An Efficient CSC-Preserving Fixed Channel Assignment Algorith

吳東光
Tung-Kunag Wu
明新技術學院
資訊管理系
臺灣省 新竹縣
tkwu@mis.mhit.edu.tw

鄭瑞恒
Rei Heng Cheng
玄奘人文社會學院
資訊管理系
臺灣省 新竹市
rhc@hcu.edu.tw

俞征武
Chang Wu Yu
中華大學
資訊工程系
臺灣省 新竹市
cwyu@chu.edu.tw

Abstract

A critical issue in the design of a cellular radio network is to determine a spectru -efficient and conflict-free allocation of channels among the cells. In this paper, w propose a novel approach for the Fixed Channel Assignment (FCA) problem. By preserving the Co-Sit channel constraint throughout our algorithm and adopting a fine-tuning procedure to escape from a local minimum, we speedup th overall execution time and improve the convergence rate. Simulation results show that our algorithm achieve 100% convergence rate to solutions on eight benchmark problems. Furthermore, the number of iterations our algorithm required is fewer than all previous results. Such significant results indicate that our ne approach is indeed an effective and efficient method for the fixed channel assignment problem.

Keywords: Channel Assignment, FCAP, Cellular Network, Algorithm, Mobile Communication.

1. Introduction

The growing popularity for mobile communication services demands efficient use of the limited frequency spec trum. A common way to increase frequency spectrum utilization is the cellular structure approach, which divides spatiall the geographical region into a number of cells. A base station (BS) is established in each cell, and each mobile station (MS) in the cell communicates through the BS via a channel. In such a frequency reuse system, MSs in different non-interfering cells may simultaneously use the same frequency channel. A critical issue in the design of such a cellular radio network is to determine a spectrum-efficient and conflict-free allocation of channels among the cells.

Generally, there are three types of constraints that need t be considered, which include (1) the *co-channel constraint* (CCC), where the same channels cannot be assigned t certain pair of cells simultaneously; (2) the *adjacent channel constraint* (ACC), where channels adjacent in the frequency spectrum cannot be assigned to adjacent cells simultaneously and (3) the *co-site constraint* (CSC), where channels assigned in the same cell must have a minimal separation in frequency among each other. We call these three constraints the *electromagnetic compatibility constraints (EMC)*. Allocating the channels among the cells while satisfying both the traffic demand and EMC is the main objective of the *channel assignment problem*.

The channel assignment problem can be classified into two categories: (1) *fixed channel assignment problem* (FCAP), where channels are permanently allocated to each cell [12]; and (2) *dynamic channel assignment problem* (DCAP), where all channels, which are available for every cell, are allocated dynamically upon request [1].

The FCAP problem has been studied extensively [4], [8], [9],[10], [11], [12]. It has been shown that this problem is equivalent to a generalized graph-coloring problem, which is NP-hard [10],[11]. Such problems require extremely time-consuming algorithms for their exact solution. It is therefore necessary to use more time-efficient algorithms that, however, cannot guarantee optimal solutions. Most of the efforts are spent in developing approximation algorithms [14]. These include graph -theoretic [7], neural network [4],[10], simulated annealing [3], and geneti algorithms approaches [12]. However, neural-network-based algorithms typically yield onl sub-optimal solutions [14]. The simulated annealin approach, although it may be more flexible, is easil trapped in local minimu , which requires a lot of computation time to escape from [14]. In view of this undesirable feature, Gamst derives some lower bounds for the minimal number of channels required [6]. Sung and Wong [14] provided another lower bound, which is tighter in some cases. In addition, they proposed an algorithm, which always finds the optimal solution for a special class of cellular network topologies. In short, different approaches have their own limitations. These reflect how hard the channel assignment problem is.

In this paper, we propose a new approach to solve the fixed channel assignment problem. By preserving the Co -Site channel constraint throughout our algorithm, we speedup the overall execution time. A number of new techniques are also devised to increase the convergence rate. Simulation results show that our algorithm achieve 100% convergence rate to solutions on eight benchmark problems. To the best of our knowledge, no previous algorithms achieved such high convergence rate. Furthermore, the number of iterations our algorithm required is fewer than all previous results. Such significant results indicate that our new approach is indeed an effective and efficient method for the fixed channel assignment problem.

