

Fixed Channel Assignment and Neural Network Algorithm 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

The channel-assignment problem is important in mobile telephone communication. Since the usable range of the frequency spectrum is limited, the optimal channel-assignment problem has become increasingly important. The goal of this problem is to find a channel assignment to requested calls with the minimum number of channels subject to interference constraints between channels. This algorithm consists of: 1) the fixed channel assignment stage; 2) the neural network stage. In the first stage, the calls in a cell determining the lower bound on the total number of channels are assigned channels at regular intervals, then the calls in adjacent six cells are assigned channels by a cluster heuristic method sequentially. In the second stage, the calls in the remaining cells are assigned channels by a binary neural network. The performance is verified through solving well-known benchmark problems. Especially for Sivarajan's benchmark problems, my algorithm first achieves the lower bound solutions in all of the 12 instances.

Keywords: Channel assignment, Fixed channel assignment, Neural network

1. 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.

The channel assignment problem 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. Three types of constraints have usually been considered in channel assignment problem.

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 channel assignment problem is to find a channel assignment to every requested call with the minimum number of channels subject to the above three constraints.

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. Since the graph coloring problem is known to be NP-complete, 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. In 1982 Gamst and Rave summarized four existing sequential approximation algorithms. 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. The second algorithm repeatedly assigns frequencies according to the assignment difficulty of requirements. 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. In 1989 Sivarajan *et al.* proposed an $O(n^2)$ time sequential heuristic algorithm, based on the first algorithm introduced by Gamst and Rave. Sivarajan *et al.* applied their algorithm to several problems, where the values of total frequencies in solutions are shown without any actual assignment results. In 1997 Smith and Palaniswami proposed two neural network algorithms for a different formulation of channel assignment problem, which treats the noninterference constraints as soft constraints in the objective function. In 1998 Ngo and Li proposed a genetic algorithm with a local search method. Unfortunately, none of the existing heuristic algorithms can find the lower bound solutions in all of the 13 instances for Sivarajan's benchmark problems.

In this paper, I propose a two-stage algorithm for channel assignment problem by combining sequential heuristic algorithms into a parallel neural network algorithm. This algorithm consists of fixed channel assignment(FCA) stage and the neural-network assignment stage. The performance is verified through solving the benchmark problems by Sivarajan, Kunz, and Kim. For Sivarajan's and Kunz's problems, my algorithm first achieves the lower bound solutions in all the instances, whereas my algorithm provides the better solution quality than existing algorithms for Kim's problems.

2. Problem Representation

Channel assignment in this paper follows the problem formulation by Funabiki, Okutani and Nishikawa. The three constraints for the channel interference in the N - cell system are described by an $N \times N$ symmetric compatibility matrix C . The nondiagonal element c_{ij} ($i \neq j$) of C represents the minimum distance between a channel assigned to cell i and a channel to cell j . The diagonal element C_{ii} of C represents the distance between a pair of channels assigned to cell i . Thus, CCC is described by $C_{ij}=1$, ACC is by $C_{ij} \geq 2$ and CSC is by $C_{ii} \geq 1$ in C , respectively. A set of requested calls in the N -cell system is given by an N -element demand vector D . The i th element d_i of D represents the number of channels for the requested calls in cell i . Let f_{ik} be the k th channel assigned to cell i for $i=1, \dots, N$ and $k=1, \dots, d_i$. Then, the total number of required channels M can be represented by:

$$M = \max \{ f_{ik} \} \text{ for } i = 1, \dots, N$$

and

$$k = 1, \dots, d_i .$$
(1)

Given a pair of a compatibility matrix C and a demand vector D , the goal of channel assignment problem is to find a channel assignment $\{f_{ik}\}$ with the minimum value of M subject to the interference constraints:

$$|f_{ik} - f_{jl}| \geq c_{ij} \text{ for } i = 1, \dots, N, \quad j = 1, \dots, N,$$

$$k = 1, \dots, d_i$$

and

$$l = 1, \dots, d_j$$

except for $i = j$ and $k = l$.

