A Neural Network Based Multi-Destination Routing Algorithm for Communication Network

Dong-Chul Park and Seung-Eok Choi
School of Electrical and Electronic Engineering
MyongJi University
parkd@icrl.myongji.ac.kr

Abstract— **A routing algorithm for large scale communication network with multi-destination problem is proposed in this paper. The proposed approach consists of three procedures : recursive Hopfield neural network for obtaining the routing order between given source and multiple destinations, a screening procedure for localizing the problem and minimizing the computational effort along with Depth-First Search method, and an improved version of Hopfield neural network for routing in the large scale communication networks. The results show improvements in both computational performance and solution optimality by the proposed approach over conventional approaches.**

I. Introduction

Recently with the increasing availability of different switching network technologies, a special class of multicast communication problems has been brought forward to the stage of recent research interest[1]-[3]. For those problems, they can be modeled as a delivery of the same message from a source node to several destination nodes in a connected communication network with constraints, such as bandwidth, link capacity, delay bound or traffic load of the connected network between each other systems. For this reason, the area of network routing has been the subject of intensive research for many years. A good routing algorithm should try to find the optimum path(s) for data(or messages) transmission within a very short time so as to satisfy the user's demand for a fast service. In this paper, we focus on the optimum routing problem with multiple destinations in a large scale network to take an increasingly important role in a communication networks, where the goal is to minimize the network-wide average link load. The use of neural networks to find the shortest path between a given source-destination pair was first introduced by Rauch and Winarske[4]. The approach by Rauch and Winarske has a constraint to have a minimum number of nodes forming the path in terms of neural network architecture. Zhang and Thomopoulos[5] proposed another approach that makes it possible to visit all the nodes in the networks. Recently, Ali and Kamoun[6] introduced an adaptive algorithm for the network structure with the topological information. However, Ali and Kamoun's algorithm[6] has its own limitation to the size of the network because the algorithm does not converge even a network with more than 20 nodes. In this paper, a novel approach for dealing with the optimal routing problem that can be used in practically large size network is proposed.

The problem of routing according to the destination set is defined in Section II. Section III summarizes the recursive Hopfield neural network for large scale network model. Section IV introduces the screening procedure using Depth-First Search method to reconstruct the given network. Section V explains the development of an improved routing algorithm based on Hopfield neural network. In Section VI, the proposed neural network algorithm and several conventional algorithms are applied to a large scale problems and the results are compared. The conclusion in Section VII summarizes the main results of this paper.

II. Problem Definition

With the definition of an undirected graph the underlying topology of the communication networks can be given[7] with $\mathbf{G} = (\mathbf{V},\mathbf{E})$, where $\mathbf{V}$ is the set of vertices(N nodes), and $\mathbf{E}$ is the set of its edges, a link cost matrix $\mathbf{L}= [L_{ij}]$ where L_{ij} is a cost from node i to node j, s is the source node, and d is the destination node. For multiple destination problem, define d_x as the set of destinations where $d_x \in \mathbf{V}$, and $x = \{1, \cdots, N\} \in \mathbf{X}$.

For each link (i,j), there exists a nonnegative number L_{ij}, called cost including time delay, bandwidth, and traffic load of the link from node i to node j. If there is no link from node i to j, L_{ij} is set to a very large value in order to exclude it from the routing path. Note that link(i,j) is symmetric with link (i,j), $L_{ij} = L_{ji}$ since the network is given by an undirected graph.

If we define an undirected path P^{sd_x} with respect to multiple destinations d_x as follows :

$$P^{sd_x} = (s, a, b, \cdots, d_1, \cdots, d_2, \cdots, i, d_x)$$

where the routing is $(s \rightarrow a \rightarrow b \rightarrow \cdots \rightarrow d_1 \rightarrow \cdots d_2 \rightarrow \cdots \rightarrow i \rightarrow d_x)$. Then the total cost TC_{sd_x} of this problem becomes : $TC_{sd_x} = L_{sa} + L_{ab} + \cdots + L_{id_x}$.

