ANALOG AND VLSI IMPLEMENTATION OF CONNECTIONIST NETWORK FOR MINIMUM SPANNING TREE PROBLEMS

H. S. Ng, K. P. Lam, and W. K. Tai

Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Abstract— **We describe a connectionist architecture which shows promise in obtaining the global optimal solution to the classical minimum spanning tree problem in a time independent of the problem size. Using commonly available analogue electronic components, a network prototype was found to give the global optimal solution within the microseconds range. Simulation results and limiting factors of the performance of analog and VLSI implementation circuits are discussed.**

I. INTRODUCTION

Owing to the fundamental nature and a broad range of applications, minimum spanning tree problems have been widely studied in the field of network flows. Minimum spanning tree (MST) problems find presence in many applications such as cluster analysis, reduced data storage, optimal message passing, etc. For example, a typical application is in designing a pipeline network to connect a number of towns using the smallest possible total length of pipeline. Efficient MST computations are also required in real-time cluster identification for computer vision tasks. There are several well-known sequential algorithms to find the minimum spanning tree for a given graph. They are Kruskal's algorithm [5], Sollin's algorithm [9] and the Prim algorithm [7]. Both Prim's and Kruskal's algorithms work with a greedy strategy when growing the spanning tree, in which one arc from a candidate list is added at each iteration. They differ in the way the candidate list is selected. Sollin's algorithm is a hybrid version of Kruskal's and Prim's. Besides these efforts, there are on-going researches in finding more efficient solutions [8], including parallel variants of the classical algorithms. However, most of these algorithms were originally developed based on a sequential computational model, and hence it is difficult to exploit parallelism effectively for efficiency improvement. In most cases, the solution time is theoretically expected to exhibit a polynomial time dependence on the problem size.

In this paper we propose a connectionist architecture based on the binary relation inference network to solve the MST problems. The architecture has been found to show promise in obtaining the global optimal solution in a time independent of the problem size. This is due to the parallel and asynchronous operating nature of the network, and each computational unit can process information asynchronously, or in the continuous-time domain, without the need of precise synchronization with other units of the network. The underlying computational model is a non-sequential but parallel one, to be defined by the specific connection pattern (or communication highway) among a set of computational units with simple and almost identical inferencing operations. Generally, the network could be implemented either in the discrete or continous-time domain. As verified by extensive simulation, the continuous-time operating ability of the binary relation inference network promises a solution time (i.e., the network's convergence time) which is independent of the problem size. However, the realization of this merit depends on the success in overcoming the limitations of different implementation platforms, which require more in-depth investigation and careful study. Two specific cases are described in this work. One is on the implementation of the inference network using commonly available analog integrated circuit components. A particular network implementation was found to give the global optimal solution within the microseconds range. The other is to consider the voltage-mode approach of using some basic analog VLSI building blocks in constructing the inference network. For the sake of reference, the minimum spanning tree problem and the basic Bellman-Ford architecture of the binary relation inference network are first reviewed in the following section.

*This work was supported by the Hong Kong Research Grant Council under Grant No. CUHK 58/93E.

0-7803-2624-5/95/$4.00 © 1995 IEEE

II. MINIMUM SPANNING TREE PROBLEM AND INFERENCE NETWORK

Given an undirected graph $G = (V, A)$ with $N = |V|$ nodes, $M = |A|$ arcs, and with a cost c_{ij} associated with each arc $(i, j) \in A$. A spanning tree T of the graph G is a connected acyclic subgraph that spans all the nodes in G. The minimum spanning tree is the spanning tree with the smallest total cost, measured as the sum of costs of the arcs in the spanning tree.

The MST problem is closely related with the all-pairs minimax path problem (MPP), which tries to determine, for each given node pair (i, j), the minimum *value* path from node i to node j. In distinct contrast with the classical shortest path problem, the *value* of a path P from node i to node j is the maximum cost arc in P and not the total arc cost of P. A well established result in MPP is that, if D_{ij} is the value (minimax) of the minimax path from node i to node j and $c_{ij} = D_{ij}$, then the arc (i, j) is an edge of the MST. It is therefore a simple task to solve the MST problem once we know the solution to the MPP.