(2)

3. Two - Stage Algorithm for Channel Assignment Problem

This algorithm consists of: 1) the fixed channel assignment stage; 2) the neural-network assignment stage. In the first stage, the calls in a cell, which determines the lower bound on the total number of channels are assigned channels at regular interval, then the calls in adjacent six cells are assigned channels by a cluster heuristic method sequentially. Initially, the cluster region is composed of a cell, which determines the lower bound on the total number of channels and its adjacent cells because the channel assignment to cells with large demand is usually more difficult than the assignment to other cells. Then, every time the whole assignment is failed, the cluster region is expanded by additionally including the cells adjacent to the original region. In the second stage, the calls in the remaining cells are assigned channels by a binary neural network in parallel. when the neural network stage is applied to the remaining cells, the channel assignment result by the fixed channel assignment stage must be fixed in this algorithm.

The overall procedure of the this algorithm is described as follows.

- 1) Input a compatibility matrix C and a demand vector D .
- 2) Initialize the total number of channels M by the lower bound or its approximation.
- 3) Apply *the fixed channel assignment stage*.
 - a) Initialize the cluster region by a cell which determines the lower bound of M and its adjacent cells.
 - b) Regular interval assignment to the cell which determines the lower bound of M .
 - c) Apply *the cluster assignment stage* to the cluster region while the assignment by b) is fixed. If this assignment is failed, then increment M by one ($M=M+1$) and go to b).
- 4) Apply *the neural-network assignment stage* to the remaining cells while the assignment by b) and c) is fixed. If this assignment is failed, then expand the cluster region by additionally including its adjacent cells and go to c). If no expandable cell exists, then increment by one and go to b).

Fig. 1 shows a cellular system of 21 cells in Sivarajan's benchmark problems. Each number inside a cell represents the number of requested calls. In this algorithm, the fixed channel assignment stage is applied to the cell with 77 calls and adjacent six cells, because the 77-call cell determines the lower bound of M . Then, the neural-network assignment stage is applied to the other cells.

3.1 Fixed Channel Assignment Stage

The fixed channel assignment stage assigns channels to the calls in the cell which determines the lower bound on M , at a regular interval of satisfying CSC in order to minimize the total number of channels M . Let us consider the channel assignment to cell $\neq i$, where C_{ii} and d_i represent CSC and the number of requested channels for cell $\neq i$, respectively. Each of d_i calls is assigned a channel at an interval of either C_{ii} or $(C_{ii}+1)$, after the first and last channels are assigned to cell $\neq i$. Actually, the first x calls are assigned at the interval of C_{ii} , whereas the rest (d_i-x) calls are assigned at the interval of $(C_{ii}+1)$.

$$\begin{aligned} f_{ik} &= 1 + c_{ii}(k-1) & \text{for } k = 1, \dots, x \\ f_{ik} &= 1 + c_{ii}(x-1) + (c_{ii}+1)(k-x) & \text{for } k = x+1, \dots, d_i \end{aligned} \quad (3)$$

The value of x is given by solving the linear equation

$$1 + c_{ii}(x-1) + (c_{ii}+1)(d_i-x) = M \quad (4)$$

After channel assignment to the calls in the cell which determines the lower bound on M , then at the cluster assignment stage assigns channels sequentially by using the assignment list, which is made of calls in the cluster region after sorted in the descending order of the assignment difficulty in Box's algorithm. Each call in this list is assigned a channel by the requirement exhaustive strategy proposed by Sivarajan. The procedure of this stage is as follows.

