A Parallel Algorithm for Time-Slot Assignment Problems in TDM Hierarchical Switching Systems

Nobuo Funabiki, *Member, IEEE*, and Yoshiyasu Takefuji

***Abstract*—This paper presents a parallel algorithm for time-slot assignment problems in TDM hierarchical switching systems, based on the neural network model. The TDM systems are operated in repetitive frames composed of several time-slots. A time-slot represents a switching configuration where one packet is transmitted through an I/O line. The goal of our algorithm is to find conflict-free time-slot assignments for given switching demands. The algorithm runs on a maximum of $n^2 \times m$ processors for m-time-slot problems in $n \times n$ TDM systems. In small problems up to a 24×24 TDM system, the algorithm can find the optimum solution in a nearly constant time, when it is performed on $n^2 \times m$ processors.**

I. Introduction

TDM (Time Division Multiplexing) hierarchical switching systems have been used widely in both terrestrial and satellite networks. The TDM systems can reduce hardware costs, enhance trunk line efficiency, and accommodate additional loads [1]–[4]. Eng *et al.* defined the general structure of TDM systems as shown in Fig. 1 [3]. A TDM system consists of three synchronously operated stages: several multiplexers, one central time-multiplex crossbar switch, and several demultiplexers. Multiplexer $\#i$ concentrates P_i inputs from end users to I_i TDM lines, then to the central switch, where P_i is usually larger than I_i. The central switch exchanges the TDM lines. Demultiplexer $\#j$ connects O_j TDM lines with Q_j outputs to end users, where Q_j is usually larger than O_j. The total number of input lines is equal to that of output lines.

We use the assumption of Eng *et al.* that signals through the TDM systems are fixed-length message packets [3]. The packets are transmitted in repetitive frames composed of several time-slots. A time-slot represents one switching configuration having a unit time, where one packet can be transmitted through one I/O line. As defined by Bonuccelli [4], packet transmission demands through the $n \times n$ TDM system are represented by an $n \times n$ traffic matrix. The ijth element t_{ij} represents the number of packets to be transmitted from input $\#i$ to output $\#j$. t_{ij} is equivalent to the number of time-slots required for the packet transmission. In order to maximize the throughput, it is necessary to find the conflict-free time-slot assignment for a given traffic matrix using the minimum number of time-slots.

Paper approved by E. G. Sable, the Editor for Communication Switching of the IEEE Communications Society. Manuscript received May 9, 1992; revised September 20, 1992.

N. Funabiki was with the System Engineering Division, Sumitomo Metal Industries, Ltd., Amagasaki 660, Japan. He is now with the Department of Information and Computer Sciences, Osaka University, Osaka 560, Japan.

Y. Takefuji is with the Department of Electrical Engineering and Applied Physics, Case Western Reserve University, Cleveland, OH 44106 USA.

IEEE Log Number 9404706.

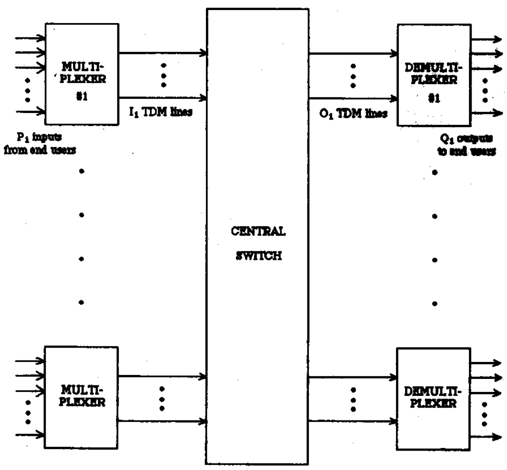


Fig. 1. A TDM hierarchical switching system.

The problem of finding conflict-free time-slot assignments for the traffic matrix has been studied both in TDM hierarchical switching systems [3], [4], and in TDM non-hierarchical switching systems [5]–[15], particularly in TDMA (Time Division Multiple Access) satellite network applications. Although the time-slot assignment problem in hierarchical systems is not NP-complete, some problems in nonhierarchical systems have proven to be NP-complete. In 1983, Gopal *et al.* proved that the traffic scheduling problem in the case of zone interference, is NP-complete [13]. In 1985, Gopal *et al.* proved that the minimization problem of switching configuration chances is NP-complete [14]. In 1987, Bertossi *et al.* proved that the minimization problem of total transmission time is NP-complete [15]. The nonhierarchical system can be seen as a special case of the hierarchical system, because the former system does not involve multiplexers and demultiplexers. In 1987 Eng *et al.* showed the four requisite conditions for valid solutions in time-slot assignment problems in hierarchical systems [3]. In 1989, Bonuccelli proposed the $O(n^5)$ sequential algorithm for $n \times n$ hierarchical systems [4]. Also in 1989, Rose proposed the $O(n)$ parallel algorithm on n^2 processing elements for simple $n \times n$ crossbar switching systems, based on a cellular automaton [16]. Because Rose's algorithm cannot deal with multiplexer/demultiplexer constraints, a lot of modifications are required to cope with time-slot assignment problems in TDM hierarchical switching systems. A fast parallel algorithm is essential to improve the switching performance.

II. Neural Network Approach for Optimization Problems

Our parallel algorithm is based on the three-dimensional artificial neural network model. A neural network model for solving optimization problems consists of a large number of interconnected processing elements. These processing elements are called neurons because they perform the function of simplified biological neurons. The interconnections between processing elements are given by the motion equation:

$$\frac{dU_{ijk}}{dt} = -\frac{\partial E}{\partial V_{ijk}} \tag{1}$$

where U_{ijk} and V_{ijk} are the input and the output respectively of processing element $\#ijk$. The computational energy function E represents all the constraints of the optimization problem. The goal of the neural network approach is to minimize energy function E, where the motion equation performs the gradient descent method [22].