The all-pairs minimax path problem bears many similarities with the all-pairs shortest path problem [4], and hence can be solved by similar methods. In the context of dynamic programming, the minimax values are related by the Bellman equation [2]:

$$D_{ij} = \min_k \; [\; \max \{c_{ik}, D_{kj}\} \;] \tag{1}$$

where $i, j, k = 1, 2, ..., N, k \neq i, k \neq j$. The Bellman-Ford algorithm is widely known to solve the Bellman equation iteratively in $O(N)$ steps, for any fixed i (single-source, multi-destinaton) or fixed j (multi-source, single destination) schemes.

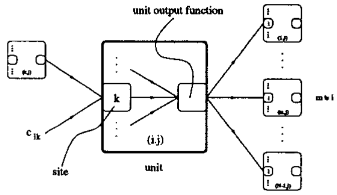

Fig. 1. Single-destination inference network with reference node j

A. AN INFERENCE NETWORK ARCHITECTURE FOR BELLMAN'S EQUATION

Previous works[10,6] have established a binary relation inference network architecture for solving all-pairs shortest path problem. The inference network considers the set of nodes that are allowed as intermediate nodes on the paths, while the Bellman-Ford algorithm tends to iterate on the number of arcs in the path. A modification of the inference network can readily be made to result in a multi-source single-destination inference network, which forms the basic architecture for solving the Bellman equation. With the single-destination inference network (Fig. 1), the site output S_k^{ij} and unit output $g(i, j)$ required in solving the minimum spanning tree problem are given by

$$S_k^{ij} = \max \{c_{ik}, g(k, j)\} \tag{2}$$

$$g(i, j) = \min_k \; S_k^{ij} \tag{3}$$

where $i, j, k = 1, 2, ..., N \quad k \neq i, k \neq j$. It is readily seen that there is a close resemblance between the Bellman equation (Eq. (1)) and Eq. (2), Eq. (3). However, the network does not specify a computational order as is required in the Bellman-Ford algorithm. In fact, such computational order is not required for the network to converge to the global optimal solution after proper initialization. A simple first-order non-linear model can be assumed for each computational unit, given by

$$\frac{dg(i, j)}{dt} = -\lambda_{ij} g(i, j) + \lambda_{ij} \; \min_k [\max\{c_{ik}, g(k, j)\}] \tag{4}$$

and the network has been found to converge to the correct solution, with $g(i, j) = D_{ij}$, arbitrarily fast by using large λ_{ij} for the units. Extensive simulation also indicates that the convergence time is independent of the problem size N. It is certainly true that the mathematical model is idealistic, and the actual performance of the network is likely to be limited by practical factors in different implementation platforms.

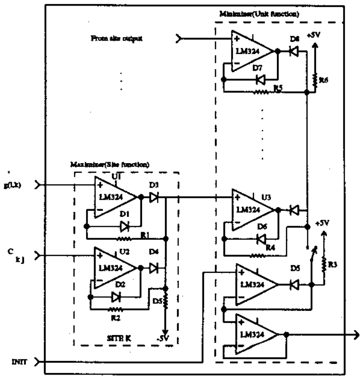

Fig. 2. Circuit of a computational unit

III. ANALOG IMPLEMENTATION

The site and unit output functions [6] as defined in Eq. (2) and Eq. (3) are maximizer and minimizer circuits [6] which can be readily connected to form a network as shown in Fig. 2. The completed network which is shown in Fig. 3 is tested with the following

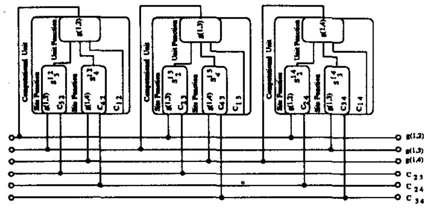

Fig. 3. The single-destination network in a bus configuration for a 4 cities problem

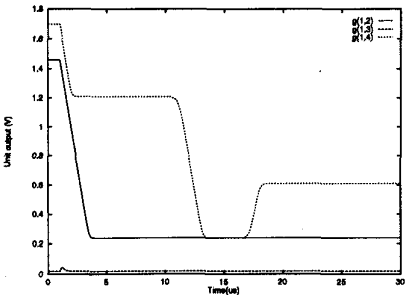

Fig. 4. Pspice simulation output of 3-unit single-destination inference network

data:

$$\begin{aligned} C_{12} &= 60, & C_{13} &= 1, & C_{14} &= 70, \\ C_{23} &= 10, & C_{24} &= 25, & C_{34} &= 50. \end{aligned}$$