- 1) Make a list of the calls in the cluster region in the ascending order of the cell index number.
- 2) Initialize the assignment difficulty of every call in the list by zero and the iteration number t by one.
- 3) Initialize the channel number j by one.
- 4) Check whether or not channel $\#j$ can be assigned to each unassigned call in the list from the top to the bottom sequentially. If it can be assigned there without interference, then assign channel $\#j$ to that call.
- 5) If every call in the list is assigned a channel, then terminate the procedure as success, else if $j < M$ then increment t by one and go to 4).
- 6) If the iteration number t reaches its upper limit G max, then terminate the procedure as failure, else increment t by one.
- 7) Add a randomized real number between zero and one to the assignment difficulty for each unassigned cell in this iteration, clear the assignment result by this stage, remake the list by sorting the calls in the descending order of the assignment difficulty, and go to 3).

3.2 Neural-Network Assignment stage

This algorithm is based on a two dimensional neural network model. Consider a channel assignment problem in a four-cell network proposed by Sivarajan. Fig.2 shows the compatibility matrix C and the demand vector D . Fig. 3(a) shows the neural network representation for solving the channel assignment problem in Fig. 2. 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 $N \times M$ 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. 3(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 $\#i$:

$$\sum_{q=1}^M V_{iq} - d_i \tag{5}$$

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 $\#i$:

$$\sum_{\substack{q=j-(c_{ii}-1) \\ q \neq j \\ 1 \leq q \leq M}}^{j+(c_{ii}-1)} V_{iq} \tag{6}$$

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

$$\sum_{\substack{p=1 \\ p \neq i \\ c_{ip} > 0}}^N \sum_{\substack{q=j-(c_{ip}-1) \\ 1 \leq q \leq M}}^{j+(c_{ip}-1)} V_{pq} \tag{7}$$

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:

$$\frac{dU_{ij}}{d_t} = - \frac{\partial E}{\partial V_{ij}} = -A \left(\sum_{q=1}^M V_{iq} - d_i \right) - B \left(\sum_{\substack{q=j-(c_{ii}-1) \\ q \neq j, 1 \leq q \leq M}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1 \\ p \neq i, c_{ip} > 0}}^N \sum_{\substack{q=j-(c_{ip}-1) \\ 1 \leq q \leq M}}^{q=j+(c_{ip}-1)} V_{pq} \right) \tag{8}$$

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 (8):

$$E = \frac{A}{2} \sum_{i=1}^N \left(\sum_{q=1}^M V_{iq} - d_i \right)^2 + B \sum_{i=1}^N \sum_{j=1}^M \left(\sum_{\substack{q=j-(c_{ii}-1) \\ q \neq j, 1 \leq q \leq M}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1 \\ p \neq i, c_{ip} > 0}}^N \sum_{\substack{q=j+(c_{ip}-1) \\ q=j-(c_{ip}-1) \\ 1 \leq q \leq M}} V_{pq} \right) V_{ij} \quad (9)$$

3.2.1 Heuristics for the Global Minimum Convergence

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:

$$-A f \left(\sum_{q=1}^M V_{iq} - d_i \right) \quad (10)$$

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 $(t \bmod T) < w$

$$\text{then } -B \left(\sum_{\substack{q=j-(c_{ii}-1) \\ q \neq j, 1 \leq q \leq M}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1 \\ p \neq i, c_{ip} > 0}}^N \sum_{\substack{q=j+(c_{ip}-1) \\ q=j-(c_{ip}-1) \\ 1 \leq q \leq M}} V_{pq} \right) V_{ij} \\ \text{else } -B \left(\sum_{\substack{q=j-(c_{ii}-1) \\ q \neq j, 1 \leq q \leq M}}^{j+(c_{ii}-1)} V_{iq} + \sum_{\substack{p=1 \\ p \neq i, c_{ip} > 0}}^N \sum_{\substack{q=j+(c_{ip}-1) \\ q=j-(c_{ip}-1) \\ 1 \leq q \leq M}} V_{pq} \right) \quad (11)$$

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

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

$$+ Ch \left(\sum_{q=1}^M V_{iq} - d_i \right) \cdot (1 - V_{ij}) \quad (12)$$

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 i 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:

$$\begin{aligned} U_{ij} &= U_{\max}, & \text{if } U_{ij} > U_{\max} \\ U_{ij} &= U_{\min}, & \text{if } U_{ij} < U_{\min} \end{aligned} \quad (13)$$

