Proc. Natl. Sci. Counc. ROC(A)
Vol. 24, No. 6, 2000. pp. 463-471

The Shortest-Path Computation in MOSPF Protocol through an Annealed Chaotic Neural Network

JZAU-SHENG LIN[*], MINGSHOU LIU[*], AND NEN-FU HUANG[**]

[*]*Department of Electronic Engineering*
National Chinyi Institute of Technology
Taichung, Taiwan, R.O.C.
[**]*Department of Computer Science*
National Tsing-Hua University
Hsinchu, Taiwan, R.O.C.

(Received November 30, 1999; Accepted April 21, 2000)

ABSTRACT

Multimedia communications have become popular in many network services, such as video conferencing, video on demand, and so on. Most multimedia applications require that the attached hosts/routers transmit data through multicasting. In order to provide efficient data routing, routers must provide the multicast capability. In this paper, a self-feedback mechanism controlled by an annealing strategy and embedded into the Hopfield neural network is proposed to calculate the shortest-path tree for the Multicast Open-Shortest Path First (MOSPF) Protocol. A multicast shortest path tree is built on demand and is rooted in the source node. To facilitate hardware implementation, the annealed chaotic neural network can be employed to deal with shortest-path (SP) problems in packet switching computer networks. In addition, the annealed chaotic neural network can avoid the local-minima solution so as to obtain near-global minima or global-minimum solutions.

Key Words: MOSPF protocol, annealed chaotic neural network, Hopfield neural network, shortest-path problem

I. Introduction

A routing algorithm is employed to find an appropriate path so that traffic can be relayed in optimal ways. One important issue in routing algorithms is the speed with which they can react to topology changes. Traditionally, TCP/IP uses the Routing Information Protocol (RIP) (Hedrick, 1988), in which routing decisions are made based on the number of hops between the source and destination. Many RIP systems have been replaced by a more powerful routing method called the Open Shortest Path First (OSPF) (Moy, 1994a, 1994b) protocol. RFC1583 is a specification of the OSPF TCP/IP internet routing protocol. OSPF is classified as an Interior Gateway Protocol (IGP). This means that it distributes routing information between routers belonging to a single Autonomous System (AS). OSPF belongs to a class of *link state* protocols in which routing information is flooded through the entire network. It provides a dynamic and adaptive routing mechanism for making topology changes. The performance of most link-state protocols depends heavily on the optimal path calculation. One classical work called the *shortest path first* algorithm proposed by Dijkstra (1959) is employed in many systems for the purpose of shortest path selection. Dijkstra's algorithm works well but is computationally intensive, especially in the case of multicast routing.

Multimedia communications have become important in many network services, such as video conferencing, video on demand, and so on. In order to provide efficient data forwarding, a routing protocol must provide the multicast capability. An enhanced version of the OSPF, called the Multicast Open Shortest Path First (MOSPF) protocol (Moy, 1994b), was proposed to deal with IP multicast routing. RFC1584 is the protocol used for multicast extensions to OSPF. MOSPF is a source/destination routing in which each router must construct a tree rooted at the source node. This is different from unicast OSPF, where the root is the computing router itself.

An Artificial Neural Network (ANN) is a system consisting of a number of simple processors running in parallel. The Hopfield neural network, a well-known network, was proposed to solve optimization problems (Hopfield, 1982; Hopfield and Tank, 1985, 1986; Cheng *et al*., 1996; Lin, 1999a, 1999b; Lin *et al*., 1996a, 1996b), and many researchers have subsequently applied it to SP (shortest path) problems in computer networks. Leung (1994) demonstrated neural scheduling algorithms for time-multiplex switches. Brown (1989) also presented neural networks for switching problems. A new neural-network model, Routron, was proposed by Lee and Chang (1993) for routing of communication networks with unreliable components. The continuous Hopfield neural network was applied by Ali and Kamoun (1993) to the optimal routing problem in packet-switched computer networks to minimize the network-wide average time delay. In addition, Pornavalai *et al*. (1996) proposed a Hopfield neural network for optimal steiner tree computation.

The chaotic neural network is a well-known technique which employs a chaotic-dynamics mechanism to solve optimization problems. It has a rich range and flexible dynamics, which give it a better ability to search for globally optimal solutions. The chaotic dynamics in chaotic neural networks have been discussed recently. Song *et al*. (1997) embedded chaotic dynamics in a self-feedback manner into the Hopfield neural network to avoid local-minima solutions for the travelling salesman problem. Adaptive predication of nonstationary signals using chaotic neural networks was proposed by Choi *et al*. (1998). Aihara *et al*. (1990) demonstrated the use of a chaotic neural network to search for the global minimum effectively using the chaotic searching mechanism without stacking in undesirable local minima. Due to random bifurcation in relation to chaotic dynamics, chaotic neural networks do not always stay in the global solution, and the problem of convergence is not satisfactorily solved. In order to control the bifurcation behavior in chaotic neurons, the simulated annealing strategy is incorporated into the chaotic neural network.

