International Journal of Computer Theory and Engineering, Vol. 2, No. 3, June, 2010
1793-8201

Delay Bound Multicast Routing Using Hopfield Neural Network

Sushma Jain and J.D. Sharma

Abstract-- **The multicasting is used to transfer data from one or more sources to a potentially large number of destinations such that overall utilization of resources is minimized in some sense. An approach based on neural network is presented to solve the delay constraint multicast routing problem, which is a NP complete problem,. The multicast tree is obtained corresponding to optimum (minimum) routing cost subjected to end-to-end delay constraints. The multicast tree is obtained by recursively obtaining the delay bound shortest paths from source to various destinations and combining them by union operator to ensure that a link is appearing only once in the multicast tree. The effectiveness of the developed algorithm is tested to obtain optimum multicast trees for different sets of source and destinations on 14-node weighted connected network.**

Index Terms-- **Multicast Routing, Neural Network, Hopfield Neural Network, Steiner Tree, Routing Algorithm.**

I. Introduction

The multimedia applications are becoming increasingly important and the computer networks are designed to handle media traffic to end users. For such applications, various requirements, which are regarded as Quality of Service (QoS) parameters, must be guaranteed. Otherwise, the real time application will be meaningless. These requirements include end-to-end delay bound, delay jitter bound, minimum bandwidth etc. The researchers have been studying for many years to develop efficient multicast routing algorithm with and without constraints. Among various QoS parameters, the routing with end-to-end delay have been explored extensively. The various minimum Steiner tree heuristics have been reported in literature [1-3]. The multicast tree, a NP complete problem, for communication network was first formulated by Kompella et al. [4]. Noronha and Tobagi [5] proposed an algorithm based on integer programming to construct the optimal source-specific delay-constrained minimum Steiner tree. The bounded shortest multicast algorithm [6, 7] was used to solve delay constrained tree optimization problem. The algorithm started by computing a least delay tree rooted at a source and spanning all group members and iteratively replaced the superedges in the tree by cheaper superedges not in the tree. The paper [8] summarized a tradeoff algorithm between the minimum Steiner tree and the least delay tree. Barathkumar and Jaffe [9] studied the algorithms to optimize the cost and delay of the routing tree. Rouskas and Baldine [10] studied the problem of constructing multicast trees subject to both delay and delay jitter constraints. Salama et al. [11] compared the performance of shortest path broadcast tree algorithm and a heuristic for tree cost, delay and delay jitter bound.

The Hopfield Neural Network (HNN) is a candidate for providing the solution to constrained optimization problem. The HNN is a parallel, distributed information processing structure consisting of many processing elements connected via weighted connections [12]. The objective function was then expressed as quadratic energy function and the associated weights between neurons were computed using the gradient descent of energy function. The formulation of energy function associated with HNN for shortest path computation was first proposed by Ali and Kamoun [13]. HNN has been used for computation of shortest path for routing in computer networks and communication systems [13-15]. The HNN was used to solve QoS Constrained real time multicast routing [16-19]. The optimization of multicast tree using HNN with delay and delay jitter has been reported in [19].

An algorithm to obtain the delay bound minimum cost multicast tree using Hopfield neural network is presented in this paper. The delay bound shortest paths from source to various destinations are obtained recursively, which are combined using union operator to ensure that a link is participating only once in the multicast tree. The effectiveness of the developed program is tested for undirected weighted 14-node network for obtaining unconstraint and delay constraint optimum multicast trees for different sets of source and destinations.