In 1943 McCulloch and Pitts proposed the first mathematical neuron model [17]. The input/output function of processing element $\#ijk$ is given by

$$\text{if } U_{ijk} > 0 \text{ then } V_{ijk} = 1$$

$$\text{if } U_{ijk} \leq 0 \text{ then } V_{ijk} = 0. \tag{2}$$

Hopfield *et al.* first introduced the neural network model for solving optimization problems by using the sigmoid neuron model [18]. The input/output function of processing element $\#ijk$ is given by

$$V_{ijk} = \frac{1}{1 + \exp\left(-\frac{U_{ijk}}{U_0}\right)} \tag{3}$$

where U_0 is a constant parameter.

In 1989 Marrakchi *et al.* proposed the Hopfield neural network model application for crossbar switching systems, which was verified only in an 8×8 crossbar system [19]. In 1989 Brown proposed the Hopfield neural network model application for multistage crossbar switching systems [20]. Also in 1990, Brown *et al.* proposed the application for Banyan network systems [21]. Unfortunately, none of these investigators discussed the time complexity and the convergence frequency of neural network models, although these factors are always controversial in neural network research. Takefuji *et al.* proved that the decay term in the Hopfield neural network model disturbs the convergence under some conditions [22].

Although the McCulloch–Pitts neuron model can drastically reduce the computation time on a digital computer, it sometimes introduces undesirable oscillatory behavior. It has been shown empirically that the newly introduced hysteresis McCulloch–Pitts neuron model can suppress this oscillatory behavior [24]. The input/output function of processing element $\#ijk$ is given by:

$$\text{if } U_{ijk} > UTP \text{ then } V_{ijk} = 1$$

$$\text{if } U_{ijk} < LTP \text{ then } V_{ijk} = 0$$

$$\text{otherwise } V_{ijk} = \text{unchanged}. \tag{4}$$

where UTP is always larger then LTP, and the initial value of V_{ijk} must be assigned 1 or 0.

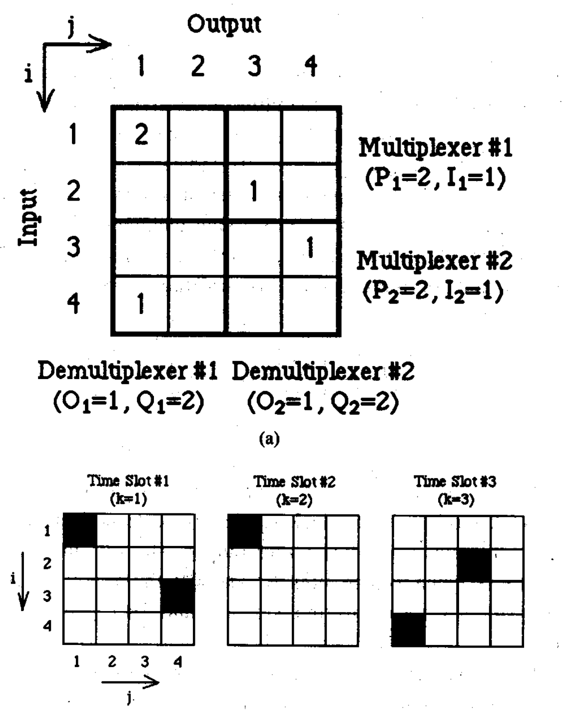


Fig. 2. A 4×4 traffic matrix problem and the neural network representation. (a) A 4-input 4-output traffic matrix problem. (b) The neural network representation with 48 processing elements.

III. Neural Network Representation for Time-Slot Assignment Problems

Fig. 2(a) shows the neural network model for the 4×4 traffic matrix problem. Each multiplexer connects two inputs ($P_i = 2$) with one TDM line ($I_i = 1$). The central switch exchanges two TDM lines. Each demultiplexer connects one TDM line ($O_i = 1$) with two outputs ($Q_i = 2$). Fig. 2(b) shows the neural network representation for the three-time-slot problem of Fig. 2(a). Because the assignment of each traffic matrix element requires three processing elements, a total of 48 ($= 4 \times 4 \times 3$) processing elements is prepared in this problem.

Generally, a total of $n^2 \times m$ processing elements is required for the m-time-slot problem in an $n \times n$ TDM system. The output of processing element $\#ijk$ represents whether or not one packet from input $\#i$ to output $\#j$ is assigned to time-slot $\#k$. The nonzero output ($V_{ijk} = 1$) indicates the assignment, and the zero output ($V_{ijk} = 0$) indicates no assignment. Because a total of t_{ij} packets is demanded by traffic matrix element $\#ij$, a total of t_{ij} processing elements from among m processing elements for t_{ij}, must have nonzero output. The energy function representing this constraint is given by

$$E_1 = \left(\sum_{r=1}^{m} V_{ijr} - t_{ij}\right)^2. \quad (5)$$

E_1 is zero if, and only if, every packet is assigned to a time-slot. Fig. 2(b) also shows the convergence state of the neural network model, where black squares indicate nonzero output and white squares indicate zero output. The two packets for t_{11} are assigned to time-slots $\#1$ and $\#2$, the packet for t_{23} to time-slot $\#3$, for t_{34} to time-slot $\#1$, and for t_{41} to time-slot $\#3$.

The TDM hierarchical switching system has two constraints: 1) in a given time-slot, each input can be connected with only one output and each output with only one input (point-to-point connections) and 2) in a given time-slot, multiplexer $\#y$ can connect at most I_y inputs with the central switch, and demultiplexer $\#z$ can connect at most O_z outputs with the central switch. The energy function representing the first constraint is given by:

$$E_2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{pjk} + \sum_{\substack{q=1\\q\neq j}}^{n} V_{iqk}\right)V_{ijk}. \quad (6)$$

E_2 is zero if, and only if, at most one packet from an input and at most one packet to an output are assigned to any time-slot. The energy function representing the second constraint is given by

$$E_3 = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\left(f\left(\sum_{\substack{p\in y\\(p,q)\neq(i,j)}}\sum_{q=1}^{n} V_{pqk} - I_y\right) + f\left(\sum_{\substack{q\in z\\(p,q)\neq(i,j)}}\sum_{p=1}^{n} V_{pqk} - O_z\right)\right)V_{ijk} \quad (7)$$

where $f(x)$ is 1 if $x \geq 0$, and $f(x)$ is 0 if $x < 0$. E_3 is zero if, and only if, I_y or fewer inputs are assigned in any time-slot for multiplexer $\#y$, and O_z or fewer outputs are assigned in any time-slot for demultiplexer $\#z$. Note that input $\#i$ is connected with multiplexer $\#y$, and output $\#j$ with demultiplexer $\#z$.

