# A Neural Network Parallel Algorithm for Channel Assignment Problems in Cellular Radio Networks

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

*Abstract*—**A parallel algorithm for channel assignment problems in cellular radio networks is presented in this paper. The channel assignment problem involves not only assigning channels or frequencies to each radio cell, but also satisfying frequency constraints given by a compatibility matrix. The proposed parallel algorithm is based on an artificial neural network composed of nm processing elements for an n-cell-m-frequency problem. The algorithm runs not only on a sequential machine but also on a parallel machine with up to a maximum of nm processors. The algorithm was tested by solving eight benchmark problems where the total number of frequencies varied from 100 to 533. The algorithm found the solutions in nearly constant time with nm processors. The simulation results showed that the algorithm found better solutions than the existing algorithm in one out of eight problems.**

## I. Introduction

Recent demand for mobile telephone service has been growing rapidly. At the same time, the electromagnetic spectrum or frequencies allocated for this purpose are limited. This makes solving the problem of channel assignment more and more critical. The channel assignment problem involves efficiently assigning channels or frequencies to each radio cell in the cellular radio network, while satisfying the electromagnetic compatibility constraints.

This paper considers the following three conditions as the electromagnetic compatibility constraints as in [5]:

1) the cochannel constraint: the same frequency cannot be assigned to certain pairs of radio cells simultaneously;
2) the adjacent channel constraint: frequencies adjacent in the frequency domain cannot be assigned to adjacent radio cells simultaneously;
3) the co-site constraint: any pair of frequencies assigned to a radio cell must have certain distance in the frequency domain.

In 1982 Gamst and Rave defined the general form of the channel assignment problem in an arbitrary inhomogeneous cellular radio network [5]. In their definition, the electromagnetic compatibility constraints in an $n$-cell network are described by an $n \times n$ symmetric matrix which is called compatibility matrix $C$. Each nondiagonal element $c_{ij}$ in $C$ represents the minimum separation distance in the frequency domain between a frequency assigned to cell $\#i$ and a frequency to cell $\#j$. The cochannel constraint is represented by $c_{ij} = 1$, and the adjacent channel constraint is represented by $c_{ij} = 2$. $c_{ij} = 0$ indicates that cells $\#i$ and $\#j$ are allowed to use the same frequency. Each diagonal element $c_{ii}$ in $C$ represents the minimum separation distance between any two frequencies assigned to cell $\#i$, which is the co-site constraint, where $c_{ii} \geq 1$ is always satisfied. The channel requirements for each cell in an $n$-cell network are described by an $n$-element vector which is called demand vector $D$. Each element $d_i$ in $D$ represents the number of frequencies to be assigned to cell $\#i$. When $f_{ik}$ indicates the $k$th frequency assigned to cell $\#i$, the electromagnetic compatibility constraints are represented by:

$$|f_{ik} - f_{j1}| \geq c_{ij}, \quad \text{for } i = 1, \cdots, n, j = 1, \cdots, n, k = 1, \cdots, d_i \text{ and } 1 = 1, \cdots, d_j \text{ except } i = j, k = 1. \tag{1}$$

The channel assignment problem in the cellular radio network is finding a conflict-free frequency assignment with the minimum number of total frequencies, where $C$ and $D$ are given.

Consider a channel assignment problem in a four-cell network in [9]. Fig. 1(a) shows the compatibility matrix $C$ and the demand vector $D$. Fig. 1(b) shows the network topology corresponding to the compatibility matrix $C$. The vertex represents a cell, and the edge represents the existence of the cochannel/adjacent constraints between two cells. For example, a frequency within distance 4 from the frequency assigned to cell $\#1$ cannot be assigned to cell $\#2$ because of $c_{12} = c_{21} = 4$. Also any two frequencies assigned to cell $\#4$ must have at least distance 5 because of $c_{44} = 5$. The minimum number of total frequencies in this problem is 11 because cell $\#4$ requires at least 11 ($= 1 + 5 \times 2$) frequencies. Fig. 1(c) shows the optimum solution in this problem, where frequency $\#10$ is assigned to cell $\#1$, frequency $\#3$ to cell $\#2$, frequency $\#4$ to cell $\#3$, and frequencies $\#1$, $\#6$, and $\#11$ to cell $\#4$.

In the simplest form of the channel assignment problem, the cochannel constraint only is considered, and the problem is known to be equivalent to a graph coloring problem [4]. Since the graph coloring problem is known to be NP-complete [1], the computation complexity of searching for the optimum solution in the channel assignment problem grows exponentially with the problem size.

Many researchers have investigated the channel assignment problem in the cellular radio network [2]–[10]. In 1982 Gamst and Rave summarized four existing sequen-

Manuscript received June 24, 1991; revised September 11, 1991, and November 22, 1991.

N. Funabiki is with the System Engineering Division, Sumitomo Metal Industries, Ltd., Amagasaki 600, Japan.

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

IEEE Log Number 9201033.

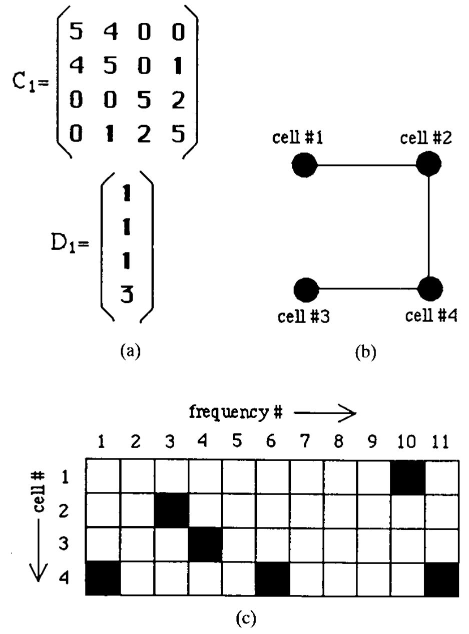


Fig. 1. A four-cell channel assignment problem and the optimum solution. (a) A four-channel assignment problem. (b) The corresponding network topology. (c) The optimum solution with 11 frequencies.