II. Delay Bound Multicast Routing

The network is simply represented as weighted connected network G=(N,E), where N denotes the set of nodes (vertices) and E denotes the set of links (arcs). Let M is a subset of N i.e. $M \subseteq N$ forms the multicast group with each node of M is a group member. The node $s \in N$ is a multicast source for multicast group M. A multicast tree $Tr(s,M) \subseteq E$ is a sub-graph of G that spans all nodes in M, while it may include non-group member nodes along a path in the tree. The link between nodes i and j, $e=(i,j) \in E$ has its properties, cost Cij and delay Dij as real positive values R+. The link cost Cij may be the monitory cost incurred by the use of the network link or may be some measure of network utilization. The cost coefficients Cij for the nonexistent arcs are defined as infinity. The delay Dij represents the time needed to transmit information through link that includes transmission,

Manuscript received October 3, 2009.
Sushma Jain is with Computer Science & Engineering Department, Thapar University, Patiala, India-147004, (e-mail: sjain@thapar.edu).
J.D. Sharma is with Electrical Engineering Department, Indian Institute of Technology, Roorkee, India-247667, (e-mail: jaydsfee@iitr.ernet.in).

queuing and propagation delays.

Delay bound multicast routing problem can be defined as to find a tree rooted at the source s and spanning to all the member of M such that the total cost of the links of the tree is minimum and the delay from source to each destination is not greater than the required delay constraint . Therefore, the end-to-end delay bound multicast routing problem is defined as –

Minimize cost of multicast tree

$$Cost(T_r(s,M)) = \sum_{i,j \in T_r(s,M)} C_{ij} \qquad (1)$$

Subjected to

End-to-end delay constraint

$$D_m(P_T(s,m)) = \sum_{i,j \in P_T(s,m)} D_{ij} \le \Delta \quad \text{for m=1,M} \qquad (2)$$

The PT(s,m) is a path between source node s and the destination node m. The path PT(s,m) is an ordered sequence of nodes connecting from source node s to destination node m, indicating the traverse of data from ‘s’ to ‘m’ as-

(s → i → j → k …..→ r → d).

III. Mapping Multicast Routing Using HNN

The computational circuit, the neuron dynamics and the energy function of Hopfield neural network are briefed in Appendix. The delay bound multicast tree using HNN is obtained by recursively obtaining the delay bound shortest paths from source to each destination in the multicast group and combining them by the union operator. The union operator ensures that a link is appearing only once in the multicast tree. This methodology is explained through the following steps –

- Mapping shortest path problem
- Mapping delay bound in shortest path
- Dynamics of the Hopfield neural network
- Union of shortest paths between various destinations

A. Mapping shortest path problem

The total cost associated with path PT(s,m) is therefore expressed as –

Csm = Csi + Cij+ Cjk+…….+Crm (3)

The shortest path problem is aimed to find the path PT(s,m) that has the minimum total cost Csm. It can be mapped in HNN through the following procedure -

A (N×N) matrix V=[Vij] is used where N is the number of nodes. The diagonal elements of matrix V are taken as zero. The each element in the matrix is representative of a neuron described by double subscripts i and j representing the node numbers. Therefore, only N(N-1) neurons are to be computed and the neuron at the location (x,i) is characterized by its output Vxi defined as follows –

$$V_{xi}^m = \begin{cases} 1 & \text{if arc from node i to node j is in path} \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Let define ρ_{xi} as –

$$\rho_{xi} = \begin{cases} 1 & \text{if arc from node x to node i does not exist} \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

The minimization of the energy function drives the neural network into its lowest energy state, which corresponds to the shortest path solution. The energy function must favor states that correspond to valid paths between the specified source-destination pairs. Among these, it must favor the one which has the shortest path (minimum cost). An energy function satisfying such requirement is given by [13] –

$$E^m = E_1^m + E_2^m + E_3^m + E_4^m + E_5^m \qquad (6)$$

Such that,

$$E_1^m = \frac{\mu_1}{2} \sum_{x=1}^{n} \sum_{\substack{i=1 \\ i \ne x}}^{n} C_{xi} V_{xi}^m \qquad (6a)$$

$$E_2^m = \frac{\mu_2}{2} \sum_{x=1}^{n} \sum_{\substack{i=1 \\ i \ne x}}^{n} \rho_{xi} V_{xi}^m \qquad (6b)$$