TABLE I
SPECIFICATIONS AND SIMULATION RESULTS IN TEN PROBLEMS

	problem specification		simulation result	
	matrix size	time-slot #	average # of iteration steps	convergence frequency
Problem #1	6 × 6	7	845.3	95%
Problem #2	8 × 8	10	860.2	93%
Problem #3	10 × 10	9	705.8	91%
Problem #4	12 × 12	9	831.3	92%
Problem #5	14 × 14	10	662.9	94%
Problem #6	16 × 16	11	820.2	90%
Problem #7	18 × 18	12	765.2	82%
Problem #8	20 × 20	13	705.2	89%
Problem #9	22 × 22	14	816.5	80%
Problem #10	24 × 24	15	728.0	76%

The total energy function for this problem is given by:

$$\begin{aligned} E &= \frac{A}{2}E_1 + BE_2 + CE_3 \\ &= \frac{A}{2}\left(\sum_{r=1}^{m} V_{ijr} - t_{ij}\right)^2 \\ &\quad + B\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{pjk} + \sum_{\substack{q=1\\q\neq j}}^{n} V_{iqk}\right)V_{ijk} \\ &\quad + C\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\left(f\left(\sum_{\substack{p\in y\\(p,q)\neq(i,j)}}\sum_{q=1}^{n} V_{pqk} - I_y\right)\right. \\ &\quad \left. + f\left(\sum_{\substack{q\in z\\(p,q)\neq(i,j)}}\sum_{p=1}^{n} V_{pqk} - O_z\right)\right)V_{ijk} \end{aligned} \quad (8)$$

where A, B, and C are constant coefficients.

From (1) and (8), the motion equation of processing element $\#ijk$ for the m-time-slot problem in the $n \times n$ TDM system is given by

$$\begin{aligned} \frac{dU_{ijk}}{dt} &= -A\left(\sum_{r=1}^{m} V_{ijr} - t_{ij}\right) - B\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{pjk} + \sum_{\substack{q=1\\q\neq j}}^{n} V_{iqk}\right) \\ &\quad - C\left(f\left(\sum_{\substack{p\in y\\(p,q)\neq(i,j)}}\sum_{q=1}^{n} V_{pqk} - I_y\right)\right. \\ &\quad \left. + f\left(\sum_{\substack{q\in z\\(p,q)\neq(i,j)}}\sum_{p=1}^{n} V_{pqk} - O_z\right)\right). \end{aligned} \quad (9)$$

The A-term forces total t_{ij} from m processing elements for traffic matrix element t_{ij}, to have nonzero output. The B-term discourages processing element $\#ijk$ from having nonzero output if other demands from input $\#i$ and/or to output $\#j$ are

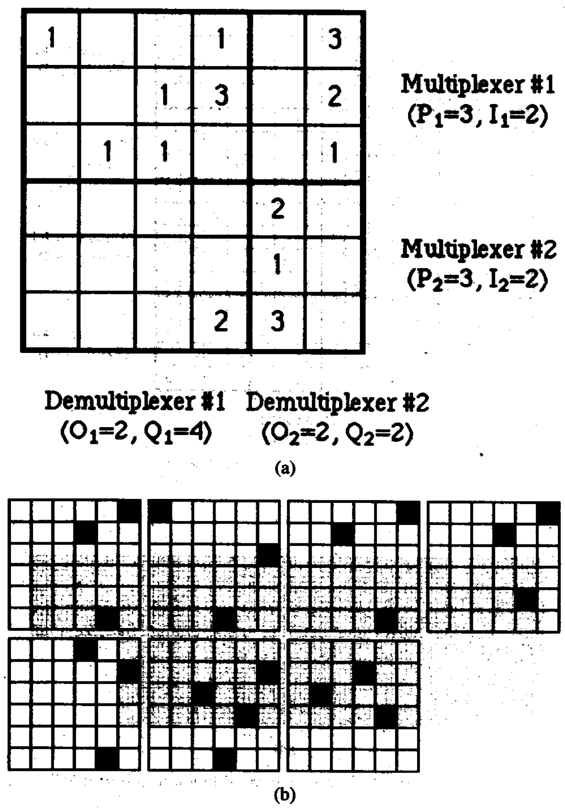


Fig. 3. The 6 × 6 traffic matrix and the solution to problem #1.

assigned to time-slot #k. The C-term has the same role as the B-term if I_y or more packets sharing multiplexer #y, and/or O_z or more packets sharing demultiplexer #z, are assigned to time-slot #k.

IV. Three Heuristics for the Global Minimum Convergence

In the neural network model, only the local minimum convergence has been proven [22]. In order to improve the global minimum convergence, as required in optimization problems, we have introduced empirically the following three heuristics:

1) The hill-climbing term heuristic [22]

The following hill-climbing term is added to the motion equation, in order to help a total of t_{ij} processing elements for t_{ij} to have nonzero output, which is necessary in the global minimum solution of the neural network model:

$$+Dh\left(\sum_{r=1}^{m} V_{ijr} - t_{ij}\right) \qquad (10)$$

where $h(x)$ is 1 if $x < 0$, and $h(x)$ is 0 if $x \geq 0$.