The Pspice simulation result is shown in Fig. 4. The horizontal portion of the curve indicates the presence of delay in the network. This delay is largely caused by the saturation of the op-amps in the minimizer and maximizer during a switch between the op-amp-diode subcircuits which act as the driver for the corresponding site and unit outputs. This delay has been reduced by adding diodes(D1, D2, D6, D7, ...) as shown in Fig. 2 as saturation guards for the associated op-amps. The network convergence time has been reduced from $18\mu s$ to $9\mu s$ with the addition of this saturation guard as shown in Fig. 5. Since there are delay in each computational unit, the network convergence time depends on both the problem size and configuration. With fixed problem size, a MST with longer paths experiences more delay and the convergence time increases drastically. However, the exact relationship still needs careful study. The best convergence time(i.e. one without time delay) is governed by the op-amp large signal response and is practical size independent. With LM324, this time is found to be about $4.9\mu s$ whereas the OP-27, which has a slew rate four times higher than that of LM324, gives a convergence time of about $1.6\mu s$. The variance in op-amp open loop gain has little effect on the convergence time. With the default gain of LM324, the percentage error between input and output of a unit is 0.2%.

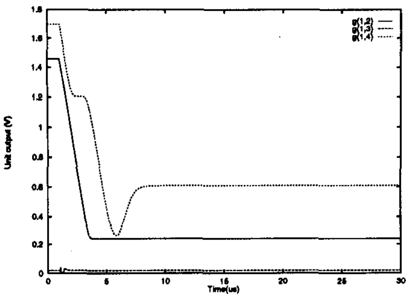

Fig. 5. Pspice simulation output after addition of saturation guard

IV. VLSI IMPLEMENTATION

Based on the inference network model, the site and the unit output functions [11] can be implemented by operational transconductance amplifier(OTA) [1] in Fig. 6. Although the OTA can provide a sufficiently large gain, it introduces non-negligible error. In our simulation,using Caltech's Chipmunk [12], the maximum input is within 1.7V. The error is taken as the maximum difference between input signal and output signal divided by the maximum input.

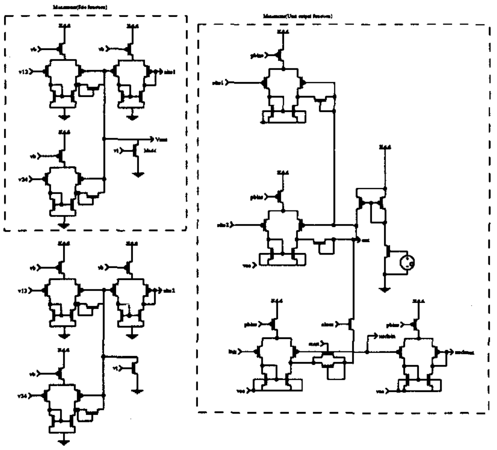

Fig. 6. The Computational Unit

The site function is a maximizer circuit [10]. With a D.C. and a pulse applied to the site inputs, an error as large as 35% is observed. This is caused by the parasitic capacitance of the output (Vout as shown in Fig. 6) holding the site output voltage at the falling edge of the pulse. It is shown in Fig. 7. In order to reduce this effect, an N-channel CMOS transistor (Madd as shown in Fig. 6) was added at the *Vout* output. The transistor speeds up the discharge rate by increasing the conductance. The induced error from the OTA maximizer reduces to 0.35%. The induced error can further be reduced to 0.26% by increasing the gain 100 times. Due to this undesirable capacitance, a wide range transconductance amplifier(WRTA) [3] have been considered as an alternative to the OTA. However, despite the much lower parasitic capacitance at the output of this amplifier and the relatively higher voltage gain when compared to those of the OTA, the error in the resulting maximizer circuit is 0.9%, about three times larger.