0-7803-4859-1/98 $10.00©1998 IEEE

To formulate the above mentioned problem, a $(n \times n)$ matrix with setting all diagonal elements to be zero is used where n is the number of nodes. Each element in the matrix is represented by a neuron which is denoted by two indices(i, j). The state of a neuron, V_{ij}, is defined as following :

$$V_{ij} = \begin{cases} 1, & \text{if the link from node } i \text{ to node } j \text{ exists} \\ & \text{in solution.} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

A matrix Γ=$[\gamma_{ij}]$ is defined for the link connection (or edge) as follows :

$$\gamma_{ij} = \begin{cases} 1, & \text{if the link from node } i \text{ to node } j \text{ does} \\ & \text{not exist.} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

By using the above definitions, the problem of multiple destinations can be represented by the logical algebraic expressions of constrained combinatorial optimization problem as follows :

$$Minimize \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} L_{ij} V_{ij} \quad \text{for all } (s \to d_x) \tag{3}$$

$$Subject\, to \left(\sum_{\substack{j=0 \\ j \neq i}}^{N-1} V_{ij} - \sum_{\substack{k=0 \\ k \neq i}}^{N-1} V_{ki} \right) = \begin{cases} 1, & \text{if } i = s \\ -1, & \text{if } i = d_x \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

III. Hopfield Neural Network for Routing Order

The use of neural networks to solve constrained optimization problems was first introduced by Hopfield and Tank[8][9] to derive good solutions to discrete combinatorial optimization problems and demonstrate the computational power of their network by applying their model to the Traveling Salesman Problem(TSP)[9]-[13]. In this paper, Hopfield Neural Network(HNN) is also employed to find the routing order between given source and destinations. By modifying the energy function for HNN, we try to find an optimal routing order for multiple destinations which means one source and multiple destinations in large scale communication network.

The main advantage of the recursive Hopfield-Neural Network[13] for finding routing order is that it is capable of escaping from the local minimum solution. Recent approaches to solve large scale TSP using HNN[13][14] give us some insights how to approach the large scale communication routing problem ; decompose the given large scale problem into several small problems, solve isolated several small problems first, and then solve the full scale problem with isolated small solution.

IV. Screening Procedure

Most of conventional routing algorithms have fixed number of nodes in neural network for a given problem. Lee and Chang first introduced a routing algorithm which is able to decompose a large scale problem into several small problems by using dependency factors[15] for unreliable network components. However, the dependency factors have to be changed from problem to problem. Since it is difficult to adjust the dependency factors for different networks, a general procedure, called screening procedure, to reduce the scale of the problem without loosing the optimality of the solution is proposed in this paper.

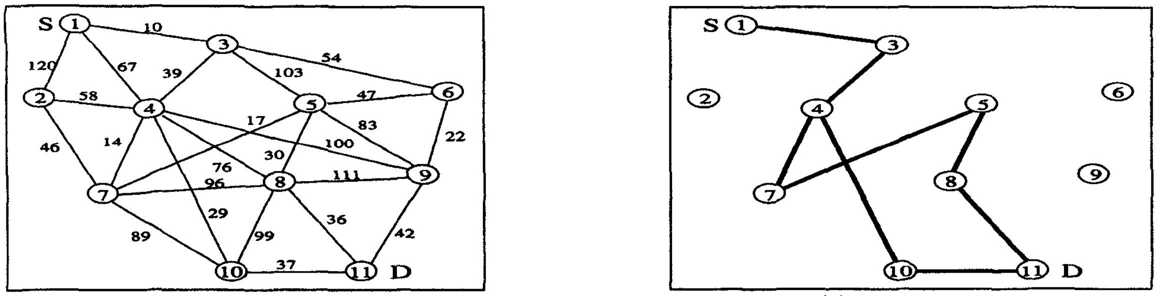
Let us consider the link cost L_{xi} and the number of nodes, N, where the link cost L_{xi} is not a geometrical distance but a quantity of physical load. Also assume that all the link capacities be the same. Before the screening procedure, find the minimum link cost L_{min}, the maximum link cost L_{max}, and the difference between L_{max} and L_{min}. Starting with the L_{min}, a threshold value, *Screening Load*(L_{sl}) is applied to the link costs in the network to eliminate any links above the given threshold value. The threshold value is increased until a valid path from the source to a destination exists regardless of the optimality of the route. The threshold value L_{sl} at n-th trial is defined as:

$$L_{sl}(n) = L_{min} + \Delta L \times (n-1) \tag{5}$$

$$\Delta L = \frac{1}{n}(L_{max} - L_{min}) \tag{6}$$

In order to find any valid paths during the screening procedure until step n $(2 \leq n < N-1)$, a graph search method called Depth-First Search(DFS)[16] is employed. The DFS is similar to Minimum Spanning Tree(MST)[17]-[20], if it is performed on a graph to avoid forming cycle(s) taking into account of the link cost. However, it can apply DFS method to the cyclic network because it is not to find the overall path for all the nodes such as a tree structure, but to find all the valid network nodes except the heavy loading node in terms of the cost quantity in this paper. A pseudo-code representing the screening process is shown below.

For example, given a 11-node problem shown in Fig.1-(a) with source node #1 and destination #11. In this example, L_{min}, L_{max}, step n, and the threshold value ΔL are $L_{13} = 10$, $L_{12} = 120$, $(2 \leq n < 10)$, and 10 respectively. After applying the screening procedure 4 times(n = 3) with DFS, the result of Fig. 1-(b) is obtained. Note that DFS results in eliminating



(a) Given Example network #1

(b) Screening result

Fig. 1. Screening procedure

```
Find L_max, L_min, and ΔL
   for n = 2 to maximum (N − 1) do
       Screen all the costs
       Apply DFS to the screened costs
       Check a network whether decomposed or not
   endfor
end
```

node #2 and the link between #6 and #9 that isolated ones from the path.

By using the screening procedure, the proposed routing neural network can be organized with $(n-m)\times(n-m)$ matrix $(0 \leq m < n)$ while $(n\times n)$ matrix is required in conventional network. As shown in Fig. 1, it needs $64(=8^2)$ neuron units while other conventional methods need $121(=11^2)$ neuron units to formulate this problem with HNN.

V. Improved Routing Algorithm using Hopfield Neural Network with modified Energy Function

In order to obtain improved routing algorithm with HNN, a modified energy function for routing algorithm is proposed in this paper. The modified energy function is intended to yield more stable result over Ali and Kamoun's energy function[6]. This stable state should correspond to the optimal cost routing from given source to destinations. A suitable energy function proposed in this paper for the optimal routing can be written as

$$
\begin{aligned}
E &= \frac{A}{2}\sum_{i=0}^{N-1}\sum_{\substack{j=0\\ j\neq i}}^{N-1} L_{ij}V_{ij} + \frac{B}{2}\sum_{i=0}^{N-1}\sum_{\substack{j=0\\ j\neq i}}^{N-1} \gamma_{ij}V_{ij} \\
&+ \frac{C}{2}\sum_{i=0}^{N-1}\left\{\sum_{\substack{j=0\\ j\neq i}}^{N-1} V_{ij} - \sum_{\substack{j=0\\ j\neq i}}^{N-1} V_{ji} - \phi_i\right\}^2 \qquad (7)\\
&+ \frac{D}{2}\sum_{i=0}^{N-1}\sum_{\substack{j=0\\ j\neq i}}^{N-1} V_{ij}(1-V_{ij}) + \frac{F}{2}\sum_{i=0}^{N-1}\sum_{\substack{j=0\\ j\neq i}}^{N-1} V_{ij}V_{ji}
\end{aligned}
$$

where ϕ_i is defined as following :

$$
\phi_i = \begin{cases} 1, & \text{if } i = s \\ -1, & \text{if } i = d_x, \text{ for all } i \in \{1,\cdots,N\},\ x \in \mathbf{X}, \\ & \text{where } \mathbf{X} \text{ is the set of destination} \\ 0, & \text{otherwise} \end{cases} \qquad (8)
$$

where n is the number of destination node.