In this paper, an annealed chaotic neural network with a cooling schedule is proposed for computation of the shortest path in the MOSPF protocol. The general goal is to formulate the computer routing algorithm as an SP problem and apply the annealed chaotic neural network to solve it. The specific objective is to simplify the constrained-cost function so as to avoid the need to determine system-dependent parameters and to improve performance. Furthermore, we want to obtain a solution that is very close to the global minimum in a very short time.

This paper is organized as follows. Section II demonstrates the use of neural networks to solve SP problem. In Section III, chaotic neural networks are described. The chaotic neural network with an incorporated annealing strategy to harness the chaotic dynamics is discussed in Section IV. Section V discusses the MOSPF shortest path tree and inter-area routing for the neural-based MOSPF. Section VI presents several experimental results to show that the annealed chaotic network is suitable for computation of the shortest-path tree in the OSPF routing domain. Finally, Section VII presents discussion and conclusions.

II. Shortest Path Computation with the Neural Networks Model

In a computer network, routing is an important factor determining transmission performance. Network routing has been a subject of intensive research for many years. This paper focuses on the MOSPF routing problem in order to minimize the average time delay in data transmission. During the past decade, the Hopfield neural network has been studied extensively because of its simple architecture and potential for parallel implementation. The Hopfield neural network is a well-known technique used to solve optimization problems based on the Lyapunov energy function. Application of the competitive Hopfield neural network to medical image segmentation was described by Cheng *et al*. (1996). Polygonal approximation using a competitive Hopfield neural network was demonstrated by Chung *et al*. (1994). In the competitive Hopfield neural network, a 2-dimensional discrete Hopfield network was employed with winner-take-all learning to eliminate the need to find weighting factors in the energy function. In the following, the Hopfield network will be reviewed. Let v_i be the state at neuron i; then the Hopfield neural network can be modelled as

$$v_i(t+1) = \begin{cases} 1 & net_i > U_i \\ v_i(t) & net_i = U_i, \\ 0 & net_i < U_i \end{cases} \tag{1}$$

where U_i represents the threshold for the ith node, and net_i is defined by

$$net_i = \sum_{j=1}^{n} w_{ij} v_j + I_i. \tag{2}$$

By employing the Lyapunov theorem, the energy function can be defined as

$$E = -\frac{1}{2}\sum_i \sum_{\substack{j \\ j \neq i}} v_i w_{ij} v_j - \sum_i I_i v_i + \sum U_i v_i. \tag{3}$$

It can be shown that the energy function is gradient decent and can reach a stable state as the system evolves.

1. Problem Mapping

In the MOSPF routing domain, the topology of a single area in the Autonomous System (AS) can be defined as a directed graph $G(N,A)$ with n nodes and l arcs, where each node in the graph represents a router or transit network. Corresponding to each $arc(x,i)$, there is a nonnegative weight C_{xi} representing the cost from node x to node i. To compute the shortest path from one source node to a group, The Hopfield model with an $n \times n$ array is proposed, where each component in the array represents a router/transit network. Each neuron in the array is identified by double indices (x,i), where x and i indicate the row and column number, respectively. The neuron at location (x,i) shows the link from node x to i in the graph of a router/transit network. Except for neurons at the diagonal, only $n(n-1)$ neurons are used in calculation in each array.

In order to characterize the neuron activities at location (x,i), we define the neuron state v_{xi} and C_{xi} as

$$v_{xi} = \begin{cases} 1 & \text{if the arc from node } x \text{ to node } i \text{ does exist} \\ 0 & \text{otherwise,} \end{cases}$$

C_{xi} = the finite nonnegative cost weight on the link from node x to node i.

2. Definition of the Objective Function

To compute the shortest path using the Hopfield model, the object function, similar to Ali and Kamoun's definition (Ali and Kamoun, 1993), can be defined as

$$E_{obj} = \frac{A}{2}\sum_{x=1}^{n}\sum_{\substack{i=1 \\ i\neq x \\ (x,i)\neq(d,s)}}^{n} c_{xi}v_{xi} + \frac{B}{2}\sum_{x=1}^{n}\left\{\sum_{\substack{i=1\\ i\neq x}}^{n} v_{xi} - \sum_{\substack{i=1\\ i\neq x}}^{n} v_{ix}\right\}^2$$

$$+\frac{C}{2}\sum_{i=1}^{n}\sum_{\substack{x=1\\ x\neq i}}^{n} v_{xi}(1-v_{xi}) + \frac{D}{2}(1-v_{ds}),$$

where the parameter A represents the minimization of the total cost. The B term is zero if the number of incoming arcs equals the number of outgoing arcs. The C term is zero if every output converges to $\{0,1\}$.

