TELKOMNIKA, Vol.11, No.3, March 2013, pp. 1489 ~ 1496
ISSN: 2302-4046

Improved Hysteretic Noisy Chaotic Neural Network for Broadcast Scheduling Problem in WMNs

Ming Sun*, Yanjun Zhao, Zhengliang Liu, Hui Zhang
College of Computer and Control Engineering, Qiqihar University
No. 42 of Wenhua Street, Jianhua District, Qiqhar, China, Telp (+86)452-2738173
*Corresponding author, e-mail: snogisunming@yahoo.com.cn

Abstract

It has been proven that the noise-tuning-based hysteretic noisy chaotic neural network (NHNCNN) can use the noise tuning factor to improve the optimization performance obviously at lower initial noise levels while can not at initial higher noise levels. In order to improve the optimization performance of the NHNCNN at initial higher noise levels, we introduce a new noise tuning factor into the activation function and propose an improved hysteretic noisy chaotic neural network (IHNCNN) model. By regulating the value of the newly introduced noise tuning factor, both noise levels of the activation function and hysteretic dynamics in the IHNCNN can be adjusted to help to improve the global optimization ability as the initial noise amplitude is higher. As a result, the IHNCNN can exhibit better optimization performance at initial higher noise levels. In order to demonstrate the advantage of the IHNCNN over the NHNCNN, the IHNCNN combined with gradual expansion scheme (GES) is applied to solve broadcast scheduling problem (BSP) in wireless multihop networks (WMNs). The aim of BSP is to design an optimal time-division multiple-access (TDMA) frame structure with minimal frame length and maximal channel utilization. Simulation results in BSP show the superiority of the IHNCNN.

Keywords*: hysteretic noisy chaotic neural network, noise tuning factor, broadcast scheduling problem, wireless multihop networks*

1. Introduction

Wireless multihop networks (WMNs) can provide easy-to-use wireless data communication services among geographically distributed nodes. In WMNs, a packet store-and-forward technique and a shared radio channel with a manner of broadcast are applied for data transformations, and the multi-hop routing technology is also used to achieve data transformations for different nodes with a large distance [1-3]. Since all nodes in WMNs share a single channel to transit for saving radio channel resources, uncontrolled transmissions often cause conflicts which easily result in damaged packets at the destination and ultimately increase network delay. In order to avoid transmission conflicts, the time division multiple access (TDMA) has been adopted. In a TDMA, time is divided into frames and each frame consists of a collection of time slots. A time slot has a unit time length required for a single packet to be transmitted to or received from each other adjacent nodes. A broadcast schedule should guarantee that each node must transmit at least once in a TDMA frame, and that any two conflict nodes must be scheduled to transmit at different time slots. The goal of broadcast scheduling problem (BSP) is to find an optimal TDMA frame structure that fulfills the following two objectives: the first is to find a minimal TDMA frame length able to schedule transmissions of all nodes without any conflict, and the second is to maximize the conflicts-free node transmissions, i.e., maximize channel utilization.

The BSP has been proven to be NP-complete [4, 5], and artificial neural networks (ANNs) have been proven to be an effective tool to solve the BSP. Funabiki et al. [1] first used Hopfield neural network (HNN) to solve the BSP. After that, many kinds of ANNs have been used to solve BSP and have obtained better results. In the following, different ANNs which have been used to solve BSP are introduced detailedly.

After Funabiki first used HNN to solve BSP, Salcedo-Sanz et al. [6] also used a HNN-based hybrid method for different attempts. However, the HNN is easy to be trapped in local minima and even converges to infeasible solutions of BSP for its gradient descent mechanism.