2) The omega function heuristic [23]

The periodic use of the following two forms of B-term in the motion equation makes local minima shallower, and allows

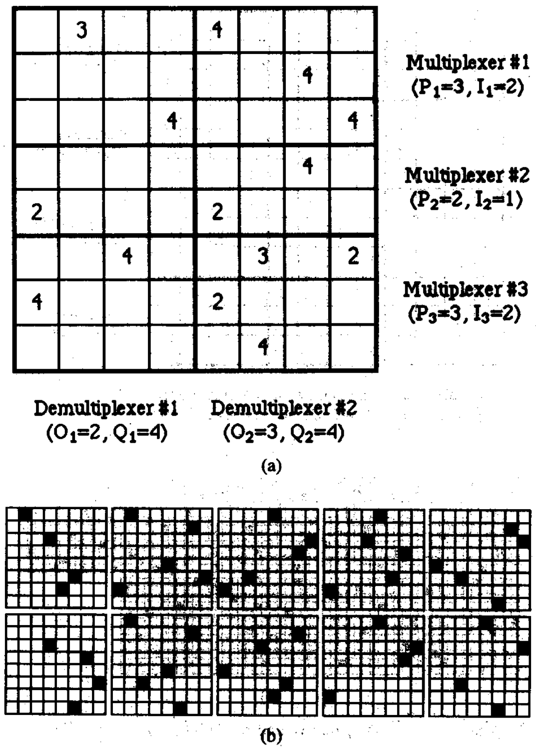


Fig. 4. The 8 × 8 traffic matrix and the solution to problem #2.

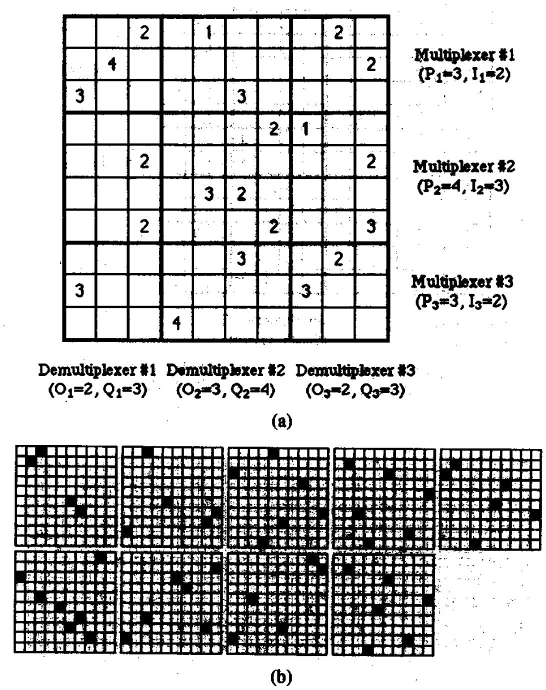
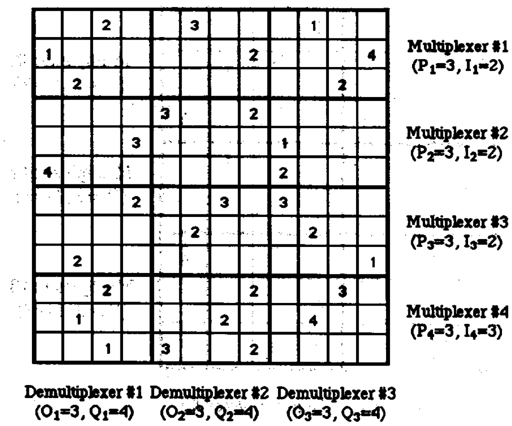
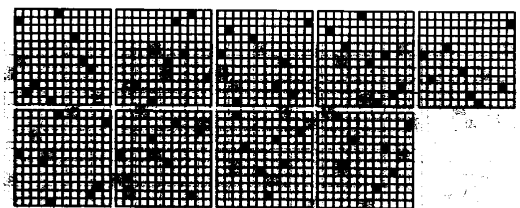


Fig. 5. The 10 × 10 traffic matrix and the solution to problem #3.

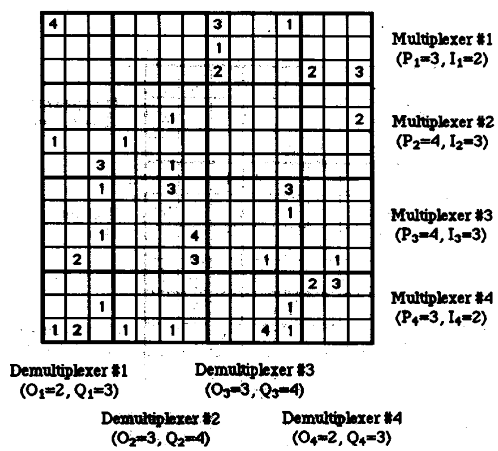


(a)

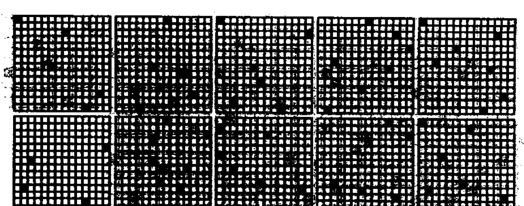


(b)

Fig. 6. The 12 × 12 traffic matrix and the solution to problem #4.

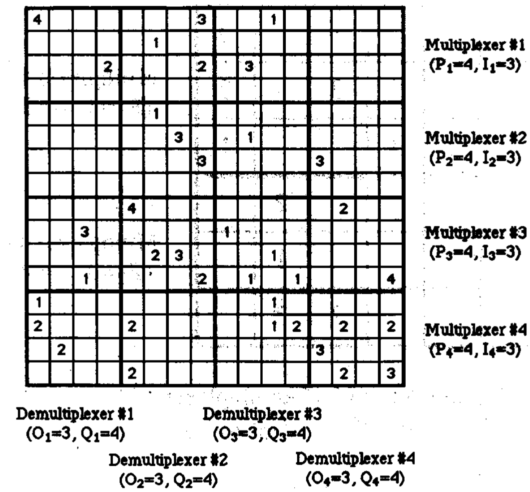


(a)

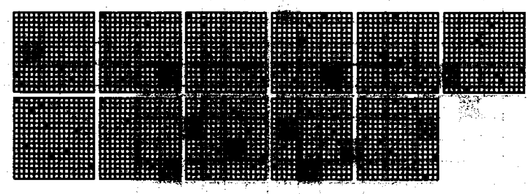


(b)

Fig. 7. The 14 × 14 traffic matrix and the solution to problem #5.

