A Neural Network Approach to Topological Via-Minimization Problems

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

Abstract—**Topological via-minimization (TVM) algorithms in two-layer channels based on the artificial neural network model are presented in this paper. TVM problems require not only assigning wires or nets between terminals without an intersection to one of two layers, but also a minimization of the number of vias, which are the single contacts of nets between two layers. The goal of our algorithms is to embed the maximum number of nets without an intersection. Two types of TVM problems are examined: split rectangular TVM (RTVM) problems and split circular TVM (CTVM) problems. Our algorithms require $3n$ processing elements for the n-net split RTVM problems, and $5n$ processing elements for the n-net split CTVM problems. The algorithms were verified by solving seven problems with 20 to 80 nets. The algorithms can be easily extended for more-than-two-layer problems.**

I. Introduction

1. Via-Minimization Problems

VIA-MINIMIZATION problems in two-layer channels are very important in the automatic design of VLSI chips and printed circuit boards (PCB's) [1], [2]. Not only must given wires be assigned to one of two layers without an intersection, but also the number of vias must be minimized. A *via* is a single contact for a wire, which establishes the connectivity between two layers. A via is sometimes called a *through hole* in PCB's. It is also called a *layer contact* in VLSI chips. More vias not only reduce the reliability of products, but also increase the manufacturing cost. Because the physical dimension of vias is usually larger than the wire width, more vias increase the chip size. The two-layer technology using a metal layer and a polysilicon layer has been widely used in VLSI chips, while the current advancement in VLSI chip technology allows us to use four layers using two metal layers and two polysilicon layers [1]. Two-layer via-minimization algorithms, which can be flexibly extended for multi-layer channels, are in strong demand.

In a via-minimization problem, a set of nets are given with a multi-layered channel. A net represents a wire interconnecting terminals which are located on the channel boundary. It is required to embed each net on one of the given layers while any two nets do not intersect each other in any layer. It is also required to minimize the number of vias.

Two kinds of via-minimization problems have been extensively studied [3]-[21]: constrained via-minimization (CVM) problems, and topological via-minimization (TVM) problems. In CVM problems, because the wire location in the channel is already fixed for any net, it is only required to find the layer number of each wire. CVM problems were introduced by Hashimoto *et al.* in 1971 [3]. It was proven that CVM problems are NP-complete when the maximum junction degree is four or more degrees [17]. *Junction degree* refers to the number of wire segments meeting at a single point for the electrical correction. CVM problems are polynomially solvable when the maximum junction degree is less than four [7], [14]. In TVM problems, it is assumed that the wire width and the via size are infinitely small, and no restriction is imposed on the shapes or locations of wires and vias. TVM problems require to find the topological location of each net. TVM problems were introduced by Hsu in 1983 [8]. It was proven that TVM problems are NP-complete even in the simple case where only two-terminal nets are routed in a two-layer channel [19].

In 1989, Rim *et al.* introduced two new TVM problems with only two-terminal nets to be routed in a k-layer channel [20]: split rectangular TVM (RTVM) problems and split circular TVM (CTVM) problems. In split RTVM problems, the channel is a rectangle where the same number of terminals are located on the top side and the bottom side. In split CTVM problems, the channel is a double circle where the same number of terminals are located on the two circles and all the nets must be embedded on the region between the two circles. In both problems, each net has one terminal on one side of the channel and another terminal on the other side. Rim *et al.* showed that split RTVM problems are solvable in $O(kn^2)$ time, and split CTVM problems in $O(n^{2k+1})$ time, where n is the number of nets and k is the number of layers. They proposed the sequential algorithms.

2. Neural Network Approach

In our approach for split RTVM/CTVM problems in two-layer channels, the problems are divided into two steps. In the first step, we maximize the number of nets

Manuscript received January 18, 1991; revised April 30, 1992. This paper was recommended by Associate Editor R. Otten.

N. Funabiki is with the Systems Engineering Division of Sumitomo Metals Industries, Ltd., Amagasaki 660, Japan.

Y. Takefuji is with the Department of Electrical Engineering and Applied Physics, Case Western Reserve University, Cleveland, OH 44106, and with the School of Environmental Information, Keio University, Japan.

IEEE Log Number 9205507.

embedded in the two-layer channel without any intersection. In the second step, we must find the locations of the vias in order to embed the remaining nets. This paper proposes parallel algorithms for the first step of our approach. The algorithms can be easily extended for multilayer channels.

Our algorithms are based on two-dimensional artificial neural network models, which are simplified mathematical models for biological neural networks of human brains. A neural network model consists of a large number of neurons and synapses. A neuron is a simple processing element having an input and an output, which are connected by a nondecreasing function. A synapse represents the interconnection from outputs of several neurons to an input, where the information is transferred.

Hopfield and Tank first introduced neural network models for solving combinatorial optimization problems. They used the sigmoid neuron model as the input/output function [22], where the output V_{ij} of processing element ij is given by

$$V_{ij} = \frac{1}{2}\left(1 + \tan h\left(\frac{U_{ij}}{U_0}\right)\right) \tag{1}$$

where U_{ij} is the input of processing element ij and U_0 is the constant parameter. Because the sigmoid model is slow for the convergence, Takefuji *et al.* have adopted the McCulloch–Pitts model [23] for optimization problems [24]–[34], where V_{ij} is given by

$$\begin{aligned}&\text{if } U_{ij} > 0 \text{ then } V_{ij} = 1\\ &\text{otherwise } V_{ij} = 0.\end{aligned} \tag{2}$$

In order to further improve the convergence, this paper adopts the modified McCulloch–Pitts model [25], where V_{ij} is given by

$$\begin{aligned}&\text{if } U_{ij} > 0 \text{ and } U_{ij} = \max\{U_{iq}\} \text{ for } q = 1, \cdots, m\\ &\text{then } V_{ij} = 1\\ &\text{otherwise } V_{ij} = 0\end{aligned} \tag{3}$$

where m is the number of choices of the layer assignment of net i. As described in the following sections, m is 3 for split RTVM problems, and 5 for split CTVM problems.

In the neural network approach to an optimization problem, first we must define the computational energy function $E(V_{11}, \cdots, V_{nm})$, which represents all the constraints of the problem. Because the solution state of the neural network model has a minimum energy function, we must minimize the energy function by using a motion equation, which is given by

$$\frac{dU_{ij}}{dt} = -\frac{\partial E(V_{11}, \cdots, V_{nm})}{\partial V_{ij}}. \tag{4}$$

It was proven that the motion equation forces the state of the neural network model to converge to the local minimum [26].