tial approximation algorithms [5]. The first algorithm has four different versions by combining two different assignment strategies—the frequency-exhaustive assignment and the requirement-exhaustive assignment, and two different ordering strategies—the node-degree order and the node-coloring order [2]. The second algorithm repeatedly assigns frequencies according to the assignment difficulty of requirements [3]. The third algorithm uses the heuristic geometric principle of maximum overlap of denial areas. It states that a frequency should be assigned to cell whose denial area has the maximum overlap with the existing denial area of that frequency. The fourth algorithm is based on the graph theory, where the clique number plays a key role. In 1986 Gamst proposed procedures to generate lower bounds on the number of total frequencies [8]. In 1989 Sivarajan *et al.* proposed an $O(n^2)$ time sequential heuristic algorithm, based on the first algorithm introduced by Gamst and Rave [9]. Sivarajan *et al.* applied their algorithm to several problems, where the values of total frequencies in solutions are shown without any actual assignment results.

## II. Neural Network Approach

This paper proposes a parallel algorithm based on the artificial neural network model approach. The neural network model is composed of a large number of massively connected simple processing elements. The processing element is called a neuron because it performs the function of a simplified biological neuron model. A processing element has an input and an output. The input of a processing element is connected with outputs of several processing elements, including the processing element itself.

From several proposed neuron models, the hysterses McCulloch–Pitts neuron model [23] was adopted in this paper where the input/output function is given by:

$$\begin{aligned} V_i &= 1, \quad \text{if } U_i > \text{UTP(upper trip point)} \\ &= 0, \quad \text{if } U_i < \text{LTP(lower trip point)} \\ &\text{unchanged otherwise} \end{aligned} \tag{2}$$

Note that $U_i$ and $V_i$ are the input and the output, respectively, of the $i$th processing element. It has been shown empirically that the hysteresis McCulloch–Pitts neuron model not only provides faster convergence to the solution than the sigmoid neuron model used by Hopfield [11], but also improves the convergence frequency by suppressing the undesirable oscillatory behavior of the McCulloch–Pitts neuron model [15].

The change of the input $U_i$ is given by the partial derivatives of a computational energy function $E(V_1, \cdots, V_n)$ which is called a motion equation:

$$\frac{dU_i}{dt} = -\frac{\partial E(V_1, \cdots, V_n)}{\partial V_i}. \tag{3}$$

Note that $n$ is the number of required processing elements in each problem. The energy function $E$ represents the distance between the current state of the neural network system and the solution state of the neural network system. The energy function is determined by considering all constraints in the problem. The goal of the neural network model for solving combinatorial optimization problems is to minimize the energy function $E$. Theorem 1 in the Appendix states that the motion equation always forces the state of the neural network system, composed of the hysteresis McCulloch–Pitts neurons, to converge to a local minimum [21], [24].

The neural network model for solving a combinatorial optimization problem was first introduced by Hopfield and Tank [11], [12]. Although Wilson and Pawley [13], and Paielli [14] criticized the stability and the solution quality of the Hopfield neural network model, Takefuji and Lee pointed out that the decay term of the motion equation in the Hopfield model disturbs the convergence of the neural network system under some conditions [21]. More than 20 successful neural network applications for solving NP-complete and optimization problems have been reported in the last two years [16]–[36]. In each case, the McCulloch–Pitts neuron model or the hystersis McCollough–Pitts neuron model was used without the decay term.

In 1991 Kunz proposed the first Hopfield neural network model for solving the special case of the channel assignment problem in the cellular radio network [10]. Kunz did not consider the adjacent channel constraint, and limited the co-site constraint to $c_{ii} = 2$. The Kunz neural network model has several disadvantages. First of all, it uses the slow sigmoid neuron model and the harmful decay term in the motion equation. In addition, careful tuning of the coefficients in the motion equation is needed for each different problem, and careful gain control of the sigmoid function has to be provided in order to obtain valid solutions. Also, Kunz did not discuss the convergence frequency of the neural network system which is always controversial in neural network research. These drawbacks of the Kunz model prevent us from solving the general channel assignment problem in the cellular radio network, for practical reasons.

As previously mentioned, the neural network model proposed in this paper is composed of the hystersis McCulloch–Pitts neurons without the decay term. In order to improve the frequency of the global minimum convergence, the four heuristics introduced in Section IV were used. The proposed neural network model was tested by eight benchmark problems where the same set of coefficients and parameters was used in the problems.

## III. Neural Network Representation

The parallel algorithm is based on a two dimensional neural network model. Fig. 2(a) shows the neural network representation for solving the channel assignment problem in Fig. 1. The frequency assignment to each cell requires 11 processing elements because there are 11 candidates or frequencies. A total of 44 ($= 4 \times 11$) are required for this problem. Generally, a total of $nm$ processing elements is required for solving an $n$-cell-$m$-frequency problem, where $n$ is the number of radio cells and $m$ is the total number of frequencies. The output of the $ij$th processing element $V_{ij}$ indicates whether or not frequency $\#j$ is assigned to cell $\#i$. The nonzero output ($V_{ij} = 1$) indicates that frequency $\#j$ is assigned to cell $\#i$. The zero output ($V_{ij} = 0$) indicates that frequency $\#j$ is not assigned to cell $\#i$. Fig. 2(b) shows the solution state of the neural network system, where black squares represent the nonzero output and white squares indicate the zero output.

