

Noisy Chaotic Neural Networks For Delay Constrained Multicast Routing

Wen Liu, *Student Member, IEEE*, Lipo Wang, *Senior Member, IEEE*, Haixiang Shi

Abstract—The QoS constrained multicast routing is studied widely these years due to the development of multimedia applications such as video-conferencing and video-on-demand. We apply noisy chaotic neural networks (NCNN) on the delay constrained multicast routing problem. The NCNN has richer and more flexible dynamics, and therefore is more efficient compared with the conventional Hopfield neural network as the latter is often trapped at local minima.

I. INTRODUCTION

The multicast routing problem, also known as the Steiner tree problem, aims to minimize the total cost of a multicast tree and is NP-complete (nondeterministic polynomial time complete) [1]. In multicast services, data generated by the source flows through the multicast tree, traversing each tree edge exactly once.

In real-time communications, the network has to find an optimum multicast route, which has enough resources to guarantee the required quality of service (QoS) from the end user [2], [3]. This problem is called QoS constrained multicast routing problem or constrained Steiner tree (CST) problem [4]–[6]. Individual QoS parameters may be conflicting and inter-dependent, thus making the problem even more challenging [7].

The application of neural networks to the routing problem has been motivated by the powerful parallel computational ability of the neural network and the fact that a hardware-implemented neural network can achieve high response speeds [8], [9]. A modified Hopfield neural network model is proposed in [10] to solve the delay constrained multicast routing problem. In this paper, we modify the energy function and solve the problem using noisy chaotic neural networks (NCNN) [11], [12].

This paper is organized as follows. We introduce the delay constrained multicast routing problem in Section 2. The noisy chaotic neural network is presented in Section 3. The Simulation result is presented and discussed in Section 4. Finally, we conclude this paper in Section 5.

II. THE DELAY CONSTRAINED MULTICAST ROUTING PROBLEM

A. Problem formulation

We use the formulation proposed in [13]. Considering an n -node communication network with D destinations, for which

Manuscript received October 6, 2006; revised December 31, 2006.

Wen Liu, Lipo Wang, and Haixiang Shi are with School of Electrical and Electronic Engineering, Nanyang Technological University, Block S1, 50 Nanyang Avenue, Singapore 639798. (phone: +65 6790 6372; fax: +65 6793 3318; email: {liuw0004, eloping, pg02782641}@ntu.edu.sg)

the neurons are arranged on $D \times n$ matrices, matrix m is used to compute the constrained unicast route from source node s to destination m , ($m = 1, \dots, D$). Each element in one matrix is treated as a neuron and neuron mxi describes the link from node x to node i for destination m in the communication network. V_{xi}^m is the output of the neuron at location (x, i) in matrix m : $V_{xi}^m = 1$, then the link from node x to node i is on the final optimal tree for destination d_m ; $V_{xi}^m = 0$, otherwise.

To characterize the connection topology of the communication network, we define P_{xi} as $P_{xi} = 1$, if the link from node x to node i does not exist; $P_{xi} = 0$, otherwise. The cost and delay of a link from node x to node i are assumed to be real non-negative numbers and are denoted by C_{xi}, L_{xi} , respectively [10]. For non-existing arcs, corresponding entries will be zero. Link costs and link delays are assumed to be different independent functions. For example, costs could be a measure of buffer space or channel bandwidth used, and the delay could be a combination of propagation, transmission, and queuing delay.

B. Problem Definition

The delay constrained multicast routing problem [14], [15] is defined to construct a tree rooted at the source s and spanning to all the destination members of $D = \{d_1, d_2, \dots, d_m\}$ such that 1) the total cost of the tree is minimum; 2) the delay from the source to each destination is not greater than the required delay constraint:

$$\sum_{x=1}^n \sum_{i=1, i \neq x}^n L_{xi} V_{xi}^m \leq \Delta \quad V_{xi}^m \in \{0, 1\} \quad (1)$$

where Δ is the delay bound. V_{xi}^m denotes the neuron output of constrained unicast route for destination d_m .