There are two kinds of users for sequential/parallel algorithms for via-minimization problems. One user prefers the global minimum solution, where the computation time is a second concern. The other user requests not only a reasonable solution, but also a computation time as short as possible. We propose algorithms for the second user to solve split RTVM/CTVM problems in order to satisfy the requested reasonable solution quality and computation time, while the algorithms find the global minimum solution in more than a 50% frequency.

II. Split RTVM Problems in Two-Layer Channels

1. *System Representation for Split RTVM Problems*

Fig. 1(a) shows a split RTVM problem in a two-layer channel with five nets: (1, 4), (2, 1), (3, 5), (4, 3), and (5, 2). Note that net i (i, n_i) represents the connection between terminal i on the top side of the channel and terminal n_i on the bottom side. For each net, three processing elements are required for the assignment. A total of 15 (=3 × 5) processing elements are required in this problem as shown in Fig. 1(b). Generally, a total of $3n$ processing elements are required for the n-net problems. The nonzero output of processing element ij ($V_{ij} = 1$) for $j = 1$ or 2 indicates that net i is assigned on layer j, while $V_{i3} = 1$ indicates that net i cannot be assigned on any layer without an intersection. The zero output ($V_{ij} = 0$) indicates no assignment.

In order to assign every net, one and only one output among three processing elements for net i, when $i = 1, \cdots, n$, must be nonzero. The energy function E_1 representing this constraint is given by

$$E_1 = \sum_{i=1}^{n}\left(\sum_{q=1}^{3} V_{iq} - 1\right)^2. \tag{5}$$

E_1 is zero if, and only if, every net is assigned on one of the three choices. Fig. 1(b) also shows one solution, where black squares indicate the nonzero output and white squares indicate the zero output. This solution shows that nets 1 and 3 are assigned on layer 1, nets 2 and 5 on layer 2, and net 3 cannot be assigned on any layer. Fig. 1(c) shows the corresponding routing solution.

Any two nets must not intesect each other on any layer. Fig. 2 shows the intersecting conditions of net i (i, n_i) and net p (p, n_p) on layer j for $i \neq p$ and $n_i \neq n_p$. The energy function E_2 rewpresenting this constraint is given by

$$E_2 = \sum_{i=1}^{n}\sum_{j=1}^{2}\sum_{\substack{p=1\\p\neq 1}}^{n} (g(p, i)g(n_i, n_p) + g(i, p)g(n_p, n_i))V_{pj}V_{ij} \tag{6}$$

where $g(x, y)$ is 1 if $x > y$, and $g(x, y)$ is 0 if $x \leq y$. E_2 is zero if, and only if, there is no intersection.

In order to minimize the number of vias, the number of third processing elements having nonzero output must be

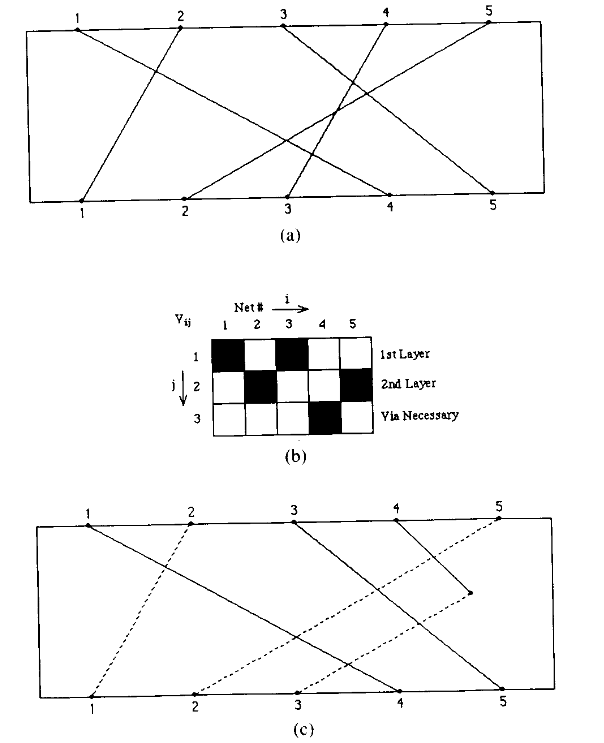


Fig. 1. System representation for a five-net split RTVM problem. (a) A five-net split RTVM problem. (b) 5×3 processing elements for the problem and the convergence to a solution. (c) The corresponding routing solution.

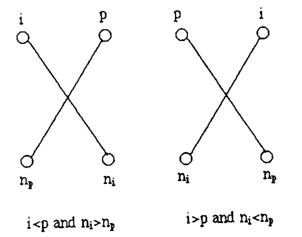


Fig. 2. The intersecting conditions for split RTVM problems.

minimized. It is assumed that the maximum number, N_{not}, of nets, which are not allowed to be embedded on any layer, is provided beforehand. By repeating our algorithm with a smaller N_{not}, it is possible to minimize the number of vias. From this constraint, at most N_{not} third processing elements can have nonzero output. The energy function is given by

$$E_3 = \sum_{i=1}^{n} u\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{p3} - N_{\text{out}} + 1\right) V_{i3} \tag{7}$$

where $u(x)$ is 3 if $x > 0$, and $u(x)$ is 0 if $x \leq 0$. E_3 is zero if, and only if, the number of third processing elements having nonzero output is N_{not} or fewer.

