A Hopfield Neural Network for Channel Assignment Problem in Cellular Radio Networks

Omid Moradi

ACECR(jahad-e-daneshgahi), Isfahan University of technology

Isfahan, Iran

Tel: 98-91-3103-2136 E-mail: moradi.acecr@gmail.com

Abstract

In wireless mobile communication system, radio spectrum is limited resource. However, efficient use of available channels has been shown to improve the system capacity. The role of a channel assignment scheme is to allocate channels to calls or mobiles in such a way as to minimize call blocking or call dropping probabilities, and also to maximize the quality of service. Channel assignment is known to be an NP-hard optimization problem. In this paper, a new channel-assignment algorithm using a modified Hopfield neural network is proposed. The channel-assignment problem is formulated as an energy-minimization problem that is implemented by a modified discrete Hopfield network. In this algorithm, an energy function is derived, and the appropriate interconnection weights between the neurons are specified. The interconnection weights between the neurons are designed in such a way that each neuron receives inhibitory support if the constraint conditions are violated and receives excitatory support if the constraint conditions are satisfied. The algorithm will be tested by solving seven benchmark problems, where the total number of frequencies varied from 73 to 533. This new algorithm, together with the proposed regular interval initialization and new interconnection weights, has better performance results than the existing algorithms in all of the seven problems.

Keywords: Wireless communication, Channel assignment, Neural network

1. Introduction

Recently, the demands for cellular mobile communication systems have rapidly increased due to the portability and the availability of the systems. On the other hand, the electromagnetic frequency spectrum allocated for this purpose has been limited. As a result, the efficient use of the frequency spectrum or channels has gained the importance in order to meet the increasing demands. The channel assignment problem has been extensively studied to solve this important task in cellular mobile communication systems.

The channel assignment problem (CAP) in this paper is based on a common model. The service area of the system is divided into a large number of hexagonal cells. A cell composes a unit area to provide communication services, where every user is located in one cell. When a user requests a call for this system, a channel or frequency spectrum is assigned there to provide the communication service. This channel assignment must satisfy the constraints to avoid the radio interference between channels. Funabiki, Okutani and Nishikawa (2000) states that “ Three types of constraints have usually been considered in CAP.

1) *The Cochannel Constraint (CCC):* The same channel cannot be reused in the cells within a certain distance from each other. A set of channel-reuse forbidden cells is called a cluster, where a different channel must be assigned to every call.

2) *The Adjacent Channel Constraint (ACC):* Adjacent channels cannot be assigned to adjacent cells simultaneously. In other words, any pair of channels in adjacent cells must have a specified distance. Note that the distance indicates the difference in the channel domain.

3) *The Cosite Constraint (CSC):* Any pair of channels in the same cell must have a specified distance. this distance for *CSC* is usually larger than that for *ACC*.

The goal of CAP is to find a channel assignment to every requested call with the minimum number of channels subject to the above three constraints.’’(p.397)

1.1 Problem formulation of CAP

Gamst and Rave(1982) defined the general form of the channel assignment problem in an arbitrary inhomogeneous cellular radio network. 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 (*CCC*) is represented by c_{ij} = 1, and the adjacent channel constraint (*ACC*) 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 cosite constraint (*CSC*), where $c_{ij} \geq 1$ 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 required channel numbers(RCN's) 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:

$$\begin{array}{ll} \left|f_{ik} - f_{jl}\right| \geq c_{ij}, \ \ for & i = 1,..., N , j = 1,..., N \\ & k = 1,..., d_i \\ and & \\ & l = 1,..., d_i \\ except & \\ & i = j , k = l. \end{array} \qquad (1)$$

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(pp.309-315). Consider a channel assignment problem in a four-cell network proposed by He *et al.*(2002):

> Fig. 1 shows the compatibility matrix *C*, The demand vector *D*, and the corresponding networks topology, as well as that of several interference-free optimum solutions with 11 channels. The network topology corresponds to the compatibility matrix *C*. The vertex represents a cell, and the edge represents the existence of *CCC* or *ACC* between two cells. The diagonal term c_{ii}=5 indicates that any two channels assigned to cell #*i* must be at least five channels apart in order to satisfy *CSC*. Channels assigned to cells #1 and #2 must be at least c_{12}=4 channels apart. Off-diagonal terms of c_{ij}=1 and c_{ij}=2 correspond to *CCC* and *ACC*, respectively. The CAP, as demonstrated by using this example, tries to find a conflict-free channel assignment that satisfies the constraint conditions with the minimum number of total channels with given *C* and *D*. Suppose that *M* represents the number of channels available. Why is the minimum number of channels needed for an interference-free assignment 11 in this example? From Fig.1, because cell #4 requires at least 11(=1+5×2) channels, the minimum number of channels needed for an interference-free assignment in this example is 11. Thus, *M*=11 is the lower bound, and we will be unable to find any interference-free assignments if *M*<11 (pp.1387-1388).

In this paper, an energy function is derived, which represents the constraints that should be satisfied in order to find the best assignment. The appropriate interconnection weights are designed such that the constraints on the channel-assignment problem are expressed in terms of inhibitory connections between neurons. A 2-D discrete Hopfield network is implemented to minimize the energy function. The state of each neuron in the Hopfield network represents the possibility of assigning a certain channel to a cell. To escape the local minima during the solution search, the assigned channel numbers (ACN's) in a cell are checked, and if the ACN's are less than the RCN's, then a forced frequency-assignment method is used. The forced frequency assignment method is an additional assignment of one or more channels although the energy is increased by the additional channel assignment since the extra assignment has violated the constraints. This simple technique dramatically increases the convergence rate of assignment since most of the local minima are due to a lack of assigned frequencies, which satisfy all three constraints for each cell. Initialization method and updating technique, which use the specific characteristics of the channel-assignment problem (e.g., *CSC*, *CCC*, and *ACC*) in cellular radio networks are investigated to reduce the number of iterations and to increase the convergence rate of channel assignment. The proposed frequency-assignment technique uses a modified discrete Hopfield network to achieve a solution that simultaneously satisfies all the constraints by using new interconnection weights and external inputs. In this paper channel assignment to the calls in a cell, which determines the lower bound on the total number of channels are fixed after regular interval initialization method.