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

3.2.2 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 , $T = 10$, $w = 5$, $U_{\max} = 30$, $U_{\min} = -30$, $UTP = 5$, $LTP = -5$, and $T_{\max} = 500$.
- 2) Randomize the initial values of input $U_{ij}(t)$ for $i=1, \dots, N$ and $j=1, \dots, M$ between 0 and U_{\min} . Assign the initial values of output $V_{ij}(t)$ for $i=1, \dots, N$ and $j=1, \dots, M$ to 0.
- 3) The channel assignment result by the fixed channel assignment stage must be fixed when the neural network stage is applied to the remaining cells. To represent this constraint correctly, all the fixed channel assignment must be reflected into the corresponding neuron outputs:

$$\begin{aligned} &\text{If the assignment of channel } \neq j \text{ to cell } \neq i \text{ be fixed,} \\ &\text{then } V_{ij} = 1 \text{ else } V_{ij} = 0. \end{aligned} \quad (14)$$

- 4) Compute the change of input $\Delta U_{ij}(t)$ based on the motion equation in (8):

if $(t \bmod T) < w$ *then*

$$\begin{aligned} \Delta U_{ij}(t) &= -A f \left(\sum_{q=1}^M V_{iq}(t) - d_i \right) \\ &\quad - B \left(\sum_{\substack{q=j-(c_{ii}-1) \\ q \neq j, 1 \leq q \leq M}}^{j+(c_{ii}-1)} V_{iq}(t) + \sum_{\substack{p=1 \\ p \neq i, c_{ip} > 0}}^N \sum_{\substack{q=j+(c_{ip}-1) \\ q=j-(c_{ip}-1) \\ 1 \leq q \leq M}} V_{pq}(t) \right) V_{ij}(t) \\ &\quad + C h \left(\sum_{q=1}^M V_{iq}(t) - d_i \right) (1 - V_{ij}(t)) \end{aligned} \quad (15)$$

else

$$\begin{aligned} \Delta U_{ij}(t) &= -A f \left(\sum_{q=1}^M V_{iq}(t) - d_i \right) \\ &\quad - B \left(\sum_{\substack{q=j-(c_{ii}-1) \\ q \neq j, 1 \leq q \leq M}}^{j+(c_{ii}-1)} V_{iq}(t) + \sum_{\substack{p=1 \\ p \neq i, c_{ip} > 0}}^N \sum_{\substack{q=j+(c_{ip}-1) \\ q=j-(c_{ip}-1) \\ 1 \leq q \leq M}} V_{pq}(t) \right) \\ &\quad + C h \left(\sum_{q=1}^M V_{iq}(t) - d_i \right) (1 - V_{ij}(t)) \end{aligned} \quad (16)$$

- 5)

$$U_{ij}(t+1) = U_{ij}(t) + \Delta U_{ij}(t) \quad (17)$$

6) Use the input saturation heuristic in (13):

$$\begin{aligned} U_{ij}(t+1) &= U_{\max}, & \text{if } U_{ij}(t+1) > U_{\max} \\ U_{ij}(t+1) &= U_{\min}, & \text{if } U_{ij}(t+1) < U_{\min} \end{aligned} \quad (18)$$

7) Update output $V_{ij}(t+1)$ based on the hysteresis Mc-Culloch-Pitts neuron model:

$$\begin{aligned} V_{ij}(t+1) &= 1, & \text{if } U_{ij}(t+1) > UTP \\ V_{ij}(t+1) &= 0, & \text{if } U_{ij}(t+1) < LTP \end{aligned} \quad (19)$$

8) Increment the number of iteration steps by one.

9) Check the termination condition: if all the constraints are resolved, then terminate this procedure as success, else if then terminate it as failure, else go to 4).