$$E_3^m = \frac{\mu_3}{2} \sum_{x=1}^{n} \left\{ \sum_{\substack{i=1 \\ i \ne x}}^{n} V_{xi}^m - \sum_{\substack{i=1 \\ i \ne x}}^{n} V_{ix}^m \right\}^2 \qquad (6c)$$

$$E_4^m = \frac{\mu_4}{2} \sum_{i=1}^{n} \sum_{\substack{x=1 \\ x \ne i}}^{n} V_{xi}^m (1 - V_{xi}^m) \qquad (6d)$$

$$E_5^m = \frac{\mu_5}{2} (1 - V_{ms}^m) \qquad (6e)$$

Where,

µ1 weight to force the minimum cost of the path by accounting cost of existing links

µ2 weight to prevent the nonexistent links being included in the chosen path

µ3 weight to ensure that if a node has been entered in, it will also be exited

µ4 weight to force the state of neural network to converge to one of the corner of the hypercube defined by $V_{xi} \in \{0, 1\}$

µ5 weight to enforce the construction of path originating at source s and terminate at destination m.

B. Mapping delay bound in shortest path

A new energy term E_6^m, where end-to-end delay constraint is referred as penalty, can be added for delay bound shortest path. This energy term is therefore expressed as –

$$E_6^m = \frac{\mu_6}{2} \sum_{x=1}^{n} \sum_{\substack{i=1 \\ i \ne x \\ (x,i) \ne (m,s)}}^{n} DL_{xi}^m V_{xi}^m \le \Delta \qquad (7)$$

Where,

μ6 weight to enforce that the delay of the constructed path is less or equal to specified delay constraint.

This inequality constraint can be taken care by converting inequality to equality constraint or by linear programming (LP) neuron [16,20]. When the delay of the path is greater than delay constraint, the LP neuron penalizes all the neurons in the corresponding matrix. This neuron contributes only when constraint is violated. The transfer function h(z) of the LP type neuron [16] is given as -

$$h(z) = \begin{cases} 0 & if\ z \le 0 \\ z & otherwise \end{cases} \tag{8}$$

Where,

$$z = \sum_{x=1}^{n} \sum_{\substack{i=1 \\ i \ne x \\ (x,i)\ne(m,s)}}^{n} DL_{xi}^{m} V_{xi}^{m} - \Delta \tag{8a}$$

The delay constraint term using LP-type neurons can be defined as

$$E_6^m = H(z) = \int h(z)d(z) \tag{9}$$

The total energy function for mth destination E^m including delay bound is therefore defined as -

$$E^m = E_1^m + E_2^m + E_3^m + E_4^m + E_5^m + E_{6,LP}^m \tag{10}$$

Using eqn (6) and (9),

$$E^m = \frac{\mu_1}{2}\sum_{x=1}^{n}\sum_{\substack{i=1 \\ i\ne x}}^{n} C_{xi}V_{xi}^m + \frac{\mu_2}{2}\sum_{x=1}^{n}\sum_{\substack{i=1 \\ i\ne x}}^{n} \rho_{xi}V_{xi}^m + \frac{\mu_3}{2}\sum_{x=1}^{n}\left\{\sum_{\substack{i=1 \\ i\ne x}}^{n} V_{xi}^m - \sum_{\substack{i=1 \\ i\ne x}}^{n} V_{ix}^m\right\}^2 + \frac{\mu_4}{2}\sum_{i=1}^{n}\sum_{\substack{x=1 \\ x\ne i}}^{n} V_{xi}^m(1-V_{xi}^m) + \frac{\mu_5}{2}(1-V_{ms}^m) + \frac{\mu_6}{2}H(z) \tag{11}$$

