ELSEVIER

Information Sciences 129 (2000) 17–30

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

The shortest path computation in MOSPF protocol using an annealed Hopfield neural network with a new cooling schedule

Jzau-Sheng Lin [a,*], Mingshou Liu [a], Nen-Fu Huang [b]

[a] *Department of Electronic Engineering, National Chin-Yi Institute of Technology, No. 35, Lane 215, Sec. 1, Chung-Shan Road, Taiping, Taichung, Taiwan, ROC*

[b] *Department of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan, ROC*

Received 16 June 1998; received in revised form 10 March 2000; accepted 4 July 2000

Abstract

Many network services, such as video conferencing and video on demand, have popularly used the multimedia communications. The attached hosts/routers are required to transmit data as multicasting in most multimedia applications. In order to provide an efficient data routing, routers must provide multicast capability. In this paper, a new cooling schedule in Hopfield neural network with annealing strategy is proposed to calculate the shortest path (SP) tree for multicast open shortest path first (MOSPF) protocol. The SP tree in multicast is built on demand and is rooted at the source node. To facilitate the hardware implementation, the annealed Hopfield neural network could be a good candidate to deal with SP problems in packet switching computer networks. In addition, it is proved that the proposed new cooling schedule is more suitable in all range of fixed temperature than the other demonstrated cooling schedules in the experimental results. © 2000 Elsevier Science Inc. All rights reserved.

Keywords: MOSPF protocol; Cooling schedule; Annealed Hopfield neural network

* Corresponding author. Tel.: +886-4-3924505 ext. 7311; fax: +886-4-3926610.
E-mail address: jslin@chinyi.ncit.edu.tw (J.-S. Lin).

0020-0255/00/$ - see front matter
PII: S0020-0255(00)00071-2

1. Introduction

Routing algorithm refers to a process of finding appropriate path so that the traffic can be relayed in some optimal ways. One important issue in routing algorithms is the speed with which they can react to topology changes. Traditionally, TCP/IP uses routing information protocol (RIP) [1] in which the routing decisions are based on the number of hops between source and destination. Many RIP systems have been replaced by a more powerful routing method called the open shortest path first (OSPF) [2,3] protocol.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 protocol in which routing information is flooded through the entire network. It provides a dynamic and adaptive routing mechanism to the topology change. The performance of most link-state protocols depends heavily on the optimal path calculation. One classical work called shortest path (SP) first algorithm proposed by Dijkstra [4] is popular in many systems for the SP selection. Dijkstra's algorithm works well, but it is computationally intensive, especially for multicast routing.

Multimedia communications have been popular in many network services, such as video conferencing, video on demand, and so on. In order to provide efficient data forwarding, routing protocol must provide multicast capability. An enhanced version of the OSPF, called the multicast open shortest path first (MOSPF) protocol [3], was proposed to deal with the IP multicast routing. RFC1584 is the protocol of 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 performed in parallel. The Hopfield neural network, a well-known network, was proposed for solving optimization problems [5–9], and many researchers have addressed the applications to the SP problems in computer networks subsequently. Leung [10] demonstrated neural scheduling algorithms for time-multiplex switches. Brown [11] also presented neural networks for switching problems. A new neural-network model, Routron, was proposed by Lee and Chang [12] for routing of communication networks with unreliable components. The continuous Hopfield neural network was applied by Ali and Kamoun [13] to the optimal routing problem in packet-switched computer networks to minimize the network wide average time delay.

In this paper, an annealed Hopfield neural network with a new cooling schedule is proposed to MOSPF protocol to compute SP. The general goal is to formulate the computer routing algorithm as an SP problem and apply the annealed Hopfield neural network to solve it. The specific objective is to simplify the constrained-cost function so as to avoid the determination of system-dependent parameters and to increase the 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 2 demonstrates the discrete Hopfield and annealed Hopfield neural networks for solving SP problem. In Section 3, the annealing techniques are described. The different cooling schedules are discussed and compared using an initial temperature in Section 4. Section 5 discusses MOSPF SP tree. Section 6 presents several experimental results to show that the annealed Hopfield network is suitable for the computation of SP tree in OSPF routing domain. Section 7 specifies the inter-area routing for neural-based MOSPF. Finally, Section 8 gives the discussion and conclusions.