4) Simulation and Discussion

For the comparison, two heuristic algorithms by Sivarajan and by Box have also been implemented. The benchmark problems by Sivarajan, Kunz and Kim are used as simulated instances in this paper, where specifications are summarized in Tables I and II. Note that instance #12 in Table I is the Kunz's problem. In the compatibility matrix of these tables, " N_c " is the cluster size for CCC, " c_{ij} " is the minimum channel distance between any pair of cells in the same cluster except for adjacent cells for CCC, " acc " is the minimum channel distance between adjacent cells for ACC, and " c_{ii} " is the minimum distance within the same cell for CSC. In the demand vector, "case 1" and "case 2" represent the corresponding channel requirements in Sivarajan's problem. " $U(x,y)$ " represents a set of uniformly randomized integers between and , and "total call" is the total number of requested channels in each simulated instance.

In the fixed channel assignment stage the cell with 77 calls in instances #1–8 and to the cell with 45 calls in instances #11,12 in Sivarajan's problems are assigned channels at regular intervals, because they determine the lower bound on the total number of channels. In the cluster assignment stage, the iteration limit is set 100 and a maximum of ten trials using different random numbers is repeated for each cluster region until the assignment succeeds. In the neural-network assignment stage, a maximum of ten trials using different initial neuron states is repeated for each assignment result by the FCA stage until the assignment succeeds.

A total of ten runs using different random numbers is executed by my algorithm and Box's algorithm for each instance, whereas one run is executed by each of Sivarajan's eight algorithms. Table III shows the lower bound in Sivarajan's and Kunz's problems, best and average solutions found by this algorithm. Here, we need to correct the lower bound on M in instances #1 and #2. Consider the channel assignment to the requests in the seven-cell cluster composed of the 77-call cell and its adjacent six cells. In this cluster, a different channel must be assigned to every call to satisfy CCC. When a channel is assigned to each call in the 77-call cell, this channel and its adjacent channels cannot be reused to the other cells in this cluster because of $acc=2$. Thus, the 77-call cell must occupy $229(=2 \times 2 + 3 \times (77-2))$ channels exclusively in this cluster. The remaining six cells need at least $198(=25+8+52+28+57+28)$ different channels. Therefore, the channel assignment for this cluster requires at least $427(=229+198)$ channels.

Table III indicates that my algorithm first achieves the lower bound solutions in all of the benchmark instances, whereas the existing algorithms cannot find them in instances #1, #2, #3, #9, and #10. One run is executed by the FCA and neural network algorithm, Sivarajan's eight algorithms, and Box's algorithm for each of nine instances in Kim's benchmark problems. Table IV compares the solutions by the FCA and neural network on the total numbers of required channels, where my algorithm can find the best solutions in all of the nine instances. Table V summarizes the simulation results, and shows the average number of iteration steps required to converge to the optimum solution and the convergence rate. Fig.4 shows the distribution of the frequency for each cell and fig.5 shows the number of iteration step required to convergence to solution in instance #3.

5. Conclusion

This paper presents a FCA and neural network algorithm of combining sequential heuristic methods into a parallel neural network for the NP-complete channel assignment problem in cellular mobile communication systems. This algorithm consists of the fixed channel assignment stage and the neural-network assignment stage to seek the near-optimum solution. The performance is verified through solving the benchmark problems by

Sivarajan, Kunz, and Kim. Especially for all the instances in Sivarajan's problems, my algorithm first achieves the lower bound solutions. For Kim's problems, my algorithm provides the better solution quality than existing algorithms.

References

- 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. 430–437
- 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
- Fujii, T. & Nakazima, N. (2003). Radio Channel Assignment System for Cellular Mobile Communications. *Electronics and Communications in Japan*. Part 1, Vol. 86, No. 5
- Gamst, A. & Rave, W. (1982). On frequency assignment in mobile automatic telephone systems. *Procedeeng on Global Communication*. pp. 309-315
- 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
- Kunz, D. (1991). Channel assignment for cellular radio using neural networks. *IEEE Trans. Veh. Technol.*,Vol.40,pp. 188-193
- Sivarajan, K. N. , McEliece, R. J., & Ketchum, J. W. (1989). Channel assignment in cellular radio. *IEEE Vehicle Technology Society Conference*. pp. 846-850
- Wang, W., & Rushforth, C. K. (1996). An adaptive local-search algorithm for the channel assignment problem (CAP). *IEEE Trans. Veh. Technol.*,vol. 45, pp. 459–466