To satisfy channel requirements, a total of $d_i$ processing elements among $m$ processing elements for cell $\#i$ must have nonzero output, because a total of $d_i$ frequencies are required for cell $\#i$:

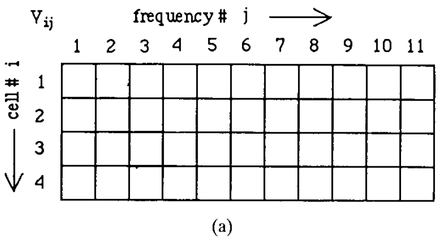
$$\sum_{q=1}^{m} V_{iq} - d_i$$

is zero if and only if $d_i$ processing elements for cell $\#i$ have nonzero output. In the co-site constraint, if frequency $\#q$ within distance $c_{ii}$ from frequency $\#j(|j-q| < c_{ii})$ is assigned to cell $\#i$, frequency $\#j$ must not be assigned to cell $\#i$:

$$\sum_{\substack{q=j-(c_{ii}-1)\\ q\neq j\\ 1\le q\le m}}^{j+(c_{ii}-1)} V_{iq}$$

is nonzero if the assignment of frequency $\#j$ to cell $\#i$ violates the co-site constraint.

In the cochannel constraint and the adjacent channel constraint, if frequency $\#q$ within distance $c_{ip}$ from frequency $\#j(|j-q| < c_{ip})$ is assigned to cell $\#p$ for $c_{ip} > 0$ and $p \neq i$,



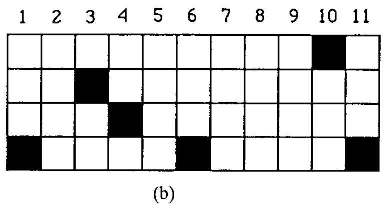


Fig. 2. Neural network representation for the channel assignment problem in Fig. 1. (a) 4 × 11 processing elements for the channel assignment problem in Fig. 1. (b) The convergence of 4 × 11 processing elements to a solution.

frequency $\#j$ must not be assigned to cell $\#i$:

$$\sum_{\substack{p=1\\ p\neq i\\ c_{ip}>0}}^{n} \sum_{\substack{q=j-(c_{ip}-1)\\ 1\le q\le m}}^{j+(c_{ip}-1)} V_{pq}$$

is nonzero if the assignment of frequency $\#j$ to cell $\#i$ violates the cochannel constraint and/or the adjacent channel constraint.

The motion equation of the $ij$th processing element in the $n$-cell-$m$-frequency problem is given by:

$$\frac{dU_{ij}}{dt} = -A\left(\sum_{q=1}^{m} V_{iq} - d_i\right) - B\left(\sum_{\substack{q=j-(c_{ii}-1)\\ q\neq j\\ 1\le q\le m}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1\\ p\neq i\\ c_{ip}>0}}^{n} \sum_{\substack{q=j-(c_{ip}-1)\\ 1\le q\le m}}^{j+(c_{ip}-1)} V_{pq}\right). \tag{4}$$

The first term ($A$-term) forces $d_i$ processing elements among $m$ candidates for cell $\#i$ to have nonzero output, where the corresponding frequencies are assigned to cell $\#i$. The second term ($B$-term) discourages the $ij$th processing element from having nonzero output if the assignment of frequency $\#j$ to cell $\#i$ violates the three constraints mentioned previously. $A$ and $B$ are constant coefficients ($A = B = 1$). The energy function $E$ for the channel assignment problem is given by considering (3) and (4):

$$E = \frac{A}{2}\sum_{i=1}^{n}\left(\sum_{q=1}^{m} V_{iq} - d_i\right)^2 + B\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\sum_{\substack{q=j-(c_{ii}-1)\\ q\neq j\\ 1\le q\le m}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1\\ p\neq i\\ c_{ip}>0}}^{n} \sum_{\substack{q=j-(c_{ip}-1)\\ 1\le q\le m}}^{j+(c_{ip}-1)} V_{pq}\right) V_{ij}. \tag{5}$$

## IV. Heuristics for the Global Minimum Convergence

As shown in the Appendix, only the local minimum convergence is guaranteed 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 four heuristics have been introduced empirically.

1) The $A$-term saturation heuristic: the following function is used for the $A$-term in order to confine the $A$-term between two values:

$$-Af\left(\sum_{q=1}^{m} V_{iq} - d_i\right) \tag{6}$$

where $f(x)$ is $A\_\max$ if $x > A\_\max$, $A\_\min$ if $x < A\_\min$, and $x$ otherwise. $A\_\max$ and $A\_\min$ are the constant upper and lower bounds, respectively of the $A$-term ($A\_\max = 5$, $A\_\min = 5$).

2) The omega function heuristic: two forms of the $B$-term are used periodically in the motion equation:

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

$$-B\left(\sum_{\substack{q=j-(c_{ii}-1)\\ q\neq j\\ 1\leq q\leq m}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1\\ p\neq i\\ c_{ip}>0}}^{n} \sum_{\substack{q=j-(c_{ip}-1)\\ 1\leq q\leq m}}^{j+(c_{ip}-1)} V_{pq}\right) V_{ij}$$

else

$$-B\left(\sum_{\substack{q=j-(c_{ii}-1)\\ q\neq j\\ 1\leq q\leq m}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1\\ p\neq i\\ c_{ip}>0}}^{n} \sum_{\substack{q=j-(c_{ip}-1)\\ 1\leq q\leq m}}^{j+(c_{ip}-1)} V_{pq}\right) \tag{7}$$

where $t$ is the number of iteration steps, and $T$ and $\omega$ are constant parameters ($T = 10, \omega = 5$).

3) The hill climbing heuristic: the following term is added to the motion equation:

$$+Ch\left(\sum_{q=1}^{m} V_{iq} - d_i\right) \cdot (1 - V_{ij}) \tag{8}$$

where $h(x)$ is 1 if $x < 0$, and 0 if $x \geq 0$. $C$ is randomly chosen among 3, 4, and 5 in each iteration step to avoid oscillation due to digital simulation, where otherwise, two or more processing elements continue to have the same states. The $C$-term encourages the $ij$th processing element to have nonzero output if fewer than $d_i$ processing elements for cell $\#i$ have nonzero output and $V_{ij} = 0$.