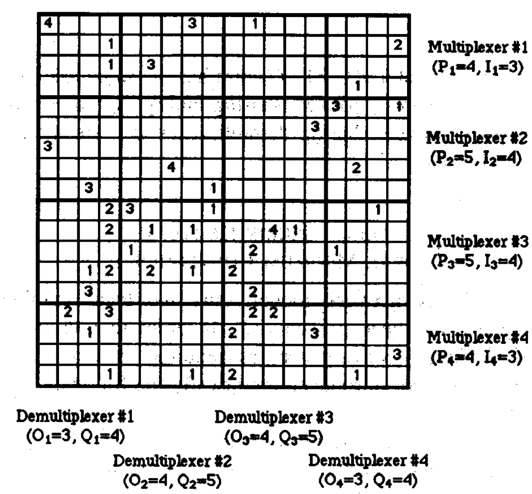


(a)

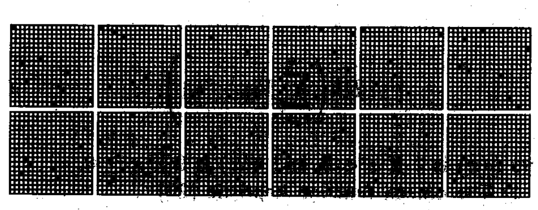


(b)

Fig. 8. The 16 × 16 traffic matrix and the solution to problem #6.



(a)



(b)

Fig. 9. The 18 × 18 traffic matrix and the solution to problem #7.

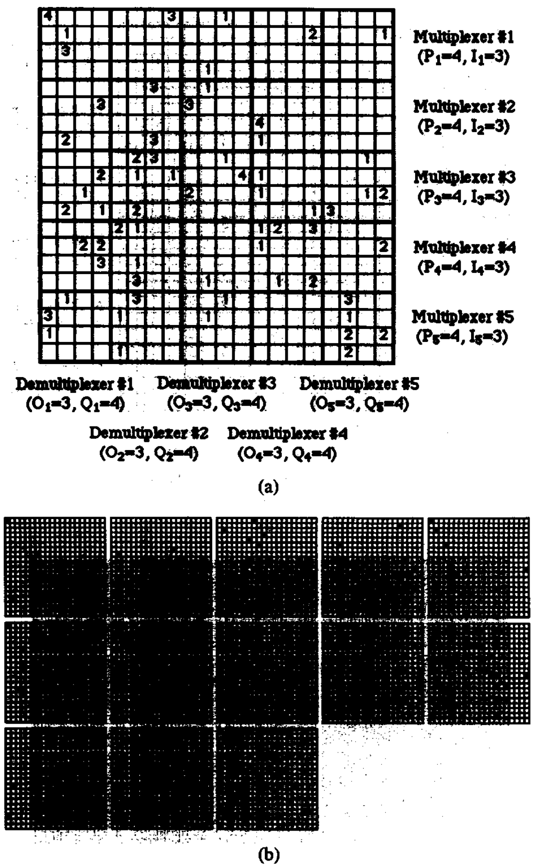


Fig. 10. The 20 × 20 traffic matrix and the solution to problem #8.

the neural network model to escape easily:

$$\text{if} \quad (t \bmod T) < \omega \quad \text{then} \quad -B\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{pjk} + \sum_{\substack{q=1\\q\neq j}}^{n} V_{iqk}\right) V_{ijk}$$

$$\text{else} \quad -B\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{pjk} + \sum_{\substack{q=1\\q\neq j}}^{n} V_{iqk}\right) \qquad (11)$$

where t is the number of iteration steps, and T and ω are omega function parameters.

3) The input saturation heuristic [25]

The neuron input is saturated between two values, which also makes local minima shallower.

$$\begin{aligned} &\text{If} \quad U_{ijk} > U\_\max \quad \text{then} \quad U_{ijk} = U\_\max \\ &\text{If} \quad U_{ijk} < U\_\min \quad \text{then} \quad U_{ijk} = U\_\min \qquad (12) \end{aligned}$$

where $U\_\max$ is the upper limit of the input, and $U\_\min$ is the lower limit.

V. Parallel Algorithm

The following procedure represents our parallel algorithm for the m-time-slot problem in the $n \times n$ TDM system. The set of parameters are empirically determined [22]–[34].

$$\begin{aligned} &0) \quad \text{Set} \quad t = 0,\ A = B = C = 1,\ D = 20, \\ &T = 5,\ \omega = 2,\ \text{UTP} = 5,\ \text{LTP} = -5, \\ &U\_\max = 200,\ U\_\min = -800, \text{ and } T\_\max = 3000. \qquad (13) \end{aligned}$$

1) Set initial values of $U_{ijk}(t)$ for $i = 1, \cdots, n$, $j = 1, \cdots, n$, and $k = 1, \cdots, m$ by uniform random numbers between $U\_\max$ and $U\_\min$, and set initial values of $V_{ijk}(t)$ by 0.