The total energy function E for n-net split RTVM problems is given by

$$\begin{aligned} E &= A/2E_1 + BE_2 + BE_3 \\ &= \frac{1}{A}\sum_{i=1}^{n}\left(\sum_{q=1}^{3} V_{iq} - 1\right)^2 \\ &\quad + B\sum_{i=1}^{n}\sum_{j=1}^{2}\sum_{\substack{p=1\\p\neq i}}^{n} (g(p, i)g(n_i, n_p) \\ &\quad + g(i, p)g(n_p, n_i))V_{pj}V_{ij} \\ &\quad + B\sum_{i=1}^{n} u\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{p3} - N_{\text{out}} + 1\right) V_{i3} \end{aligned} \tag{8}$$

where A and B are constant coefficients. The motion equation of the first and second processing elements ($j = 1$ or 2) for net i is given by

$$\begin{aligned} \frac{dU_{ij}}{dt} &= -A\left(\sum_{q=1}^{3} V_{iq} - 1\right) - B\sum_{\substack{p=1\\p\neq i}}^{n} (g(p, i)g(n_i, n_p) \\ &\quad + g(i, p)g(n_p, n_i))V_{pj}. \end{aligned} \tag{9}$$

The A-term forces one processing element among the three for net i to have nonzero output in order to decide the assignment. The B-term discourages processing element ij from having nonzero output if other nets intersect net i. The motion equation for the third processing element is given by

$$\begin{aligned} \frac{dU_{i3}}{dt} &= -A\left(\sum_{q=1}^{3} V_{iq} - 1\right) \\ &\quad - Bu\left(\sum_{\substack{p=1\\p\neq 1}}^{n} V_{p3} - N_{\text{not}} + 1\right). \end{aligned} \tag{10}$$

The B-term discourages processing element $i3$ to have nonzero output if N_{out} or more processing elements already have nonzero output.

2. *Three Heuristics for the Global Minimum Convergence*

Only the local minimum convergence is proven in the neural network model, although we must consider the global minimum convergence. In order to increase the frequency of the global minimum convergence, the following three heuristics have been empirically introduced [25]:

1. *The hill-climbing heuristic:* the following C-term is added to the motion equations

$$+Ch\left(\sum_{q=1}^{3} V_{iq}\right) \tag{11}$$

where $h(x)$ is 1 if $x = 0$, and $h(x)$ is 0 if $x \leq 0$, and C is a constant coefficient. The C-term encourages processing element ij to have nonzero output if no processing elements for net i have nonzero output. It provides a positive force for the state of the system to escape the local minimum.

2. *The omega function heuristic:* the following two forms of the B-terms are used periodically:

if (t mod T) $< \omega$ then (B-term)

$$= -B \sum_{\substack{p=1\\p\neq i}}^{n} (g(p, i)\, g(n_i, n_p) + g(i, p) \cdot g(n_p, n_i))\, V_{pj} V_{ij}$$

otherwise (B-term)

$$= -B \sum_{\substack{p=1\\p\neq i}}^{n} (g(p, i) g(n_i, n_p) + g(i, p) g(n_p, n_i))\, V_{pj}$$

$$\text{for } j = 1 \text{ or } 2; \tag{12}$$

if (t mod T) $< \omega$ then (B-term)

$$= -Bu \left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{p3} - N_{\text{not}} + 1 \right) V_{i3}$$

otherwise (B-term)

$$= -Bu \left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{p3} - N_{\text{not}} + 1 \right)$$

$$\text{for } j = 3; \tag{13}$$

where t is the number of iteration steps, and ω and T are constant parameters. The omega function heuristic empirically makes a local minimum shallower according to the ratio between T and ω, thus the state of the system can easily escape the local minimum.

3. *The input saturation:* the input of the processing element is confined between two values:

$$\text{if } U_{ij} > U\_\text{max} \text{ then } U_{ij} = U\_\text{max}$$
$$\text{if } U_{ij} < U\_\text{min} \text{ then } U_{ij} = U\_\text{min} \tag{14}$$

where U_max and U_min are constant upper and lower limits of U_{ij} respectively. Because the input saturation reduces the number of possible states of the system, the searching space is limited.

3. Parallel Algorithm for Split RTVM Problems

The following procedure represents our parallel algorithm for n-net split RTVM problems. It maximizes the number of nets embedded on one of two layers without an intersection. The parameter set in step 0 are empirically determined, where T_max1 and T_max2 are the maximum numbers of iteration steps for the global and local minimum convergence respectively.

0) Set $t = 0$, $A = B = 1$, $C = 5$, U_max $= 20$, U_min $= -20$, $N_{\text{not}} = 1$, $T = 10$, $\omega = 5$, T_max1 $= 500$, and T_max2 $= 1000$.

1) Initialize $U_{ij}(t)$ for $i = 1, \cdots, n$ and $j = 1, 2, 3$ by using random numbers between 0 and U_min, and initialize $V_{ij}(t)$ by 0.

2) Use the motion equations with the hill-climbing heuristic and the omega function heuristic to compute $\Delta U_{ij}(t)$, where $N_{\text{not}} = 3$ if $t > T$_max1:

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

$$\Delta U_{ij}(t) = -A\left(\sum_{q=1}^{3} V_{iq}(t) - 1\right) - B \sum_{\substack{p=1\\p\neq i}}^{n} (g(p, i) g(n_i, n_p) + g(i, p) g(n_p, n_i))\, V_{pj}(t) V_{ij}(t) + Ch\left(\sum_{q=1}^{3} V_{iq}(t)\right) \tag{16}$$

for $j = 1$ or 2;
and if $t < 100$ then $\Delta U_{i3}(t) = 0$

$$\text{otherwise } \Delta U_{i3}(t) = -A\left(\sum_{q=1}^{3} V_{iq}(t) - 1\right) - Bu \cdot \left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{p3} - N_{\text{not}} + 1\right) V_{ij}(t) + Ch\left(\sum_{q=1}^{3} V_{iq}(t)\right)$$

if (t mod T) $\geq \omega$ then

$$\Delta U_{ij}(t) = -A\left(\sum_{q=1}^{3} V_{iq}(t) - 1\right) - B \sum_{\substack{p=1\\p\neq i}}^{n} (g(p, i) g \cdot (n_i, n_p) + g(i, p) g(n_p, n_i))\, V_{pj}(t) + Ch\left(\sum_{q=1}^{3} V_{iq}(t)\right) \tag{18}$$

for $j = 1$ or 2;
and if $t < 100$ then $\Delta U_{i3}(t) = 0$

$$\text{otherwise } \Delta U_{i3}(t) = -A\left(\sum_{q=1}^{3} V_{iq}(t) - 1\right) - Bu \cdot \left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{p3} - N_{\text{not}} + 1\right) + Ch\left(\sum_{q=1}^{3} V_{iq}(t)\right). \tag{19}$$

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

$$\Delta U_{ij}(t + 1) = U_{ij}(t) + \Delta U_{ij}(t). \tag{20}$$

4) Use the input saturation heuristic:

$$\text{if } U_{ij}(t+1) > U\_\text{max then } U_{ij}(t+1) = U\_\text{max}$$
$$\text{if } U_{ij}(t+1) < U\_\text{min then } U_{ij}(t+1) = U\_\text{min}. \quad (21)$$

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

$$\text{if } U_{ij}(t+1) > 0 \text{ and } U_{ij}(t+1) = \max\{U_{iq}(t+1)\}$$
$$\text{for } q = 1, \cdots, m \text{ then } V_{ij}(t+1) = 1$$
$$\text{otherwise } V_{ij}(t+1) = 0. \quad (22)$$

6) If all conflicts are resolved, or $t = T\_\text{max2}$, then terminate this procedure, otherwise increment t by 1, and go to step 2.