4) The input saturation heuristic: the input of the processing element is confined between two values:

$$\begin{aligned} U_{ij} &= U\_\max, && \text{if } U_{ij} > U\_\max \\ U_{ij} &= U\_\min, && \text{if } U_{ij} < U\_\min \end{aligned} \tag{9}$$

where $U\_\max$ and $U\_\min$ are the constant upper and lower bounds, respectively, of the input value $U_{ij}$ ($U\_\max = 30$, $U\_\min = -30$).

## V. Parallel Algorithm

The following procedure describes the proposed parallel algorithm based on the motion equation in (4) with the four heuristics, where the first-order Euler method is used. The data set of coefficients and parameters are determined empirically.

1) Set $t = 0$, $A = B = 1$, $C = 3$, 4, or 5, $T = 10$, $\omega = 5$, $U\_\max = 30$, $U\_\min = -30$, $UTP = 5$, $LTP = -5$, and $T\_\max = 500$.

2) Randomize the initial values of input $U_{ij}(t)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$ between 0 and $U\_\min$. Assign the initial values of output $V_{ij}(t)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$ to 0.

3) Compute the change of input $\Delta U_{ij}(t)$ based on the motion equation in (4):

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

$$\begin{aligned}\Delta U_{ij}(t) = &-Af\left(\sum_{q=1}^{m} V_{iq}(t) - d_i\right) \\ &- B\left(\sum_{\substack{q=j-(c_{ii}-1)\\ q\neq j\\ 1\leq q\leq m}}^{j+(c_{ii}-1)} V_{iq}(t) + \sum_{\substack{p=1\\ p\neq i\\ c_{ip}>0}}^{n} \sum_{\substack{q=j-(c_{ip}-1)\\ 1\leq q\leq m}}^{j+(c_{ip}-1)} V_{pq}(t)\right) V_{ij}(t) \\ &+ Ch\left(\sum_{q=1}^{m} V_{iq}(t) - d_i\right)(1 - V_{ij}(t))\end{aligned} \tag{10}$$

else

$$\begin{aligned}\Delta U_{ij}(t) = &-Af\left(\sum_{q=1}^{m} V_{iq}(t) - d_i\right) \\ &- B\left(\sum_{\substack{q=j-(c_{ii}-1)\\ q\neq j\\ 1\leq q\leq m}}^{j+(c_{ii}-1)} V_{iq}(t) + \sum_{\substack{p=1\\ p\neq i\\ c_{ip}>0}}^{n} \sum_{\substack{q=j-(c_{ip}-1)\\ 1\leq q\leq m}}^{j+(c_{ip}-1)} V_{pq}(t)\right) \\ &+ Ch\left(\sum_{q=1}^{m} V_{iq}(t) - d_i\right)(1 - V_{ij}(t))\end{aligned} \tag{11}$$

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

$$U_{ij}(t+1) = U_{ij}(t) + \Delta U_{ij}(t) \tag{12}$$

5) Use the input saturation heuristic in (9):

$$\begin{aligned} U_{ij}(t+1) &= U\_\max, && \text{if } U_{ij}(t+1) > U\_\max \\ U_{ij}(t+1) &= U\_\min, && \text{if } U_{ij}(t+1) < U\_\min. \end{aligned} \tag{13}$$

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

$$\begin{aligned} V_{ij}(t+1) &= 1, && \text{if } U_{ij}(t+1) > \text{UTP} \\ &= 0, && \text{if } U_{ij}(t+1) < \text{LTP} \\ &\text{unchanged otherwise.} \end{aligned} \tag{14}$$

7) Check the termination condition:

$$\text{if } \left( \left( \sum_{q=1}^{m} V_{iq} - d_i \right) = 0 \right) \text{and } (V_{ij}(t) = 1 \text{ and}$$

$$\left( \sum_{\substack{q=j-(c_{ii}-1) \\ q \neq j \\ 1 \leq q \leq m}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1 \\ p \neq i \\ c_{ip}>0}}^{n} \sum_{\substack{q=j-(c_{ip}-1) \\ 1 \leq q \leq m}}^{j+(c_{ip}-1)} V_{pq} \right) = 0),$$

$$\text{for } i = 1, \cdots, n \text{ and } \exists j \in \{1, \cdots, m\} \text{ or } t = T\_\max$$

then terminate this procedure, otherwise increment $t$ by 1, and go on to step 2.

The state of $nm$ processing elements for the $n$-cell-$m$-frequency problem can be updated synchronously or asynchronously. In this paper the synchronous parallel system is simulated on a sequential machine. The synchronous parallel system can be performed on up to a maximum of $nm$ processors. The following procedure/program outlines how to simulate the synchronous parallel system using a sequential machine as if the program were running on the parallel machine:

```
Program parallel-simulator-on-a-sequential-machine
. . . . .
initialization of $U_{ij}$ and $V_{ij}$ for $i$ := 1 to $n$ and for $j$ := 1 to $m$;
. . . . .
{***Main Program***}
while (a set of conflicts is not empty) do
begin
. . . . .
{***Updating all input values***}
    for $i$ := 1 to $n$ do
    for $j$ := 1 to $m$ do
        $U_{ij} := U_{ij} + \Delta U_{ij}$;
{***End of the first loop***}
. . . . .
{***Updating all output values***}
    for $i$ := 1 to $n$ do
    for $j$ := 1 to $m$ do
        If $U_{ij} > 0$ then $V_{ij} := 1$ else $V_{ij} := 0$;
{***End of the second loop***}
. . . . .
end;
{***Main Program end***}
```

In the first loop, all input values $U_{ij}$ are sequentially updated, while all output values $V_{ij}$ are fixed. Then in the second loop, all output values $V_{ij}$ are sequentially updated, while all input values $U_{ij}$ are fixed. It is equivalent to simultaneously updating the values of all inputs and outputs.