C. Dynamics of the Hopfield neural network

The Hopfield dynamic equations are described in Appendix as eqns. (A.1, A.3, A.4). These equations are rewritten herewith to account the double subscript representation as –

$$V_{xi}^m = g_{xi}^m(U_{xi}^m) = \frac{1}{1+e^{-\lambda_{xi}^m U_{xi}^m}} \tag{12}$$

$$\frac{dU_{xi}^m}{dt} = \sum_{y=1}^{n}\sum_{\substack{j=1 \\ j\ne y}}^{n} T_{xi,yj}V_{yj}^m - \frac{U_{xi}^m}{\tau} + I_{xi} \tag{13}$$

$$\frac{dU_{xi}^m}{dt} = -\frac{U_{xi}^m}{\tau} - \frac{\partial E}{\partial V_{xi}^m} \tag{14}$$

Substituting the energy expression i.e. eqn. (10) in the eqn. (14) results into the expression as –

$$\frac{dU_{xi}^m}{dt} = -\frac{U_{xi}^m}{\tau} - \frac{\mu_1}{2}C_{xi}(1-\delta_{xm}\delta_{is}) - \frac{\mu_2}{2}\rho_{xi}(1-\delta_{xm}\delta_{is}) - \mu_3\sum_{\substack{y=1 \\ y\ne x}}^{n}(V_{xy}^m - V_{yx}^m) + \mu_3\sum_{\substack{y=1 \\ y\ne i}}^{n}(V_{iy}^m - V_{yi}^m) - \frac{\mu_4}{2}(1-2V_{xi}^m) + \frac{\mu_5}{2}\delta_{xm}\delta_{is} \tag{15}$$

The δ is the Kronecker delta defined by –

$$\delta_{ab} = \begin{cases} 1 & if\ a = b \\ 0 & otherwise \end{cases} \tag{15a}$$

By comparing the connection strengths, the biases $T_{xi,yj}$ are derived as –

$$T_{xi,yj} = \mu_4\delta_{xy}\delta_{ij} - \mu_3\delta_{xy} - \mu_3\delta_{ij} + \mu_3\delta_{jx} + \mu_3\delta_{iy} \tag{16}$$

The first term represents the excitatory self-feedbacks and the second and third terms represent local inhibitory connections among neurons in the same row and in the same column respectively. The last two terms represent excitatory cross connections among neurons.

$$I_{xi} = -\frac{\mu_1}{2}C_{xi}(1-\delta_{xm}\delta_{is}) - \frac{\mu_2}{2}\rho_{xi}(1-\delta_{xm}\delta_{is}) - \frac{\mu_4}{2} + \frac{\mu_5}{2}\delta_{xm}\delta_{is} = \begin{cases} \frac{\mu_5}{2} - \frac{\mu_4}{2} & if\,(x,i) = (m,s) \\ -\frac{\mu_1}{2}C_{xi} - \frac{\mu_2}{2}\rho_{xi} - \frac{\mu_4}{2} & otherwise \end{cases} \tag{17}$$

D. Union of shortest paths between various destinations

The V^m obtained above represents the output matrix of unicast route or shortest path from source s to destination m. Final output of multicast tree is obtained by the union of unicast routes from source to various destinations. The element V_{xi} in multicast tree is obtained as -

$$V_{xi} = V_{xi}^1 \cup V_{xi}^2 \cup V_{xi}^3 \cup \ldots\ldots\ldots\ldots \cup V_{xi}^m \tag{18}$$

With this implementation, there will be only one entry for a link even if it exists in multiple unicast routes. This is explained with the help of an example graph shown as Fig. 1 where the source node is '0' and the destination nodes are {3,4}.

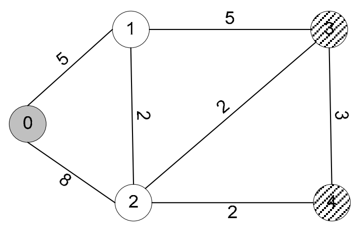

Fig. 1 An example graph with source node '0' and destinations (3,4)

The unicast paths for each of the destination nodes (3,4) are explained through matrix shown in Table 1(a) and Table 1(b) respectively. The resulting matrix of the multicast tree after the union is shown in Table 1(c). The links between nodes (0,1) and (1,2) appearing in both the paths PT(0,3) and PT(0,4) are accounted only once in multicast tree.