The remainder of this paper is organized as follows. In secti 2, some necessary definitions are given. Our ne approach is described in Section 3, with the e xperimental results and analysis given in Section 4. Finally, we make some conclusion in Section 5.

2. Definitions and notations

A system of n cells is represented by an n vectors $X = \{x_1, x_2, \ldots, x_n\}$. We assume that the channels are equall spaced in the frequency domain and are ordered from the low-frequency band to the high -frequency band with numbers 1, 2, ..., *m*. We use an $n \times n$ nonnegative symmetric matrix *C*, called *compatibility matrix* [7], to represent EMC. Each diagonal element c_{ii} (where $i=j$) in *C* represents the CSC, and the rest of elements, c_{ij} (where $i \neq j$), represents the ACC or CCC. A demand vector $R=\{r_1, r_2, \ldots, r_n\}$ describes the channel requirements for each cell. Each element r_i in *R* represents the minimal number of channels to be assigned to cell x_i.

The FCAP is specified by the triple (*X*, *R*, *C*), where *X* is a cell system, *R* is a requirement vector, and *C* is a compatibility matrix. Let $N=\{1, 2, \ldots, n\}$ be a set of available channels, and H_i the subset of *N* assigned to x_i. The objective of FCAP is to find an assignment $H=\{H_1, H_2, \ldots, H_n\}$, which satisfies the following conditions

$|H_i| = r_i$, for $1 \leq i \leq n$,

and

$|h-h| \geq c_{ij}$, for all $h \in H_i, h \in H_j$, where $1 \leq i \leq n$ and $1 \leq j \leq n$,

where $|H_i|$ denotes the number of channels in the set of H_i. We call such assignment an *admissible assignment.* Notations used in this paper are summarized in Table 1.

Table 1. Notations used in this paper.

n	Number of cells in the mobile network
m	Number of available channels in the mobile network
R	Required channel number matrix, $R=(r_i)$
r_i	Required number of channels for cell i, $1 \leq i \leq n$
C	Compatibility matrix, $C=(c_{ij})$
c_{ij}	Minimal channel separation between channels in cell i and j, $1 \leq i,j \leq n$
d_{csc}	Minimal distance between channels of the same cell for CSC. The d_{csc} for cell i is c_{ii}.
ACC	Adjacent channel constraint
CCC	Co-channel constraint
CSC	Co-site constraint

In our channel assignment algorithm, we define an energy function *E* to keep track of the status of the simulation system. When *E* is down to zero, we find an admissible assignment. In addition to the *energy function E*, two more functions, the *feasibility* and *support function*, are defined and used with our algorithm. In the following, we give more specific definition and a brief description of these three functions.

The energy function E

The energy function *E* is defined as follows.

$$E = \sum_{i=1}^{n} \sum_{a=1}^{m} \sum_{j=1}^{n} \sum_{b=1}^{m} p(i,a) * Obj(i,a,j,b) * p(j,b)$$

Where

$$p(i,a) = \begin{cases} 1, & \text{if channel } a \text{ is assigned to cell } x_i. \\ 0, & \text{otherwise.} \end{cases}$$

$$Obj(i,a,j,b) = \begin{cases} 0, & \text{if distance between channel } a \text{ and } b \text{ is greater or equal to } c_{ij}, \text{ that is, } |a-b| \geq c_{ij}, \\ 1, & \text{otherwise.} \end{cases}$$

$Obj(i, a, j, b)$ is set to one if the assignment of channel *a* to cell x_i and channel *b* to cell x_j violates the EMC. As a result, *E* represents the total number of these violations in an assignment. We can then use *E* to evaluate the extent of violations in current assignment.