## VI. Simulation Results and Discussion

To test the proposed parallel algorithm, the simulator was developed on a Macintosh SE/30 and a Macintosh IIfx. The program was coded in Turbo Pascal. Eight benchmark problems were examined where problems #1, and #3–#8 are taken from [9] and problem #2 is from [10]. Table I shows the specifications of the problems, where the total number of frequencies varied from 11 to 533. Figs. 1 and 3 show the compatibility matrices and the channel requirement vectors for these problems.

TABLE I
Specifications of Simulated Problems

| Problem # | Number of Radio Cells $n$ | Number of Frequencies $m$ | Compatibility Matrix $C$ | Demand Vector $D$ |
|---|---|---|---|---|
| 1 | 4 | 11 | $C_1$ | $D_1$ |
| 2 | 25 | 73 | $C_2$ | $D_2$ |
| 3 | 21 | 381 | $C_3$ | $D_3$ |
| 4 | 21 | 533 | $C_4$ | $D_3$ |
| 5 | 21 | 533 | $C_5$ | $D_3$ |
| 6 | 21 | 221 | $C_3$ | $D_4$ |
| 7 | 21 | 309 | $C_4$ | $D_4$ |
| 8 | 21 | 309 | $C_5$ | $D_4$ |

The number of required frequencies $m$ must be determined before simulating the system. In general, $m$ is roughly determined by multiplication of the value of $c_{ii}$ and the maximum value in the demand vector. Computing the lower bound of required frequencies is equivalent to solving a $k$-colorability problem in a graph which is NP-complete. If a problem is not solved with the value of $m$, we must monotonically increase it until the system finds a solution. Fortunately, in our simulation we were able to start with the lower bounds given from [9], and [10]. In order to accelerate the convergence time in some problems, the frequency assignment was fixed in a certain cell or certain cells with the largest number of required frequencies. For example, in Fig. 1(a) cell #4 has the largest element in the demand vector, so the frequency assignment of cell #4 was fixed in our simulation. Fixing a single cell frequency assignment can drastically reduce the searching space and consequently the convergence time is shortened.

To investigate the computation time and the convergence frequency, 100 simulation runs were performed from different random initial values of $U_{ij}(t)$ for each of the eight problems. Table II summarizes the simulation results, and shows the average number of iteration steps required to converge to the optimum solution and the convergence frequency. Fig. 4 shows the distribution of the number of iteration steps required to converge to the solution in two problems.