Many researchers (Sivarajan, McEliece, and Ketchum, 1989; Gamast and Rave, 1982; Kunz, 1991; Funabiki and Takefuji, 1992; Chan, Palaniswami and Everitt, 1994; Kim *et al*., 1997; Funabiki, Okutani and Nishikawa, 2000;

Vidyarthi, Ngom and Stojmenovic, 2005) have investigated the CAP in telephone networks. Sivarajan, McEliece and Ketchum (1989) proposed an *0*(n2) time sequential heuristic algorithm, based on the first algorithm introduced by Gamst and Rave (1982). Sivarajan, McEliece and Ketchum (1989) applied their algorithm to several problems, where the values of total frequencies in solutions are shown without any actual assignment results(pp.846-850). Kunz (1991) used the continuous Hopfield network, where the output of each neuron V_i was a fixed function *f* of the internal state u_i , i.e., $V_i=f(u_i)$, where $f(x)=1/2(1+tanh(\lambda x))$. The Kunz neural-network model required a large number of iterations in order to reach the final solution, and there were also difficulties in finding the proper values for and the parameters in the interconnection weights and energy function. Kunz (1991) considered cochannel and cosite interference in his neural-network model(pp.188-193). Funabiki and Takefuji (1992) proposed a neural-network parallel algorithm for channel-assignment problem. All input values are sequentially updated, while all output values are fixed. Then, all output values are sequentially updated, while all input values are fixed. Their neural-network model is composed of the hysteresis McCulloch–Pitts neurons. In the Funabiki and Takefuji model(1992), four heuristics were used to improve the convergence rate of channel assignment. They also fixed the channel assignment in one or more certain cells in order to accelerate the convergence time. The results were favorable in some cases, but not in others (PP.430-436). Chan, Palaniswami and Everitt (1994) used a feed forward neural network, which had a learning process prior to actual channel assignment. For the learning process, they used training data that was dependently obtained by other assignment methods. The performance of their algorithm is totally dependent on the used training data. Also, only the cochannel constraint (CCC) was considered (PP.279-288). Kim *et al.*(1997) used a modified Hopfild network. In their algorithm, used new interconnection weights an external inputs to the neurons with the initialization and updating methods(pp.957-967). Funabiki *et al.* (2000) proposed a three-stage algorithm of combining sequential heuristic methods into a parallel neural network for the NP-complete channel assignment problem(p.397). Vidyarthi *et al.*(2005) proposed a hybrid channel assignment approach using an efficient evolutionary strategy. They developed an evolutionary strategy(ES) which optimized the channel assignment(pp.1887-1895).

In this paper, a modified discrete Hopfield neural-network algorithm for the channel-assignment problem is proposed in order to improve the convergence rate and to reduce the number of iterations. My experimental results are also compared with that of Kim *et al.*(1997). In this paper, the channel-assignment problem is formulated as an energy-minimization problem such that the energy is at its minimum when all the constraints are satisfied and the number of assigned frequencies are the same as the required channel numbers (RCN's) in each cell.

2. Modified Hopfield Network for Channel Assignment

Each processing element (neuron) is fully interconnected in the Hopfield network. The *i*th neuron is described by its state, which is denoted by V_i. Each neuron has two possible states. The value of each state is determined by the total input from other neurons followed by a thresholding operation. The input of the *i*th neuron is derived from two sources: the outputs of other neurons scaled by the connection weights and an appropriate external input. The total input to neuron #*i* is denoted by U_i

$$U_i = \sum_j W_{ij} V_j + I_i \tag{2}$$

where W_{ij} is the connection weight from neuron #*j* to neuron #*i* and I_i is the external input. Each neuron updates its own state according to a thresholding rule with threshold *THD* as shown by

$$V_i = \begin{cases} 1 & if\, U_i > THD \\ 0 & otherwise \end{cases} \tag{3}$$

The thresholding rule can be applied asynchronously (in series) or synchronously (in parallel). In the asynchronous mode, this rule is applied sequentially to each neuron, and the state of each neuron is updated individually. In the synchronous mode, this thresholding operation is simultaneously applied to every neuron, and the states of all neurons are updated at the same time. The updating operation is terminated when the states are unchanged or the energy has reached a minimum value. According to Kim *et al.*(1997), The energy function *E* is defined as

$$E = -\frac{1}{2}\sum_i \sum_j W_{ij} V_i V_j - \sum_i V_i I_i \tag{4}$$

This energy function is minimized by the Hopfield neural network updating procedure. The successive application of the updating procedure will force the network to converge such that the energy of the network becomes smaller during the updating procedure. When the network reaches a stable state, it has fallen into minimum energy state, where this could be a local or global minimum. W_{ij} and I_i should be set appropriately for the applications so that E represents the function, which is minimized to solve the combinatorial optimization problems. The energy function should represent all the constraints of the problem.

In my algorithm, a discrete Hopfield network is implemented, which is updated until the energy is equal to zero or a predefined maximum iteration number has been reached. In the simulation, I considered a mobile radio network that has N cells (or base stations) and M available frequency channels. The value of M is set to be the lower bound of required number of channels LB for a given channel assignment problem. Fig.2 shows a two-dimensional (2-D) Hopfield network for the channel-assignment problem. Each ith cell can carry any of the d_i frequencies among the M frequencies if the carriage of frequencies does not violate the imposed constraints of the channel-assignment problem. The value of the processing unit V_{ij} , $1 \leq i \leq N$ and $1 \leq j \leq M$ indicates if frequency #*j* is assigned to cell #*i*: $V_{ij}=1$ meaning that the frequency j is assigned to cell #*i* and $V_{ij}=0$, which means that the frequency #*j* cannot be assigned to cell #*i.* In Fig.2, i and p represent the cell number; j and q indicate the frequency number, respectively. The state of the current processing neuron to be updated is represented by V_{ij}. An energy function is derived to represent the three constraints of the channel-assignment problem. In *CSC*, a frequency f_{iq} cannot be assigned to cell #*i* if the distance of f_{iq} and any assigned frequency f_{ij}, $1 \leq j \leq M$ is less than c_{ii} (minimum frequency distance for *CSC*). The energy function for cell #*i* (for *CSC*) can be defined as

$$E^i_{CSC} = \sum_{j=1}^{M}\sum_{q=1}^{M} V_{ij} V_{iq} X_{ijq} \tag{5}$$

where

$$X_{ijq} = \begin{cases} 1, & if\ q \neq j \quad and \\ & j-(c_{ii}-1) \leq q \leq j+(c_{ii}-1) \\ 0, & otherwise. \end{cases} \tag{6}$$