2) Use the motion equation with two heuristics to compute $\Delta U_{ijk}(t)$:

if $(t \bmod T) < \omega$ then

$$\begin{aligned} \Delta U_{ijk}(t) = &-A\left(\sum_{r=1}^{m} V_{ijr}(t) - t_{ij}\right) \\ &- B\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{pjk}(t) + \sum_{\substack{q=1\\q\neq j}}^{n} V_{iqk}(t)\right) V_{ijk}(t) \\ &- C\left(f\left(\sum_{\substack{p\in y\\(p,q)\neq(i,j)}} \sum_{q=1}^{n} V_{pqk}(t) - I_y\right)\right. \\ &\left. + f\left(\sum_{\substack{q\in z\\(p,q)\neq(i,j)}} \sum_{p=1}^{n} V_{pqk}(t) - O_z\right)\right) \\ &+ Dh\left(\sum_{r=1}^{m} V_{ijr}(t) - t_{ij}\right) \end{aligned}$$

else

$$\begin{aligned} \Delta U_{ijk}(t) = &-A\left(\sum_{r=1}^{m} V_{ijr}(t) - t_{ij}\right) \\ &- B\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{pjk}(t) + \sum_{\substack{q=1\\q\neq j}}^{n} V_{iqk}(t)\right) \\ &- C\left(f\left(\sum_{\substack{p\in y\\(p,q)\neq(i,j)}} \sum_{q=1}^{n} V_{pqk}(t) - I_y\right)\right. \\ &\left. + f\left(\sum_{\substack{q\in z\\(p,q)\neq(i,j)}} \sum_{p=1}^{n} V_{pqk}(t) - O_z\right)\right) \\ &+ Dh\left(\sum_{r=1}^{m} V_{ijr}(t) - t_{ij}\right) \qquad (14) \end{aligned}$$

3) Update $U_{ijk}(t+1)$ based on the first-order Euler method:

$$U_{ijk}(t+1) = U_{ijk}(t) + \Delta U_{ijk}(t) \qquad (15)$$

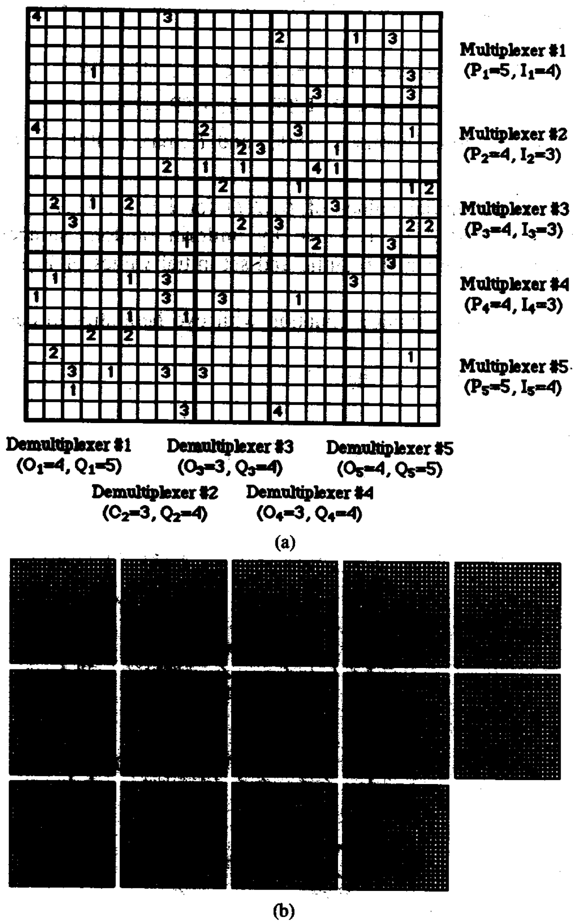


Fig. 11. The 22 × 22 traffic matrix and the solution to problem #9.

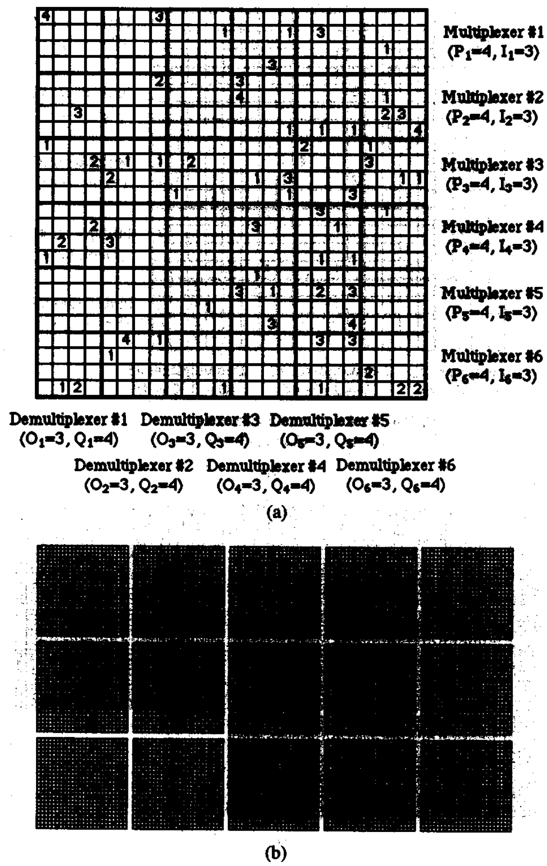


Fig. 12. The 24 × 24 traffic matrix and the solution to problem #10.

4) Use the input saturation heuristic:

$$\text{If} \quad U_{ijk}(t+1) > U\_\text{max} \quad \text{then} \quad U_{ijk}(t+1) = U\_\text{max}$$
$$\text{If} \quad U_{ijk}(t+1) < U\_\text{min} \quad \text{then} \quad U_{ijk}(t+1) = U\_\text{min} \qquad (16)$$

5) Update $V_{ijk}(t+1)$ based on the hysteresis McCulloch–Pitts neuron model:

$$\text{if} \quad U_{ijk}(t+1) > UTP \quad \text{then} \quad V_{ijk}(t+1) = 1$$
$$\text{if} \quad U_{ijk}(t+1) < LTP \quad \text{then} \quad V_{ijk}(t+1) = 0 \qquad (17)$$

6) If all the constrains are satisfied ($E = 0$), or $t = T\_\text{max}$, then terminate this procedure, else increment t by 1, and go to step 2.