To compute the optimal weight, we replace Eqs. (1) and (3) with double indices and let $U_i = 0$ for all i; then

$$v_{xi}(t+1) = \begin{cases} 1 & net_{xi} > 0 \\ v_{xi}(t) & net_{xi} = 0 \\ 0 & net_{xi} < 0 \end{cases}$$

and

$$E = -\frac{1}{2}\sum_{x=1}^{n}\sum_{\substack{i=1\\ i\neq x}}^{n}\sum_{y=1}^{n}\sum_{\substack{j=1\\ j\neq y}}^{n} v_{xi}w_{xi;yj}v_{yj} - \sum_{x=1}^{n}\sum_{\substack{i=1\\ i\neq x}}^{n} I_{xi}v_{xi}. \tag{4}$$

3. Interconnection Weights and Input Bias

In order to map the objective function into the Lyapunov energy function, the derivatives of these two functions with respect to the state change of neuron (x,i) are equal:

$$\frac{\partial E}{\partial v_{xi}} = \frac{\partial E_{obj}}{\partial v_{xi}} \tag{5}$$

such that E and E_{obj} decrease at the same rate as v_{xi} changes. By comparing each component in Eq. (5), we can obtain the interconnection weights and input bias as follows:

$$w_{xi;yj} = C\delta_{xy}\delta_{ij} - B\delta_{xy} - B\delta_{ij} + B\delta_{jx} + B\delta_{ij}$$

and

$$I_{xi} = -\frac{A}{2}c_{xi}(1-\delta_{xd}\delta_{is}) - \frac{C}{2} + \frac{D}{2}\delta_{xd}\delta_{is}\delta_{ij},$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise.} \end{cases}$$

Floreen and Orponen (1993) indicated that determining the attraction radius of a stable vector in a discrete (binary) Hopfield network is an NP-hard problem. This might hamper convergence of the discrete Hopfield net in training complex and large data sets, and the Hopfield neural network may become trapped in a local minimum. In this application using a discrete Hopfield network, a neuron (x,i) in a firing state indicates that an arc from node x to node i does exist. In the annealed chaotic neural network, however, a neuron (x,i) in a probable state indicates that an arc from node x to node i exists with a degree of uncertainty described by a probability function. The annealed chaotic network, which is a continuous model with a probability function, can overcome the NP-hard problem exhibited in binary Hopfield nets.

III. Chaotic Neural Networks (CNN)

Chaos is a revolutionary technique employed in engineering applications. Unlike the conventional neural network, the CNN has a rich range and flexible dynamics, so it can be expected to have a better ability to search for globally optimal or near-optimum results. In addition to having the characteristics of conventional neural units, chaotic neural networks exhibit an extensive range of behavior reminiscent of that observed in neurons. The chaotic dynamic in chaotic neural networks has been discussed in previous reports (Aihara *et al*., 1990; Choi *et al*., 1998; Song *et al*., 1997) due to its potential biological functional role.

The 2-D chaotic neural network based on the Hopfield net, embedded into which are Feigenbaum's bifurcation formula and a self-feedback connection weight, has the capability of parallel synchronous computation in bifurcation states. The model of the chaotic neural network can be given as follows:

$$u_{xi}(k) = \frac{1}{1+e^{-v_{xi}(k)/\lambda}}, \tag{6}$$

$$v_{xi}(k+1) = \delta \sin[\pi v_{xi}(k)] + [\textstyle\sum_{yj} w_{xi;yj} u_{yj}(k) + I_{xi}] - T_{xi}(k) u_{xi}(k), \tag{7}$$

where

u_{xi} = the output of neuron (x,i),

v_{xi} = the internal state of neuron (x,i),

$w_{xi;yj}$ = the connection weight from neuron (y,j) to neuron (x,i),

I_{xi} = the input bias of neuron (x,i),

I_0 = a positive parameter,

δ = the damping factor of the nerve membrane ($0 \le \delta \le 1$),

$T_{xi}(k)$ = the self-feedback connection weight,

λ = the steepness parameter of the output function of neuron (x,i) ($\lambda > 0$),

π = the ratio of the circumference of a circle to its diameter.

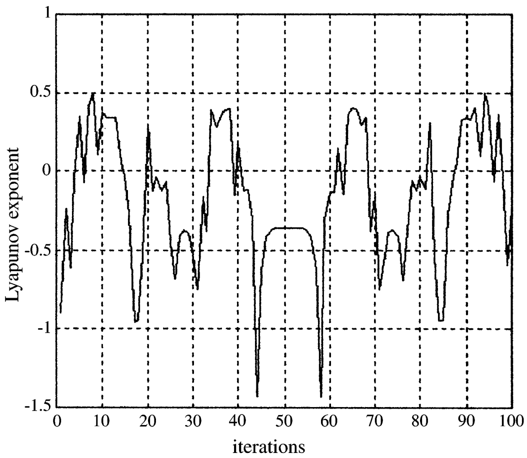


Fig. 1. Lyapunov exponent during 100 iterations.

