”, IEEE Trans. Systems, man, and cybernetics—part C: applications and reviews, vol. 38, no. 2, Mar. 2008.

[7] Lipo Wang, Wen Liu, and Haixiang Shi, “Delay-Constrained Multicast Routing Using the Noisy Chaotic Neural Networks”, IEEE Trans. Comput., vol. 58, no. 1, Jan. 2009.

[8] Ming Sun, Lin Zhao and Wei Cao etc. “Novel Hysteretic Noisy Chaotic Neural Network for Broadcast Scheduling Problems in Packet Radio Networks”. IEEE Trans. Neu. Net., vol. 21, no. 9, pp. 1422-1433, Sep. 2010.

[9] S. Tabarce, V. Grade Tavares and P. Guedes de Oliveira, “Programmable analogue VLSI implementation for asymmetric sigmoid neural activation function and its derivative”, Electronics Letters, vol. 41, no. 15, July. 2005.