In Eq.(7), the A term minimizes the total link cost of a routing by taking into account the cost of existing links, the B term prevents nonexistent links from being included in the chosen routing path, the C term is zero if the constraints is satisfied, (meaning if for every node in the routing solution), and the number of incoming direction links equals the number of outgoing direction links, the D term enforce the state of the neural network to converge to a valid routing, and the F term, newly introduced in this paper, makes sure that the flow vector $\mathbf{V}=[V_{ij}]$ is one directional for each destinations. It means that $V_{ij}V_{ji}$ is always zero for all (i,j) when flow vector $\mathbf{V}=[V_{ij}]$ is only one directional when compared with the energy function of Ali and Kamoun. The energy function with F term gives much faster convergence. Note that the C term is used to make it sure that if a node entered into other node also its node will be excited by a routing path. The B term D term also enforce the state of the neural network to converge to a valid routing. The output function of the proposed neural network is given by the sigmoid function.

As shown in Eq. (7), an advantage of this model is that it maps the link costs into the biases rather than into the neural interconnections. It is also a flexibility reflected by the fact that the link cost $\mathbf{L}=[L_{ij}]$ and the network topological information $\Gamma=[\gamma_{ij}]$ can be changed through the biases. In other words, the interconnection weights do not depend on a particular source or destination. This will make the proposed neural network routing algorithm very attractive to operate in real time. Additionally, after finding all the valid links using the screening procedure, its model is more effective to solve the multiple destinations routing problem than the conventional methods. Because the proposed neural network model is adaptive according to the changed network, it can easily find the optimal routing of the problem.

VI. Experiments and Results

A. Proposed Approach for Multiple Destinations

For enhancing the performance of the proposed routing algorithm to find a solution of the routing problem, the screening procedure makes a simplified network composed of lower link costs from the complex network and it decreases the computational time for a given network because the link connection between all the nodes are reduced by the screening procedure. After the screening procedure, it decomposes a given problem into the simplified problems. Then the proposed routing algorithm is applied to a given network after the screening procedure. And the following pseudocode describes the general procedure for the proposed approach for a problem of multiple destination routing in this paper:

Find the routing order
Initialize all the units
  for $X = 1$ **to** N *destination* **do**
    Screen for all the link costs for (S→D) or (D→D)
    Apply the routing algorithm to (S→D) or (D→D)
  endfor

The ability of the proposed routing model to reshape the energy landscape through the link cost bias term, while keeping the interactions among the neurons is a salient feature. Although there is a possibility for the neural state to get fell into a local minimum state, it will be shown that, by proper tuning of the weight coefficients, such a possibility could be reduced if not excluded. So when selecting the coefficients, large value of B ensures that the solution quality is high while large value of C guarantees that the network converges to a valid one. By taking into account of these requirements in experiments, the set of parameter is chosen as follows :

$$A = 550, B = 2550, C = 2150, D = 250, F = 1350,$$

$$\tau = 1, \Delta t = 10^{-5}, \lambda = 1$$

In order to compare the performance of the proposed algorithm with some of the conventional algorithms, Dijkstra's *Shortest Path*(SP) algorithm and Kruskal's algorithm of *Minimum Spanning Tree*(MST) for *Steiner tree problem* are applied to the same problem. And in general, Dijkstra's algorithm is not used to solve a problem concerned with combinatorial optimization problem since the computational time for solving a large scale problem is highly increased by considering a number of given nodes. Then the Kruskal's algorithm of MST to be used to find a routing in the multicast data communication is compared with other algorithms including the proposed algorithm. However, it has some drawbacks ; First, all the nodes in the network must participate in the execution of algorithm. This may be impractical in a large scale communication network with sparse multiple destinations. Second, it does not consider the overall link cost but the gradual link cost as the theoretical upper link bound in a given network. Therefore it can be easily trapped in an incorrect routing solution[20].