In problem #2, our neural network model found the solution within 200 iteration steps, while the Kunz neural network model required 2450 iteration steps to the solution [10]. Our neural network algorithm achieves a significant improvement in the computation time. In problem #8, our simulator found better solutions than the best existing algorithms [9]. This does not mean that our algorithm always finds better solutions than the best existing algorithms. The simulation results in Table II show that our neural network model converged to the solution in a nearly constant number of iteration steps. We conclude that with nm processors, the proposed parallel algorithm finds a solution for an $n$-cell-$m$-frequency channel assignment problem in a cellular radio network, in nearly constant time.

```
2 1 1 0 1 0 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0    10
1 2 1 0 1 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0    11
1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0     9
0 0 1 2 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1     5
1 1 1 0 2 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0     9
0 0 1 0 0 2 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0     4
1 1 1 1 0 1 2 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0     5
1 1 1 1 0 1 1 2 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0     7
1 0 1 1 0 1 1 1 2 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1     4
1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 0 0 0 0 0 1 0 1 0     8
0 0 1 1 1 0 1 1 1 1 2 0 1 1 1 1 0 1 1 1 1 1 1 1 1     8
1 1 1 1 1 0 1 1 1 1 0 2 1 1 0 0 0 0 0 0 0 0 0 0 0     9
1 1 1 1 1 0 1 1 0 1 1 1 2 1 1 1 1 1 1 1 0 0 0 0 0    10
1 1 1 0 1 0 0 0 0 1 1 1 1 2 1 1 1 1 1 1 0 0 0 0 0     7
1 1 0 0 1 0 0 0 0 1 1 0 1 1 2 1 1 1 1 1 1 1 0 0 0     7
0 0 0 0 1 0 0 0 0 1 1 0 1 1 1 2 1 1 1 1 0 0 0 0 0     6
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 2 1 1 0 0 0 0 0 0     4
0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 2 1 1 0 0 0 0 0     5
0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 2 1 1 1 1 0 0     5
0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 1 2 1 1 1 0 0     7
0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 2 1 1 0 0     6
0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 1 1 2 1 1 1     4
0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 2 1 1     5
0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 2 1     7
0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 2     5

                        (a)                          (b)
```

```
5 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0     8
1 5 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0    25
1 1 5 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0     8
0 1 1 5 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0     8
0 0 1 1 5 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0     8
1 0 0 0 0 5 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0    15
1 1 0 0 0 1 5 1 1 0 0 0 1 1 1 1 0 0 1 0 0    18
1 1 1 0 0 1 1 5 1 1 0 0 0 1 1 1 1 0 1 1 0    52
1 1 1 1 0 0 1 1 5 1 1 0 0 0 1 1 1 1 1 1 1    77
0 1 1 1 1 0 0 1 1 5 1 1 0 0 0 1 1 1 0 1 1    28
0 0 1 1 1 0 0 0 1 1 5 1 0 0 0 0 1 1 0 0 1    13
0 0 0 1 1 0 0 0 0 1 1 5 0 0 0 0 0 1 0 0 0    15
0 0 0 0 0 1 1 0 0 0 0 0 5 1 1 0 0 0 0 0 0    31
1 0 0 0 0 1 1 1 0 0 0 0 1 5 1 1 0 0 1 0 0    15
1 1 0 0 0 1 1 1 1 0 0 0 1 1 5 1 1 0 1 1 0    36
1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 5 1 1 1 1 1    57
0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 5 1 1 1 1    28
0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 5 0 1 1     8
0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 5 1 1    10
0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 5 1    13
0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 5     8

                    (c)                     (d)
```

```
7 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0     5
1 7 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0     5
1 1 7 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0     5
0 1 1 7 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0     8
0 0 1 1 7 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0    12
1 0 0 0 0 7 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0    25
1 1 0 0 0 1 7 1 1 0 0 0 1 1 1 1 0 0 1 0 0    30
1 1 1 0 0 1 1 7 1 1 0 0 0 1 1 1 1 0 1 1 0    25
1 1 1 1 0 0 1 1 7 1 1 0 0 0 1 1 1 1 1 1 1    30
0 1 1 1 1 0 0 1 1 7 1 1 0 0 0 1 1 1 0 1 1    40
0 0 1 1 1 0 0 0 1 1 7 1 0 0 0 0 1 1 0 0 1    40
0 0 0 1 1 0 0 0 0 1 1 7 0 0 0 0 0 1 0 0 0    45
0 0 0 0 0 1 1 0 0 0 0 0 7 1 1 0 0 0 0 0 0    20
1 0 0 0 0 1 1 1 0 0 0 0 1 7 1 1 0 0 1 0 0    30
1 1 0 0 0 1 1 1 1 0 0 0 1 1 7 1 1 0 1 1 0    25
1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 7 1 1 1 1 1    15
0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 7 1 1 1 1    15
0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 7 0 1 1    30
0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 7 1 1    20
0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 7 1    20
0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 7    25

                    (e)                     (f)
```

```
7 2 1 0 0 1 2 2 1 0 0 0 0 1 1 1 0 0 0 0 0
2 7 2 1 0 0 1 2 2 1 0 0 0 0 1 1 1 0 0 0 0
1 2 7 2 1 0 0 1 2 2 1 0 0 0 0 1 1 1 0 0 0
0 1 2 7 2 0 0 0 1 2 2 1 0 0 0 0 1 1 0 0 0
0 0 1 2 7 0 0 0 0 1 2 2 0 0 0 0 0 1 0 0 0
1 0 0 0 0 7 2 1 0 0 0 0 2 2 1 0 0 0 0 0 0
2 1 0 0 0 2 7 2 1 0 0 0 1 2 2 1 0 0 1 0 0
2 2 1 0 0 1 2 7 2 1 0 0 0 1 2 2 1 0 1 1 0
1 2 2 1 0 0 1 2 7 2 1 0 0 0 1 2 2 1 1 1 1
0 1 2 2 1 0 0 1 2 7 2 1 0 0 0 1 2 2 0 1 1
0 0 1 2 2 0 0 0 1 2 7 2 0 0 0 0 1 2 0 0 1
0 0 0 1 2 0 0 0 0 1 2 7 0 0 0 0 0 1 0 0 0
0 0 0 0 0 2 1 0 0 0 0 0 7 2 1 0 0 0 0 0 0
1 0 0 0 0 2 2 1 0 0 0 0 2 7 2 1 0 0 1 0 0
1 1 0 0 0 1 2 2 1 0 0 0 1 2 7 2 1 0 2 1 0
1 1 1 0 0 0 1 2 2 1 0 0 0 1 2 7 2 1 2 2 1
0 1 1 1 0 0 0 1 2 2 1 0 0 0 1 2 7 2 1 2 2
0 0 1 1 1 0 0 0 1 2 2 1 0 0 0 1 2 7 0 1 2
0 0 0 0 0 0 1 1 1 0 0 0 0 1 2 2 1 0 7 2 1
0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 2 2 1 2 7 2
0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 2 2 1 2 7

                    (g)
```

Fig. 3. Compatibility matrices and demand vectors in simulated problems. (a) Compatibility matrix $C_2$. (b) Demand vector $D_2$. (c) Compatibility matrix $C_3$. (d) Demand vector $D_3$. (e) Compatability matrix $C_4$. (f) Demand vector $D_4$. (g) Compatibility matrix $C_5$.

## VII. Conclusion

This paper proposes a parallel algorithm for solving channel assignment problems in cellular radio networks. The parallel algorithm is based on a neural network model composed of nm processing elements for an $n$-cell-$m$-frequency problem. The algorithm runs not only on a sequential machine, but also on a parallel machine with up to a maximum of nm processors. Unlike conventional parallel algorithms, our algorithm does not require a rigorous synchronization procedure. Eight benchmark problems were examined in which our algorithm found the solution in nearly constant time with nm processors. In one of the benchmark problems, one algorithm found better solutions than the best existing algorithms.

## Appendix

*Theorem 1:* The system always satisfies $\Delta E/\Delta t \leq 0$ under two conditions such as $\Delta U_i/\Delta t = -\Delta E/\Delta V_i$ and $V_i = f(U_i)$ where $E$ is the computational Liapunov energy function and $f(U_i)$ is a hysteresis binary function:

$$\begin{aligned} f(U_i) &= 1 \text{ if } U_i > \text{UTP} \\ &= 0 \text{ if } U_i < \text{LTP} \\ &\text{unchanged otherwise.} \end{aligned}$$

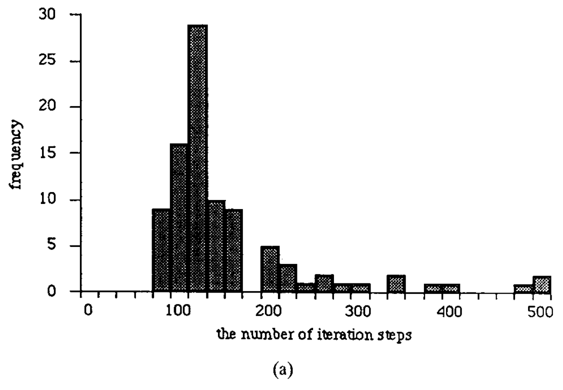
*Proof:* Consider the derivatives of the computational energy $E$ with respect to time $t$.