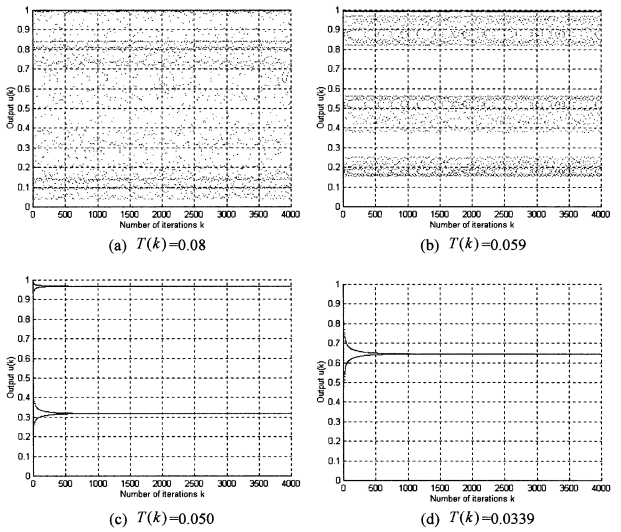


Fig. 2. The bifurcation states of a neuron with various self-feedback connection weights $T(k)$ during 4000 iterations: (a) $T(k)$ = 0.08; (b) $T(k)$ = 0.059; (c) $T(k)$ = 0.050; and (d) $T(k)$ = 0.0339.

Furthermore, the Lyapunov exponent is defined as

$$L = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n} \ln\left|\frac{dy(k+1)}{dy(k)}\right|. \tag{8}$$

The chaotic dynamics in a single neuron exist in accordance with the Lyapunov exponent. Chaotic behavior is displayed if the Lyapunov exponent is positive. Figure 1 shows the Lyapunov exponents in a chaotic neuron during 100 iterations. It shows that chaotic characteristics appear in some intervals.

The self-feedback connection weight $T_{xi}(k)$ is a fixed value in the system. Figure 2 shows the bifurcation characteristics with several fixed values of $T_{xi}(k)$. Chaotic neural networks do not always stay in the global solution because the chaotic dynamical mechanism is not clear. Figure 2 shows that it is difficult to decide how to harness the chaotic behavior in a chaotic neuron for convergence to a stable equilibrium point corresponding to an acceptably near-optimum result. Because the bifurcation behavior in the chaotic neuron is difficult to control, the simulated annealing strategy is introduced into the chaotic neural network in order to control the chaotic dynamical mechanism as described in next section.

IV. Annealed Chaotic Neural Network

An unsupervised parallel approach, called the annealed chaotic neural network, to calculate the shortest-path tree for the MOSPF Protocol is proposed in this section. The goal is to modify an unsupervised scheme based on the Hopfield neural network using the chaotic technique governed by an annealing strategy in order to control the chaotic dynamic and convergence process and make parallel implementation for finding a near-global solution feasible. In addition to retaining the characteristics of conventional neural units, the annealed chaotic network displays a rich range of behavior reminiscent of that observed in chaotic neurons. Unlike the conventional Hopfield neural network, the annealed chaotic neural network has a rich range of flexible dynamics, so it can be expected to have better ability in searching for globally optimal or near-optimum solutions.

In order to control the chaotic dynamics in the annealing process, a feasible cooling schedule is required. Reaching a thermal equilibrium at a low temperature might take a very long time. The search for adequate cooling schedules has been a subject of active research for several years. In this paper, a decrement function for a cooling schedule that can make the objective function converge a global minimum rapidly under all fixed temperatures is proposed as follows:

$$T_k = \frac{1}{\beta+1}[\beta + \tanh(\alpha)^k]T_{k-1}, \quad k = 1,\ 2,\ \ldots, \qquad \textbf{(9)}$$

where α ($0.8 \le \alpha \le 0.99$) is a constant smaller but close to unit and β is another constant. The simulated annealing technique has non-zero probability during transition from one state to another and moves temporarily toward a worse state so as to escape from local traps. The probability function depends on the temperature and the energy difference between the two states. With the probabilistic hill-climbing search approach, the simulated annealing technique has a better probability of going to a higher energy state at a higher temperature.

In the annealed chaotic network, the self-feedback connection weight decreases with the decrement function as shown in Eq. (9). The structure of the annealed chaotic network is defined as follows:

$$u_{xi}(k) = \frac{1}{1+e^{-v_{xi}(k)/\lambda}} \qquad \textbf{(10)}$$

$$v_{xi}(k+1) = \delta \sin[\pi v_{xi}(k)] + [\textstyle\sum_{yj} w_{xi;yj} u_{yj}(k) + I_{xi}] - T_{xi}(k)[u_{xi}(k) - I_0] \qquad \textbf{(11)}$$

and

$$T_{xi}(k) = \frac{1}{\beta+1}[\beta + \tanh(\alpha)^k]T_{xi}(k-1), \quad k = 1,\ 2,\ \ldots, \qquad \textbf{(12)}$$