B. Routing problem with 20 nodes

In order to show the performance of the proposed algorithm for the routing problem of multiple destinations, an exemplar problem shown in Fig. 2 is used. In Fig. 2, the numbers on line represent link costs. The source is given as node #2 and destinations are node #9, #15, and #20. First, we have to find the routing order between source(s) and destinations with respect to node-to-node distance for large scale network using recursive Hopfield neural network. In this experiment, the optimal routing order among a source and multiple destinations is found first ($S : 2 \rightarrow D1 : 9 \rightarrow D2 : 15 \rightarrow D3 : 20$). An advantage of finding the optimal routing order in the first step is that it makes easier to find the optimal routing without concerning the network. Without such a routing order among source and multiple destinations firsthand, we have to find the optimal path always with the total number of nodes(neural network size $= N^2$, where N is the number of nodes) for each routing between source and destination pair or destinations. After finding the routing order among source and destinations through the recursive Hopfield neural network, a number of node for the neural network is reduced by the screening procedure in accordance with the routing order.

The optimal routing path through the proposed routing algorithm was compared with the method introduced by Ali and Kamoun, Dijkstra algorithm, and Kruskal MST algorithm in Table I in terms of total link cost and in Fig. 3. Fig. 3-(a) shows the shortest routing path from the Dijkstra SP algorithm while Fig. 3-(b) shows the routing by Kruskal's MST method to be visited all the nodes as a tree. Fig. 3-(c) and (d) show the routing path from Ali and Kamoun's routing algorithm and the proposed algorithm respectively. As

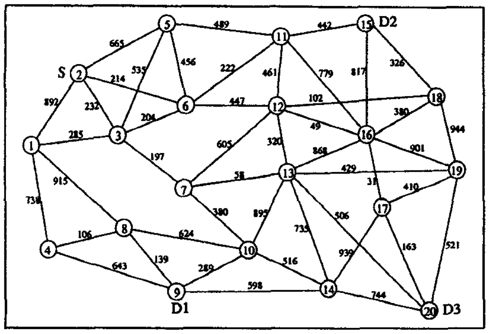


Fig. 2. Example #2 network for experiment (the numbers on lines represent link cost : $\times 10^{-4}$)

shown in Fig. 3 and Table I, the performance of the proposed algorithm is better than Ali and Kamoun's algorithm and Kruskal's algorithm which is widely employed at the computer network and the data communication. A comparison for the convergence time in terms of epoch number with the same problem is shown in Fig. 4. The routing of the proposed algorithm finds the optimal routing path more faster than Ali and Kamoun's algorithm because of the proposed neural network approach for decomposing network as shown in Fig. 4. The proposed method, on the other hand, can be run concurrently to find the individual routing between (source to destination) or (destination to destination) after finding routing order. This significantly shortens the convergence time furthermore. Similar results were obtained with 50 node problem. Fig. 5 shows the energy state in accordance with time with respect to the proposed routing model. In this experiments, the proposed routing algorithm shows both flexible and optimal for finding the results rather than other conventional models.

VII. Conclusion

A routing algorithm for large scale communication network using neural networks is proposed. The proposed approach has largely three processes ; first, the recursive Hopfield neural network is used to determine the optimal routing order among source and multiple destinations, next the screening procedure to get a simplified network for routing with multiple destinations of the network is applied, and then the improved routing algorithm based on Hopfield neural network is simultaneously applied to the simplified network to find the overall path.

The characteristics of the proposed routing model are especially suitable for large scale networks if the coefficient parameters in the energy function are properly chosen, while other conventional methods are very difficult to find the solution of the problem of multiple destinations with large number of nodes. When compared to other algorithms such as Ali & Kamoun's algorithm, Dijkstra's SP algorithm, Kruskal's MST algorithm, and the proposed algorithm in this paper shows very promising results especially for large scale problems.