We examined both the global minimum convergence and the local minimum convergence in simulated problems. Because the global minimum solutions have only one via, we assigned N_{out} as 1 for the global minimum solution, and as 3 for the local minimum solution. We changed N_{out} from 1 to 3 when t is greater than T_max1 in step 2.

4. *Simulation Results and Discussion for Split RTVM Problems*

The simulator has been developed on a Macintosh in order to verify our algorithm. The newly created seven problems shown in Table I were simulated. Figs. 3–5 show one global minimum solution for three of the problems respectively. Our algorithm found several other solutions in the same problems from the different initial values of $U_{ij}(t)$. Table I summarizes the average number of required iteration steps and the convergence frequency in the global/local minimum solutions, and the average number of nets that could not be assigned in any solutions. In each problem, 100 simulation runs were performed from differential initial values of $U_{ij}(t)$ in order to avoid the initial value dependence of neural network models. In split RTVM problems, our simulation results empirically show that the parallel algorithm can embed the maximum number of nets in the two-layer channel in a nearly constant time with $3n$ processors.

III. Split CTVM Problems in Two-Layer Channels

1. *System Representation for Split CTVM Problems*

Fig. 6(a) shows a split CTVM problem in a two-layer channel with five nets: (1, 3), (2, 2), (3, 1), (4, 5), and (5, 4). Note that net i (i, n_i) represents the connection between terminal i on the outside circle of the channel and terminal n_i on the inside circle. Because two routing routes shown in Fig. 6(b), a clockwise route and a counterclockwise route, are available on one layer for each net, a total of four routing routes are available for each net. For each net, five processing elements are required for the assignment. A total of 25 (=5 × 5) processing elements are required in this problem as shown in Fig. 6(c). Generally, a total of $5n$ processing elements are required for the n-net problems. The nonzero output of processing element $i1$ ($V_{i1} = 1$) indicates that net i is assigned on the clockwise route of layer 1, $V_{i2} = 1$ on the counterclockwise route of layer 1, $V_{i3} = 1$ on the clockwise route of layer 2, and $V_{i4} = 1$ on the counterclockwise route of layer 2, while $V_{i5} = 1$ indicates that net i cannot be assigned on any layer route without intersection. The zero output ($V_{ij} = 0$) indicates no assignment.

In order to assign every net, one and only one output among five processing elements for net i for $i = 1, \cdots, n$ must be nonzero. The energy function E_4 representing this constraint is given by

$$E_4 = \sum_{i=1}^{4} \left(\sum_{q=1}^{5} V_{iq} - 1 \right)^2. \quad (23)$$

E_4 is zero if, and only if, every net is assigned in one of five choices. Fig. 6(c) also shows one solution where nets 1 and 4 are assigned on the clockwise route of layer 1, net 2 on the clockwise route of layer 2, net 5 on the counterclockwise route of layer 2, and net 3 cannot be assigned on any layer route. Fig. 6(d) shows the corresponding routing solution.

Any two nets must not be intersected on any layer. Fig. 7 shows the intersecting conditions for clockwise routes of net i (i, n_i) and net p (p, n_p) for $i \neq p$ and $n_i \neq n_p$. The energy function E_5 representing this constraint in clockwise routes is given by

$$\begin{aligned}
E_5 = &\sum_{i=1}^{n} \sum_{j=1,3} \sum_{\substack{p=1\\p\neq i}}^{n} (f(n_i, i) f(p, n_p) g(p, i) g(n_i, n_p) \\
&+ f(n_i, i) g(n_p, p)(g(p, i) + g(n_i, n_p))) V_{pj+1} V_{ij} \\
&+ \sum_{i=1}^{n} \sum_{j=1,3} \sum_{\substack{p=1\\p\neq i}}^{n} (g(i, n_i) f(p, n_p)(g(p, i) \\
&+ g(n_i, n_p)) + g(i, n_i) g(n_p, p)) V_{pj+1} V_{ij} \\
&+ \sum_{i=1}^{n} \sum_{j=1,3} \sum_{\substack{p=1\\p\neq i}}^{n} (f(n_i, i) f(n_p, p)(g(i, p) g(n_p, n_i) \\
&+ g(p, i) g(n_i, n_p)) + f(n_i, i) g(p, n_p) \\
&\cdot (g(i, p) + g(n_p, n_i))) V_{pj} V_{ij} \\
&+ \sum_{i=1}^{n} \sum_{j=1,3} \sum_{\substack{p=1\\p\neq i}}^{n} (g(i, n_i) f(n_p, p)(g(p, i) \\
&+ g(n_i, n_p)) + g(i, n_i) g(p, n_p)(g(p, i) g(n_i, n_p) \\
&+ g(i, p) g(n_p, n_i))) V_{pj} V_{ij}
\end{aligned} \quad (24)$$

TABLE I
SUMMARY OF SIMULATION RESULTS FOR THE SPLIT RTVM PROBLEMS

	No. of Nets n	Convergence to Global Minimum: Average No. of Iteration Steps	Convergence to Global Minimum: Frequency	Convergence to Global Minimum or Local Minimum: Average No. of Iteration Steps	Convergence to Global Minimum or Local Minimum: Frequency	Convergence to Global Minimum or Local Minimum: Average No. of Non-Assigned Nets
Problem 1	20	117.7	99%	121.7	100%	1.0
Problem 2	30	116.4	100%	116.4	100%	1.0
Problem 3	40	124.7	55%	300.6	100%	1.6
Problem 4	50	125.4	98%	125.4	98%	1.0
Problem 5	60	126.1	99%	126.1	99%	1.0
Problem 6	70	129.5	63%	285.3	99%	1.7
Problem 7	80	141.9	68%	275.4	100%	1.6

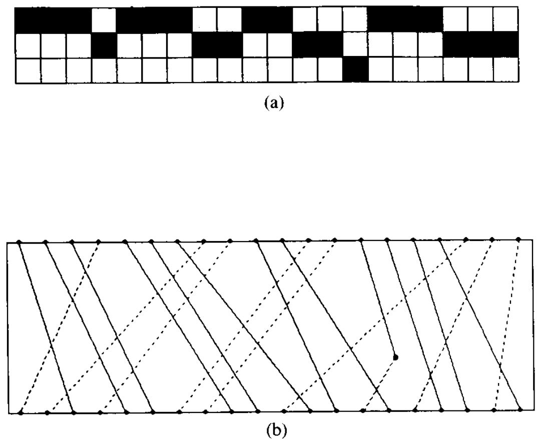


Fig. 3. A simulation result for Problem 1. (a) The convergence of 20×3 processing elements to a solution. (b) The corresponding routing solution.