$$\frac{\Delta E}{\Delta t} = \sum_i \frac{\Delta V_i}{\Delta t}\frac{\Delta E}{\Delta V_i} = \sum_i \frac{\Delta V_i}{\Delta t}\left(-\frac{\Delta U_i}{\Delta t}\right)$$

$$\text{where } \frac{\Delta E}{\Delta V_i} \text{is replaced by} \left(-\frac{\Delta U_i}{\Delta t}\right)$$

$$= -\sum_i \left(\frac{\Delta U_i}{\Delta t}\frac{\Delta V_i}{\Delta U_i}\right)\left(\frac{\Delta U_i}{\Delta t}\right)$$

$$= -\sum_i \left(\frac{\Delta V_i}{\Delta U_i}\right)\left(\frac{\Delta U_i}{\Delta t}\right)^2$$



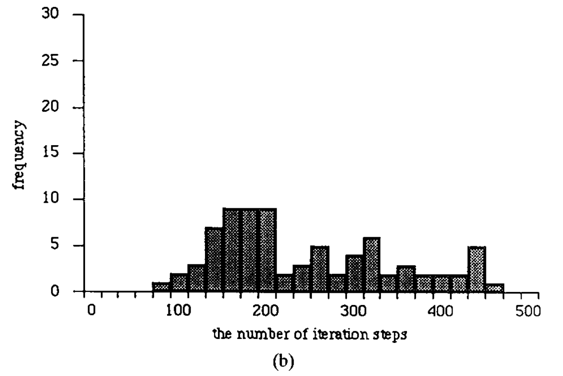


Fig. 4. The distribution of the number of iteration steps required to converge to solutions. (a) Problem #3. (b) Problem #6.

TABLE II
SUMMARY OF SIMULATION RESULTS

| Problem # | Average Number of Iteration Steps to Solutions | Convergence Frequency to Solutions |
|---|---|---|
| 1 | 21.2 | 100% |
| 2 | 294.0 | 9% |
| 3 | 147.8 | 93% |
| 4 | 117.5 | 100% |
| 5 | 100.3 | 100% |
| 6 | 234.8 | 79% |
| 7 | 85.6 | 100% |
| 8 | 305.6 | 24% |

Let $\Delta U_i/\Delta t$ be $U_i(t+\Delta t) - U_i(t)/\Delta t$. Let $\Delta V_i/\Delta U_i$ be $V_i(t+\Delta t) - V_i(t)/U_i(t+\Delta t) - U_i(t)$. It is necessary and sufficient to consider the following four regions:

Region 1: $U_i(t) > \text{UTP}$ and $V_i(t) = 1$
Region 2: $\text{LTP} \leq U_i(t) \leq \text{UTP}$ and $V_i(t) = 1$
Region 3: $\text{LTP} \leq U_i(t) \leq \text{UTP}$ and $V_i(t) = 0$
Region 4: $U_i(t) < \text{LTP}$ and $V_i(t) = 0$.

In region 1: We must consider the four possible cases for $U_i(t+\Delta t)$:

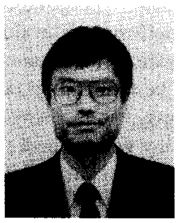
(a) $U_i(t+\Delta t) > U_i(t)$
(b) $\text{LTP} \leq U_i(t+\Delta t) < U_i(t)$
(c) $U_i(t+\Delta t) < \text{LTP} < U_i(t)$
(d) $U_i(t+\Delta t) = U_i(t)$.

In (a) and (b), $V_i(t+\Delta t) = V_i(t) = 1 \Rightarrow \frac{\Delta V_i}{\Delta U_i} = 0$. Therefore $\Delta E/\Delta t = 0$. In (d), $\Delta U_i/\Delta t = 0 \Rightarrow \Delta E/\Delta t = 0$. In (c), $V_i(t+\Delta t) = 0 \Rightarrow \Delta V_i/\Delta U_i = 0 - 1/\text{negative number} > 0$ and $\Delta U_i/\Delta t < 0$. Therefore $\Delta E/\Delta t < 0$. It is concluded that $\Delta E/\Delta t \leq 0$ is always satisfied in region 1. Similarly, in regions 2–4, $\Delta E/\Delta t \leq 0$ is always satisfied. This completes the proof. Q.E.D.