Table 1. Specifications for Sivarajan'S and Kunz'S Benchmark Problems

Instance	Compatibility matrix				Demand Vector	
	N_c	c_{ij}	acc	c_{ii}	Call distrib.	Total calls
#1	12	1	2	5	case1	481
#2	7	1	2	5	case1	481
#3	12	1	2	7	case1	481
#4	7	1	2	7	case1	481
#5	12	1	1	5	case1	481
#6	7	1	1	5	case1	481
#7	12	1	1	7	case1	481
#8	7	1	1	7	case1	481
#9	12	1	2	5	case2	470
#10	7	1	2	5	case2	470
#11	12	1	2	7	case2	470
#12	7	1	2	7	case2	470

Table 2. Specifications For Kim'S Benchmark Problems

Instance	Compatibility matrix				Demand Vector	
	N_c	c_{ij}	acc	c_{ii}	call distrib.	total calls
#1	7	1	1	3	U(10,15)	619
#2	7	2	3	5	U(10,15)	619
#3	7	3	4	7	U(10,15)	619
#4	7	1	1	3	U(10,20)	760
#5	7	2	3	5	U(10,20)	760
#6	7	3	4	7	U(10,20)	760
#7	7	1	1	3	U(10,30)	976
#8	7	2	3	5	U(10,30)	976
#9	7	3	4	7	U(10,30)	976

Table 3. Simulation Results For Sivarajan’S And Kunz’S Benchmark Problems

Instance	LB	FCA and neural network algorithm		Sivarajan		Box	
		best	average	Best	average	best	average
1	427	427	427.0	460	498.3	442	443.6
2	427	427	427.0	447	498.3	442	443.4
3	533	533	533.0	536	552.8	533	533.2
4	533	533	533.0	533	550.1	533	533.0
5	381	381	381.0	381	381.0	381	381.0
6	381	381	381.0	381	381.0	381	381.0
7	533	533	533.0	533	533.0	533	533.0
8	533	533	533.0	533	533.0	533	533.0
9	258	258	258.0	283	315.9	270	271.7
10	258	258	258.0	269	294.4	260	261.6
11	309	309	309.0	310	338.5	309	309.0
12	309	309	309.0	310	330.6	309	309.0

Table 4. Simulation Results For Kim’S Benchmark Problems

Instance	FCA and Neural Network Algorithm	Sivarajan		Box
#1	96	106	112.3	102
#2	253	270	307.5	280
#3	363	389	439.4	402
#4	127	132	146.5	130
#5	319	343	388.8	350
#6	467	501	555.3	505
#7	168	178	186.8	169
#8	434	473	524.3	457
#9	631	673	735.0	650

Table 5. Summary of Simulation Results

Instance	LB	FCA and neural network algorithm		Average Number of Iteration step to Solutions	Convergence Rate
		best	average		
#1	427	427	427.0	150.4	100%
#2	427	427	427.0	170.2	100%
#3	533	533	533.0	128.33	100%
#4	533	533	533.0	137.88	100%
#5	381	381	381.0	166.55	100%
#6	381	381	381.0	152.62	100%
#7	533	533	533.0	100.33	100%
#8	533	533	533.0	140.5	100%
#9	258	258	258.0	11.36	100%
#10	258	258	258.0	115.2	99%
#11	309	309	309.0	120.3	98%
#12	309	309	309.0	128.33	91%

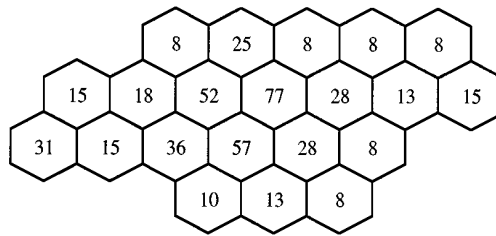


Figure 1. The 21-cell system in Sivarajan’s benchmark problems.