The feasibility function

Without considering the CCC and the ACC, we notice that cells with high channel requirement or with large d_{csc} are more difficult to obtain an assignment that satisfies the CSC. According to our experimental experience, these difficult ones are the bottleneck of the channel assignment procedure. If we can tackle these difficult ones first, it is more likely to obtain an admissible assignment quickly. Based upon this intuition, we define the *feasibility function* of a cell as follows.

$$feasibility(x_i) = m - r_i * c_{ii}$$

A cell with more feasible combinations of channel assignment that satisfies CSC has a larger *feasibility* value. Intuitively, the value of *feasibility* function indicates the flexibility of a cell for channel assignment. Channel assignment procedure will then proceed according to the ascending order of the calculated *feasibility* values of each cell.

The support function

The degree of support for assigning a channel to a cell is computed by counting the votes from the neighbors of that cell. A channel *a* that is assigned to a cell x_i without violating the EMC will be given a positive vote by its neighboring cells. On the other hand, a negative vote will be given. Accordingly, the *support* of channel *a* assigning to cell x_i can be defined as follows

$$support\ (a, x_i) = \sum_{i=1}^{n} \sum_{b=1}^{m} S(i,a,j,b) * p(j,b),$$

where

$$p(j,b) = \begin{cases} 1, & \text{if channel } b \text{ is assigned to cell } x_j, \\ 0, & \text{otherwise.} \end{cases}$$

$$S(i,a,j,b) = \begin{cases} 1, & \text{if distance between channel } a \text{ and } b \text{ is greater or equal to } C_{ij}, \\ -1, & \text{otherwise.} \end{cases}$$

Intuitively, the value of support function of a channel assigned to a particular cell represents the degree of supports from its neighboring cells for the particular channel assignment. In our algorithm, the support function is used to select the most promising channels for each cell.

3. The CSC-Preserving FCA Algorithm

The most essential idea of our algorithm is preserving the CSC for each cell throughout the channel assignment procedure. Assigning channels to cells in the FC problem involves many inter-dependent factors. Our thought was that if we could limit one or more of these factors, we might have more control over the FCA process and improve the efficiency of our algorithm. We choose to preserve CSC because it is the only EMC constraints that can be locally determined with each cell. The other essential feature of our algorithm is adopting a fine-tuning procedure to escape from a local minimum if certain conditions are met (i.e., if the simulation system is ver close to reach a successful assignment). As will be shown later, by preserving the CSC and adopting a fine-tuning procedure, the energy function E defined in Section 2 does converge to zero much more quickly than those without.

Our new FCAP algorithm includes mainly three procedures. At first, the *initializing procedure* assigns a set of channels that satisfy the CSC to each cell without considering ACC or CCC. Then the *updating procedure* tries to reduce the energy function E by changing the current assignment without violating the CSC. If the energy function E is equal to zero then we successfully obtain an admissible assignment. On the other hand, if E keeps unchanged, the system is called in a locked state. An *unlocking procedure* is then activated. The unlocking procedure first tries t find a free channel from neighboring cells to resolve these conflicts. If it does not work, we randomly reset the channel assignment of each cell, again without violating the CSC. Then go back to the updating procedure until simulating system terminates. If the energy function of the system is greater than zero af ter the maximum number of iterations is reached, the simulating system terminates and fails to converge (the maximum number of iterations is set to 500 in our simulations).

Our new approach is summarized in the CSC -preserving algorithm as follows. (The flow chart of our algorithm is shown in Figure 1.) Note that, throughout the entire process, none of these temporary assignments violates co-site constraint.

CSC-preserving algorithm.

Step 1. Determine an assignment to each cell with the initializing procedure.

Step 2. Process all cells sequentially according to the ascending order of value of the feasibility function of each cell. For each cell we determine the ne assignment by applying the *updating procedure.*

Step 3. If E=0, stop the procedure. An admissible assignment is then successfully obtained.

Step 4. If E makes no change when comparing to the assignment prior to Step 2, launch the *unlocking procedure* and go to Step 3. Otherwise, go to Step 2 for another iteration.

In the following, we give more detailed descriptions of the three major procedures described above.

3.1 Initializing procedure

Initially, we assign channel set H_i={1, 1 c_{ii} , 1+2× c_{ii}, ..., 1+(r_i −1)× c_{ii}} to the cell x_i. It is a straightforward way to assign each cell such that the assignment satisfies the CSC and the channel requirement. The procedure is shown below.