## REFERENCES

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W. H. Freeman, 1979.
[2] J. A. Zoellner and C. L. Beall, "A breakthrough in spectrum conserving frequency assignment technology," *IEEE Trans. Electromag. Compat.*, vol. EMC-19, pp. 313–319, Aug. 1977.
[3] F. Box, "A heuristic technique for assigning frequencies to mobile radio nets," *IEEE Trans. Veh. Technol.*, vol. VT-27, pp. 57–64, May 1978.
[4] W. K. Hale, "Frequency assignment: Theory and applications," *Proc. IEEE*, vol. 68, pp. 1497–1514, Dec. 1980.
[5] A. Gamst and W. Rave, "On frequency assignment in mobile automatic telephone systems," in *Proc. GLOBECOM '82* 1982, pp. 309–315.
[6] A. Gamst, "Homogeneous distribution of frequencies in a regular hexagonal cell system," *IEEE Trans. Veh. Technol.*, vol. VT-31, pp. 132–144, Aug. 1982.
[7] A. Gamst, R. Beck, R. Simon, and E. G. Zinn, "An integrated approach to cellular radio network planning," in *Proc. 35th IEEE Veh. Technol., Soc. Conf.*, pp. 21–25, May 1985.
[8] A. Gamst, "Some lower bounds for a class of frequency assignment problems," *IEEE Trans. Veh. Technol.*, vol. VT-35, pp. 8–14, Feb. 1986.
[9] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchum, "Channel assignment in cellular radio," in *Proc. 39th IEEE Veh. Technol. Soc. Conf.*, May 1989, pp. 846–850.
[10] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 188–193, Feb 1991.
[11] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybern.*, vol. 52, pp. 141–152, 1985.
[12] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: A model," *Science*, vol. 233, pp. 625–633, Aug. 1986.
[13] G. V. Wilson and G. S. Pawley, "On the stability of the traveling salesman problem algorithm of Hopfield and Tank," *Biological Cybern.*, vol. 58, pp. 63–70, 1988.
[14] R. A. Paielli, "Simulation tests of the optimization method of Hopfield and Tank using neural networks," NASA Tech. Mem. 101047, Nov. 1988.
[15] W. S. McCulloch and W. H. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.
[16] Y. Takefuji and K. C. Lee, "A near-optimum parallel planarization algorithm," *Science*, 245, pp. 1221–1223, Sept. 1989.
[17] Y. Takefuji and K. C. Lee, "A parallel algorithm for tiling problems," *IEEE Trans. Neural Networks*, vol. 1, pp. 143–145, Mar 1990.
[18] Y. Takefuji, C. W. Lin, and K. C. Lee, "A parallel algorithm for estimating the secondary structure in Ribonucleic Acids," *Biological Cybern.*, vol. 63, no. 5, pp. 337–340, 1990.
[19] Y. Takefuji, L. L. Chen, K. C. Lee, and J. Huffman, "Parallel algorithms for finding a near-maximum independent set of a circle graph," *IEEE Trans. Neural Networks*, vol. 1, pp. 263–267, Sept. 1990.
[20] Y. Takefuji and K. C. Lee, "A super parallel sorting algorithm based on neural networks," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1425–1429, Nov. 1990.
[21] ——, "Artificial neural networks for four-coloring map problems and K-colorability problems," IEEE Trans. Circuits Systems, Vol. 38, 3, pp. 326–333, March 1991.
[22] T. Kurokawa *et al.*, "CMOS layout design of the hysteresis McCulloch-Pitts neuron," *Electron. Lett.*, vol. 26, no. 25, pp. 2093–2095, Dec. 1990.
[23] Y. Takefuji and K. C. Lee, "An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics," *Biological Cybern.*, vol. 64, pp. 353–356, 1991.
[24] N. Funabiki and Y. Takefuji, "A parallel algorithm for spare allocation problems," *IEEE Trans. Reliability*, vol. 40, pp. 338–346, Aug. 1991.
[25] ——, "A parallel algorithm for channel routing problems," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 464–474, Apr. 1992.
[26] ——, "A parallel algorithm for solving "Hip" games," *Neurocomput.*, vol. 3, pp. 97–106, 1991.
[27] ——, "A parallel algorithm for traffic control problems in TDM hierarchical switching systems," *IEEE Trans. Commun.*

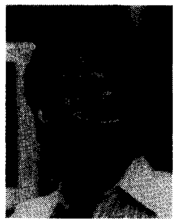
[28] ——, "A parallel algorithm for traffic control problems in three-stage connecting networks," *J. Parallel Distrib. Comput.*, to appear.
[29] ——, "A parallel algorithm for broadcast scheduling problems in packet radio networks," *IEEE Trans. Commun.*, to be published.
[30] Y. Takefuji, T. Tanaka, and K.C. Lee, "A parallel string search algorithm," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 332–336, Mar./Apr. 1992.
[31] Y. Takefuji, "Neural network parallel algorithms for combinatorial optimization problems," *J. Intelligent Manuf.*,vol. 5, 1992
[32] 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.
[33] K.C. Lee and Y. Takefuji, "Neural network computing for knight's tour problems," *Neurocomput.,* vol. 4, no. 5, pp. 249–254, 1992.
[34] N. Funabiki, Y. Takefuji, and K.C. Lee, "A neural network parallel algorithm for clique vertex-partition problems," *Int. J. Electron.*, vol. 72, no. 3, pp. 357–372, 1992.
[35] K.C. Lee and Y. Takefuji, "A generalized maximum neural network for the module orientation problem," *Int. J. Electron.*, vol. 72, no. 3, pp. 331–355, 1992.
[36] S.C. Amartur, D. Piraino, and Y. Takefuji, "Optimization neural networks for the segmentation of magnetic resonance images," *IEEE Trans. Medical Imaging,* vol. 11, pp. 215–220, 1992.



**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 the M.Sc. degree in electrical engineering from Case Western Reserve University, Cleveland, OH, in 1991.

He is currently a Senior Engineer at the System Engineering Division, Sumitomo Metal Industries, Ltd., Japan. His research interests include neural network applications for optimization problems and process control problems, and the industrial use of Petri net theory.

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



**Yoshiyasu Takefuji** received the B.S., M.S., and Ph.D. degrees in electrical engineering from Keio University, Japan, in 1978, 1980, and 1983, respectively.

He taught at the University of South Florida and the University of South Carolina before joining the Department of Electrical Engineering and Applied Physics, Case Western Reserve University, Cleveland, OH, in 1988. Since April 1992 he has also been an Associate Professor at Keio University, Japan. His research interests focus on neural network parallel computing for solving real-world problems. He is also interested in VLSI applications and silicon architecture.

Dr. Takefuji received the Information Processing Society of Japan's best paper award in 1980. He coauthored two books, *Digital Circuits* (in Japanese, Ohm-Sha Publishers) in 1984 and *Neural Network Computing* (in Japanese, Baifukan Publishers) in 1992. He received the National Science Foundation/Research Initiation Award and Darpa Award in 1989 and is an NSF Advisory panelist. He was an Editor of the *Journal of Neural Network Computing,* is an Associate Editor of the *IEEE Transactions on Neural Networks,* and is a Guest Editor of a special issue on analog VLSI neural network *International Journal of Analog Integrated Circuits and Signal Processing.* He has published more than 90 papers and is the author of a book, *Neural Networth Parallel Computing* (Kluwer Academic Publishers). He is a member of the IEEE Computer Society, ACM, the International Neural Network Society, and the American Association for the Advancement of Science.