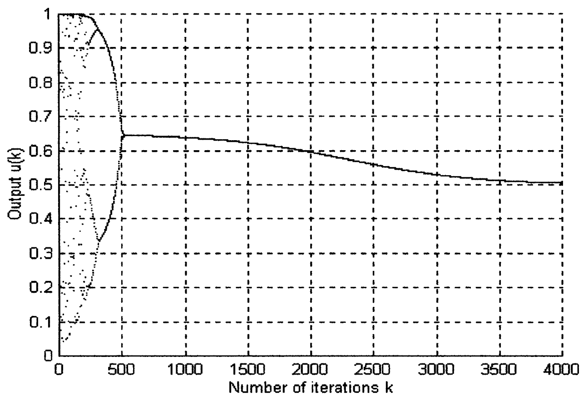
where I_0 = a positive constant and all the other parameters have the same definitions as in the chaotic neural network described in the previous section. Therefore, a single neural model in the annealed chaotic network can be expressed as

$$u(k) = \frac{1}{1+e^{-v(k)/\lambda}}, \qquad \textbf{(13)}$$

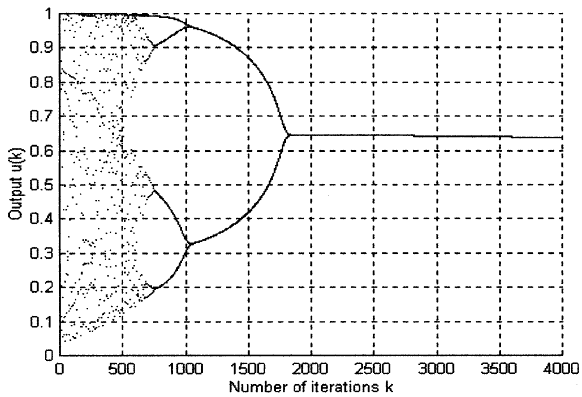
$$v(k+1) = \delta \sin[\pi v(k)] + net - T(k)[u(k) - I_0], \qquad \textbf{(14)}$$

$$T_k = \frac{1}{\beta+1}[\beta + \tanh(\alpha)^k]T_{k-1}, \quad k = 1,\ 2,\ \ldots, \qquad \textbf{(15)}$$

where $net = \Sigma_{yj} w_{xi;yj} u_{yj}(k) + I_{xi}$. To show the chaotic dynamics of a single interconnection strength model in the annealed chaotic network, the values of the parameters are fixed as $\delta = 0.3$, $\lambda = 1/250$, $I_0 = 0.65$ and $T(0) = 0.08$, and only α, β, and *net* vary, respectively. The dynamics regimes are illustrated by means of bifurcation diagrams for the transient state with respect to the self-feedback connection weight in a single interconnection strength. Figure 3(a) and (b) show the time evolutions of the transient



(a) ($\alpha = 0.98, \beta = 500$ and $net = 0$)



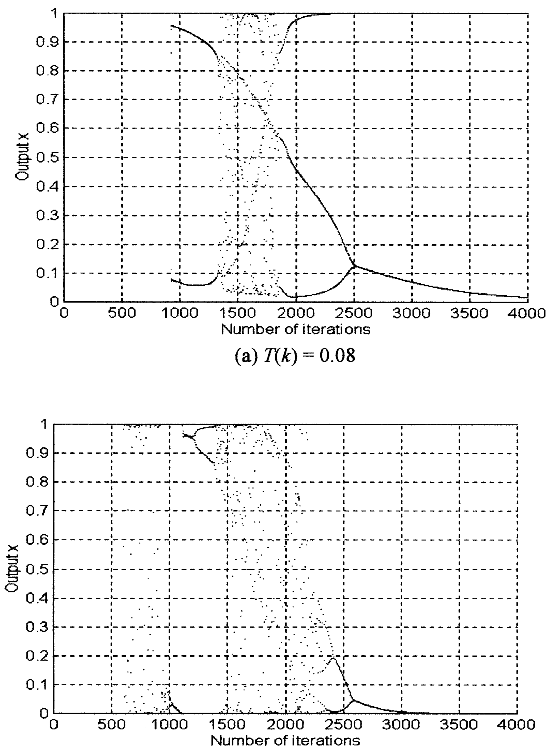
(b) ($\alpha = 0.9998, \beta = 500$ and $net = 0$)

Fig. 3. Time evolutions of a single neuron model with various values of α in the annealed chaotic network : (a) $\alpha = 0.98$, $\beta = 500$ and *net* = 0; (b) $\alpha = 0.9998$, $\beta = 500$ and *net* = 0.

state for a single interconnection strength during 4000 iterations and the decreasing self-feedback connection weight when α = 0.98, β = 500 and *net* = 0, and when α = 0.9998, β = 500 and *net* = 0, respectively. Fixed points, periodic orbits and complex oscillation can be detected in $T(k)$ during the decreasing process in these figures. The chaotic dynamics also disappear quickly because the self-feedback connection weight decreases rapidly with a small value of α as shown in Fig. 3(a). On the other hand, as shown in Fig. 3(b), the chaotic dynamics appear for a longer period of time owing to the larger value of α. Therefore, the constant α can govern the bifurcation speed of the chaotic neuron. In order to improve the computation performance in the 2-D neuron array using the annealed chaotic network, the probability grade of the neural state can be modified from that in Eq. (10) and normalized as in the following equation:

$$u_{x,i}(k) = \frac{e^{-v_{x,i}(k)/\lambda}}{\sum_{j=1}^{c} e^{-v_{x,j}(k)/\lambda}}. \quad \textbf{(16)}$$