2. Shortest path computation with neural networks model

Let v_i be the state at neuron i. Then the Hopfield neural network can be modelled as

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

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

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

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

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

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

In MOSPF routing domain, the topology of a single area in AS can be defined as a directed graph $G(N, A)$, with n nodes and l arcs, each node in the graph represents a router or transit network. Corresponding to each $arc(x, i)$, there is a nonnegative weight $c(x, i)$, representing the cost from node x to node i. To compute the SP from one source node to a group, the Hopfield model with $n \times n$ array is proposed, 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 neurons at the diagonal, only $n(n-1)$ neurons are used to calculate in each array.

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

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

To compute the SP using Hopfield model, the object function, which is similar to Ali and Kamoun's definition, can be defined as

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

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 optimal weight, we replace Eqs. (1) and (3) by double indices and let $U_i = 0$ for all i, then

$$v_{xi}(t+1) = \begin{cases} 1 & \text{net}_{xi} > 0, \\ v_{xi}(t) & \text{net}_{xi} = 0, \\ 0 & \text{net}_{xi} < 0 \end{cases}$$

and

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

Let

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

such that E and E_{obj} decrease at the same rate as v_{xi} change. By comparing each component in Eq. (5), we can obtain inter-connection 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 [14] indicated that to determine the attraction radius of a stable vector in a discrete (binary) Hopfield network is a NP-hard problem. It might hamper the convergence of the discrete Hopfield net to train with complex and large data sets, and the Hopfield neural network may trap on to a local minimum. In this application using a discrete Hopfield network, a neuron (x, i) in a firing state indicates that the arc from node x to node i does exist. But in the annealed Hopfield neural network, a neuron (x, i) in a probable state indicates that the arc from node x to node i does exist with a degree of uncertainty described by a probability function. The annealed Hopfield network, which is a continuous model with probability function, can overcome the NP-hard problem exhibit in binary Hopfield net.

3. Annealing techniques

Simulated annealing is a stochastic relaxation algorithm which has been used successfully to resolve the optimization problems including computer network topology problems [15], traveling salesman problems [16], circuit routing problems [17], image processing problems [18,19], and clustering problems [20]. Instead of the other optimization methods such as steepest descent approach used in the Hopfield neural network, the simulated annealing technique, which allows the search to move away from a local minimum, seeks the global or near global minimum of an energy function without getting trapped in local minimum. The simulated annealing technique had nonzero probability to go from one state to another, moves temporarily towards 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 to go to a higher energy state at a higher temperature.

Although the simulated annealing method can yield the global minimum, it is very time-consuming with asymptotical iterations. The annealed Hopfield neural network, presented to the SP tree for MOSPF protocol, which incorporate the characteristics of the annealing strategy with a new cooling schedule and the Hopfield neural network, can converge much faster than the simulated annealing. Each row of this modified Hopfield network represents a source node x and each column represents a destination node i. The network reaches a stable state when the Lyapunov energy function is minimized. For example, a neuron (x, i) in a maximum probability state indicates the arc from x to node i does exist. Based on Bibro et al. [21], each state $v_{x,i}$ is looked upon as the probabilities of finding the arc from node x to node i existing undergo random thermal perturbations. The probability of the arc from node x to node i does exist at a given temperature T conforms to a Boltzmann distribution

$$v_{x,i} \propto \mathrm{e}^{-\Delta E_{x,i}/T}. \tag{6}$$

Then the total input of net(x, i) $\mathrm{net}_{x,i}$ and mean field $E_{x,i}$ can be calculated from Eq. (4) to be

$$\mathrm{net}_{x,i} = -\Delta E_{x,i} = \sum_{y=1}^{n} \sum_{j=1}^{n} w_{x,i;y,j} v_{y,j} + \sum_{x=1}^{n} I_{x,i}. \tag{7}$$

The probability of the arc from node x to node i does exist and can be normalized as follows:

$$v_{x,i} = \frac{\mathrm{e}^{-E_{x,i}/T}}{\sum_{j=1}^{c} \mathrm{e}^{-E_{x,j}/T}}. \tag{8}$$

As the temperature is reduced, the training arc will begin to approach in a feasible path that will minimize the total cost.