For both *ACC* and *CCC*, the frequency f_{ij} cannot be assigned to cell #*i* if the distance of f_{ij} and any other assigned frequency f_{pq} is less than c_{ip} (minimum frequency separation between the frequencies in cell #*i* and cell #*p*). The energy function of *i*th cell for *ACC* and *CCC* can be defined as

$$E^i_{(ACC,CCC)} = \sum_{j=1}^{M}\sum_{p=1}^{N}\sum_{q=1}^{M} V_{ij} V_{pq} Y_{ijpq} \tag{7}$$

Where

$$Y_{ijpq} = \begin{cases} 1, & if\ p \neq i \quad and \\ & c_{ip} > 0\, and\ j-(c_{ip}-1) \leq q \leq j+(c_{ip}-1) \\ 0, & otherwise. \end{cases} \tag{8}$$

In addition to the three constraint conditions, the total ACN's in the *i*th cell must be the same as the RCN's for cell #*i.* The energy function for the ACN can be defined as

$$E^i_{ACN} = \left(d_i - \sum_{j=1}^{M} V_{ij} \right)^2 \tag{9}$$

From (5), (7), and (9), the energy function for *i*th cell can be defined as

$$E^i = \left(d_i - \sum_{j=1}^{M} V_{ij} \right)^2 + \sum_{j=1}^{M}\sum_{q=1}^{M} V_{ij} V_{iq} X_{ijq} + \sum_{j=1}^{M}\sum_{p=1}^{N}\sum_{q=1}^{M} V_{ij} V_{pq} Y_{ijpq}. \tag{10}$$

The total energy function for the channel-assignment problem is as follows:

$$E = \sum_{i=1}^{N} \left(\left(d_i - \sum_{j=1}^{M} V_{ij} \right)^2 + \sum_{j=1}^{M}\sum_{q=1}^{M} V_{ij} V_{iq} X_{ijq} + \sum_{j=1}^{M}\sum_{p=1}^{N}\sum_{q=1}^{M} V_{ij} V_{pq} Y_{ijpq} \right). \tag{11}$$

The channel-assignment problem is now formulated as an energy-minimization problem. This energy function can be minimized by an equivalent Hopfield neural network with the appropriate interconnection weights and the external inputs. The interconnection weights must represent the constraints of the optimization problem such as the RCN's for each cell and the three constraint of the channel-assignment problem. Each of the constraints are invoked by inhibitory and excitatory support. If neuron V_{ij} takes a value of unity, which means frequency f_{ij} can be used at cell #*i*, then the neuron V_{iq} within the interference must be inhibited by the *CSC* condition. The constraint can be specified in the form

$$W_{CSC} = -\delta_{ip}\alpha_{jq}(c_{ii}) \tag{12}$$

where δ is the Kronecker delta function and defined with α_{ij} as follows:

$$\delta_{ij} = \begin{cases} 1. & if\ i = j \\ 0 & otherwise \end{cases} \tag{13}$$

$$\alpha_{ij}(x) = \begin{cases} 1, & if\left|i - j\right| < x \\ 0, & otherwise. \end{cases} \tag{14}$$

In both *ACC* and *CCC*, if neuron V_{ij} takes a value of unity, then a neuron V_{iq} within the interference must be inhibited. This constraint can be specified in the form

$$W_{(ACC,CCC)} = -\left|(1-\delta_{ip})\right|\alpha_{jq}(c_{ij}) \tag{15}$$

To make ACN's the same as the RCN's, I have to feed inhibitory support to the neuron by an amount proportional to the number of assigned frequencies. If RCN is less than or equal to the ACN's, the additional frequencies cannot be assigned to a cell. This constraint can be expressed as

$$W_{ACN} = -\delta_{ip}\left|(1-\delta_{jq})\right| \tag{16}$$

Equation (16) shows that the self-inhibition is not allowed. The purpose of W_{ACN} is to give the inhibitory support to neurons in the same cell in order to avoid the case of ACN>RCN.

From (12), (15), and (16), the total interconnection weight is

$$W_{ijpq} = -\delta_{ip}\alpha_{jq}(c_{ii}) - \left|(1-\delta_{ip})\right|\alpha_{jq}(c_{ij}) - \delta_{ip}\left|(1-\delta_{jq})\right| \tag{17}$$

The interconnection weight W_{ijpq} between V_{ij} and V_{pq} is symmetrical, i.e., $W_{jpq}= W_{pqij}$ for $1 \le i$, $p \le N$ and $1 \le j$, $q \le M$. Self-feedback is not allowed, i.e., $W_{ijij}=0$.

In this paper, to increase the convergence rate, a nonlinear function is applied on the weights as follows

$$W_{ijpq(new)} = A \times sign\left(W_{ijpq}\right) \times \exp\left(\left|W_{ijpq}\right|\right) \tag{18}$$

The external input I_i is defined as

$$I_i = (d_i - 1) \tag{19}$$

The external input I_i is used to give excitatory support to neurons in the same cell to make them to be satisfied by the traffic demand constraint. If a frequency assignment satisfies all the three constraints (*CSC*, *CCC*, and *ACC*) for a cell and the ACN's are less than the RCN's, then that cell must receive excitatory support as it relates to reinforcing that assignment. The summation of inputs from all neurons for the current updating neuron is $-(d_i$ -1) which should be given an excitatory bias. The input to each neuron (*i,j*) in the original Hopfield neural net is defined as

$$U_{ij} = \sum_{p=1}^{N}\sum_{q=1}^{M} W_{ijpq}V_{pq} + I_i \tag{20}$$

When a neuron receives an input, only frequencies which are satisfied by the three channel-assignment constraints are selected as usable frequencies for the cell. A local minimum can be achieved for the channel assignment given in (20) by using the Hopfield updating procedure. If all the assigned frequencies are satisfied by the three channel-assignment constraints, but one or more of the cells have fewer channels than the RCN's, the frequency assignment will never change and the energy value cannot reach the global minima even though more iterations are performed. To prevent this from occurring, I incorporate a forced assignment method, which allows for a frequency to be assigned to a cell by another excitatory input, even though the channel-assignment constraints are violated and the energy is increased. The forced assigned frequency can change the frequency assignment for other cells too. The algorithm will then search for another solution space in order to attempt to reach the global minima when the current assigned channels does not satisfied the traffic demand constraint. The algorithm checks for the ACN with the value of RCN . The difference between the RCN and the ACN constraints is used as an additional excitatory input, which is given by

$$I_{E_{ij}} = \left(d_i - \sum_{q=1}^{M} V_{iq} \right) \tag{21}$$

The values of W_{ACN} and I_i are constant for the neurons in the same cell. However, the value of I_{Eij} is variable based on the current neuron states. In I_{Eij}, the difference of (RCN-CAN) is fed into the neurons in cell *#i.* To count ACN, V_{ij} is included in ACN of (21).