C. The energy function

When solving the optimization problem by neural networks, one needs to define a proper energy function which determines the associated weights between neurons as in (2). The energy function of a Hopfield neural network (HNN) decrease monotonically as neurons update [16]. Hence the energy function should be designed with its minima value corresponding to the optimal solution.

$$-\frac{\partial E}{\partial V_{xi}^m} = \sum_{y=1}^N \sum_{j=1, j \neq y}^N w_{yj, xi} V_{yj}^m(t) + I_{xi} \quad (2)$$

Where $w_{yj,xi}$ is the connection weight from neuron (y, j) to neuron (x, i) and I_{xi} is the input bias.

Ali and Kamoun [13] proposed the energy function for the shortest path computation. Pornavalai et al [10] then extended it to suit the delay constraint multicast routing problem. We change the neuron update rule by using the average value of neuron outputs as the threshold to fire the neuron, i.e., to separate the final outputs into 0 or 1. In the conventional model, the outputs are forced to be 0 or 1 by an energy term $\sum_{x=1}^n \sum_{i=1}^n V_{xi}^m (1 - V_{xi}^m)$.

The total energy function E for the delay constrained multicast routing is the sum of energy functions E^m ($m \in D$) for delay constrained unicast routing to every destination. Here E^m is the energy function of matrix m , which is used to find the constrained unicast route from source node s to destination d_m . It consists of a cost term E_1^m and several constraint terms E_2^m, \dots, E_5^m [10]:

E_1^m is the total cost of the unicast route for destination d_m . Neurons from different matrices that represent the same link in the communication network try to cooperate together to minimize the cost of the multicast tree [10]. The total cost is reduced when unicast routes for different destinations choose the same link.

$$E_1^m = \sum_{x=1}^n \sum_{i=1, i \neq x}^n C_{xi} \frac{1}{1 + \sum_{j=1, j \neq m}^n V_{xi}^j} V_{xi}^m \quad (3)$$

$E_2^m = (1 - V_{ms}^m)$ creates a virtual link from destination d_m to source s [13]. E_3^m ensures that for every node, the number of incoming links is equal to the number of outgoing links [13]. The pre-condition is a deliberate link from the destination to the source created by the E_2^m term.

$$E_3^m = \sum_{x=1}^n \left\{ \sum_{i=1, i \neq x}^n V_{xi}^m - \sum_{i=1, i \neq x}^n V_{ix}^m \right\}^2 \quad (4)$$

$E_4^m = \sum_{x=1}^n \sum_{i=1, i \neq x}^n P_{xi} V_{xi}^m$ penalizes neurons that represent non-existing links [13]. $E_5^m = \int h(z) dz$ is used to satisfy the delay constraint. Where $h(z)$ is the transfer function of the linear programming (LP) type neuron [17] and $h(z) = 0$, if $z \leq 0$; $h(z) = z$, otherwise. Furthermore, $z = \sum_{x=1}^n \sum_{i=1, i \neq x}^n L_{xi} V_{xi}^m - \Delta$. Thus the LP neuron contributes positively only when the delay constraint is violated.

As described above, when the neural network reaches the minimal value of the energy function, all the constraints are fulfilled and the route cost is optimized at the same time.

III. SIMULATION RESULTS

Wang et al [11], [12], [18] proposed a stochastic chaotic simulated annealing (SCSA) method which can solve the traveling salesman problem (TSP) efficiently. The model is a noisy chaotic neural network (NCNN) resulted from adding decaying stochastic noise into the transiently chaotic neural network (TCNN) [19].

Substituting the energy function in the NCNN and TCNN model, we will get the neural network dynamics function. The algorithm is implemented in VC++ and run on a Linux cluster (16-node dual Xeon 3.06 GHz, Intel IA32). We control the

TABLE I
COMPARISON OF THREE SOLUTIONS WHEN $\Delta = 15$.