4. Cooling schedule

In order to converge to a near global minimum in annealing process, a feasible cooling schedule is required. Reaching thermal equilibrium at low temperature might take a very long time. The search for adequate cooling schedules has been the subject of an active research field for several years [22]. Geman and Geman [23] demonstrated that if the temperature is lowered at the rate

$$T_{\mathrm{rate}} = \frac{T_0}{\log(k+1)}, \tag{9}$$

where T_0 is a constant and k is the number of iterations, the algorithm will converge to the set of states of least energy. Jalali and Boyce [19] presented that the value of the constant T_0 for which Geman and Geman were able to guarantee convergence is in general very high, so that the convergence time becomes impractically slow. Jalali et al. used a schedule very similar to that of Geman and Geman given in Eq. (9), but with a steeper descent at higher iterations as follows:

$$T_{\text{rate}} = \frac{T_0}{\log(k+1)^3}. \tag{10}$$

Jalali and Boyce showed that the value of T_0 in Eq. (10) has to be kept as small as possible, so that the number of iterations can be held within a reasonable limit. Unfortunately, the cooling schedules specified by Eqs. (9) and (10) with high value of T_0 are too slow to be of practical use [24]. Kirkpatrick et al. [17] proposed a cooling schedule which specified a finite sequence of values of the temperature and a finite number of transitions attempted at each value of the temperature. The decrement function of cooling schedule is defined by

$$T_{\text{rate}} = (\alpha)^k T_0, \quad k = 1, 2, \ldots, \tag{11}$$

where α $(0.8 \leqslant \alpha \leqslant 0.99)$ is a constant smaller but close to unit.

In this paper, a new decrement function of cooling schedule is proposed as follows:

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

where α is a constant same as the one in Eq. (11) and β is another constant which needs to be defined. $\beta = 4$ has been selected in this paper, Eq. (12) can result in a faster decrement speed than those resulted from Eq. (11). The decrement results with initial temperature $T_0 = 4000$ in 100 iterations are shown in Fig. 1.

To demonstrate the power of annealed Hopfield model, we choose an example from Ali and Kamoun's paper [13] as shown in Fig. 2. While this

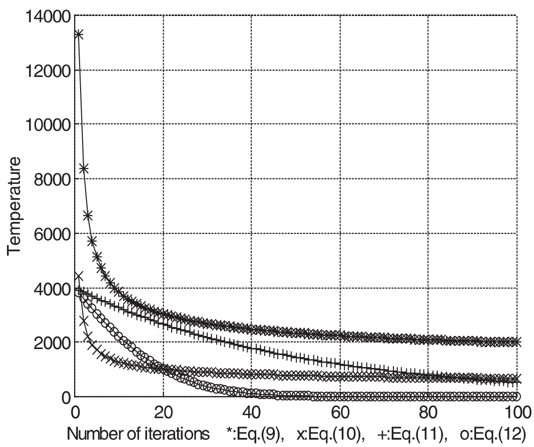

Fig. 1. The reduction process using different decrement functions described from Eqs. (9)–(12) with $T_0 = 4000$ and 100 iterations.

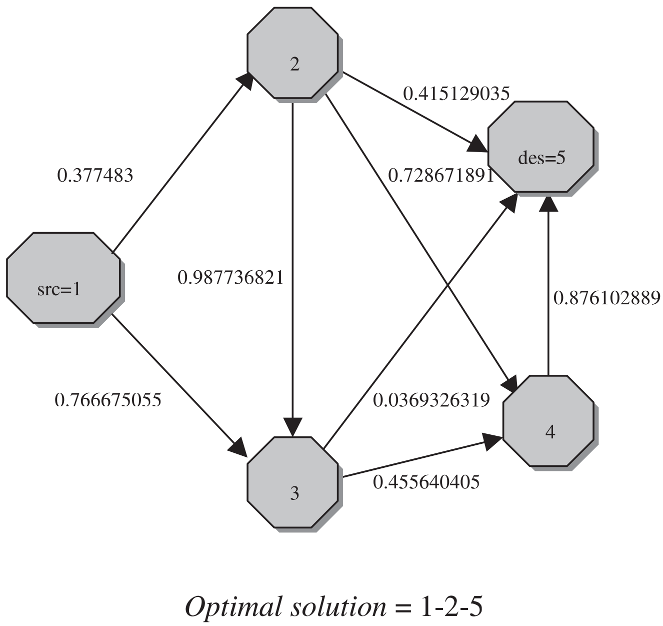

Fig. 2. Testing example for the SP problem.

example is not suited for MOSPF routing, it is mentioned here for comparison basis. First, randomly initialized neurons'states with a probability value from 0 to 1 at time $t = 0$ fixed the source and destination nodes 1 and 5, respectively. Then the minimum value for changed states δ and weighting factors were set up as $\delta = 0.00001$, $A = 350$, $B = 25$, $C = 450$, and $D = 5000$.