$S \to D$	$2 \to 9$	$9 \to 15$	$15 \to 20$	Total
SP	1098	1475	671	3114
MST	1284	1475	671	3330
Ali's	1946	2385	1101	5342
P&C	1098	1475	671	3114

TABLE I. Total cost comparison for four methods ($\times 10^{-4}$), P&C(the proposed algorithm)

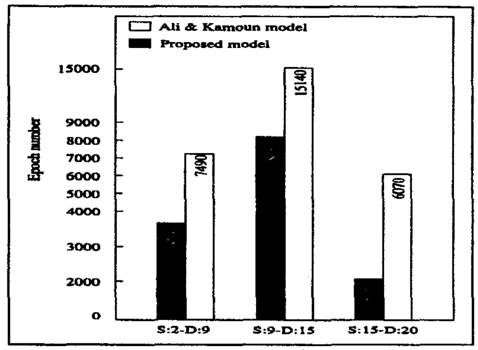


Fig. 4. Comparison of convergence time(Epoch number) between the proposed model and Ali & Kamoun's model for each ($S \to D$) pair

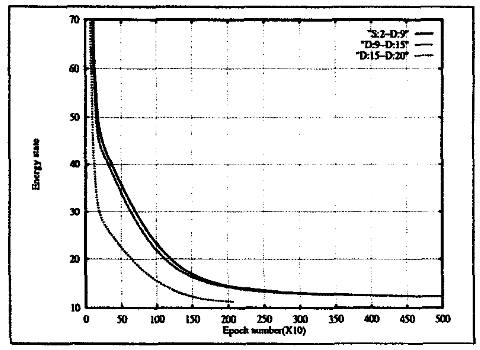
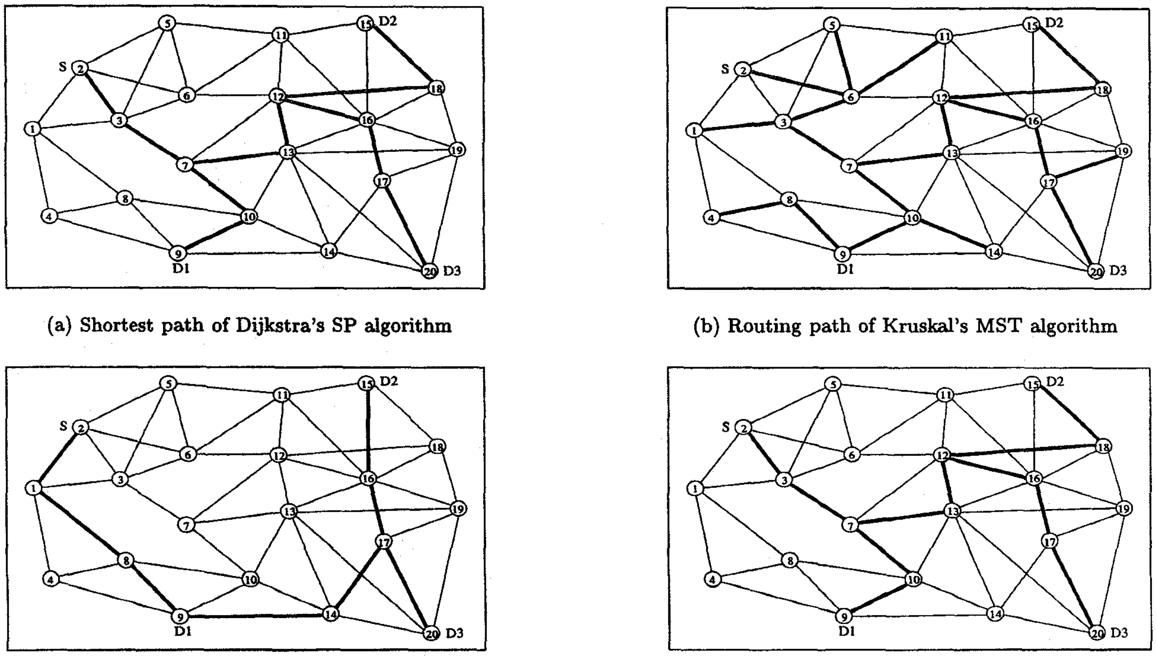


Fig. 5. Energy state change for the proposed model in Example #2 network

Acknowledgement

The research described in this paper was supported by the Korea Science and Engineering Foundation (Grant # KOSEF 951-0906-06202).



(a) Shortest path of Dijkstra's SP algorithm

(b) Routing path of Kruskal's MST algorithm

(c) Routing path of Ali and Kamoun's algorithm

(d) Routing path of the proposed neural network algorithm

Fig. 3. Routing comparison for each algorithm with the same problem

References

[1] X. L. Lin and L. M. Ni, " Multicast Communication in Microcomputer Networks," *IEEE Tr. on Parallel and Distributed Systems*, Vol. 4, No. 10, pp.1105-1117, 1993.