VI. Simulation Results and Discussion

The simulator based on the algorithm has been developed on a Macintosh. The ten problems shown in Table I were examined. Problem #1 was originally described by Gopal [14], where it was modified from a nonhierarchical to a hierarchical problem. Problem #2 was described by Bonuccelli [4]. Problems #3–#10 were newly created in this paper, where traffic matrix elements were randomly generated. Each problem satisfies the following conditions: 1) a maximum of four packets are demanded in any traffic matrix element, 2) the number of time-slots is the optimum condition to have valid solutions, and 3) the multiplexer/demultiplexer size is similar to problem #2.

Figs. 3–12 show traffic matrices and multiplexer/demultiplexer size, and the global minimum solutions found by our simulator in the ten problems. The simulator found several solutions to one problem from different initial values of U_{ijk}. In order to avoid initial value dependency of the neural network model, 100 simulation runs were performed from different initial values of U_{ijk}. Table I also summarizes the average number of iteration steps required for global minimum convergence, and the frequency. Fig. 13 shows the distribution of the number of iteration steps required for global minimum convergence in two problems. The simulation results show that in a nearly constant number of iteration steps, our algorithm can find optimum solutions, in small size problems

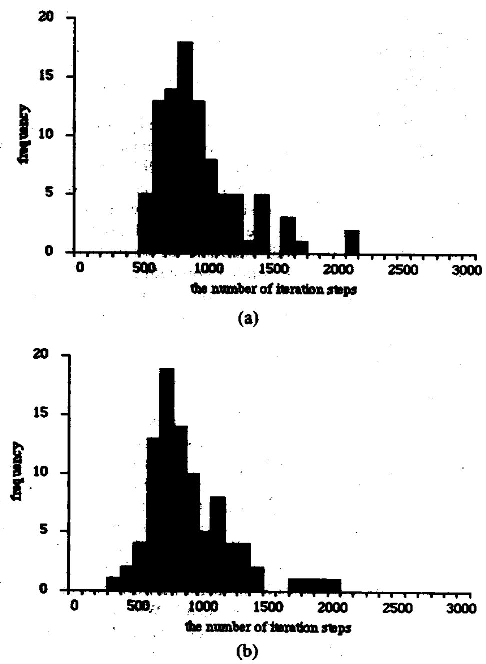


Fig. 13. The distribution of the number of iteration steps required for global minimum convergence (a) Problem #2. (b) Problem #6.

up to a 24×24 TDM system. We therefore conclude that our parallel algorithm for m-time-slot assignment problems in $n \times n$ TDM hierarchical switching systems, is able to find optimum solutions in a nearly constant time, when it is performed on a parallel machine using $n^2 \times m$ processors.

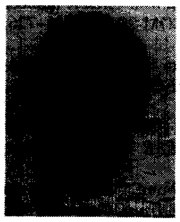
VII. Conclusion

Based on the neural network model composed of $n^2 \times m$ processing elements, we propose a parallel algorithm for m-time-slot assignment problems in $n \times n$ TDM hierarchical switching systems. The algorithm runs not only on a sequential machine, but also on a parallel machine with a maximum of $n^2 \times m$ processors. In small size problems (up to a 24×24 TDM system), our simulation results show that the algorithm is able to find optimum solutions in a nearly constant time, when it is performed on a parallel machine with $n^2 \times m$ processors. The algorithm can be easily modified for other time-slot assignment problems which have multipoint connections, and/or rectangular traffic matrices.