The training process was terminated if the minimum value for changed states of the adjustment iterations is lower than δ. The optimal solution, path 1:1–2–5, can be obtained after 7358 iterations by Ali and Kamoun [13], but the optimal result can be completed just after four iterations in an annealed Hopfield neural network using the proposed cooling schedule.

In this paper, an iteration represents all the neurons' states that are changed one time. The convergence curve of energy function for src = 1 and des = 5 is shown in Fig. 3. This network can rapidly converge to valid results that are often globally optima. The experimental result with different cooling schedules is shown in Table 1. In Eq. (9), an optimal solution can be obtained with a lot of iterations in a high initial temperature, while an overflow error may be trapped with a low initial temperature. Although an optimal path can be completed during a few iterations with $T_0 = 50$ in Eq. (10), an overflow error is met with $T_0 = 5$ and the optimal result was obtained with higher initial temperatures after a lot of iterations. From experimental results, we find that

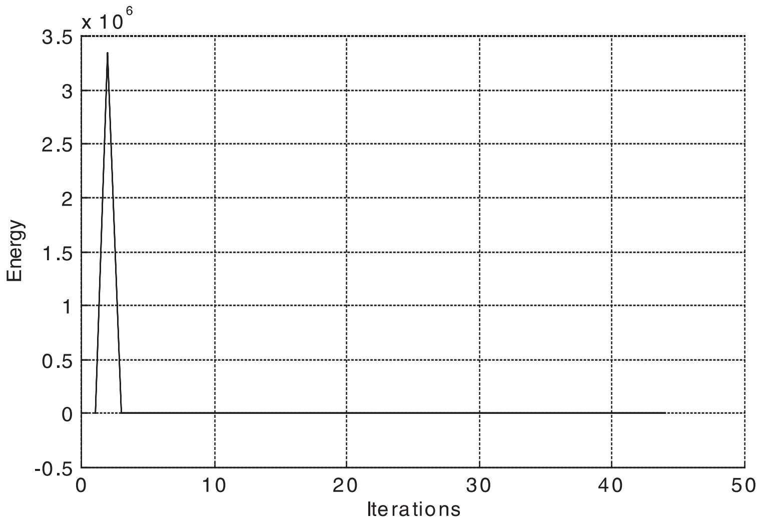

Fig. 3. Energy curve using the proposed cooling schedule with $T_0 = 450$.

Table 1
The consumed iterations for Fig. 1 using different cooling schedule with distinct temperatures

Cooling schedules	Temperatures			
	5	50	250	450
Eq. (9)	Overflow	17	916	5535
Eq. (10)	Overflow	17	53	315
Eq. (11)	4	11	69	125
Eq. (12)	4	9	36	44

Eqs. (11) and (12) are suitable in different temperatures. However, Eq. (12) can rapidly converge to the optimal path with a few iterations for all range temperatures.

5. MOSPF shortest path tree

In multicast forwarding, packets are usually sent to multiple destinations. The RFC 1584 algorithm is to construct a SP tree which span all nodes, and then those nodes which does not belong to the destination group are pruned out. However, the optimal path found by neural networks is for specific source and destination nodes, thus the construction of SP tree must start from the reverse direction. Different SPs for specific source/destination pairs must be combined into a single tree.

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. 4. The algorithm of updated tree.

To construct SP tree, each node in the tree has $n-1$ downstream interfaces, each indexed as node down(i), $1 \leqslant i < n$. The number n represents the total number of interfaces if the node is a router, or total number of attached routers if the node is a multiaccess network. The multicast SP tree begins with root node, usually this is the source network. Initially, all the routers/transit networks are not marked. After the SP has been found for a given destination by neural networks, which we call it path[i] for i from 0 to $N-1$, and path[0] is always for root node, the tree will be updated as shown in Fig. 4. After all the SPs for different destination have been computed, a multicast SP tree will be constructed.