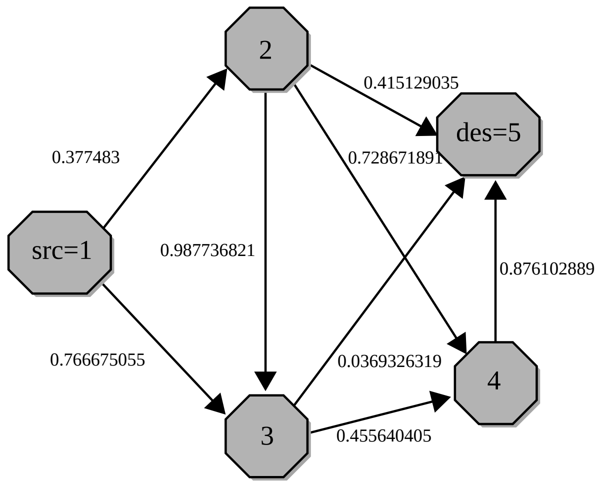
From another viewpoint, the chaotic dynamics may be influenced by different parameters. In simulation results, the neuron states display the bifurcation phenomenon when $0.01 \le T(k) \le 1.6$ and $0 \le \delta \le 1$. The chaotic characteristics are shown in Fig. 4 during 4000 iterations in which δ was changed with $\delta(k+1) = 0.995\delta(k)$ and $\delta(0) = 0.6$. Instead of controlling $T(k)$, the bifurcation processes exhibit various behavior with different fixed values of $T(k)$ when a cooling schedule is used to decrease δ. Figure 4 shows unstable chaotic dynamics. Therefore, controlling the chaotic dynamics using δ is not suitable for the chaotic neural network.



(a) $T(k)$ = 0.08

(b) $T(k)$ = 0.2

Fig. 4. Chaotic dynamics of a single neuron with different fixed values of $T(k)$ within $0.08 \le \delta \le 0.6$.



Optimal solution = 1-2-5

Fig. 5. Test example for the SP problem.

To demonstrate the potential of the annealed chaotic network, we will use an example from Ali and Kamoun (1993), shown in Fig. 5. Since this example is not suitable for MOSPF routing, it is employed here for comparison purposes. First, randomly initialized neurons' states with probability values from 0 to 1 at time t = 0 were used, and the source and destination nodes were fixed at 1 and 5, respectively. Then, the minimum values for the changed states Δ and weighting factors were set to be Δ = 0.00001, A = 350, B = 25, C = 450 and D = 5000.

The training process was terminated if the minimum values for the changed states of the adjustment iterations lower than Δ. The optimal solution, path 1:1-2-5, could be obtained after 7358 iterations by Ali and Kamoun (1993). However, the optimal result could be obtained after just 12 iterations in the annealed chaotic neural network using the proposed cooling schedule.

In this article, an iteration means that all the neurons' states are changed one time. This network can converge rapidly to valid results that are near-globally optimal.

V. MOSPF Shortest Path Tree

In multicast forwarding, packets are usually sent to multiple destinations. The RFC 1584 algorithm is used to construct a shortest path tree which spans all the nodes, and then those nodes which do not belong to the destination group are pruned out. However, neural networks find the optimal path for specific source and destination nodes; thus, construction of a shortest path tree must start in the reverse direction. Different shortest paths for specific source/destination pairs must be combined into a single tree. To construct a shortest path tree, each node in the tree must have n-1 downstream interfaces, each indexed as node.*down*(i), $1 \le i < n$. The number n represents the total number of interfaces if the node is a router, or the total number of attached routers if the node is a multi-access network. The multicast shortest path tree begins with a root node, which usually is the source network. Initially, none of the routers/transit networks are marked. After the shortest path is found for a given destination by neural networks, which we call *path*[i] for i from 0 to N–1, and *path*[0] for a root node, the tree is updated as shown in Fig. 6. After all the shortest paths for different destination have been computed, a multicast shortest-path tree is constructed.

In many situations, the source and destinations are not in the same area; thus, the OSPF must perform inter-area routing. Generally, construction of a shortest path tree for inter-area forwarding is the same as for intra-area routing above. Each router in an area computes the shortest path tree for that area and constructs its forwarding cache. On the other hand, area border routers compute the shortest path tree for each attached area and combine them into a forwarding cache. However, there are two exceptions in intra-area routing. First, when a router receives a datagram with its source address located in the same area as this router, we need to compute the shortest path to all the wild-card multicast receivers since some of the group members are outside the working area and the destinations are, thus, unknown. In the second case, the source node is in a different area. As specified by Moy (1994b), we use the summary link to create the routing topology, and all the cost values used for neural network computations proceed in the reverse direction; that is, all the cost values proceed from the destinations to the source instead of from the source node.

```
Current_node=root
For each source/destination pair
    For each node in path[i]
If the router/transit network specified by path[i] is not marked
    Mark it (Add this node to the tree)
    Current node_down[j]=path[i] for appropriate j
Set current_node=path[i]
```

Fig. 6. The algorithm for an updated tree.