The initializing procedure.

For each cell x_i, (i:= 1 to n)

For channelnum := 0 to ri-1 d

Assign channel {1+channelnum × cii } to cell i.

3.2 Updating procedure

The updating procedure is intended to reduce the energy function efficiently and to preserve the CSC at the same time. The updating procedure proceeds by processin each cell sequentially. And the support function is used in this procedure for selecting the most promising channel set for each cell. The updating procedure is described as follows.

The updating procedure.

For each cell x_i, (base on the ascending order of value of the feasibility function)

Step 1. Calculate the support function of the cell f or eac channel.

Step 2. Find out a set of channels that have high support value and preserve CSC at the same time. The following algorith is propos ed to accomplish the requirement.

2.1. Create a sorted list of channel set {1, 2, ..., m} according to the descending order of the support function calculated in Step 1. Let the list be *ChList*.

2.2. Create an empty list *SeList*.

2.3. Pick the first channel a from *ChList*. Delete channel a from *ChList*.

2.4. Insert the channel a into the proper position of *SeList* so that the order of *SeList* is ascending.

2.5. Determine whether the channel a can be preserved in *SeList* or not. By function f, we can calculate that the maximal possible number of channels, which preserves the CSC and contains *SeList*. If the value of f is less than r_i, the channel a is removed from *SeList*. The function f is defined below.

$$f = \lfloor Sel_1 / d_{csc} \rfloor + \sum_{n=2}^{top} \lfloor (Sel_n - Sel_{(n-1)}) / d_{csc} \rfloor + \lfloor (m - Sel_{top}) / d_{csc} \rfloor$$

Where Sel_i is the ith channel in *SeList* and top is the number of channels in *SeList*.

2.6. If the number of channels in *SeList* is equal to r_i, then go to Step 2.7. Otherwise, go to Step 2.3

2.7. The channels in *SeList* are a set of channels that have high support value and preserve the CSC.

Step 3. If energy function E can be reduced by assigning the channels obtained in Step 2 to cell x_i, then go ahead and assign these channels to cell x_i. Otherwise, the channel assignments of cell x_i keep unchanged.

3.3 Unlocking procedure

When the value of energy function E is not equal to zero and can not be reduced any further, the system converges into a local minimum. Usually, some of channel assignments would be changed randomly to make syste unstable (to escape from the local minimum) [9], and then re-apply the updating procedure. However, we use an unlocking procedure, which consists of fine -tuning and randomizing procedures, to escape from the local minimum.