Received June 17, 2012; Revised September 10, 2012; Accepted October 19, 2012

The optimization ability of HNN can be improved effectively by introducing simulated annealing into HNN. For example, Chen et al. [7] proposed a transiently chaotic neural network (TCNN) with chaotic simulated annealing (CSA) by introducing transient chaotic dynamics into the HNN. Since chaotic attracting set has a fractal structure and covers only a very small fraction of the entire state space, CSA is more efficient in searching for better solutions for optimization problems compared to other global search algorithms. Although CSA searches in an efficient manner, CSA has completely deterministic dynamics and is not guaranteed to settle down at a global optimum no matter how slowly the annealing parameter is reduced. Fortunately, stochastic simulated annealing (SSA) tends to find a global optimum if the annealing process is carried out sufficiently slowly. To combine the advantages of CSA and SSA, Wang et al. [8] added noises with exponentially decaying noise amplitudes into the TCNN and proposed noisy chaotic neural network (NCNN) with stochastic chaotic simulated annealing (SCSA). The NCNN has better global optimization performance than the TCNN, because it combines stochastic wandering of noise and efficient ergodic searching of chaos. However, the sigmoid activation function adopted in the NCNN may cause neurons prematurely saturate easily, which is not beneficial to overcome local minimum and weakens the optimization ability of neural networks [9]. The hysteretic dynamics can cause the neuron to jump discontinuously which can effectively prevent the neuron prematurely saturating easily and overcome the weakness of the sigmoid activation function. However, all the hysteretic activation functions reported up to now have deficiencies. For instances, the hysteretic chaotic neural network (HCNN) in [10] proposed by Liu adopted two sigmoid functions deviated from each other to construct a hysteretic activation function. This kind of hysteretic activation function is realized via introducing additional sigmoid center parameters, which, no doubt, increases the number of tuning parameters. Reference [2] suggested a kind of hysteretic noisy chaotic neural network (HNCNN), where the hysteretic activation function is constructed by first taking noises as sigmoid center parameters and then controlling the noises. By this way, the optimization performance of the NCNN can be effectively improved without increasing the number tuning parameters. Based on the HNCNN in [2], the NHNCNN is proposed in [3] by introducing a noise tuning factor in input items. Compared to the HNCNN for BSP, the NHNCNN at lower noises can improve the results of BSP significantly, while can not at higher noises.

In order to improve the optimization performance of the NHNCNN at initial higher noise levels, an improved hysteretic noisy chaotic neural network (IHNCNN) is constructed by introducing a new noise tuning factor into the activation function in this paper. By adjusting the newly added noise tuning factor, both noise levels of the activation function and hysteretic dynamics can be adjusted to help to improve the global optimization ability as the initial noise amplitude is higher. As a result, the IHNCNN can exhibit better optimization performance than the HNCNN as the level of noises is higher. The simulation results in BSP prove that the proposed IHNCNN is super to the NHNCNN as the initial noise amplitude is higher.

2. Broadcast Scheduling Problem

A WMN can be represented by a graph $G = (V, E)$, where vertices in $V=\{1, 2, \ldots, N\}$ are network nodes, N being the total number of nodes in the network, and E represents the set of transmission links. Two nodes i and j $(i, j) \in E$ are connected by an undirected edge $e_{ij} \in E$ if and only if they can receive each other's transmission. In such a case, the two nodes i and j are said to be one hop away. A symmetric binary matrix $C = \{c_{ij}\}(i, j = 1,2,\ldots, N)$ can be used to describe the connectivity of nodes in WMN. C is defined as follows [1].

$$c_{ij} = \begin{cases} 1, & \textit{if node i and } j \textit{ are one hop away}; \\ 0, & \textit{otherwise.} \end{cases} \quad (1)$$

If $e_{ij} \notin E$, but there is an intermediate node k such that $e_{ik} \in E$ and $e_{kj} \in E$, then nodes i and j are two hops away. There are two constraints in BSP. 1) two nodes that one hop away or two hops away cannot transmit in the same time slot of a TDMA frame. 2) each node must be scheduled to transmit at least once in a TDMA frame.

The compatibility binary matrix $D=\{d_{ij}\}(i,j=1,2,\ldots N)$ can be obtained from matrix C. D can be described as follows [1].

$$d_{ij}=\begin{cases}1, & if\ nodes\ \ i\ and\ \ j\ are\ within\ \ two-hop\ away\ for\ i\neq j;\\ 0, & otherwise.\end{cases} \tag{2}$$

A $N\times M$ binary matrix $T=\{t_{ij}\}$ can be used to describe the TDMA frame, where M is the total number of time slots in the TDMA frame.

$$t_{ij}=\begin{cases}1, & if\ \ time\ slot\ \ j\ \ is\ assigned\ to\ node\ i;\\ 0, & otherwise.\end{cases} \tag{3}$$

The total channel utilization ρ for the entire network and the average time delay η for each node to broadcast packets, which can be used to evaluate different algorithms, are described as follows [1].

$$\rho=\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}t_{ij} \tag{4}$$

$$\eta=\frac{M}{N}\sum_{i=1}^{N}\left(\frac{1}{\sum_{j=1}^{M}t_{ij}}\right) \tag{5}$$

The goal of the BSP is to find an optimal or suboptimal TDMA frame with a minimal frame length $M_{\min}$ and maximal channel utilization $\rho_{\max}$, satisfying the following two constraints [11, 12].

$$\sum_{j=1}^{M}t_{ij}>0\ \ (i=1,2,\cdots,N) \tag{6}$$

$$\sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{k=1,k\neq i}^{N}d_{ik}t_{ij}t_{kj}=0 \tag{7}$$

The constraint (6) means that each node must transmit at least once in a TDMA frame, while the constraint (7) means that any two nodes with one hop or two hops away cannot be scheduled in the same time slot.