6. Simulation result

To illustrate the application of neural network, the single area routing performed by annealed Hopfield network model in OSPF domain is examined as shown in Fig. 5. Hosts belong to group A is represented 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 2. The SP tree for each group A member is listed as follows:

Ha1: R1–N1–R4–R5–N2
Ha2: R1–N1–R4–R5–N3
Ha3: R1–N1–R4–R5–N3

Following the algorithm in Fig. 4, the multicast shortest tree is shown in Fig. 6. From experimental results, the optimal path Ha1 is found after 19285, 670, 166, and 51 iterations, while Ha2 and Ha3 can be discovered after 7564, 367,

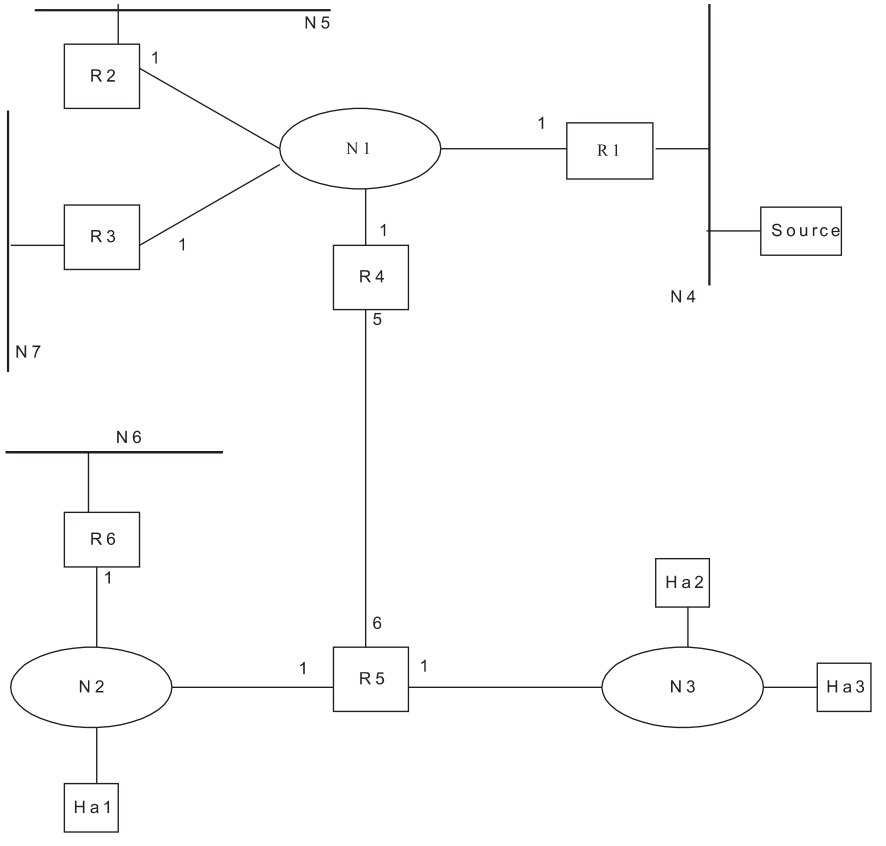

Fig. 5. The network architecture in OSPF domain.

Table 2
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							

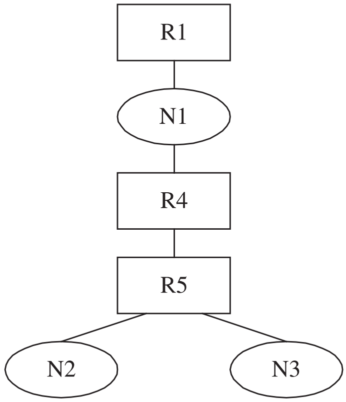

Fig. 6. Multicast shortest tree.

78, and 30 iterations with $T_0 = 450$ using Eqs. (9)–(12), respectively. These results prove that the new cooling schedule is a robust one, which can rapidly converge to an optimal solution in the SP calculation problem of MOSPF using an annealed Hopfield neural network.

7. Inter-area routing

In many situations, the source and destinations are not in the same area, thus the OSPF must perform inter-area routing. Generally, the construction of SP tree for inter-area forwarding is the same as we have done for intra-area routing. Each router in an area computes the SP tree for that area, and constructs its forwarding cache. On the other hand, area border routers compute the SP tree for each attached area, and combine them into their forwarding cache. But there are two exceptions in the intra-area routing. The first, when a router receives a datagram with its source address located in the same area as this router, then we need to compute the shortest path to all wild-card multicast receivers since some of the group members are outside the working area and thus the destinations are unknown.