[2] V. P. Kompella, J. C. Pasquale, and G.C. Polyzos, " Multicast Routing for Multimedia Communication," *IEEE/ACM Tr. on Networking*, Vol. 1, No. 3, pp.286-292, 1993.

[3] K. B. Kumar and J. M. Jaffe, " Routing to Multiple Destinations in Computer Networks," *IEEE Tr. on Communications*, Vol. Com-31, No. 3, pp.343-351, 1983.

[4] H. E. Rauch and T. Winarske, " Neural Networks for Routing Communication Traffic," *IEEE Cont. Syst. Mag.*, pp.26-30, April 1988.

[5] L. Zhang and S. C. A. Thomopoulos, " Neural Network Implementation of the Shortest Path Algorithm for Traffic Routing in Communication Networks," *Proc. Int. Joint Conf. Neural Networks*, p. II-591, June 1989.

[6] M. K. Mehmet Ali and F. Kamoun, " Neural Networks for Shortest Path Computation and Routing in Computer Networks," *IEEE Tr. on Neural Networks*, Vol. 4, No. 6, pp.941-953, 1993.

[7] R. K. Ahuja, T. L. Magnati and J. B. Orlin, *Network Flows-Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[8] J. J. Hopfield and D. W. Tank, " Neural computations of decisions in optimization problems," *Biol., Cybern.*, Vol. 52, pp.141-152, 1980.

[9] D. W. Tank and J. J. Hopfield, " Simple Neural optimization networks:An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Tr. Circuits Syst.*, Vol. CAS-33, No. 5, pp.531-541, 1986.

[10] N. Funabiki and Y. Takefuji, " A Parallel Algorithm for Broadcast Scheduling Problems in Packet Radio Networks," *IEEE Tr. on Communications*, Vol. 41, No. 6, pp.828-831, 1993.

[11] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, " A Neural Network Approach to Routing Without Interference in Multihop Radio Networks," *IEEE Tr. on Communications*, Vol. 42, No. 1, pp.166-177, 1994.

[12] Y. Watanabe, K. Yoshino, and T. Kakeshita, " Solving Combinatorial Opimization Problems Using the Oscillatory Neural Network," *IEICE Trans. Inf. & Syst.*, Vol. E80-D, No. 1, 1997.

[13] D. C. Park, A. Figueras, and C. Chen, " A Hierarchical Approach for Solving Large-Scale Traveling Salesman Problem," *ICNN' 94* , Vol. 7, pp. 4613-4618, 1994.

[14] S. Noel, H. Szu, " Multiple-resolution divide and conquer neural networks for large-scale TSP-like energy minimization problems," *ICNN' 97*, Vol.2, pp.1278-1283, 1997.

[15] S. L. Lee and S. Chang, " Neural Networks for Routing of Communication Networks with Unreliable Components," *IEEE Tr. on Neural Networks*, Vol. 4, No. 5, 1993.

[16] M. A. Weiss, *Data Structures and Algorithm Analysis in C*, The Benjamin/Cummings Pub. Co. Inc., 1993.

[17] G. N. Rouskas and I. Baldine, " Multicast Routing with End-to-End Delay and Delay Variation Constraints,", *Proc. IEEE INFOCOM'96 : The Conf. on Computer Comm.*, Vol. 3c, pp.353-360, March 1996.

[18] F. Hwang and D. Richards, " Steiner tree problems," *Networks* , Vol. 22, pp.55-89, 1992.

[19] H. Mehlhorn, " A Faster Approximation Algorithm for the Steiner Problem in Graph," *Inform. Process Lett.*, Vol. 27, No. 3, pp.125-128, 1988.

[20] F. Bauer and A. Varma, " Distributed Algorithms for Multicast Path Setup in Data Networks," *IEEE/ACM Tr. on Networking*, Vol. 4, No. 2, pp.181-191, 1996.