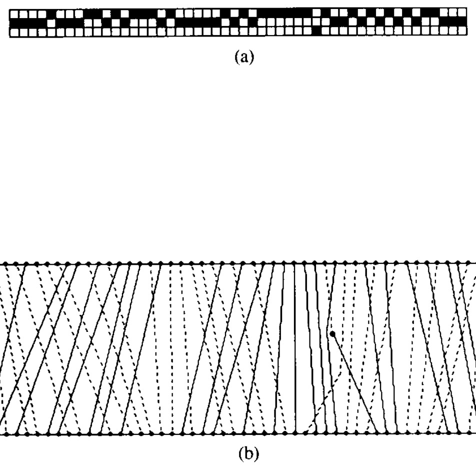


Fig. 4. A simulation result for Problem 4. (a) The convergence of 50×3 processing elements to a solution. (b) The corresponding routing solution.

where $f(x, y)$ is 1 if $x \geq y$, and $f(x, y)$ is 0 if $x < y$. E_5 is zero if, and only if, there is no intersection in clockwise routes. Similarly, the energy function E_6 representing the

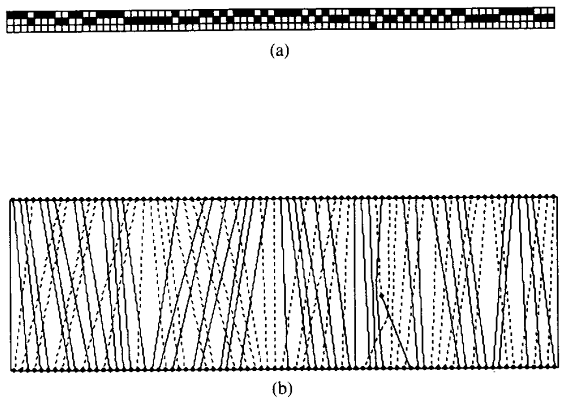


Fig. 5. A simulation result for Problem 7. (a) The convergence of 80×3 processing elements to a solution. (b) The corresponding routing solution.

constraints in counterclockwise routes is given by

$$\begin{aligned} E_6 = &\sum_{i=1}^{n} \sum_{j=2,4} \sum_{\substack{p=1\\p\neq i}}^{n} (f(i, n_i) f(n_p, p) g(i, p) g(n_p, n_i) \\ &+ f(i, n_i) g(p, n_p)(g(i, p) + g(n_p, n_i))) V_{pj-1} V_{ij} \\ &+ \sum_{i=1}^{n} \sum_{j=2,4} \sum_{\substack{p=1\\p\neq i}}^{n} (g(n_i, i) f(n_p, p)(g(i, p) \\ &+ g(n_p, n_i)) + g(n_i, i) g(p, n_p)) V_{pj-1} V_{ij} \\ &+ \sum_{i=1}^{n} \sum_{j=2,4} \sum_{\substack{p=1\\p\neq i}}^{n} (f(i, n_i) f(p, n_p)(g(p, i) g(n_i, n_p) \\ &+ g(i, p) g(n_p, n_i)) + f(i, n_i) g(n_p, p)(g(p, i) \\ &+ g(n_i, n_p))) V_{pj} V_{ij} \\ &+ \sum_{i=1}^{n} \sum_{j=2,4} \sum_{\substack{p=1\\p\neq i}}^{n} (g(n_i, i) f(p, n_p)(g(i, p) \\ &+ g(n_p, n_i)) + g(n_i, i) g(n_p, p)(g(p, i) g(n_i, n_p) \\ &+ g(i, p) g(n_p, n_i))) V_{pj} V_{ij}. \end{aligned} \tag{25}$$

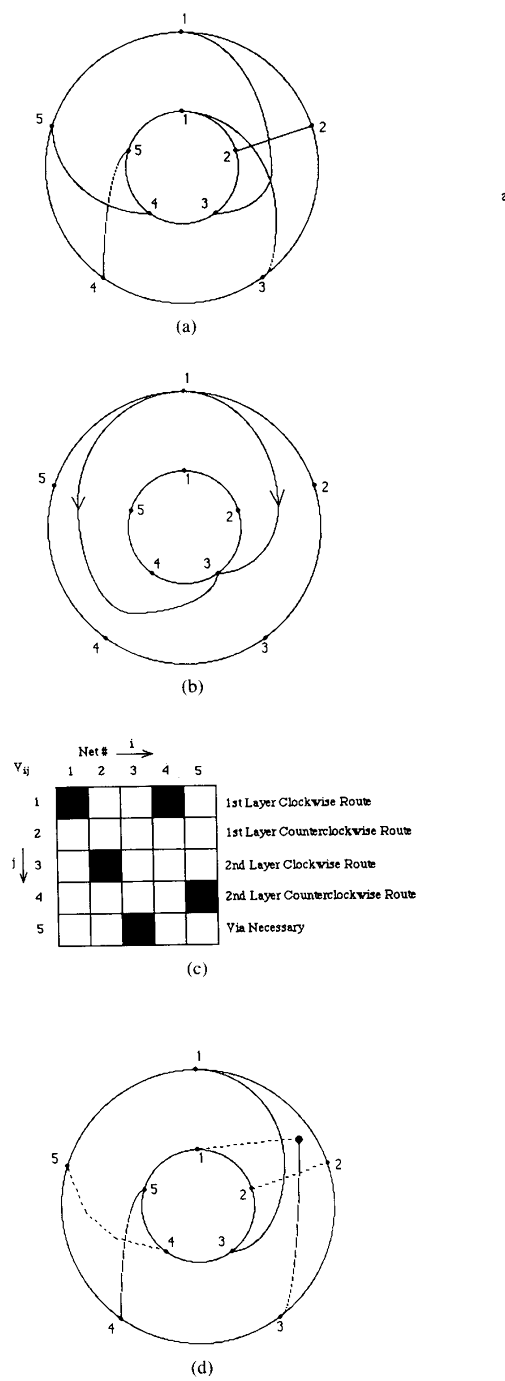

(a)
(b)
Net # i
V_{ij}
j
1st Layer Clockwise Route
1st Layer Counterclockwise Route
2nd Layer Clockwise Route
2nd Layer Counterclockwise Route
Via Necessary
(c)
(d)