TABLE 1(A) OUTPUT MATRIX OF DESTINATION '3'

N o					
0					
1					
2					
3					
4					

TABLE 1(B) OUTPUT MATRIX OF DESTINATION '4'

N o					
0					
1					
2					
3					
4					

TABLE 1(C) UNION OF MATRICES FOR DESTINATIONS '3' AND '4'

N o					
0					
1					
2					
3					
4					

IV. RESULTS AND DISCUSSION

Initially the input of neurons U_{xi} are assumed zero and the state of the neural network is simulated by the solution of N(N-1) differential equation for each destination m ∈ M in multicast group and output of neuron is taken as V_{xi}^{m}. For faster convergence and to prevent the neural network from adopting an undesirable state, very small noise $-0.0001 \le \delta U_{xi} \le +0.0001$ is added to the initial input U_{xi}. Proposed model is simulated using fourth order Runge-Kutta method. The convergence of the algorithm and the outcome depends on the proper tuning of μi, which is one of the hardest task in HNN. The following criteria is used to select these coefficients-

$2\mu_3 - \mu_4 > 0$, $\mu_2 = \mu_5$, $\mu_1 >> 2\mu_3 / (C_{xi})_{max}$, $\mu_4 = \mu_6$, $\mu_1 < \mu_6$ and $\mu_3 > \mu_4 (C_{xi})_{max}$

The following values of coefficient are selected for the simulation

$\mu_1 = 100$, $\mu_2 = 5000$, $\mu_3 = 3500$, $\mu_4 = 250$, $\mu_5 = 5000$, $\mu_6 = 250$

When the difference between the outputs of the neurons in two consecutive iterations is less than 0.00002, it is assumed that stable state is attained. The algorithm is tested to obtain optimum multicast trees for different sets of source and destinations on 14 node weighted connected network, as shown in Fig. 2.

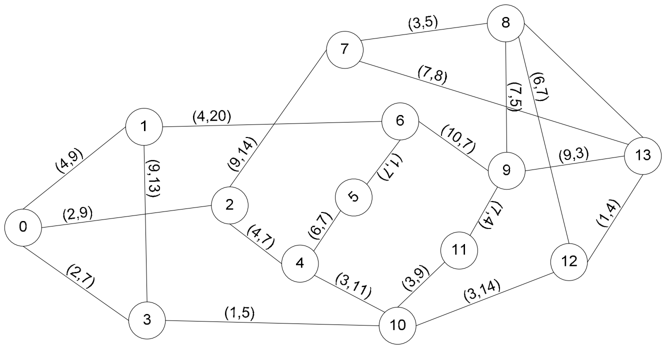

Fig. 2. The 14 node weighted connected network

The unconstrained and delay constrained multicast trees are obtained for the source at '0' and destinations {5,9} and correspondingly the results are shown in Fig. 3 and Fig. 4 respectively.

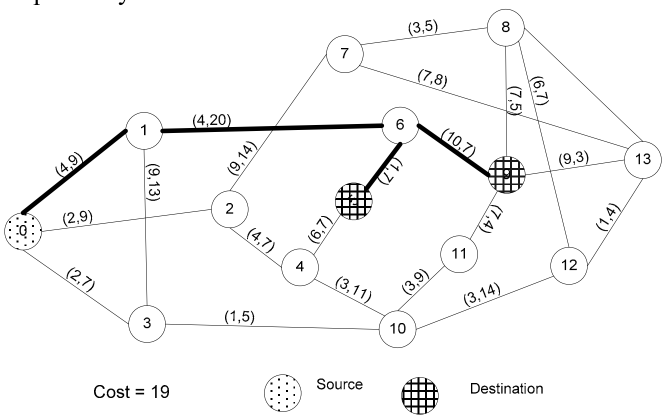

Fig. 3. Unconstraint multicast tree for destinations {5,9} using HNN

The delay bound is taken as 25. As expected, the cost with the unconstrained multicast tree will be lower than the cost with the delay bound multicast tree. This is mainly because the shortest paths between source and destinations for unconstrained tree are different than the respective shortest paths under the delay bound constrained.