For the second case, the source node is in different area. Just as specified in [3], we use the summary link to create the routing topology, and all the cost values used for neural network computations are in reverse direction, that is, all cost values are from destinations toward source instead of from the source node.

8. Discussion and conclusions

In this paper, a two-dimensional annealed Hopfield neural network with a new cooling schedule for SP computation of MOSPF has been presented. The proposed cooling schedule in the annealed Hopfield model appears to converge rapidly to the optimal solution. In addition, the proposed algorithm conforms to Internet standard ([2,3]) 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 inter-connected and can be implemented in a parallel manner. It can also be easily designed for hardware devices to achieve very high-speed implementation.

Acknowledgements

This work was partially supported by the National Science Council of the ROC under grant NSC 87-2213-E-167-005.

References

[1] C. Hedrick, Routing Information Protocol, RFC 1058, Proteon, Inc., June, 1988.
[2] J. Moy, OSPF version 2 RFC 1583 Proteon, Inc., 1994a.
[3] J. Moy, Multicast Extensions to OSPF, RFC 1584, Proteon, Inc., 1994b.
[4] E.W. Dijkstra, A note on two problems in connection with graph, Numerische Mathematik 1 (1959) 269–271.
[5] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. 79 (1982) 2554–2558.
[6] J.J. Hopfield, D.W. Tank, Neural computation of decisions in optimization problems, Biol. Cybern. 52 (1985) 141–152.
[7] J.J. Hopfield, D.W. Tank, Computing with neural circuits: a model, Science 8 (1986) 625–633.
[8] K.S. Cheng, J.S. Lin, C.W. Mao, The application of competitive Hopfield neural network to medical image segmentation, IEEE Trans. Medical Images 15 (1996) 560–567.
[9] J.S. Lin, K.S. Cheng, C.W. Mao, A fuzzy Hopfield neural network for medical image segmentation, IEEE Trans. Nucl. Sci. 43 (1996) 2389–2398.
[10] Y.W. Leung, Neural scheduling algorithms for time-multiplex switches, IEEE J. Selected Areas Commun. 12 (1994) 1481–1487.
[11] T.X. Brown, Neural networks for switching, IEEE Commun. Mag. (1989) 72–81.
[12] S.L. Lee, S. Chang, Neural Networks for routing of communication networks with unreliable components, IEEE Trans. Neural Networks 4 (1993) 854–863.
[13] M.K.M. Ali, F. Kamoun, Neural networks for shortest path computation and routing in computer networks, IEEE Trans. Neural Networks 4 (1993) 941–954.
[14] P. Floreen, P. Orponen, Attraction radii in binary Hopfield nets are hard to compute, Neural Comput. 5 (1993) 812–821.
[15] S. Pierre, M.A. Hyppolite, J.M. Bourjolly, O. Dioume, Topological design of computer communication networks using simulated annealing, Eng. Appl. Artif. Intell 8 (1995) 61–69.
[16] D.E. Van Den Bout, T.K. Miller III, A traveling salesman objective function that works, Int. Conf. Neural Network II (1988) 299–303.
[17] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simuted annealing, Science 220 (1983) 671–680.

[18] H.L. Tan, S.B. Gelfand, E.J. Delp, A cost minimization approach to edge detection using simulated annealing, IEEE Trans. PAMI 14 (1991) 3–18.
[19] S. Jalali, J.F. Boyce, Determination of optimal general edge detectors by global minimization of a cost function, Image Vision Comput. 13 (1995) 683–693.
[20] R.W. Klein, R.C. Dubes, Experiments in projection and clustering by simulated annealing, Pattern Recognition 22 (1989) 213–220.
[21] G. Bibro, R. Mann, T.K. Miller III, W. Snyder, D.E. Van Den Bout, M. White, Mean field annealing and neural networks, in: Advances in Neural Information Processing Systems, Morgan-Kaufmann, Los Altos, CA, 1989, pp. 91–98.
[22] Van Laarhoven, E.H.L. Aarts, Simulated Annealing: Theory and Applications, Kluwer Academic Publishers, Boston, MA, 1988.
[23] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions and the Baysian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 721–741.
[24] S. Haykin, Neural Network: a Comprehensive Foundation, Macmillan, New York, 1994.