Destination No.	solution 1		solution 2		solution 3	
	cost	delay	cost	delay	cost	delay
4	5	5	5	11	5	5
5	1	5	1	5	1	5
7	3	13	3	13	6	11
8	4	6	4	6	4	6
total	10	13	10	13	10	11

iteration of the algorithm through the change in the energy function between two steps: $\Delta E = E(t) - E(t - 1)$. The output is accepted, i.e. iterations stop, if ΔE is smaller than a threshold (0.002) in three continuous steps.

Values for the weighting coefficients are chosen as follows based on [10]:

$$\begin{aligned} \mu_1 &= 200 & \mu_2 &= 5000 & \mu_3 &= 1500 \\ \mu_4 &= 5000 & \mu_5 &= 250 \end{aligned}$$

Without loss of generality [12], we choose the parameters of the neural network as follows: $\epsilon_{xi}^m = \epsilon = 0.004$, independent of neuron location (x, i, m) . k corresponds to time step Δt used in the Euler approximation, and in our simulations $k = 0.9999$, i.e., $\Delta t = 0.0001$. In addition, $I_0 = 0.65, \beta_1 = \beta_2 = 0.001, z(0) = 0.1$ and $A[n(0)] = 0.06$. The initial value for self-feedback z and the amplitude of random noise n are set properly to keep balance of every term in the energy function. Initial inputs of neural networks $U_{xi}(0)$ are randomly generated between $[-1, 1]$.

At the end of each iteration, we set each neuron on or off according to the average value (V_T) of the output matrix. If $V_{xi} \geq V_T$, the neuron is on, i.e., $V_{xi} = 1$, which means the link from x to i is chosen in the final optimal tree, and vice versa.

The simulation is carried out on an eight-node communication network [10]. The source node s is labeled 1, and destination nodes are labeled 4, 5, 7 and 8, respectively. The algorithm finds three solutions when the delay constraint (Δ) is 15. The final optimal route is presented in Fig.1. The output matrix of the neural network is also showed in Fig.1(d). We compare these three solutions in Table I (the solutions 1, 2, and 3 corresponds to (a), (b), and (c) in Fig.1, respectively).

When the delay constraint (Δ) is 20, the NCNN finds the optimal tree shown in Fig. 2, with total cost= 8.

We run the algorithm 10000 times with randomly generated initial states for delay constraint $\Delta = 15$ and 20, respectively, to compare with conventional Hopfield networks used in [10] and TCNN [19]. The result is listed in Table II. "Route opt" denotes the ratio that the algorithm achieves the optimal result in 10000 runs. The NCNN is capable to jump out of local minima and achieves the global optimal due to its complex dynamics, including chaos and stochastic noise. As a trade off, the execution time increases.