3. Improved Hysteretic Noisy Chaotic Neural Network

References [2] and [12] have pointed out that higher noises can destroy the reverse bifurcation of chaos, which can cause the optimization performance of HNCNN to be weakened and is not beneficial for solving the BSP. In the NHNCNN, CSA and SSA can be balanced to improve the optimization performance at different noise levels by adjusting the value of the noise tuning factors. The simulation results in [3] suggest that tuning of δ can significantly enhance the optimization performance of the NHNCNN at lower noise levels. However, the improvement is not so significant as the initial noise amplitude is higer. In fact, the motion equation of HNCNN is composed by four equations, i.e. inputs motion equations, outputs motion equations, CSA and SSA. And the NHNCNN in [3] only controlled the noises in inputs equations, while cannot control the noise in output equations effectively. In order to overcome the mentioned problem above, we introduce a new noise tuning factor g into the output equations and propose the IHNCNN, described as follows.

$$x_{ij}(t) = \frac{1}{1+\exp\{-[y_{ij}(t) + g\eta_{ij}(t)]/\varepsilon\}} \tag{8}$$

$$y_{ij}(t+1) = ky_{ij}(t) + \delta\eta_{ij}(t) + \alpha\left[\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{\substack{l=1\\l\neq j}}^{M} w_{ij,kl}x_{kl}(t) + I_{ij}\right] - z(t)[x_{ij}(t) - I_0] \tag{9}$$

$$z(t+1) = (1-\beta_1)z(t) \tag{10}$$

$$A[n(t+1)] = (1-\beta_2)A[n(t)] \tag{11}$$

$$\eta_{ij}(t) = \begin{cases} 0, & t = 0 \\ +|n(t-1)|, & t > 0,\ y_{ij}(t) < y_{ij}(t-1) \\ -|n(t-1)|, & t > 0,\ y_{ij}(t) \geq y_{ij}(t-1) \end{cases} \tag{12}$$

where g is the newly added noises tuning factor in the activation function (8); $x_{ij}(t)$ is the output of neuron ij; $y_{ij}(t)$ is the input of the neuron ij; $k(0<k<1)$ is a damping factor of nerve membrane; δ is the noise tuning factor of input; α is positive scaling parameter for inputs; $w_{ij,kl}$ is the connection weight from neuron kl to neuron ij, with $w_{ij,kl} = w_{kl,ij}$ and $w_{ij,ij} = 0$; I_{ij} is an input bias of neuron ij; $z(t)$ is self-feedback connection weight; I_0 is a positive parameter; $n(t)$ is stochastic noise in $[-A[n(t)], A[n(t)]]$ with a uniform distribution, and $A[n(t)]$ is the noise amplitude; $\beta_1(0<\beta_1<1)$ is the simulated annealing of $z(t)$; $\beta_2(0<\beta_2<1)$ is the simulated annealing of $n(t)$.

Different from the NHNCNN in [3], there is a new noise tuning factor g in the activation function (8) of the IHNCNN. Noise levels of the activation function in the IHNCNN can be effectively adjusted by adjusting the value of g. Obviously, the IHNCNN will be same with the NHNCNN in [3] if $g=1$. Since the optimization performance of the NHNCNN in [3] is significantly improved at lower noise amplitudes while not at higher noise amplitudes, the IHNCNN is studied only at higher noise levels.

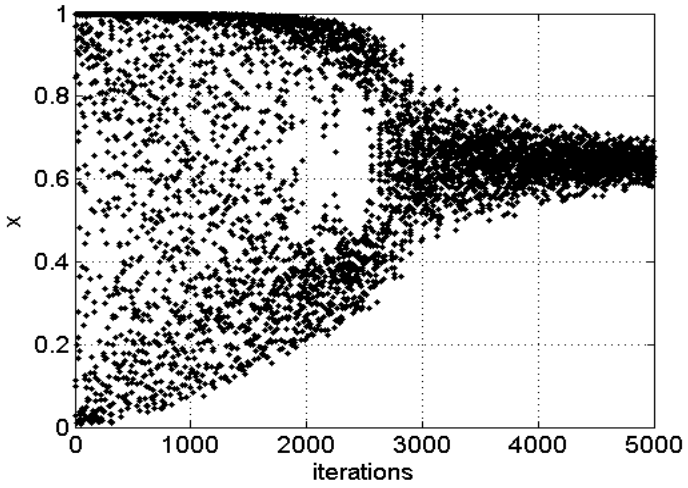

Figure 1. Dynamics of the NHNCNN neuron with $A[n(0)] = 0.01$, $\delta = 0.7$, $g = 1$.