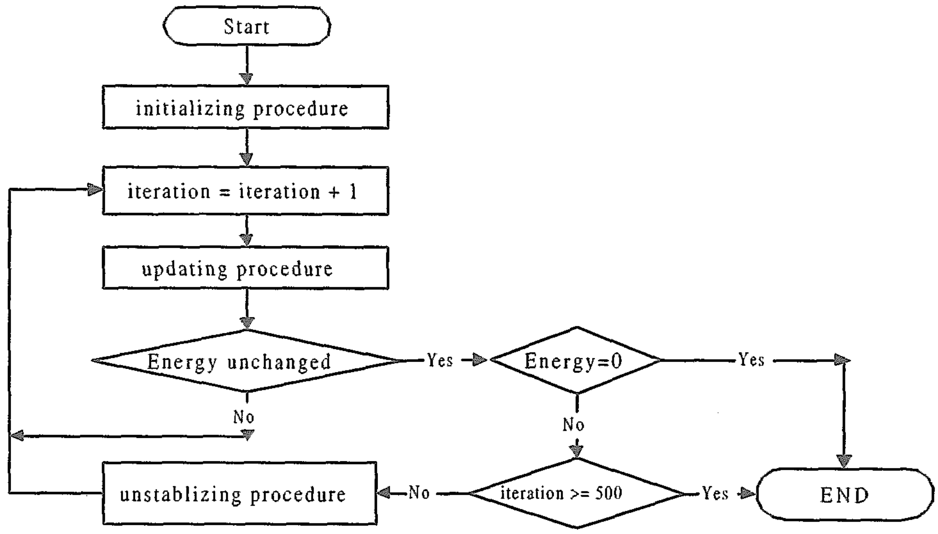

Figure 1. Flow chart of the CSC-preserving FCAP algorithm

To further illustrate the idea of the unlocking procedure, a few terms are defined as follow. A channel a is in conflict with channel b if the assignment of channel a to cell x_i and b to cell x_j does not satisfy ACC or CCC. The number of conflict pairs t determines whether to launch the fine-tuning or randomizing procedure. If the simulating system contains less than or equal to t pairs of conflicting channels, we launch the fine tuning procedure. Otherwise, when there are more than t pairs of conflicting channels or the fine-tuning procedure does not change any assignment, we launch the randomizing procedure.

3.3.1 Fine-tuning procedure

The fine-tuning procedure first finds out (if there is any) those channels that do not satisfy ACC or CCC and then tries to resolve these conflicts. If a channel a were in conflict with some other channel assigned to cell x_i. It is evident that we can replace channel a with some channel c, provided that c meet one of the two conditions (Step 3) mentioned in the following fine-tuning procedure. When this happens, conflicts can be resolved immediately without raising the energy function E. According to our experimental experience, this frequently leads to an admissible assignment. The procedure of fine -tuning stage is shown below.

The fine tuning procedure.

Step 1. Let H_i be the current channel assignment in cell x_i. Remove a from H_i.

Step 2. Try to select an unused channel c in cell x_i such that $H_i \cup \{c\}$ satisfy the CSC of cell x_i.
If such a channel c can not be found, add a back to H_i and go to Step 5.
Otherwise, mark c so that it will not be selected again.

Step 3. Define the *important set* I_i of cell x_i as follows
$I_i=\{x_k \mid \text{where } k\neq i \text{ and } c_{ik}\neq 0\}$.
If either of the following two conditions is true, we assign channel c to cell x_i and go to Step 5.
Condition1. The channel c of cell x_i does not conflict with any channel used in each cell in I_i.
Condition2. Each cell in I_i has at least one free channel. The *free channel* is unused in this cell and conflicts with no channel in every other cell.

Step 4. Go to Step 2.

Step 5. END.

3.3.2 Randomizing procedure

The randomizing procedure processes cells one by one. At first, we separate all available channels into r_i channel intervals for each cell x_i. The kth channel interval of x_i can be represented by the pair ($minpos_k$, $maxpos_k$), k=0, r_i-1. Next, we randomly select an arbitrary channel interval of the processed cell. With the selected channel interval, we select an arbitrary channel. The neighboring channel interval will be adjusted so that the next selection of channel from this channel interval does not violate CSC. The remaining channels can be decided similarly. Note that the procedure randomly changes the channel assignment of each cell to make system unstable, but still preserves CSC for each cell. The procedural steps for changing the channel assignments of cell x_i are listed as follows.

The randomizing procedure.

Step 1. Set every channel interval ($minpos_k$, $maxpos_k$) for the processed cell, where $0 \leq k \leq r_i-1$, $minpos_k = k \times c_{ii}$, and $maxpos_k = m - (r_i - k - 1) \times c_{ii}$.

Step 2. Set $currentpos_x$ be the xth channel according to the increasing order of the values of channel set assigned to the processed cell current.

Step 3. Randomly select a channel interval ($minpos_x$, $maxpos_x$), where $0 \leq x \leq r_i-1$.

Step 4. Randomly select a channel y from ($minpos_x$, $maxpos_x$). That is, $minpos_x \leq y \leq maxpos_x$).

Step 5. If $currentpos_x \leq y$, set $dir = 1$ and $limit = r_i$. Otherwise, set $dir = -1$ and $limit = -1$.

Step 6. Set $currentpos_x$=y.

Step 7. If dir=1, let $minpos_{x+1}$= $max(minpos_{x+1}, y+c_{ii})$.

Otherwise, let $maxpos_{x-1} = min(maxpos_{x-1}, y-c_{ii})$.

Step 8. Determine the next channel interval by Settin x=x+dir. If x=$limit$, then stop this procedure.