Fig. 6. System representation for a five-net split CTVM problem. (a) A five-net split CTVM problem. (b) The two routing routes on one layer. (c) 5×5 processing elements for the problem and the convergence to a solution. (d) The corresponding routing solution.

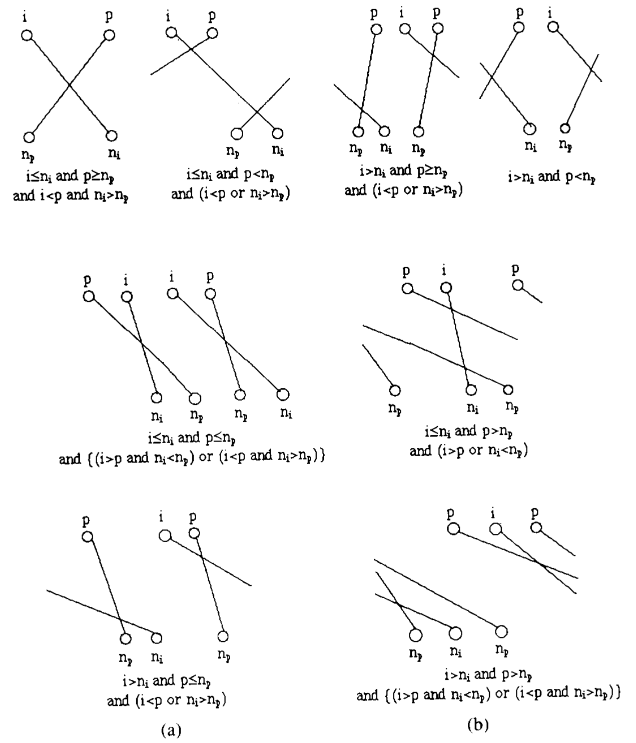

i≤nᵢ and p≥nₚ and i<p and nᵢ>nₚ
i≤nᵢ and p<nₚ and (i<p or nᵢ>nₚ)
i>nᵢ and p≥nₚ and (i<p or nᵢ>nₚ)
i>nᵢ and p<nₚ
i≤nᵢ and p≤nₚ and {(i>p and nᵢ<nₚ) or (i<p and nᵢ>nₚ)}
i≤nᵢ and p>nₚ and (i>p or nᵢ<nₚ)
i>nᵢ and p≤nₚ and (i<p or nᵢ>nₚ)
i>nᵢ and p>nₚ and {(i>p and nᵢ<nₚ) or (i<p and nᵢ>nₚ)}
(a)
(b)

Fig. 7. The intersecting conditions for split CTVM problems. (a) Intersecting conditions between V_{ij} and V_{pj+1} for $j = 1$ or 3, (b) Intersecting conditions between V_{ij} and V_{pj} for $j = 1$ or 3.

E_5 is zero if, and only if, there is no intersection in counterclockwise routes.

The fifth processing element of each net has the same role as the third processing element in split RTVM problems. The energy function E_7 for the fifth processing elements is given by

$$E_7 = \sum_{i=1}^{n} u\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{p5} - N_{\text{not}} + 1\right) V_{i5} \tag{26}$$

where N_{not} is the maximum number of nets that are not allowed to be embedded on any layer. The total energy function E for n-net split CTVM problems is given by

$$E = A/2E_4 + BE_5 + BE_6 + BE_7 \tag{27}$$

where A and B are constant coefficients.

The motion equation of processing element ij for the clockwise route assignment ($j = 1$ or 3) of net i in the n-net problem with the hill-climbing heuristic is given by

$$\frac{dU_{ij}}{dt} = -A\left(\sum_{q=1}^{5} V_{iq} - 1\right)$$
$$- B \sum_{\substack{p=1\\p\neq i}}^{n} (f(n_i, i)f(p, n_p)g(p, i)g(n_i, n_p)$$
$$+ f(n_i, i)g(n_p, p)(g(p, i) + g(n_i, n_p)))V_{pj+1}$$
$$- B \sum_{\substack{p=1\\p\neq i}}^{n} (g(i, n_i)f(p, n_p)(g(p, i) + g(n_i, p))$$
$$+ g(i, n_i)g(n_p, p))V_{pj+1}$$

TABLE II
SUMMARY OF SIMULATION RESULTS FOR THE SPLIT CTVM PROBLEMS

	No. of Nets n	Convergence to Global Minimum: Average No. of Iteration Steps	Convergence to Global Minimum: Frequency	Convergence to Global Minimum or Local Minimum: Average No. of Iteration Steps	Convergence to Global Minimum or Local Minimum: Frequency	Convergence to Global Minimum or Local Minimum: Average No. of Non-Assigned Nets
Problem 8	20	129.1	84%	189.3	100%	1.6
Problem 9	30	125.4	74%	224.4	100%	1.4
Problem 10	40	120.6	83%	186.9	100%	1.3
Problem 11	50	109.8	100%	109.8	100%	1.0
Problem 12	60	116.9	99%	120.9	100%	1.0
Problem 13	70	116.7	100%	116.7	100%	1.0
Problem 14	80	115.3	60%	271.7	100%	1.7

$$
\begin{aligned}
&- B \sum_{\substack{p=1\\p\neq i}}^{n} (f(n_i, i) f(n_p, p)(g(i, p) g(n_p, n_i)\\
&+ g(p, i) g(n_i, n_p)) + g(n_i, i) g(p, n_p)(g(i, p)\\
&+ g(n_p, n_i))) V_{pj}\\
&- B \sum_{\substack{p=1\\p\neq i}}^{n} (g(i, n_i) f(n_p, p)(g(p, i) + g(n_i, n_p))\\
&\quad + g(i, n_i) g(p, n_p)(g(p, i) g(n_i, n_p)\\
&\quad + g(i, p) g(n_p, n_i))) V_{pj} + Ch\left(\sum_{q=1}^{5} V_{iq}\right).
\end{aligned}
\tag{28}
$$