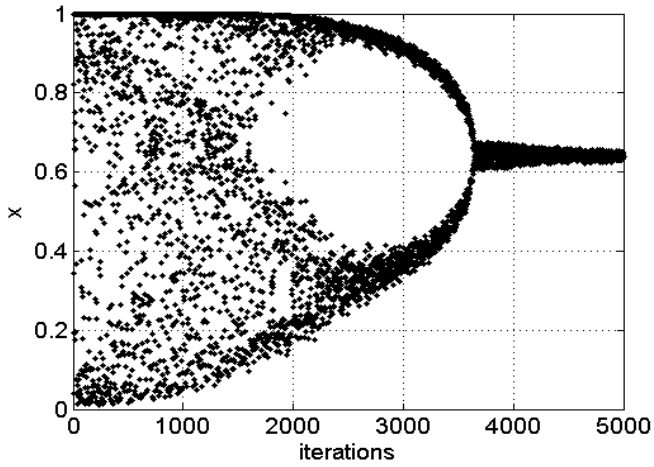

Figure 2. Dynamics of the IHNCNN neuron with $A[n(0)] = 0.01$, $\delta = 0.7$, $g = 0.4$.

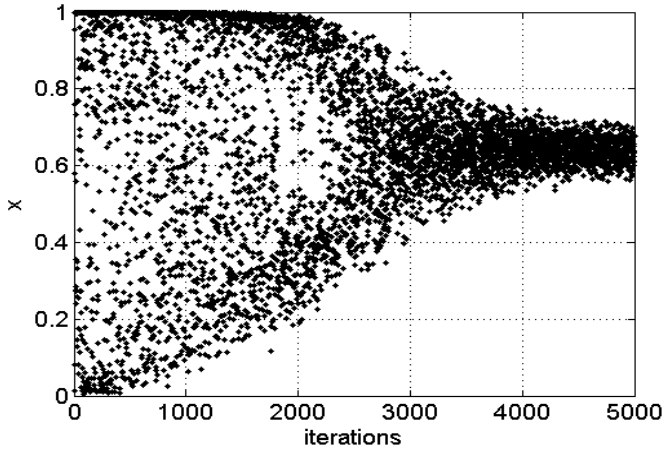

Figure 3. Dynamics of the NHNCNN neuron with $A[n(0)] = 0.01$, $\delta = 0.4$, $g = 1$.

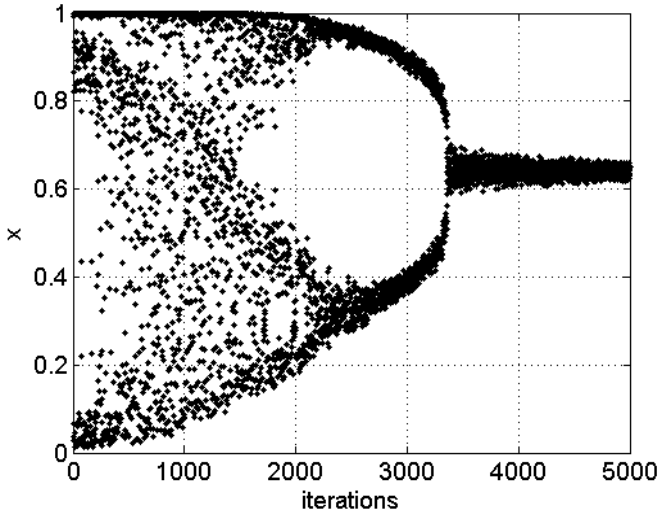

Figure 4. Dynamics of the IHNCNN neuron with $A[n(0)] = 0.01$, $\delta = 0.4$, $g = 0.4$.