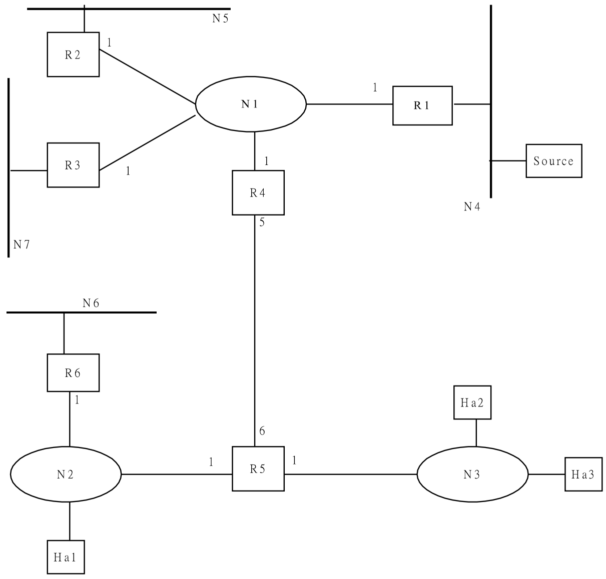


Fig. 7. The network architecture in the OSPF domain.

VI. Simulation Results

To illustrate application of the neural network, single area routing performed using the annealed chaotic network model in the OSPF domain is shown in Fig. 7. Hosts which belong to group A are represent as Hax, where “x” is the index of the host. If the source node located at N4 is going to send a multicast datagram to group A, then the OSPF database can be rewritten as the matrix shown in Table 1. The shortest path trees for group A members are listed as follows:

Ha1: R1-N1-R4-R5-N2;

Ha2: R1-N1-R4-R5-N3;

Ha3: R1-N1-R4-R5-N3.

Based on the algorithm shown in Fig. 6, the shortest mul-

Table 1. The OSPF Database

From \ To	N1	N2	N3	R1	R2	R3	R4	R5	R6
N1	0			0	0	0	0		
N2		0						0	0
N3			0					0	
R1	1								
R2	1								
R3	1								
R4	1							5	
R5		1	1				6		
R6		1							

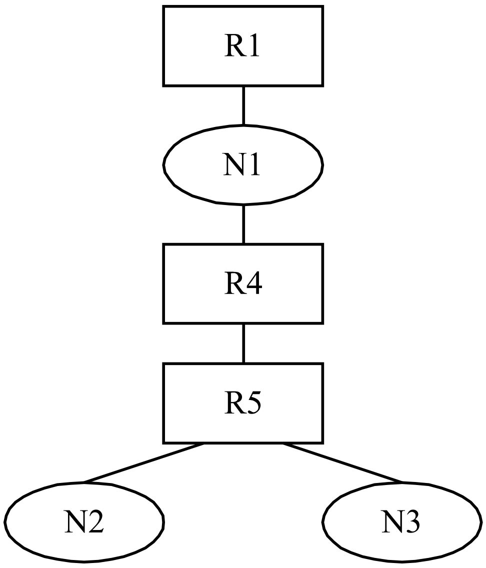


Fig. 8. Multicast shortest tree.

ticast tree is shown in Fig. 8. From the experimental results, the optimal path Ha1 is found after 36575 and 49 iterations while Ha2 and Ha3 can be discovered after 8987 and 27 iterations using Ali and Kamoun's method and the proposed annealed chaotic neural network, respectively. The simulation results are given in detail in Table 2. These results prove that the annealed chaotic neural network controlled using a simulated annealing strategy is suitable for the optimization problem because it can rapidly coverge to a near global solution when applied to the shortest path calculation problem of MOSPF.

VII. Discussion and Conclusions

In this paper, a two-dimensional Annealed Chaotic Neural Network with a chaotic dynamic mechanism controlled using a simulated strategy for shortest-path computation of MOSPF has been presented. The annealed chaotic network can converge from the chaotic state at a high temperature through successive bifurcations during temperature decreasing process to an equilibrium state at a low temperature. The proposed cooling schedule for controlling the chaotic dynamics in the annealed chaotic network model appears to converge rapidly to the optimal solution. In addition, the annealed chaotic network conforms to Internet standards (Moy, 1994a, 1994b) for both unicast and multicast forwarding, and it can coexist with other standard OSPF/MOSPF routers. Moreover, the designed neural-network-based approach is a self-organized structure that is highly interconnected and can be implemented in a parallel manner. It can also be easily applied to hardware devices so as to achieve very high-speed implementation.