The motion equation of processing element ij for the counterclockwise route assignment ($j = 2$ or 4) of net i is given by

$$
\begin{aligned}
\frac{dU_{ij}}{dt} &= -A\left(\sum_{q=1}^{5} V_{iq} - 1\right)\\
&\quad - B \sum_{\substack{p=1\\p\neq i}}^{n} (f(i, n_i) f(n_p, p) g(i, p) g(n_p, n_i)\\
&\quad + f(i, n_i) g(p, n_p)(g(i, p) + g(n_p, n_i))) V_{pj-1}\\
&\quad - B \sum_{\substack{p=1\\p\neq i}}^{n} (g(n_i, i) f(n_p, p)(g(i, p) + g(n_p, n_i))\\
&\quad + g(n_i, i) g(p, n_p)) V_{pj-1}\\
&\quad - B \sum_{\substack{p=1\\p\neq i}}^{n} (f(i, n_i) f(p, n_p)(g(p, i) g(n_i, n_p)\\
&\quad + g(i, p) g(n_p, n_i)) + f(i, n_i) g(n_p, p)\\
&\quad \cdot (g(p, i) + g(n_i, n_p))) V_{pj}\\
&\quad - B \sum_{\substack{p=1\\p\neq i}}^{n} (g(n_i, i) f(p, n_p)(g(i, p) + g(n_p, n_i))\\
&\quad + g(n_i, i) g(n_p, p)(g(p, i) g(n_i, n_p)\\
&\quad + g(i, p) g(n_p, n_i))) V_{pj} + Ch\left(\sum_{q=1}^{5} V_{iq}\right).
\end{aligned}
\tag{29}
$$

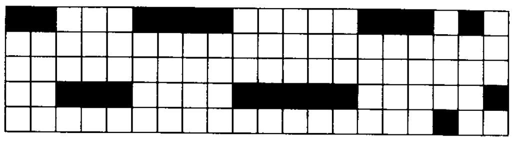
The motion equation of the fifth processing element for net i is given by

$$
\begin{aligned}
\frac{dU_{i5}}{dt} &= -A\left(\sum_{q=1}^{5} V_{iq} - 1\right)\\
&\quad - Bu\left(\sum_{\substack{p=1\\p\neq i}}^{n} V_{p5} - N_{\text{out}} + 1\right) + Ch\left(\sum_{q=1}^{5} V_{iq}\right).
\end{aligned}
\tag{30}
$$

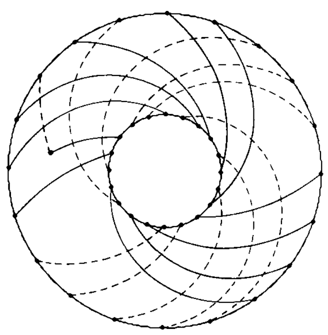
2. Simulation Results and Discussion for Split CTVM Problems

The simulator based on the procedure in Section II.3. has been developed with the motion equations in order to verify our algorithm. We examined the global and local minimum convergence in simulated problems. Because the global minimum solutions have only one via, we assigned N_{out} as 1 for the global minimum solutions, and as 5 for the local minimum solutions.

The newly created seven problems shown in Table II were simulated. Figs. 8–10 show one global minimum solution for three of the problems respectively. Our algorithm found several other solutions in the same problems from the different initial values of $U_{ij}(t)$. Table II summarizes the average number of iteration steps and the convergence frequency in the global/local minimum solutions, and the average number of nets that could not be assigned in any solutions. For each problem, 100 simulation runs were also performed from the different initial values of $U_{ij}(t)$. In split CTVM problems, our simulation results empirically show that our parallel algorithm can

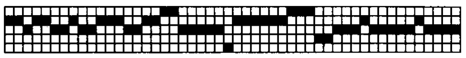


(a)

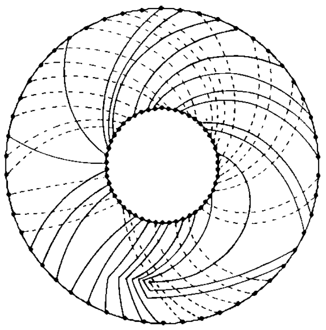


(b)

Fig. 8. A simulation result for Problem 8. (a) The convergence of 20×5 processing elements to a solution. (b) The corresponding routing solution.



(a)



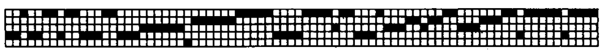
(b)

Fig. 9. A simulation result for Problem 11. (a) The convergence of 50×5 processing elements to a solution. (b) The corresponding routing solution.

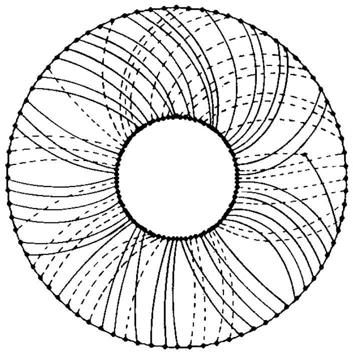
embed the maximum number of nets in the two-layer channel in a nearly constant time with $5n$ processors.

IV. Conclusion

Parallel algorithms for split RTVM problems and for split CTVM problems in two-layer channels are proposed in this paper, based on two-dimensional neural network models. The simulation results empirically show that our algorithms can embed the maximum number of nets in two-layer channels in a nearly constant time, when they run on a parallel machine. The algorithms can be easily modified and extended for solving more-than-two-layer channel problems.



(a)



(b)

Fig. 10. A simulation result for Problem 14. (a) The convergence of 80×5 processing elements to a solution. (b) The correponding routing solution.

References

[1] Mosis user manual, supplements, 1990.

[2] R. L. Geiger, P. E. Allen, and N. R. Strander, *VLSI Design Techniques for Analog and Digital Circuits.* New York: McGraw-Hill, 1990.

[3] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures," in *Proc. 8th Design Automation Workshop*, pp. 155-169, 1971.