Step 9. Randomly select a channel y from ($minpos_x$, $maxpos_x$). G to Step 6.

4. Experimental results and analysis

Our simulator is developed on a personal computer with AMD K6-233 CPU. The program is coded in Delphi 3.0. Eight benchmark problems taken from [4] are examined. Table II summarizes the characteristics of these eight problems, where all the demand vectors and the compatibility matrices are also given in [4]. The iteration number is the number of updating procedure for all cells executed before E=0. To acquire the average nûmber of iterations and the convergence rates, one hundred simulation runs are performed.

Table II. Problems specifications.

Problem #	number of radio cells n	number of channels m	Compatibilit 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

Simulation results show that our algorithm achieves 100% convergence rate to solutions on eight benchmark problems. Furthermore, the number of iterations our algorith required is fewer than all previous results (see Table III, with t=3). To the best of our knowledge, no previous results achieve such high convergence rate and performance. Such significant results indicate that our approach is indeed an effective and efficient method for the fixed channel assignment problem. The average iteration number and the convergence rate to the solution are shown in Table III.

Figure 2 shows a comparison of the FCA simulations for Problem 6 [4] between ones with and without preservin CSC. In this particular case, the one that preserves CS (bottom solid curve) converges at iteration 319, while the one without (top dotted curve) fails to converge even after 500 iterations. Figure 2 also reveals that by preservin CSC, the energy level reduces to a much lower level durin the course of simulation. This does help the simulatin system converge at a faster pace.

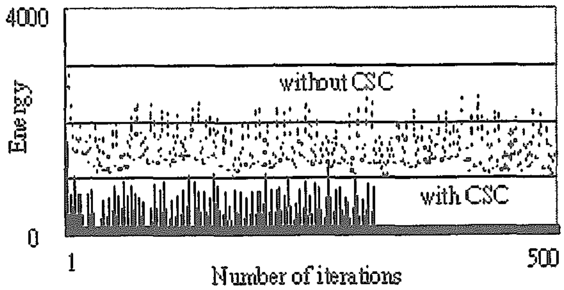

Figure 2. Comparison of the FCA simulation results between ones with and without preserving CSC.

Figure 3 compares performance for simulation of Proble 3 between the one adopting fine-tuning procedure and the one without (both preserving CSC). As we can see, the former one (dotted curve) converges at about iteration 25. However, the other (solid curve) fails to converge after 500 iterations (not shown). In fact, throughout our experiments, we notice that the fine -tuning procedure always successfully leads the simulating system away from local minimum. The threshold in determining whether to launch the fine-tuning procedure, that is, the number of conflict pairs t, could also affect overall performance of the simulation. Table 4 shows the comparison by setting t to 2 and 3. It seems that the performance would be better i we set t larger. However, the results are still quite preliminary. More simulations need to be done to gain more insight of the effects of t to the performance of fine-tuning.

Table III. Simulation results comparison with recent reported results.

Problem	Our method		Result from [4]		Result from [9]		Result from [12]	
	# of iteration	convergence rate(%)	# of iteration	convergence rate(%)	# of iteration	Convergence rate(%)	# of iteration	Convergence rate(%)
1	1	100	21.2	100	NA	NA	1	100
2	33.3	100	294.0	9	279.9	62	26382	100
3	15.4	100	147.8	93	67.4	99	NA	NA
4	1	100	117.5	100	64.2	100	NA	NA
5	40.3	100	100.3	100	126.8	98	NA	NA
6	70.9	100	234.8	79	62.4	97	63152	92
7	15.3	100	85.6	100	127.7	99	NA	NA
8	36.5	100	305.6	24	151.9	52	79502	80

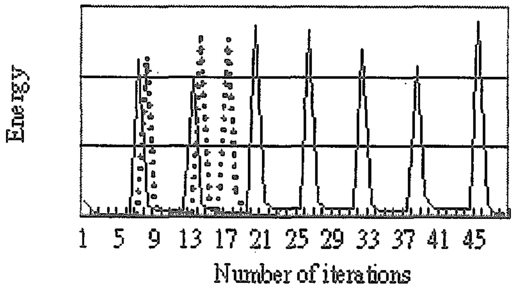

Figure 3. Comparison of the FCA simulation results between the one adopting fine-tuning procedure and the one without (both preserving CSC).