It is possible that the forced reassignment fails, i.e., ACN<RCN although the additional term I_{Eij} is introduced. If ACN<RCN even after the forced reassignment is applied, it means that the assignment for some calls are failed, which will cause the call drops. The number of callers who could not have the channels and their calls are dropped is (RCN-ACN). This case is considered as a *nonconvergence* case. On the other hand, when all calls have the assigned frequencies without any call dropping, it is considered as *convergence* case. In the simulation results, the convergence rate is the number of cases that all calls have the assigned frequencies without any call dropping before the maximum number of iterations is reached when 100 simulation runs were performed. Input to each neuron of the modified network is defined as

$$U_{ij} = \sum_{p=1}^{N} \sum_{q=1}^{M} W_{ijpq} V_{pq} + I_i + I_{Eij} \tag{22}$$

The schematic description of the input to a neuron is shown in Fig.3. The output function consists of a threshold operation. The final output state of the neuron is

$$V_{ij} = f_{out}(U_{ij}) \tag{23}$$

Where

$$f_{out} = \begin{cases} 1, & \mathit{if}\, U_{ij} \geq THD \\ 0, & otherwise \end{cases} \tag{24}$$

and *THD* is the threshold value and *THD*=0 in this paper.

3. Applying Modified Hopfield Network

In this section, the implementation of the algorithm is discussed. The overall algorithm is summarized in the following steps.

1) The initial state of neurons is set to one or zero according to the initialization method.

2) Repeat the following steps until all neurons are picked.

a) Pick neuron V_{ij} according to the updating method.

b) Calculate the input to this neuron by (22).

c) Decide the new state of this neuron by (23) and (24).

3) Compute the energy *E* of the current assignment. If *E=0*, stop and go to Step 4), otherwise, repeat the process from Step 2).

4) The output state of the neurons V_{ij} will be the final assignment based on the compatibility matrix *C* and demand vector *D*.

3.1 Initialization method

In general, a random initialization method is used as the initial states for the neurons in the Hopfield network. In my algorithm, initialization technique with the frequency assignment constraints are investigated in order to increase the convergence rate and to decrease the iteration number. The total frequency spectrum is composed of a certain number of blocks for the initialization with consideration of the constraints. Fig.4(a) shows the frequency block structures. *M* is the total number of frequencies, and the total frequency spectrum consists of *B* blocks. Every frequency block except the last one has the same number of frequency slots. The last block might not have the same number of frequency slots depending on the *LB* and the block width (*BW*), which is the number of channels in the block. In Fig.4(a), for example, all blocks have BW_i=5, $1 \le i \le (B\text{-}1)$ except the last block B, which has $\mathrm{BW_B}$=2. Frequency j refers to the number of each frequency slot, with each slot identified by f_{ij} , $1 \le i \le N,\ 1 \le j \le M.$ The channel number in block (CNB) is the frequency number in the block, and the sequence is repeated for each block.

In my algorithm, the initial state of neurons, V_{ij} = 0 or V_{ij} =1, where $1 \le i \le N,\ 1 \le j \le M$ indicates whether or not the frequency slot f_{ij} is to be assigned to the jth call in the cell i. Fig. 4(b) shows the fixed interval initialization method, which will be explained later in detail. In Fig. 4(b), the frequency slots with the initial state value V_{ij} =1 are indicated by the dark slots. In the frequency-assignment network, the lower bound (LB) on the total of channels can be obtained from demand vector D and the compatibility matrix C for the cell C_{dmax} . The distance between the frequencies for a certain cell should be at least c_{ii} . LB for the network is defined by

$$LB = (d_{\max} - 1).c_{ii} + 1 \tag{25}$$

The BW is defined by

$$BW = \begin{cases} c_{ii} & if \quad M = LB \\ \left\lfloor \dfrac{M}{d_{\max} - 1} \right\rfloor & otherwise \end{cases} \tag{26}$$

where $\lfloor a \rfloor$ is the largest integer, which is equal or less than a. The number of blocks B is defined by

$$B = \left\lceil \frac{M}{BW} \right\rceil \tag{27}$$

Where $\lfloor a \rfloor$ is the smallest integer, which is equal or greater than a. The procedure of regular interval initialization method is as follows.

1) For the cell #i with d_i calls, randomly choose the frequency block and channel number in the chosen block (CNB) k, $1 \le k \le BW$ in the chosen frequency block, which does not violate CCC and ACC with the previously assigned frequencies of other cells. Assign a "one" to the chosen frequency slot, V_{ik}=1. If the cell #i is C_{dmax} , k=1.

2) For the remaining calls of the cell, assign a "one"to the frequency slot, which has the distance BW with the previous assigned slots, $V_{ik'} = 1$, where $k' = (k + d_f.BW)$ mod M for d_f=1,2,…,d_i -1.

3) Fixed channel assignment for the cell C_{dmax}.

3.2 Updating Method

In an asynchronous Hopfield network, the neurons are selected randomly or sequentially by a certain order for the updating. In my algorithm, both random and sequential selection techniques are used along with updating method.

According to Kim *et al.*(1997), The procedural step for updating method is as follows.

1) Make a list of cells according to the descending order of RCN for each cell.