[4] K. R. Stevens and W. M. vanCleemput, "Global via elimination in a generalized routing environment," in *Proc. ISCAS*, pp. 689-692, 1979.

[5] M. J. Ciesielski and E. Kinnen, "An optimum layer assignment for routing in ICs and PCBs," in *Proc. 18th Design Automation Conf.*, pp. 733-737, 1981.

[6] R.-W. Chen, Y. Kajitani, and S.-P. Chan, "Topological considerations of the via minimization problem for two-layer PC boards," in *Proc. ISCAS*, pp. 968-971, 1982.

[7] R.-W. Chen, Y. Kajitani, and S.-P. Chan, "A graph-theoretic via minimization algorithm for two-layer printed circuit boards," *IEEE Trans. Circuits Syst.*, vol. CAS-30, pp. 284-299, May 1983.

[8] C.-P. Hsu, "Minimum-via topological routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 235-246, Oct. 1983.

[9] M. Marek-Sadowska, "An unconstrained topological via minimization problem for two-layer routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 184-190, July 1984.

[10] H. Shin and A. Sangiovanni-Vincentelli, "MIGHTY: A 'rip-up and reroute' detailed router," in *Proc. ICCAD*, pp. 2-5, 1986.

[11] K. C. Chang and D. H.-C. Du, "Efficient algorithms for layer assignment problem," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 67-78, Jan. 1987.

[12] K. J. Supowit, "Finding a maximum planar subset of a set of nets in a channel," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 93-94, Jan. 1987.

[13] K. C. Chang and H. C. Du, "Layer assignment problem for three-layer routing," *IEEE Trans. Comput.*, vol. 37, pp. 625-632, May 1988.

[14] Y. S. Kuo, T. C. Chern, and W.-K. Shih, "Fast algorithm for optimal layer assignment," in *Proc. 25th ACM/IEEE Design Automation Conf.*, pp. 554-559, 1988.

[15] X.-M. Xiong and E. S. Kuh, "The constrained via minimization problem for PCB and VLSI design," in *Proc. 25th ACM/IEEE Design Automation Conf.*, pp. 573-578, 1988.

[16] X.-M. Xiong, "A new algorithm for topological routing and via minimization," in *Proc. ICCAD*, pp. 410-413, 1988.

[17] H.-A. Choi, K. Nakajima, and C. S. Rim, "Graph bipartization and via minimization," *SIAM J. Discrete Math.*, vol. 2, pp. 38-47, Feb. 1989.

[18] X.-M. Xiong and E. S. Kuh, "A unified approach to the via minimization problem," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 190–204, Feb. 1989.

[19] M. Sarrafzadeh and D. T. Lee, "A new approach to topological via minimization," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 890-900, Aug. 1989.

[20] C. S. Rim, T. Kashiwabara, and K. Nakajima, "Exact algorithms for multilayer topological via minimization," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 1165–1173, Nov. 1989.

[21] N. J. Naclerio, S. Masuda, and K. Nakajima, "The via minimization problem is NP-complete," *IEEE Trans. Comput.*, vol. 38, pp. 1604–1608, Nov. 1989.

[22] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, pp. 141–152, 1985.

[23] W. S. McCulloch and W. H. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, p. 115, 1943.

[24] Y. Takefuji and K. C. Lee, "A near-optimum parallel planarization algorithm," *Science*, vol. 245, pp. 1221–1223, Sept. 1989.

[25] Y. Takefuji and K. C. Lee, "A parallel algorithm for tiling problems," *IEEE Trans. Neural Networks*, vol. 1, pp. 143–145, Mar. 1990.

[26] Y. Takefuji and K. C. Lee, "Artificial neural networks for four-coloring map problems and K-colorability problems," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 326-333, Mar. 1991.

[27] Y. Takefuji and K. C. Lee, "An artificial hystersis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics," *Biological Cybernetics*, vol. 64, pp. 353–356, 1991.

[28] N. Funabiki and Y. Takefuji, "A parallel algorithm for spare allocation problems," *IEEE Trans. Rel.*, vol. 40, pp. 338–346, Aug. 1991.

[29] K. C. Lee, N. Funabiki, and Y. Takefuji, "A parallel improvement algorithm for the bipartite subgraph problem," *IEEE Trans. Neural Networks*, vol. 3, pp. 139–145, Jan. 1992.

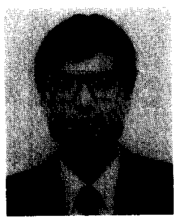
[30] N. Funabiki and Y. Takefuji, "A parallel algorithm for channel routing problems," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 464-474, Apr. 1992.

[31] N. Funabiki and Y. Takefuji, "A parallel algorithm for traffic control problems in three-stage connecting networks," to appear in *J. Parallel Distrib. Comput.*

[32] N. Funabiki, Y. Takefuji, and K. C. Lee, "A neural network model for finding a near-maximum clique," *J. Parallel Distrb. Comput.*, vol. 14, pp. 340-344, 1992.

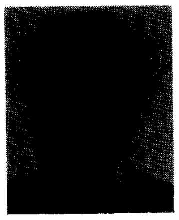
[33] N. Funabiki, Y. Takefuji, K. C. Lee, and Y. B. Cho, "A neural network parallel algorithm for clique vertex-partition problems," *Int. J. Elect.*, vol. 72, pp. 357-372, 1992.

[34] 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, 1993.



Nobuo Funabiki (S'90-M'92) received the B.Sc. degree in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984, and received the M.Sc. degree in electrical engineering from Case Western Reserve University, Ohio, in 1991.

He is currently a senior engineer at the System Engineering Division, Sumitomo Metal Industries, Ltd., in Japan. His research interests include neural network applications for optimization problems and process control problems, and the industrial use of Petri net theory. He has published more than 15 papers.



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, Japan. He has been an Associate Professor of Environmental Information at Keio University, Japan, since April 1992, and has also been on the Electrical Engineering faculty at Case Western Reserve University, Ohio, 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 *Neural Network Parallel Computing*, (Kluwer, 1992) and coauthor of *Digital Circuits* (Ohm-Sha, 1984) and *Neuro Computing* (Baifukan, 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 of Analog Integrated Circuits and Signal Processing* in the special issue on analog VLSI neural networks, and guest editor of *Neurocomputing* in the special issue on neural network optimization. He has published more than 100 papers.