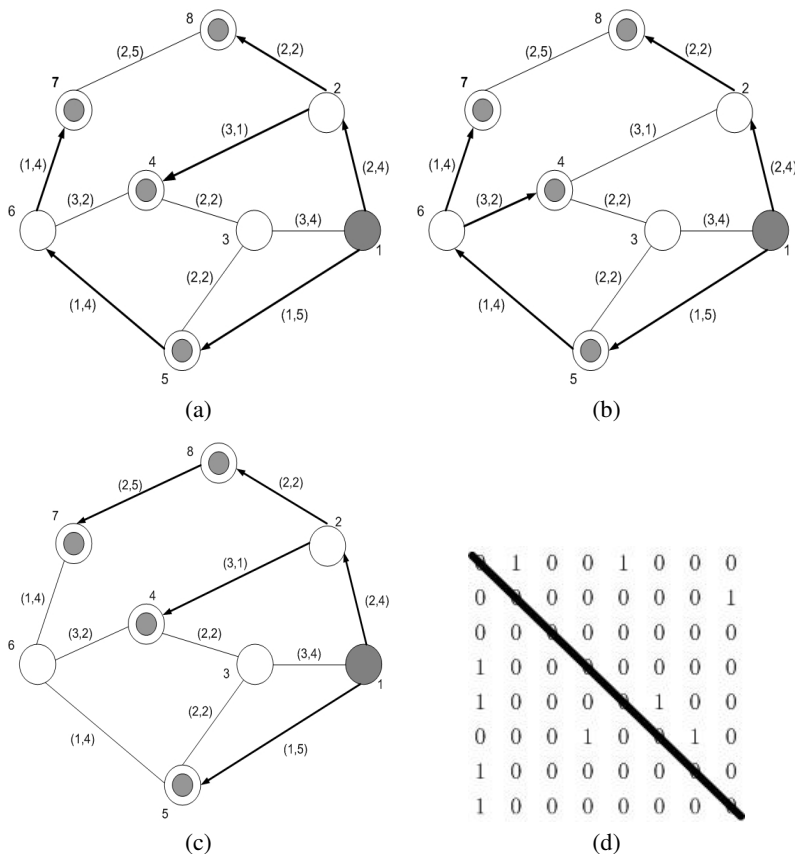


Fig. 1. Network topology of the optimal delay constrained multicast route for an eight-node communication network when the delay constraint is 15. (a), (b), and (c), presents the solutions 1, 2, and 3 in Table I, respectively; (d) the output matrix for solution 2.

TABLE II
PERFORMANCE COMPARISON BETWEEN CONVENTIONAL HNN, TCNN, AND NCNN. SD STANDS FOR STANDARD DEVIATION.

Method		Pornavalai [10]	TCNN [19]	NCNN	
$\Delta = 15$	Time (s)	mean	0.92	1.24	1.39
		SD	1.74	0.34	0.49
	Route opt %	82.25	91.48	93.75	
$\Delta = 20$	Time (s)	mean	1.07	1.36	1.55
		SD	0.89	0.61	0.5
	Route opt %	71.31	90.76	96.27	

The effect of the damping factor β_1 and β_2 is analyzed and results are shown in Table III and IV. In applications, we can balance the “route opt” ratio and the execution time through parameters β_1 and β_2 to control the decay of chaotic dynamics and noise. The larger β_1 and β_2 , the faster the NCNN converges, while the smaller the parameters, the more probable the NCNN can reach the global optimal.

IV. CONCLUSION

The equation governing dynamics of neural networks and the formulation of the delay constrained multicast routing

problem have been discussed in this paper. The algorithm can be applied to directed topology, i.e., asymmetric communication networks or applications requiring different delay bounds for different destinations. Simulation results show that the noisy chaotic neural network is efficient in solving the delay constraint multicast routing problem.

REFERENCES

[1] B. Wang and C. J. Hou, “A survey on multicast routing and its QoS extension: problems, algorithms, and protocols,” *IEEE Network Magazine*, vol. 14, no. 1, pp. 22–36, 2000.

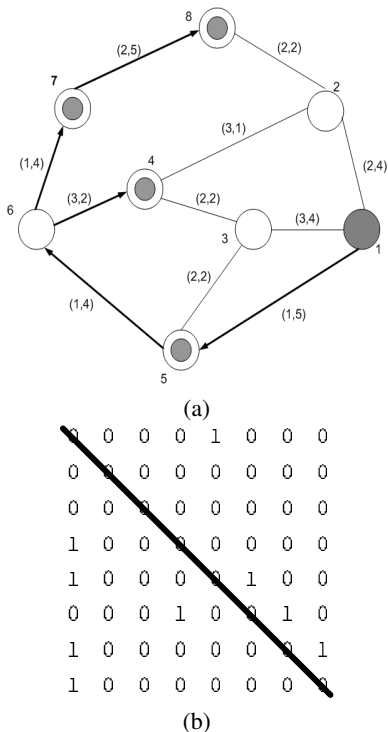


Fig. 2. The optimal delay constrained multicast route when $\Delta = 20$ (a) The optimal route; (b) the output matrix.