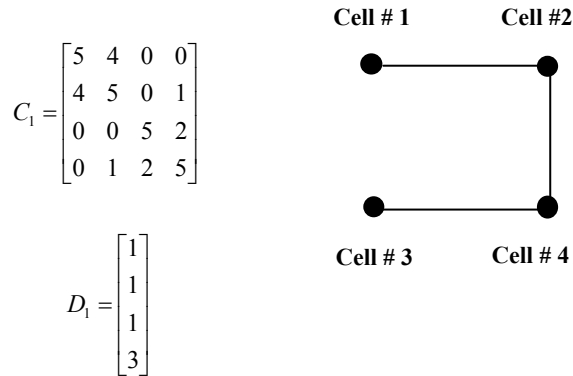


Figure 2. A four-cell channel assignment problem.

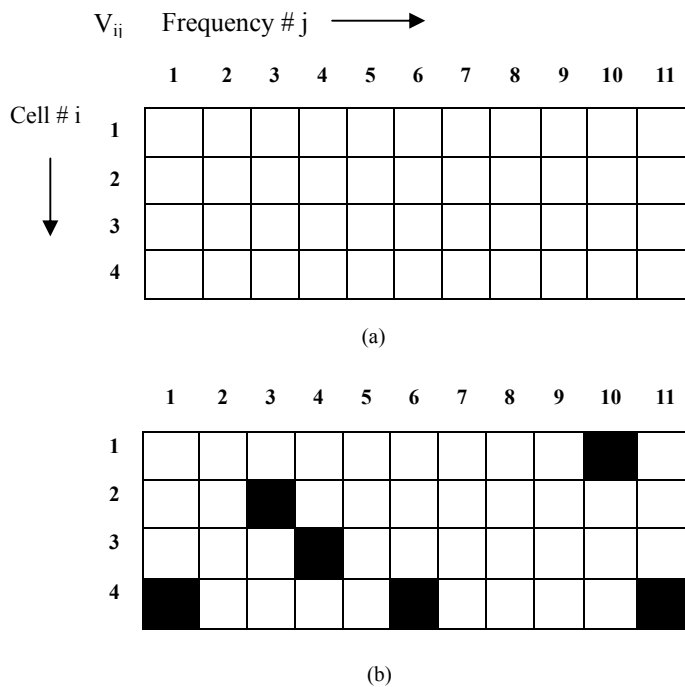


Figure 3. Neural network representation for the channel assignment problem in Fig. 2.

(a) . 4 x 11 processing elements for the channel assignment problem in Fig. 2.

(b) . The convergence of 4 x 11 processing elements to a solution.