Table 2. Simulation Results for Multicast Shortest Path Computation Shown in Fig. 8

Algorithms \ Path	Iterations		
	Ha1	Ha2	Ha3
The proposed algorithm	49	27	27
Ali and Kamoun's method	36575	8927	8927

Acknowledgment

This work was partially supported by the National Science Council of the R.O.C. under grants NSC 87-2213-E-167-005 and NSC 89-2213-E-167-009.

References

Aihara, K., T. Takabe, and M. Toyoda (1990) Chaotic neural networks. *Phys. Lett. A*, **144**, 333-340.

Ali, M. K. M. and F. Kamoun (1993) Neural networks for shortest path computation and routing in computer networks. *IEEE Trans. Neural Networks*, **4**, 941-954.

Brown, T. X. (November 1989) Neural networks for switching. *IEEE Commun. Magazine*, 72-81.

Cheng, K. S., J. S. Lin, and C. W. Mao (1996) The application of competitive Hopfield neural network to medical image segmentation. *IEEE Trans. Medical Images*, **15**(4), 560-567.

Choi, H. G., H. S. Lee, and S. H. Kim (1998) Adaptive prediction of nonstationary signals using chaotic neural networks. *IEEE Neural Network Proceedings*, **3**, 1943-1947.

Chung, P. C., C. T. Tsai, E. L. Chen, and Y. N. Sun (1994) Polygonal approximation using a competitive Hopfield neural network. *Pattern Recognition*, **27**, 1505-1512.

Dijkstra, E. W. (1959) A note on two problems in connection with graph. *Numerische Mathematik*, **1**, 269-271.

Floreen, P. and P. Orponen (1993) Attraction radii in binary Hopfield nets are hard to compute. *Neural Computation*, **5**, 812-821.

Hedrick, C. (1988) *Routing Information Protocol*. RFC 1058, Rutgers Univ., Camden, NJ, U.S.A.

Hopfield, J. J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, **79**, 2554-2558.

Hopfield, J. J. and D. W. Tank (1985) Neural computation of decisions in optimization problems. *Biological Cybernetics*, **52**, 141-152.

Hopfield, J. J. and D. W. Tank (1986) Computing with neural circuits: a model. *Science*, **8**, 625-633.

Lee, S. L. and S. Chang (1993) Neural Networks for routing of communication networks with unreliable components. *IEEE Trans. Neural Networks*, **4**, 854-863.

Leung, Y. W. (1994) Neural scheduling algorithms for time-multiplex switches. *IEEE Journal on Selected Areas in Communications*, **12**(9), 1481-1487.

Lin, J. S. (1999a) An annealed Hopfield neural network to vector quantization for image compression. *Optical Engineering*, **38**, 599-605.

Lin, J. S. (1999b) Fuzzy clustering using a compensated fuzzy Hopfield network. *Neural Processing Letter*, **10**, 35-48.

Lin, J. S., K. S. Cheng, and C. W. Mao (1996a) A fuzzy Hopfield neural network for medical image segmentation. *IEEE Trans. on Nuclear Science*, **43**(4), 2389-2398.

Lin, J. S., K. S. Cheng, and C. W. Mao (1996b) Multispectral magnetic resonance images segmentation using fuzzy Hopfield neural network. *J. Biomed. Comput.*, **42**, 205-214.

Moy, J. (1994a) *OSPF*, Version 2. RFC 1583, Proteon, Inc., Westbo-

rough, MA, U.S.A.
Moy, J. (1994b) *Multicast Extensions to OSPF*. RFC 1584, Proteon, Inc., Westborough, MA, U.S.A.
Pornavalai, C., N. Shiratori, and G. Chakraborty (1996) Neural network for optimal Steiner tree computation. *Neural Processing Letters*, **3**, 139-149.
Song, Z. C., C. T. Lun, and H. W. Qun (1997) Chaotic neural network with nonlinear self-feedback and its application in optimization. *Neurocomputing*, **14**, 209-222.

退火混沌神經網路於MOSPF協定最佳路徑計算之研究

林灶生[*]　劉明壽[*]　黃能富[**]

[*]國立勤益技術學院電子工程系
[**]國立清華大學資訊科學系

摘　要

多媒體通訊在很多網路服務上變得非常普遍，諸如視訊會議、視訊點選等等均是。大部分多媒體應用均需要主機電腦或路由器，如同多重廣播一般傳輸資料。為了提供有效的資料分布，路由器必須擁有多重廣播的能力。在本篇論文中，將一個由退火策略控制的自我反饋機制植入霍普式神經網路被提出，用以計算MOSPF 協定最短路徑樹。多重廣播最短路徑樹依照需求被建立並根植於資料源頭。用來解決最短路徑問題並適合於硬體實現，退火混沌神經網路會是一個好的選擇。此外，退火混沌神經網路可以跳脫區域最小值而得到接近整體或整體最小值之解。