2) Execute the iteration subroutine as follows until the counter reached to a prespecified maximum iteration number 500 or E=0.

a) Choose the cell #i according to the order of the cell list.

b) Randomly choose one neuron #j in cell #i [neuron(i,j)] and update that neuron.

c) For the next updating neuron, the direction is randomly chosen whether in favor of the left- [neuron(i , j-1)] or the right-side neuron [neuron(i , j+1)].

d) After the initial direction is decided by Step *c*), the next neurons are updated sequentially.

e) Repeat Steps a) – *d*) until all frequencies for all the cells are assigned.

3) Repeat Step 2) for the next cell in the list (pp.963-964).

4. Simulation and Discussion

A mobile system consisting of 21 or 25 cells is used in my simulation. The 21-cell system is repeated in Fig. 5. The output threshold (THD) in (24) is fixed at THD=0. The simulation results according to the regular interval initialization method and new interconnection weight will be discussed and then compared with that of Kim *et al.*(1997) in this paper.

Table I shows the specifications of the problems, which are also used by Kim *et al.*(1997). *LB* is the lower bound on required frequencies for each problem. Figs.6 shows the compatibility matrices and the demand vectors for these problems. In Table II, the result of this paper with new interconnection weight and initialization method are compared with the result in Kim *et al.*'s (1997) algorithm. The average iteration number and the convergence rate to the solution are also shown in Table II. The average iteration number is the average number of iterations, which are increased until E=0. Convergence rate is the probability that the network has E=0 before the maximum number iteration is reached. In these simulations, the maximum number of iterations is fixed at 500. To investigate the number of iterations and the convergence rates, 100 simulation runs were performed with different initial seed values using a random number generator for each of the seven problems. In table II, the simulation results in this paper have a smaller average iteration number and a higher convergence rate than Kim *et al.*'s (1997) results. For example, in problem #1, my algorithm found the solution with 156.23 average iteration number and 95% convergence rate, but Kim *et al.*'s (1997) algorithm found the solution with 263.4 average iteration number and 61% convergence rate. It demonstrates that generally, my algorithm has a better performance (i.e., smaller iteration numbers and higher convergence rates) than Kim *et al.*'s (1997) algorithm. It also shows that my proposed initialization method with fixed channel assignment for the cell $C_{dmax,}$ has better performance than the random initialization method which is usually used. For example, fig.7 and Table III shows the distribution of the frequency for each cell in problem #1 and cell #9 has the largest element in the demand vector, so the frequency assignment of cell #9 was fixed in this algorithm. Fixing a single cell frequency assignment can drastically reduce the searching space and consequently the convergence time is shortened. In addition, Fig.8 shows two typical trajectory of problem #1 and #2 that demonstrates the convergence rate.

5. Conclusion

In this paper, a modified Hopfield neural-network algorithm is proposed in order to obtain an optimal solution for the channel-assignment problem in the mobile cellular environment. This improved algorithm drives itself to the solution by using new interconnection weights and external inputs to the neurons with the initialization and updating method. The interconnection weight is set initially to take into account the RCN in each cell and the three channel-assignment constraints. The three constraints that are considered are the *CCC*, *ACC*, and *CSC*. The compatibility matrix C, demand vector D, and the lower bound number of total frequencies are presented as input. In the simulation results, the average iteration number and convergence rates are shown and compared with that of Kim *et al.*'s(1997) results. The proposed algorithm using the suggested initialization method and new interconnection weight has a better performance than Kim *et al.*'s(1997) algorithm. This algorithm is simple to implement in very large scale integration (VLSI) since the algorithm has only simple functions such as adder, inverter, and comparator. Also, it has a modular structure since each neuron can be constructed as a module.

References

Chan, P. T. H., Palaniswami, M. & Everitt, D. (1994). Neural network based dynamic channel assignment for cellular mobile communication systems, *IEEE Trans. Veh. Technol.,* vol. 43, pp. 279–288.

Funabiki, N., & Takefuji, Y. (1992). A neural network parallel algorithm for channel assignment problems in cellular radio networks. *IEEE Trans. Veh. Technol.,* vol.41,pp. 430–436.

Funabiki, N., Okutani, N. & Nishikawa, S. (2000). A Three-Stage Heuristic Combined Neura –Network Algorithm for Channel Assignment in Cellular Mobile Systems. *IEEE Trans. Veh. Technol.,*vol.49,pp. 397–403.

Gamst, A. & Rave, W. (1982). On frequency assignment in mobile automatic telephone systems, in *roc. GLOBECOM*, pp. 309-315.

He, Z., Zhang, Y., Wei, C. & Wang, J. (2002). A multistage self-organizing algorithm combined transiently chaotic neural network for cellular channel assignment, *IEEE Trans. Veh. Technol.,* vol.51,NO.4, pp. 1386-1396.

Kunz, D. (1991). Channel assignment for cellular radio using neural networks. *IEEE Trans. Veh. Technol.,*Vol.40,pp. 188-193.

Kim, J. S., Park, S. H., Dowd, P. W. & Nasrabadi, N. M. (1997). Cellular radio channel assignment using a modified hopfield network. *IEEE Trans. Veh. Technol.,*vol.46,pp. 957–967.

Moradi, O. (2010). Fixed channel assignment and neural network algorithm for channel assignment problem in cellular radio networks. *Canadian CIS Journal,* vol.3,pp.93-103.

Sivarajan, K. N. , McEliece, R. J., & Ketchum, J. W. (1989). Channel assignment in cellular radio. *IEEE Vehicle*

Technology Society Conference. pp. 846-850.

Vidyarthi, G., Ngom, A. & Stojmenovic, I. (2005). A hybrid channel assignment approach using an efficient evolutionary strategy in wireless mobile networks, *IEEE Trans. Veh. Technol.,* vol.54,PP.1887-1895.

Table 1. Specification of Simulation Problems.