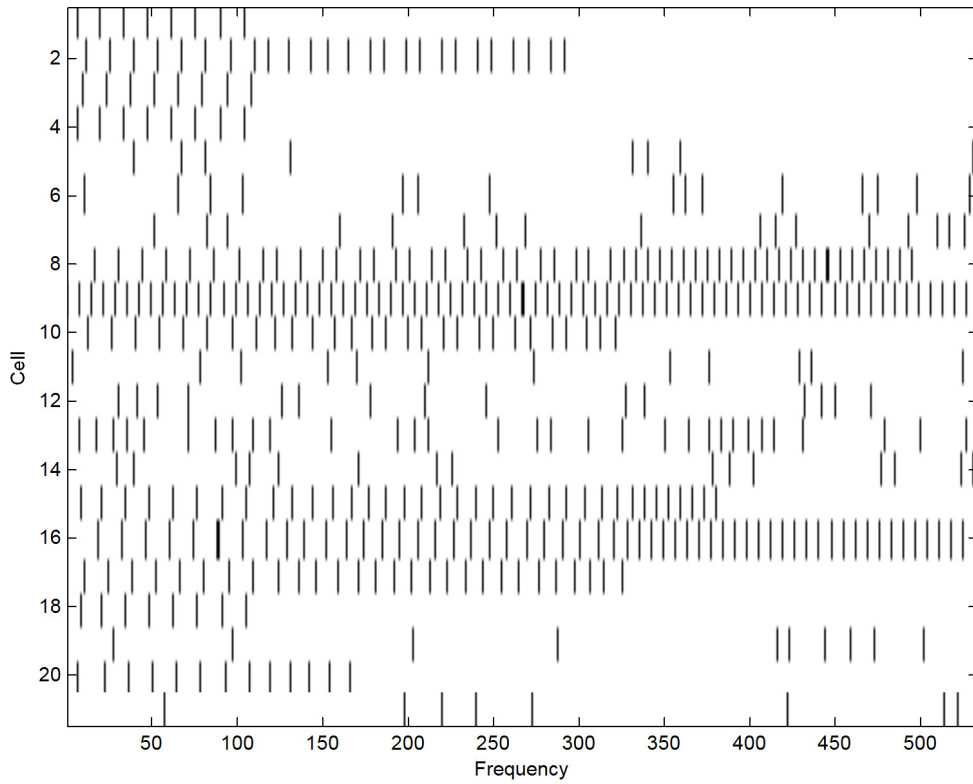


Figure 4. Distribution of the frequency for each cell with 533 frequency in instance #3.

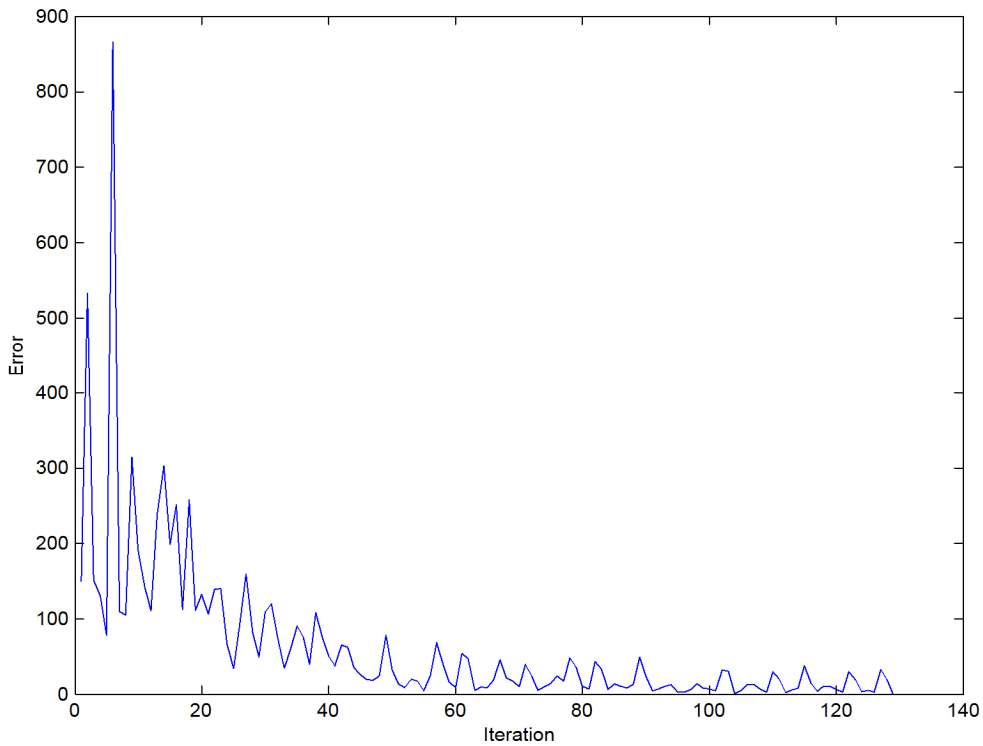


Figure 5. The number of iteration step required to convergence in instance #3.