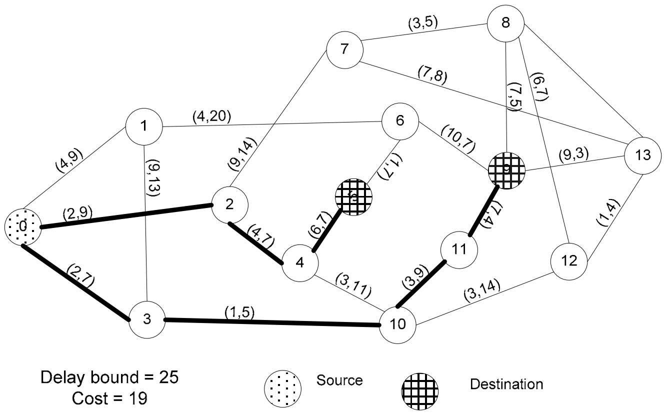

Fig. 4. Delay constrained multicast tree for destinations {5,9} using HNN

The obtained delay constrained multicast tree for destinations {8, 9, 12} with delay bound as 30, is shown in Fig. 5.

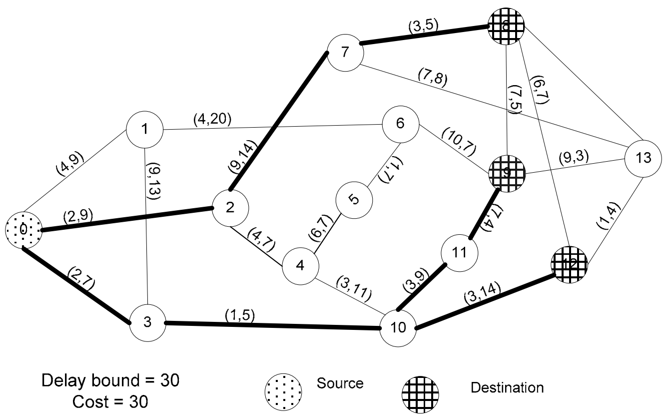

Fig. 5. Delay constrained multicast tree for destinations {8,9,12} using HNN

The results for different sets of source and destinations on 14-node sample graph for specified delay constraint of '35' are summarized in Table 2. As the number of destinations increase, the iterations and execution time also increase. The cost of multicast tree however depends on the relative position of source and destinations.

TABLE 2 RESULTS FOR DIFFERENT SETS OF SOURCE AND DESTINATIONS FOR 14 NODE GRAPH WITH DELAY CONSTRAINT '35'

Source	*Set of destination*	*Path in multicast tree*	*Cost*	*Iterations*	*Time in ms*
0	{4,12}	0→3→10→4 0→3→10→12	9	5180	3.00
6	{3,9}	6→5→4→10→3, 6→9	21	5583	3.73
1	{5,8,12}	1→6→5, 1→6→9→8 1→3→10→12	35	15002	10.2
0	{6,8,11,12}	0→1→6 0→3→10→12→8 0→3→10→11 0→3→10→12	23	18077	15.2

Few statistics for obtaining the multicast tree on random graphs are summarized in Table 3. As the graph size is increasing, the higher amount of delay bound is specified which is needed for obtaining the feasible multicast tree. The iterations needed for convergence also increase with the increase in the size of the graph.