Problem #	Number of Radio Cells *N*	Lower bound *LB*	Compatibility Matrix *C*	Demand Vector *D*
1	25	73	C_2	D_2
2	21	381	C_3	D_3
3	21	533	C_4	D_3
4	21	533	C_5	D_3
5	21	221	C_3	D_4
6	21	309	C_4	D_4
7	21	309	C_5	D_4

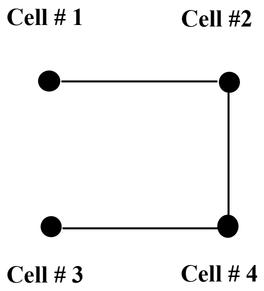
Table 2. Summary of Simulation Results

Problem #	*LB*	*M*	Hopfield network		Kim's *et al.*	
			Average Iter. No.	Convergence Rate	Average Iter. No.	Convergence Rate
1	73	73	156.23	95%	263.4	61%
2	381	381	11.59	100%	67.1	100%
3	533	533	3.34	100%	85.9	98%
4	533	533	44.78	100%	129.5	99%
5	221	221	28.59	98%	65.6	95%
6	309	309	32.44	100%	118.2	96%
7	309	309	64.77	60%	119.1	40%

Table 3. Distribution of the frequency for each cell with 533 frequency in problem #3.

										Cell number										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
16	17	188	24	33	43	18	3	1	2	20	5	5	1	4	5	7	17	16	6	48
61	25	201	167	89	97	34	11	8	10	30	50	31	29	19	13	21	151	51	14	76
124	40	215	196	189	137	45	20	15	28	46	92	58	104	28	23	31	173	126	130	124
212	94	275	227	197	244	93	27	22	42	84	99	88	113	53	32	38	327	277	150	332
457	149	322	346	235	260	100	35	29	52	214	117	118	162	63	46	58	348	304	198	369
487	185	329	499	306	275	140	47	36	86	299	138	126	180	70	59	74	410	364	231	392
511	192	455	513	375	295	150	69	43	93	326	148	159	201	83	72	103	454	462	282	404
525	200	485	530	497	330	255	82	50	123	341	176	168	242	102	81	112	504	482	349	499
	209				352	300	91	57	137	364	201	176	269	125	88	122		517	377	
	221				379	322	101	64	144	371	222	202	282	142	97	199		529	395	
	242				431	378	108	71	168	408	264	209	327	151	107	212			406	
	276				482	385	118	78	194	416	278	223	411	164	117	257			515	
	294				499	395	132	85	208	508	405	230	486	181	128	301			523	
	339				526	404	139	92	237		415	253	496	207	136	334				
	349				533	412	146	99	248		458	262	509	216	143	345				
	364					493	167	106	258			271		224	154	352				
	375					503	177	113	272			291		236	161	360				
	392					530	184	120	283			312		247	175	367				
	422						193	127	318			323		261	186	387				
	459						206	134	328			342		272	196	403				
	469						214	141	342			351		287	203	413				
	482						226	148	366			358		317	210	423				
	496						238	155	390			386		328	217	437				
	504						245	162	439			409		335	227	447				
	515						256	169	448			417		346	234	460				
							263	176	457			440		354	241	478				
							271	183	468			472		373	251	486				
							279	190	503			490		410	259	506				
							290	197				497		418	268					
							298	204				506		427	280					
							305	211				521		434	293					
							314	218						441	303					
							332	225						480	312					
							350	232						488	321					
							357	239						510	333					
							369	246						524	340					
							380	253							347					
							388	260							355					
							397	267							362					
							405	274							374					
							417	281							381					
							425	288							389					
							443	295							396					
							454	302							409					
							465	309							416					
							472	316							429					
							481	323							438					
							490	330							446					
							500	337							453					
							507	344							467					
							516	351							474					
							532	358							483					
								365							492					
								372							502					
								379							513					
								386							520					
								393							528					
								400												
								407												
								414												
								421												
								428												
								435												
								442												
								449												
								456												
								463												
								470												
								477												
								484												
								491												
								498												
								502												
								512												
								519												
								526												
								533												

$$C = \begin{bmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{bmatrix} \qquad D = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 3 \end{bmatrix}$$



V_{ij} Frequency # *j* →

Cell # *i* ↓

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1										■	
2			■								
3				■							
4	■					■					■

Figure 1. A CAP: compatibility matrix *C*, required *D*, the corresponding network topology, and the optimum solution with 11 channels