References

[1] H. E. Vaughan, "Research model for time-separation integrated communication," *IEEE Trans. Commun.*, vol. COM-27, pp. 940–947, July 1979.
[2] A. E. Joel, Jr., "Digital switching—how it has developed," *IEEE Trans. Commun.*, vol. COM-27, pp. 948–959, July 1979.
[3] K. Y. Eng and A. S. Acampora, "Fundamental conditions governing TDM switching assignments in terrestrial and satellite networks," *IEEE Trans. Commun.*, vol. COM-35, pp. 755–761, July 1987.
[4] M. A. Bonuccelli, "A fast time slot assignment algorithm for TDM hierarchical switching systems," *IEEE Trans. Commun.*, vol. 37, pp. 870–874, Aug. 1989.
[5] T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Trans. Commun.*, vol. COM-27, pp. 1449–1455, Oct. 1979.
[6] ——, "SS-TDMA networking via ISL," in *Proc. ICC*, 1981, pp. 70.6.1–70.6.5.
[7] W. G. Schmidt, "An on-board switched multiple-access system for millimeter-wave satellites," in *Conf. Rec.*, 1969 INTELSAT/IEE Int. Conf. Digital Satellite Commun., pp. 399–407.
[8] D. O. Reudink, "Spot beams promise satellite communication breakthrough," *IEEE Spectrum*, pp. 36–42, Sept. 1978.
[9] G. Bongiovanni, D. T. Tang, and C. K. Wong, "A general multibeam satellite switching algorithm," *IEEE Trans. Commun.*, vol. COM-29, pp. 1025–1036, July 1981.
[10] G. Bongiovanni, D. Coppersmith, and C. K. Wong, "An optimum time slot assignment algorithm for an SS/TDMA system with variable number of transponders," *IEEE Trans. Commun.*, vol. COM-29, pp. 721–726, May 1981.
[11] I. S. Gopal *et al.*, "An optimal switching algorithm for multibeam satellite systems with variable bandwidth beams," *IEEE Trans. Commun.*, vol. COM-30, pp. 2475–2481, Nov. 1982.
[12] J. L. Lewandowski, J. W. S. Liu, and C. L. Liu, "SS/TDMA time slot assignment with restricted switching modes," *IEEE Trans. Commun.*, vol. COM-31, pp. 149–154, Jan. 1983.
[13] I. S. Gopal, M. A. Bonuccelli, and C. K. Wong, "Scheduling in multibeam satellites with interfering zones," *IEEE Trans. Commun.*, vol. COM-31, pp. 941–951, Aug. 1983.
[14] I. S. Gopal, and C. K. Wong, "Minimizing the number of switchings in an SS/TDMA system," *IEEE Trans. Commun.*, vol. COM-33, pp. 497–500, June 1985.
[15] A. A. Bertossi, G. Bongiovanni, and M. A. Bonuccelli, "Time slot assignment in SS/TDMA systems with intersatellite links," *IEEE Trans. Commun.*, vol. COM-35, pp. 602–608, June 1987.
[16] C. Rose, "Rapid optimal scheduling for time-multiplex switches using a cellular automaton," *IEEE Trans. Commun.*, vol. 37, pp. 500–509, May 1989.
[17] W. S. McCulloch and W. H. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin Math. Biophys.*, vol. 5, p. 115, 1943.
[18] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biolog. Cybern.*, vol. 52, pp. 141–152, 1985.
[19] A. Marrakchi and T. Troudet, "A neural net arbitrator for large crossbar packet-switches," *IEEE Trans. Circuits Syst.*, vol. 36, no. 7, pp. 1039–1041, July 1989.
[20] T. X. Brown, "Neural networks for switching," *IEEE Commun. Mag.*, vol. 27, pp. 72–81, Nov. 1989.
[21] T. X. Brown and K. H. Liu, "Neural network design of a Banyan network controller," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1428–1438, Oct. 1990.
[22] Y. Takefuji and K. C. Lee, "Artificial neural networks for four-coloring map problems and K-colorability problems," *IEEE Trans. Circuits Syst.*, vol. 38, no. 3, pp. 326–333, Mar. 1991.
[23] ——, "A parallel algorithm for tiling problems," *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 143–145, Mar. 1990.
[24] ——, "An artificial hysteresis binary neuron: a model suppressing the oscillatory behaviors of neural dynamics," *Biolog. Cybern.*, vol. 64, pp. 353–356, 1991.
[25] N. Funabiki and Y. Takefuji, "A parallel algorithm for spare allocation problems," *IEEE Trans. Reliability*, vol. 40, no. 3, pp. 338–346, Aug. 1991.
[26] ——, "A parallel algorithm for solving the 'Hip' games," *Neurocomputing*, vol. 3, pp. 97–106, July 1991.
[27] ——, "A parallel algorithm for channel routing problems," *IEEE Trans. CAD/CAS*, vol. 11, no. 4, pp. 464–474, April 1992.
[28] ——, "A parallel algorithm for traffic control problems in three-stage connecting networks," *J. Parallel Distrib. Comput.*, vol. 20, pp. 236–240, Feb. 1994.
[29] K. C. Lee, N. Funabiki, and Y. Takefuji, "A parallel improvement algorithm for the bipartite subgraph problem," *IEEE Trans. Neural Networks*, vol. 3, no. 1, pp. 139–145, Jan. 1992.
[30] N. Funabiki, Y. Takefuji, K. C. Lee, and Y. B. Cho, "A neural network parallel algorithm for clique vertex-partition problems," *Int. J. Electron.*, vol. 72, no. 3, pp. 357–372, Mar. 1992.
[31] N. Funabiki, Y. Takefuji, and K. C. Lee, "A neural network model for finding a near-maximum clique," *J. Parallel Distributed Computing*, vol. 14, no. 3, pp. 340–344, Mar. 1992.
[32] N. Funabiki and Y. Takefuji, "A parallel algorithm for broadcast scheduling problems in packet radio networks," *IEEE Trans. Commun.*, vol. 41, pp. 828–831, June 1993.

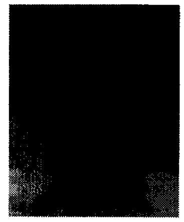
[33] N. Funabiki, Y. Takefuji, and K. C. Lee, "Comparisons of seven neural network models on traffic control problems in multistage interconnection networks," *IEEE Trans. Comput.*, vol. 42, pp. 497–501, Apr. 1993.



Nobuo Funabiki (S'90–M'92) received the B.Sc. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively, and the M.Sc. degree in electrical engineering from Case Western Reserve University, OH, in 1991.

From 1984 to 1994, he was with the System Engineering Division, Sumitomo Metal Industries, Ltd., Japan, where he was engaged in developing advanced control systems in the steel-making process. In 1994, he joined the Department of Information and Computer Sciences at Osake University, Japan, where he is now an assistant professor. His research interests include neural networks and its applications for optimization, signal processing, pattern recognition, and control. He is also interested in the Petri-net theory.

Dr. Funabiki is a member of the IEEE Computer Society.



Yoshiyasu Takefuji received the B.S. degree in 1978, the M.S. degree in 1980, and the Ph.D. degree in 1983 in electrical engineering from Keio University under the supervision of Professor H. Aiso.

He has been a tenured associate professor on the faculty of environmental information at Keio University since April 1992 and also on the faculty of electrical engineering at Case Western Reserve University since 1988. Before joining Case, he taught at the University of South Florida for two years and the University of South Carolina for three years. His research interests focus on neural network parallel computing for solving real-world problems. He is interested in VLSI applications and silicon architecture.

Dr. Takefuji received the National Science Foundation/Research Initiation Award in 1989 and is an NSF advisory panelist. A member of the IEEE Computer Society, ACM, International Neural Network Society, and American Association for the Advancement of Science, he received the Information Processing Society of Japan's best paper award in 1980. He is the author of a book entitled *Neural Network Parallel Computing,* (Kluwer Publishers; January 1992) and coauthored two books *Digital Circuits,* (Ohm-Sha Publishers; 1984) and *Neuro Computing,* (Baifukan Publishers; 1992). He was an Editor of the *Journal of Neural Network Computing* and is currently an associate editor of IEEE TRANSACTIONS ON NEURAL NETWORKS, *Neural/parallel/scientific computations,* and *Neurocomputing,* and a guest editor of *Journal Analog Integrated Circuits and Signal Processing* in the special issue on analog VLSI neural networks and also guest editor of *Neurocomputing* in the special issue on neural network optimization. He has published more than 100 papers. He is included in Who's Who in America and Who's Who in the Midwest. He will edit two books entitled *Neural networks for manufacturing* and *Neural networks for optimization,* from World Scientific Publishers in 1993.