TABLE III
PERFORMANCE COMPARISON OF THE NCNN MODELS WITH DIFFERENT β_1 ON THE 8-NODE NETWORK WITH $\Delta = 20$, $\beta_2 = 0.001$

β_1		0.01	0.005	0.001	0.0005
Time (s)	mean	1.10	1.27	1.55	3.35
	SD	0.30	0.37	0.50	0.48
Route opt %		93.02	95.30	96.27	96.85

TABLE IV
PERFORMANCE COMPARISON OF THE NCNN MODELS WITH DIFFERENT β_2 ON THE 8-NODE NETWORK WITH $\Delta = 20$, $\beta_1 = 0.001$

β_2		0.01	0.005	0.001	0.0005
Time (s)	mean	0.89	1.01	1.55	1.85
	SD	0.45	0.45	0.50	0.47
Route opt %		94.04	94.70	96.27	97.23

[2] H. Y. Tyan, J. C. Hou, and B. Wang, "Many to many multicast routing with temporal quality of service guarantees," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 826-832, 2003.

[3] M. Charikar, J. Naor, and B. Schieber, "Resource optimization in qos multicast routing of real-time multimedia," *IEEE Transactions on networking*, vol. 12, no. 2, pp. 340-347, 2004.

[4] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, "A source-based algorithm for near-optimum delay-constrained multicasting," in *Proc. IEEE Infocom 95*, 1995, pp. 377-385.

[5] R. Widyono, "The design and evaluation of routing algorithms for real-time channels," University of California at Berkeley & International Computer Science Institute, Tech. Rep. 94-024, June 1994.

[6] M. Parsa, Q. Zhu, and J. J. Garcia-Luna-Aceves, "An iterative algorithm for delay-constrained minimum-cost multicasting," *IEEE Transactions on Networking*, vol. 6, no. 4, pp. 461-474, 1998.

[7] A. Roy, N. Banerjee, and S. K. Das, "An efficient multi-objective qos routing algorithm for real-time wireless multicasting," in *Proceedings of IEEE Vehicular Technology Conferenc*, May. 2002, pp. 1160-1164.

[8] H. E. Rauch and T. Winarske, "Neural networks for routing communication traffic," *IEEE Cont. Syst. Mag.*, vol. 8, no. 2, pp. 26-31, 1988.

[9] P. Venkataram, S. Ghosal, and B. P. V. Kumar, "Neural network based optimal routing algorithm for communication networks," *Neural Networks*, vol. 15, no. 10, pp. 1289-1298, 2002.

[10] C. Pornavalai, G. Chakraborty, and N. Shiratori, "A neural network approach to multicast routing in real-time communication networks," in *International Conference on Network Protocols (ICNP-95)*, 1995, pp. 332-339.

[11] L. P. Wang and F. Y. Tian, "Noisy chaotic neural networks for solving combinatorial optimization problems," in *Proc. International Joint Conference on Neural Networks*, vol. 4, Como, Italy, July 24-27 2000, pp. 37-40.

[12] L. P. Wang, S. Li, F. Y. Tian, and X. J. 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*, vol. 34, no. 5, pp. 2119-2125, 2004.

[13] M. K. M. Ali and F. Kamoun, "Neural networks for shortest path computation and routing in computer networks," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 941-954, 1993.

[14] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast routing for multimedia communication," *IEEE/ACM Transactions Networking*, vol. 1, no. 3, pp. 286-292, June 1993.

[15] H. F. Salama, D. S. Reeves, and Y. Viniotis, "Evaluation of multicast routing algorithms for real-time communication on high-speed networks," *IEEE J. Select. Areas Communication.*, vol. 15, no. 3, pp. 332-345, 1997.

[16] S. Haykin, *Neural Networks-A comprehensive Foundation (second edition)*. Hamilton, Canada: Prentice Hall International Inc., 1999.

[17] D. W. Tank and J. J. Hopfield, "Simple neural optimization networks: An a/d converter, signal decision circuit," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 5, pp. 533-541, 1986.

[18] L. P. Wang and K. Smith, "On chaotic simulated annealing," *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 716-718, July. 1998.

[19] L. N. Chen and K. Aihara, "Chaotic simulated annealing by a neural network model with transient chaos," *Neural Networks*, vol. 8, no. 6, pp. 915-930, 1995.