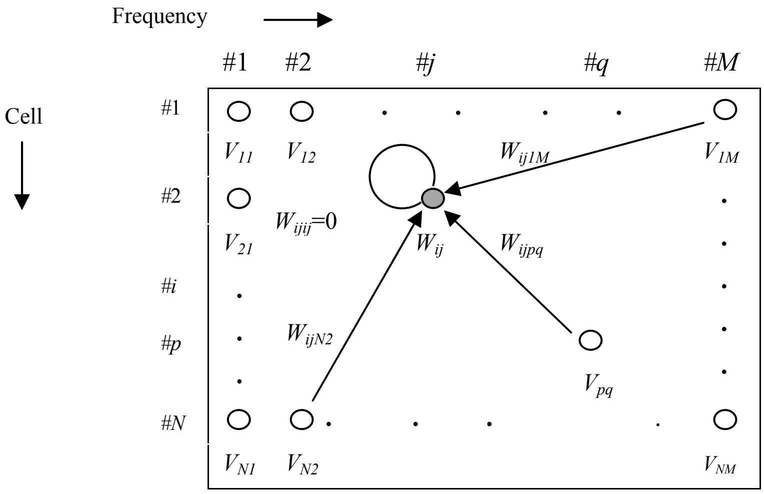


Figure 2. A 2-D Hopfield network for the channel-assignment problem

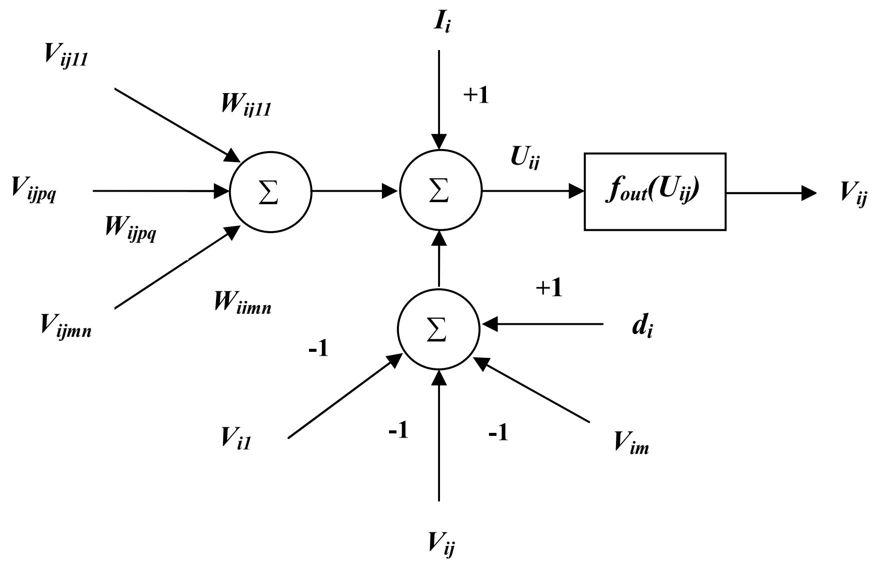
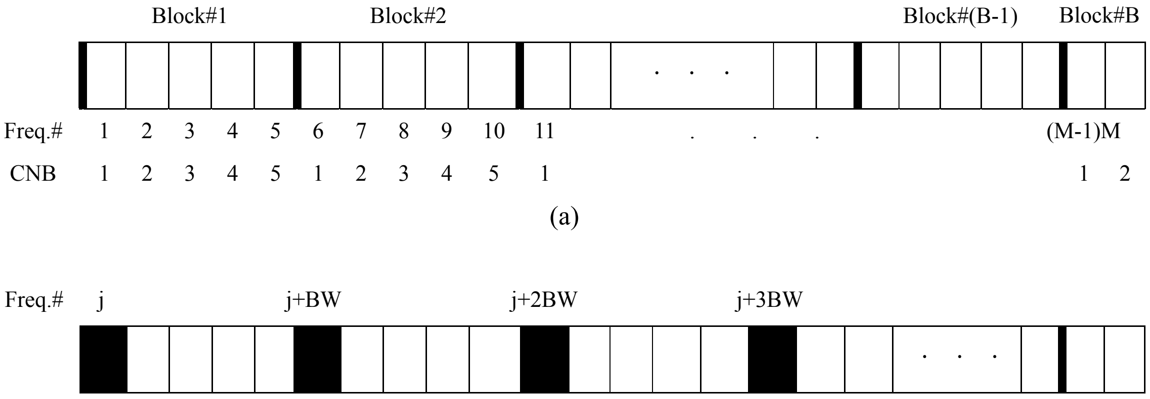


Figure 3. A 2-D Hopfield network for the channel-assignment problem


Block#1
Block#2
Block#(B-1)
Block#B
. . .
Freq.# 1 2 3 4 5 6 7 8 9 10 11 . . . (M-1)M
CNB 1 2 3 4 5 1 2 3 4 5 1 1 2
Freq.# j j+BW j+2BW j+3BW
. . .

(a)

(b)

Figure 4. Structure of frequency domain and initialization: (a) frequency and block structure, (b) regular interval initialization method

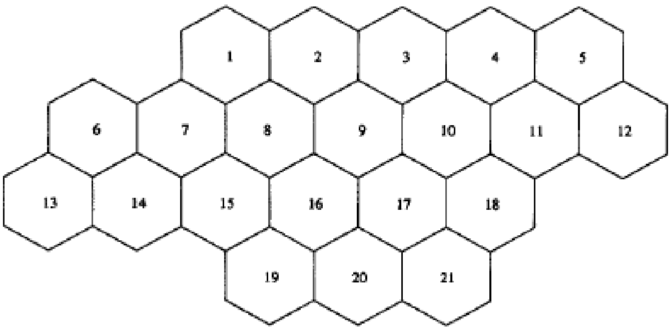
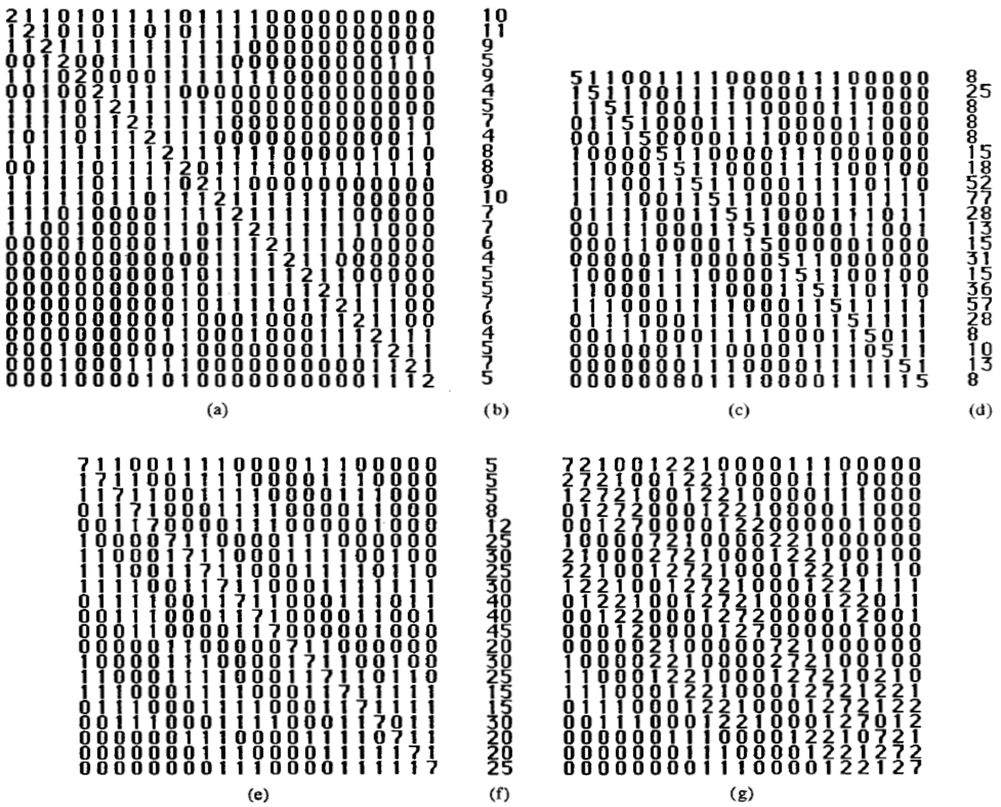