Table 4. Effect of t to the fine-tuning procedure

Problem	t=2		t=3	
	# of iteration	convergence rate(%)	# of iteration	convergence rate(%)
1	1	100	1	100
2	75.8	100	33.3	100
3	19.7	100	15.4	100
4	1	100	1	100
5	51.3	100	40.3	100
6	103.9	100	70.9	100
7	19.5	100	15.3	100
8	61.5	100	36.5	100

5. Conclusions

In this paper, we propose an algorithm that always preserves CSC throughout the FCA process. We use many techniques to improve the convergence rate and t speedup the simulating process. According to the experimental results, our algorithm has a much better performance than every other existing algorithm. Though the simulations are run in sequential mode, all the operations can be arranged for parallel executing by carefully design. How to improve initializing and fine tuning stages in the new approach is our future work. It is also quite interesting to see whether techniques developed in this paper can be applied to solve other optimization problems.

本研究係由國家科學委員會工程處專題計畫經費補助，專題計畫編號 NSC 89-2213-E-364-002

References

[1] P. T. H. Chan, M. Palaniswami, D. Everitt, "Neural network-based dynamic channel assignment for cellular mobile communication systems," *IEE Transactions on Vehicular Technology*, vol. 43, no. 2, pp. 279-287, 1994.

[2] K. Chang, J. Kim, C. Yim, and S. Kim, "An efficient borrowing channel assignment scheme for cellular mobile system," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 2, pp. 602-608, 1998.

[3] M. Duque-Anton, D. Kunz, and B. Ruber, "Channel assignment for cellular radio using simulated annealing," *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 14-21, 1993.

[4] N. Funabiki and Y. Takefuji, "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Transactions on Vehicular Technology*, vol. 41, pp. 430-436, 1992.

[5] H. Furukawa and Y. Akaiwa, "Design of underlaid microcells in umbrella cell system," *IEICE Transactions on Communication*, vol. E81 -B, no. 4, pp. 762-769, 1998.

[6] A. Gamst, "Some lower bounds for a class of frequency assignment problem," *IEEE Transactions on Vehicular Technology*, vol. VT-35, no. 1, pp. 8-14, 1986.

[7] A. Gamst and W. Rave, "On frequency assignment in mobile automatic telephone system," in *Proc. IEEE GLOBECOM'82*, pp. 309-315.

[8] S. Kim and S. L. Kim, "A two-phase algorithm for frequency assignment in cellular mobile systems," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 542-548, 1994.

[9] J. S. Kim, S. H. Park, P. W. Dowd, and N. M. Nasrabadi, "Cellular radio channel assignment using a modified Hopfield network," *IEEE Transactions on Vehicular Technology*, vol. 46, pp. 957-967, 1997.

[10] D. Kunz, "Channel assignment for cellular radio usin neural network," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 188-193, 1991.

[11] R. Mathar and J. Mattfeldt, "Channel assignment in cellular radio networks," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, 647-656, 1993.

[12] Y. Ngo, andV. O. K. Li, "Fixed channel assignment in cellular radio networks using a modified geneti algorithm," *IEEE Transactions on Vehicular Technology*, vol. 47, pp. 163-171, 1998.

[13] K. Sallberg, B. Stavenow, B. Eklundh, "Hybrid channel assignment and reuse partitioning in a cellular mobile telephone system, " in Proc. 37 th *IEE Vehicular Technology Society Conference*, pp. 405-411, 1987.

[14] W. Sung and W. S. Wong, "Sequential packing algorithm for channel assignment under cochannel and adjacent channel interference constraint," *IEE Transactions on Vehicular Technology*, vol. 46, no. 3, pp. 676-685, 1997.

[15] K. L. Yeung, T. S. Peter Yum, "Compact pattern based dynamic channel assignment for cellular mobile systems," *IEEE Transactionson Vehicular Technology*, vol. 43, no. 4, pp. 892-891