TABLE 3 RESULTS FOR RANDOM GRAPHS

Number of nodes	*Number of links*	*Delay bound*	*Iterations*
5	5	5	589
10	25	15	2438
15	59	25	4078
20	100	30	5037

V. CONCLUSION

An approach based on Hopfield neural network is presented to obtain minimum cost delay constrained multicast tree. The multicast tree is obtained by recursively obtaining the delay bound shortest path from source to various destinations and combining them by union operator. The union operator ensures that a link is appearing only once in the multicast tree. The effectiveness of the developed algorithm is tested to obtain optimum multicast trees for different sets of source and destinations on 14-node weighted connected network. As the number of destinations is increasing, the iterations needed for convergence also increase. The iterations also increase for the increase in the size of the random graph.

APPENDIX

The computational circuit of Hopfield neural network [12] is shown in Fig. A.1. The each neuron is modeled as operational amplifier with sigmoid monotonically increasing function g_i relating the output V_i to input U_i. The V_i takes value between 0 and 1.

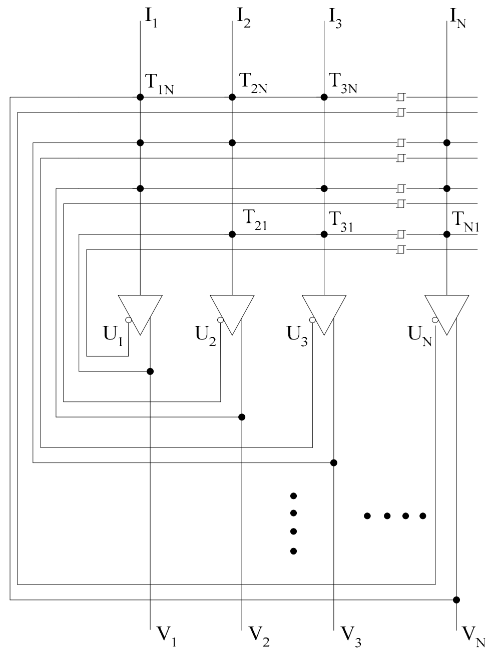

Fig. A.1 The computational circuit of Hopfield Neural Network

Function g_i is expressed as –

$$V_i = g_i(U_i) = \frac{1}{1+e^{-\lambda_i U_i}} \tag{A.1}$$

Where, λ_i is the gain of amplifier for ith neuron.

The each neuron is connected to other neurons and the connection among neurons is fully described by the interconnection matrix $\mathbf{T} = [T_{ij}]$. Each neuron receives an external bias input current I_i representing user defined input to neural network. The dynamics of the neural network is therefore described by the following equation -

$$\frac{dU_i}{dt} = \sum_{j=1}^{N} T_{ij} V_j - \frac{U_i}{\tau} + I_i \tag{A.2}$$

Where, τ is the time constant of the circuit and N is the number of neurons.

For symmetric interconnection matrix T and sufficiently high gains of amplifier ($\alpha_i \rightarrow \infty$), the dynamics follow a gradient descent of quadratic energy function, which is expressed as –

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} I_iV_i \qquad \text{(A.3)}$$

Therefore, the dynamics of ith neuron is describe as -

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} - \frac{\partial E}{\partial V_i} \qquad \text{(A.4)}$$

The minima of the energy function occurs at 2N corners in N-dimensional hypercube defined by $V_i \in \{0,1\}$