Figure 5. The 21-cell system used in this paper



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

(a)

```
10
11
9
5
9
4
5
7
4
8
8
9
10
7
7
6
4
5
5
7
6
4
5
7
5
```

(b)

```
5 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0
1 5 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0
1 1 5 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0
0 1 1 5 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0
0 0 1 1 5 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0
1 0 0 0 0 5 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0
1 1 0 0 0 1 5 1 1 0 0 0 1 1 1 1 0 0 1 0 0
1 1 1 0 0 1 1 5 1 1 0 0 0 1 1 1 1 0 1 1 0
1 1 1 1 0 0 1 1 5 1 1 0 0 0 1 1 1 1 1 1 1
0 1 1 1 1 0 0 1 1 5 1 1 0 0 0 1 1 1 0 1 1
0 0 1 1 1 0 0 0 1 1 5 1 0 0 0 0 1 1 0 0 1
0 0 0 1 1 0 0 0 0 1 1 5 0 0 0 0 0 1 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 5 1 1 0 0 0 0 0 0
1 0 0 0 0 1 1 1 0 0 0 0 1 5 1 1 0 0 1 0 0
1 1 0 0 0 1 1 1 1 0 0 0 1 1 5 1 1 0 1 1 0
1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 5 1 1 1 1 1
0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 5 1 1 1 1
0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 5 0 1 1
0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 5 1 1
0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 5 1
0 0 0 0 0 0 8 0 1 1 1 0 0 0 0 1 1 1 1 1 5
```

(c)

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

(d)

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

(e)

```
5
5
5
8
12
25
30
25
30
40
40
45
20
30
25
15
15
30
20
20
25
```

(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)

Figure 6. 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) Compatibility matrix C_4. (f) Demand vector D_4. (g) Compatibility matrix C_5

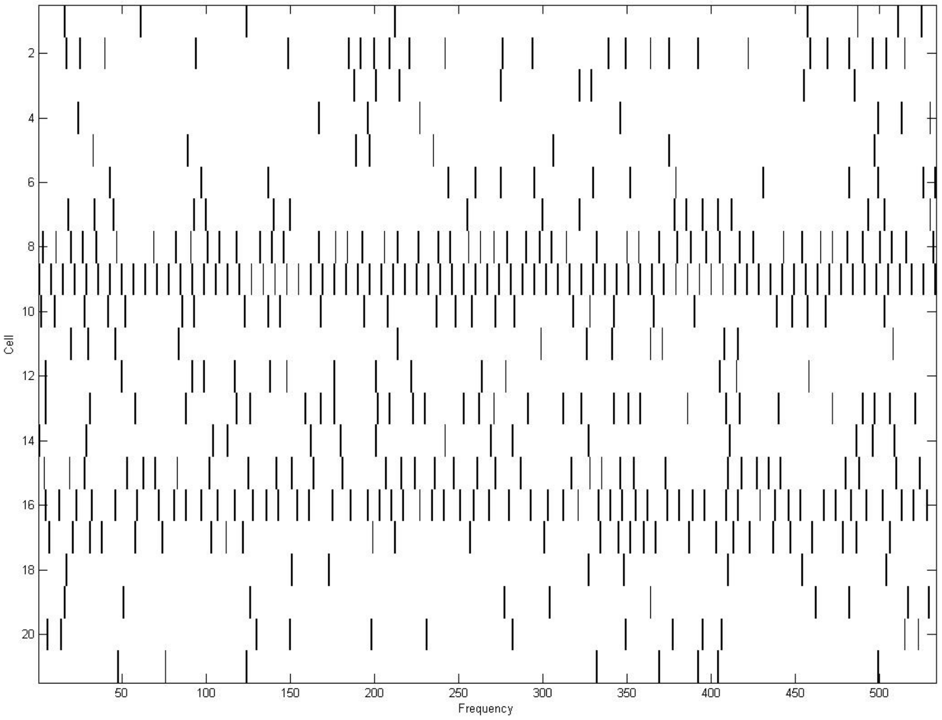
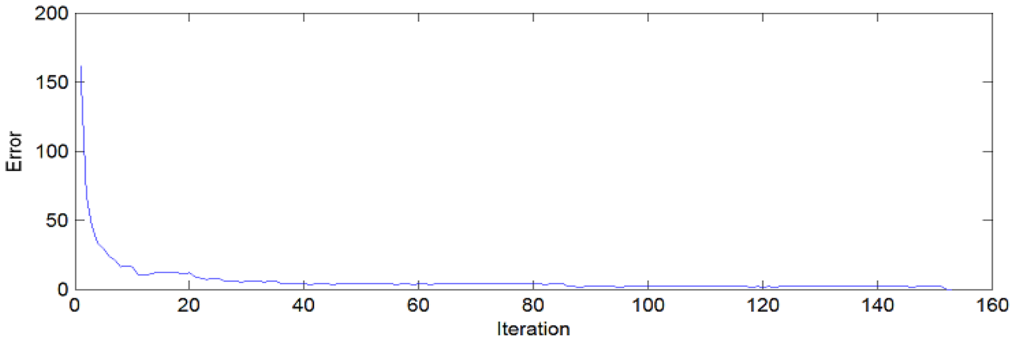
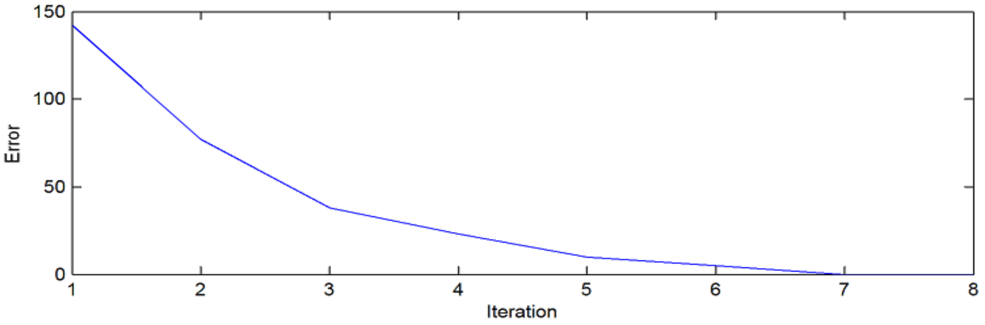


Figure 7. Distribution of the frequency for each cell with 533 frequency in problem #3.



(a)



(b)

Figure 8. two typical convergence rate trajectory: (a) Problem #1 and (b) Problem #2