In the following, dynamics of the IHNCNN neuron are used to prove the advantage of the IHNCNN over the NHNCNN. For convenient to compare neuron dynamics of the IHNCNN with the NHNCNN, the parameters are set same with reference [3], i.e. $k = 0.9$, $\varepsilon = 0.004$, $z(0) = 0.1$, $I_0 = 0.65$, $\beta_1 = 0.0003$, $\beta_2 = 0.0005$. In order to observe effects of g and δ on dynamics as the initial noise amplitude is higher, the initial noise amplitude $A[n(0)]$ is fixed as 0.01, while g is changed from 1 to 0.4 and δ is changed form 0.7 to 1, respectively. Dynamics of the IHNCNN are shown in Figure 1, Figure 2, Figure 3, and Figure 4.

As seen from Figure 1 and Figure 3, when the noise amplitude is higher ($A[n(0)] = 0.01$), there are almost no effects on dynamics of the NHNCNN neuron by only adjusting the noise tuning factor δ. Compared Figure 1 with Figure 2 or compared Figure 3 with Figure 4, dynamics of the IHNCNN neuron vary obviously, and the reverse bifurcations of chaos of Figure 2 or Figure 4 are more evident than Figure 1 or Figure 3. That is to say, when δ is set as a fixed value while adjusting the newly added noise tuning factor g, the reverse bifurcations of chaos in Figure 2 and Figure 4 become evident. These suggest that when noise amplitude is higher, the NHNCNN cannot balance the hysteretic dynamics and CSA well by adjusting the noise tuning factor δ, while the IHNCNN can balance he hysteretic dynamics and CSA well by adjusting the noise tuning factor g. Namely, the IHNCNN can balance hysteretic dynamics and CSA better by adjusting the newly added noise tuning factor g than the NHNCNN by adjusting the noise tuning factor δ. From the above analyses, we can obtain the following conclusions. When the noise amplitude $A[n(0)]$ and the noise tuning factor δ are fixed, the optimization dynamics of the IHNCNN can be improved by adjusting the noise tuning factor g. That is to say, the optimization performance of the IHNCNN would be better than that of the NHNCNN.

4. Simulation Results and Analysis

We use the same method as references [2] and [3] to find the TDMA frame. The results of BSP obtained by the NHNCNN in [3] are obviously better than other neural-network algorithms as the noise amplitude is lower. In addition, the IHNCNN will be the same with the NHNCNN as $g = 1$, so the IHNCNN will also obtain better BSP results as those obtained by the NHNCNN in refence [3] when the noise amplitude is lower. Therefore, we will not compare the proposed IHNCNN with other neural-network algorithms any more, but only compare the proposed IHNCNN with the NHNCNN in reference [3] as the noise amplitude is higher.

For convenient to compare with the results obtained by the NHNCNN in [3], the parameters are set the same with [3], i.e., $k = 0.9$, $\varepsilon = 0.004$, $z(0) = 0.08$, $I_0 = 0.65$, $\beta_1 = 0.001$, $\beta_2 = 0.0001$, $\alpha = 0.015$. The 100-node-200-edge BSP [4] is solved by the IHNCNN and NHNCNN, and the initial noise amplitude $A[n(0)]$ is set as $A[n(0)] = 0.03$ and $A[n(0)] = 0.04$, respectively. Different δ and g are be set for solving the BSP, and each case is simulated for 50 times. Solutions with the minimum time slot length ($M_{\min}$) and the maximum packets transmission ($N_{\max}$) are selected as the best solutions, and the average time delay η is also selected as an important indicator to evaluate the optimization performance of IHNCNN and NHNCNN. The simulation results are shown in table 1 and table 2.

Table 1. Comparisons of best solutions obtained by the NHNCNN in [3] and the proposed IHNCNN with different δ and g as $A[n(0)] = 0.03$

$A[n(0)]$	δ	the proposed IHNCNN		the NHNCNN in [3]
		g	Best solutions $M_{\min} / N_{\max} / \eta$	Best solutions $M_{\min} / N_{\max} / \eta$
0.03	0.8	0.4	9/139/7.47	9/128/7.83
	0.7	0.4	9/140/7.42	9/130/7.78
	0.6	0.3	9/139/7.40	**9/132/7.72**
	0.5	0.3	9/140/7.42	9/128/7.87
	0.4	0.3	**9/141/7.34**	9/130/7.78

Table 2. Comparisons of best solutions obtained by the NHNCNN in [3] and the proposed IHNCNN with different δ and g as $A[n(0)] = 0.04$