REFERENCES

[1] L. Kou, G. Markowsky, and L. Berman, “A fast algorithm for Steiner trees", Acta Informatica, vol. 15, no. 2, pp. 141-145, 1981.
[2] H. Takahashi and A. Matsuyama, “An approximate solution for the Steiner problem in graphs”, Mathematica Japonica, vol. 24, no. 6, pp. 573-577, 1980.
[3] V. R. Smith, “The computation of nearly minimal Steiner trees in graphs”, International Journal of Mathematical Education in Science and Technology, vol. 14, no. 1, pp. 15-23, 1983.
[4] V. Kompella, J. Pasquale, and G. Polyzos, “Multicast routing for multimedia communication”, IEEE/ACM Transactions on Networking, vol. 1, no. 3, pp. 286-292, 1993.
[5] C. Noronha and F. Tobagi, “Optimum routing of multicast streams”, Proceedings of IEEE INFOCOM '94, pp. 865-873, 1994.
[6] Q. Zhu, M. Parsa, and J. G. L. Aceves, “A source-based algorithm for delay-constrained minimum-cost multicasting”, Proceedings of IEEE INFOCOM '95, pp. 377-385, 1995.
[7] M. Parsa and Q. Zhu, “An iterative algorithm for delay-constrained minimum-cost multicasting”, IEEE/ACM Transactions on Networking, vol. 6, no. 4, pp. 461 – 474, 1998.
[8] L. Wei and D. Estrin, “The trade-offs of multicast trees and algorithms”, Proceedings of the Third International Conference on Computer Communications and Networking (IC 3N '94), pp. 17-24, 1994.
[9] K. B. Kumar and J. Jaffe, “Routing to multiple destinations in computer networks”, IEEE Transactions on Communications, vol. 31, no. 3, pp. 343-351, 1983.
[10] G. N. Rouskas and I. Baldine, “Multicast routing with end-to-end delay and delay variation constraints”, Proceedings of IEEE INFOCOM '96, pp. 353-360, 1996.
[11] H.. F. Salama, D. S. Reeves and Y. Viniotis, “Evaluation of multicast routing algorithms for real-time communication on high-speed networks”, IEEE Journal on Selected Areas in Communications, vol. 15, no. 3, pp. 332 – 345, 1997.
[12] J. J. Hopfield and D. W. Tank, “Neural computation of decisions in optimization problems”, Biological Cybernetics, vol. 52, pp 141-152, 1985.
[13] M. K. 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 941-953, 1993.
[14] H. E. Rauch and T. Winarske, “Neural network implementation of the shortest path algorithm for routing communication traffic”, IEEE Computer System Magazine, pp. 26-30, 1988.
[15] L. Zhang and S. C. A. Thomopoulos, “Neural network implementation of the shortest path algorithm for traffic routing in communication network”, Proceedings of International Conference on Neural Networks, 1999.
[16] C. Pornavalai, G. Chakraborty and N. Shiratori, “A neural network approach to multicast routing in real-time communication networks”, Proceedings of International Conference on Network Protocols, pp. 332 – 339, 1995.
[17] P. Venkataram, S. Ghosal and B. P. Vijaykumar, “Neural network based optimal routing algorithm for communication networks”, Neural Networks, vol. 15, no. 10, pp. 1289-1298, 2002.
[18] T. L. Hemminger, C. Coulston and C. A. Pomalaza-Raez, “Routing strategies for multicast packet radio networks”, International Journal of Smart Engineering System Design, vol. 4, no. 3, 2002.
[19] A. Ghanwani, “Neural and delay based heuristics for the Steiner problem in networks”, European Journal of Operational Research, vol. 108, no. 2, pp. 241-265, 1998.
[20] D. W Tank and J. J. Hopfield, “Simple neural optimization technique: An A/D converter, signal decision circuit, and a linear programming circuit”, IEEE Transactions on Circuits and Systems, vol. 33, no. 5, pp. 533-546, 1986.

Sushma Jain is working as Lecturer in Computer Science & Engineering Department, Thapar University, Patiala, India and pursuing Ph.D in area of routing algorithms for computer networks. The research interest includes computer network communication, evolutionary algorithms and neural network.

J.D. Sharma received the B.E. from Jiwaji University, Gwalior, India, in 1968, and the M.E. and Ph.D. from the University of Roorkee (UOR), Roorkee, India, in 1971 and 1974, respectively. He is working as Professor in Department of Electrical Engineering, Indian Institute of Technology Roorkee, India since 1985. He is actively involved in research in the area of power system planning and operation, security analysis and optimization, small hydro plant design and simulation, use of Artificial Intelligent Techniques. He has many research papers in journals of repute.