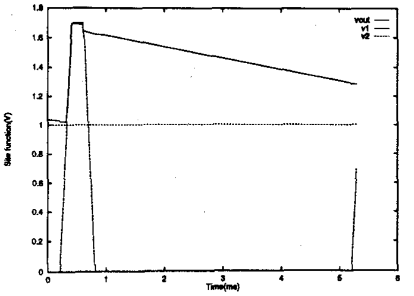

Fig. 7. Error in simulated OTA Maximizer output response caused by parasitic capacitance

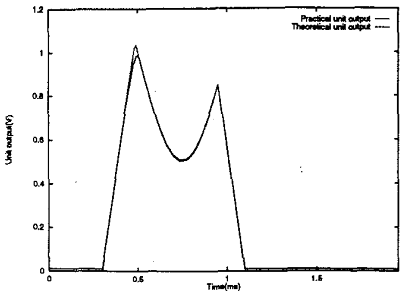

Fig. 8. The practical and theoretical unit output of a sinusoidal and a pulse input

The unit output function is a minimizer circuit [11]. With a sinusoidal and pulse input, the theoretical and the practical minimum output are shown in Fig. 8. The induced error from the circuit is about 4.47%. As the gain increases about three times, the induced error decreases to 1.94%. Besides, the convergence behaviour of a single computational unit was studied. The computational unit shows the fastest convergence rate in an operating range between 1.0V and 2.7V. The convergence time is less than 200ns. Based on their similar functional blocks of VLSI and analog implementation, the relationship between their convergence time can be estimated by a scaling factor. In a computational unit,the saturation time for analog implementation is about 4μs. The scaling factor is 20. Therefore, the network convergence time for a 4-cities problem would be approximately 245ns in VLSI implementation when that of LM324 is about 4.9μs.

V. CONCLUSION

The limiting factors of performance in different implementation platforms for the binary relation inference network in solving the MST problems are presented. According to the parallel and asynchronous operating nature of the network, the solution time should be independent of the problem size and is only limited by the physical characteristics of the circuit components. From the simulated network reponses using commonly available analog IC, it is noted that the factors affecting the network convergence rate are opamp slew rate, unit operating voltages and the propagation delay. The most serious problem is the propagation delay which is controlled by the op-amp large signal response. This can be reduced by using the saturation guard diodes.

In the VLSI implementation, the performance of the network is affected by the parasitic capacitance of the CMOS transistor. Therefore we compared the OTA and WRTA maximizer and found that OTA is better than WRTA in terms of complexity and accuracy. Also the performance of the unit output function is affected by the gain of OTA. The solution speed of VLSI implementation is much faster than that of analog.

Though the circuits presented may not completely reflect the mathematical model, they provide the basis for future work. In particular, a delay in the min-max function of mathematical model could be added to better reflect the practical behaviour of analog circuits. Also, a faster and more accurate implementation platform could be sought for, especially in the field of VLSI implementations. Actual prototype constructions are also necessary for verifying some of our simulation results.

REFERENCES

[1] P. E. Allen, and Douglas R. Holdberg, *CMOS Analog Circuit Design.* Holt, Rinehart and Winston New York, 1987.

[2] R. Bellman. 'On a Routing Problem'. *Quarterly of Applied Mathematics*, vol.16, pp. 87-90, 1958.

[3] Carver Mead, *Analog VLSI and Neural Systems.* Addison Wesley, 1989.

[4] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, *Introduction to Algorithms.* The MIT Press, 1990.

[5] J. B. Kruskal, 'On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem'. *Proc. of the Aemrican Mathematical Society*, vol. 7, pp. 48–50, 1956.

[6] K. P. Lam and C. W. Tong. 'Optimisation circuits for the Bellman-Ford computation algorithm'. *IEE Electronic Letters* vol. 30, No.19, pp. 1584-1586, 1994.

[7] R. C. Prim. 'Shortest Connection Networks and some Generalizations'. *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[8] X. Shen, and W. Liang, 'A Parallel Algorithm for Multiple Edge Updates of Minimum Spanning Trees' *Proc. of IEEE on the seventh International Parallel Processing Symposium*, pp 310-317, 1993.

[9] Sollin, An algorithm attributed to Sollin in *Programming, Games and Transportation Networks.* by Berge, C., and Choulia-Houri, A. Wiley, NY 1965.

[10] C. W. Tong, and K. P. Lam, 'VLSI Implementation of Binary Relation Inference Network in Solving Shortest Path Problems'. *Proc. IEEE International Conference on Neural Networks*, Orlando, pp. 2143-2148, 1994.

[11] C. W. Tong, and K. P. Lam, 'Closed-Semiring Optimization Circuits for Parallel and Distributed Computation'. *Manuscript submitted.*

[12] D. Gillespie, and J.Lazzaro, *'An Introducation to LOG with Analog'.* ISL, Electrical Engineering Department, Stanford University, 1993