$A[n(0)]$	δ	the proposed IHNCNN		the NHNCNN in [3]
		g	Best solutions $M_{\min} / N_{\max} / \eta$	Best solutions $M_{\min} / N_{\max} / \eta$
0.04	0.8	0.4	9/136/7.56	9/122/8.03
	0.7	0.4	9/138/7.51	9/125/7.97
	0.6	0.3	9/137/7.48	9/125/7.91
	0.5	0.3	9/140/7.42	**9/129/7.92**
	0.4	0.2	**9/140/7.35**	9/127/7.85

As seen from table 1 and table 2, as the noise level is higher, the optimization performance of the IHNCNN is better than the NHNCNN in [3] under the same conditions. i.e., the number of the total packets transmission achieved by the IHNCNN is more than HNCNN, and that the average time delay achieved by IHNCNN is less than HNCNN. As seen from table 1, the optimization performance of the IHNCNN is better than the NHNCNN as $A[n(0)] = 0.03$. The best solution obtained by the NHNCNN in [3] is 9/132/7.72, while the best solution obtained by the IHNCNN is 9/141/7.34. Obviously, the latter is much better than former. And the improvements in the number of transmission and the average time delay are 9 and 0.38, respectively. As seen from table 2, the best solution obtained by the NHNCNN is 9/129/7.92, while the best solution obtained by the IHNCNN is 9/140/7.35. The improvements in the number of transmission and the average time delay are 11 and 0.57, respectively. These suggest that the larger the initial noise amplitude is, the more obvious in the optimization performance advantages the IHNCNN is.

5. Conclusion

In this paper, an IHNCNN was proposed to solve the BSP in WMNs. Different from the NHNCNN in [3], the proposed IHNCNN has a new noise tuning factor g in the activation function. By adjusting the original noise tuning factor δ and the newly added noise tuning factor g, the IHNCNN not only can balance CSA and SSA but also can balance CSA and hysteretic dynamics. Therefore, the IHNCNN has better optimization performance than the NHNCNN. The simulation results show the superiority of the proposed IHNCNN.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant 61100103, the scientific research of Heilongjiang province teachs hall under Grant 12511600, the Program for Young Teachers Scientific Research in Qiqihar University under Grant 2010K-M14.

References

[1] N Funabiki, J Kitamichi. A gradual neural network algorithm for broadcast scheduling problem in packet radio networks. *IEICE Transactions on Fundamentals*. 1999; E82-A(5): 815-824.
[2] M Sun, L Zhao, W Cao, Y Xu, X Dai, X Wang. Novel hysteretic noisy chaotic neural network for broadcast scheduling problems in packet radio networks. *IEEE Transactions on Neural Networks*. 2010; 21(9): 1422-1433.
[3] M Sun, Y Xu, X Dai, Y Guo. Noise-tuning-based hysteretic noisy chaotic neural network for broadcast scheduling problem in wireless multihop networks. *IEEE Transactions on Neural networks and Learning Systems*. 2012; 23(12): 1905-1918.
[4] G Wang, N Ansari. Optimal broadcast scheduling in packet radio networks using mean field annealing. *IEEE Journal on Selected Areas in Communications*. 1997; 15(2): 250-260.
[5] A Ephremides, T Truong. Scheduling broadcast in multihop radio networks. *IEEE Transactions on Communications*. 1990; 38(6): 456-460.
[6] S Salcedo-Sanz, C Bousono-Calzon, A R Figueiras-Vidal. A mixed neural-genetic algorithm for the broadcast scheduling problem. *IEEE Transactions on Wireless Communications*. 2003; 2(2): 277-283.

[7] L Chen, K Aihara. Chaotic simulated annealing by a neural network model with transient chaos. *Neural Network*. 1995; 8(6): 915-930.
[8] L Wang, S Li, F Tian, X Fu. A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing. *IEEE Transactions on System, Man and Cybernetics, Part B: Cybernetics*. 2004; 34(5): 2119-2125.
[9] S Bharitkar, JM Mendel. The hysteretic hopfield neural network. *IEEE Transactions on Neural Networks*. 2000; 11(4): 879-888.
[10] X Liu, C Xiu. A novel hysteretic chaotic neural network and its applications. *Neurocomputing*. 2007; 70(13-15): 2561-2565.
[11] S Salcedo-Sanz, C Calzon. A mixed neural-genetic algorithm for the broadcast scheduling problem. *IEEE Transactions on Wireless Communications*. 2003; 2: 277-283.
[12] L Wang, H Shi. A gradual noisy chaotic neural network for solving the broadcast scheduling problem in packet radio networks. *IEEE Transactions on Neural Networks*. 2006; 17(4): 989-1000.