

# Sequential Packing Algorithm for Channel Assignment Under Cochannel and Adjacent-Channel Interference Constraint

Chi Wan Sung, *Student Member, IEEE*, and Wing Shing Wong, *Senior Member, IEEE*

**Abstract**—Generally, the channel-assignment problem (CAP) for mobile cellular systems is solved by graph-coloring algorithms. These algorithms, though sometimes can yield an optimal solution, do not supply any information on whether an optimal solution has been found or how far away it is from the optimum. In view of these undesirable features, two relevant results are presented in this paper. First, a lower bound for the minimum number of channels required to satisfy a given call-traffic demand is derived. This lower bound is tighter than the existing ones under certain conditions and can be used as a supplement for other approximate algorithms. Second, we propose an efficient heuristic algorithm to solve this problem. Although the CAP is nondeterministic polynomial (NP) complete in general, our algorithm provides an optimal solution for a special class of network topologies. For the general case, promising results are obtained, and numerical examples show that our algorithm has a better performance than many existing algorithms.

**Index Terms**—Cellular systems, channel assignment, graph coloring, maximum packing.

## I. INTRODUCTION

THE LIMITING availability of the radio spectrum imposes an inherent bound on the capacity of a mobile cellular system. As demands for various mobile communication services grow, the question of how to utilize the valuable bandwidth in the most efficient way becomes more and more critical. To maximize the system capacity, one typically tries to reuse the frequencies as much as possible. However, this may increase the mutual interferences among the cellular users. To maintain a certain quality of service, one has to keep the interference below a predefined level. For systems using frequency division multiple access (FDMA) or time division multiple access (TDMA), this requirement usually translates into compatibility constraints—stating for an arbitrary cell site what channels may be used for new calls based on what channels are currently used in other cell sites. Allocating the channels in an efficient way, which does not violate the compatibility constraints, is the main objective of the channel-assignment problem (CAP). A lot of research can be found in the literature. Most of the investigations are based on graph theoretic or heuristic approaches [1], [5], [6], [16],

[17]. Recently, algorithms employing neural networks [4], [12] and simulated annealing [3], [15] have also been proposed. However, neural-network-based algorithms typically yield only suboptimal solutions [13]. The simulated annealing approach, although it may be more flexible, is easily trapped in a local *minima*, which requires a lot of computation time to be escaped from [3]. In short, different approaches have their own limitations. It reflects how hard the CAP is.

In the simplest formulation of the CAP, only cochannel interference is considered. The problem is known to be equivalent to the classical graph-coloring problem. Since the graph-coloring problem is nondeterministic polynomial (NP) complete [10], an exact search for the optimal solution is impractical for a large-scale system due to its exponentially growing computation time. Hence, most of the efforts are spent in developing approximation algorithms [1], [5], [16]. These algorithms occasionally can find optimal solutions, but, in general, provide only suboptimal ones with no information on how far away they are from the optimal solution. In view of this undesirable feature, Gamst derives some lower bounds for the minimum number of channels required [7]. Our paper will provide another lower bound, which is tighter in some cases. We also propose an algorithm, which always finds the optimal solution for a special class of cellular network topologies. This optimality not only is significant in its own right, but it also yields a clue on which circumstances our algorithm has good performance. Finally, an overall better performance compared to other existing algorithms will be demonstrated by numerical examples.

## II. PROBLEM FORMULATION

Frequency sharing among different users is an important issue in mobile cellular systems. Many different multiple-access schemes have been proposed. Among them, the most popular ones are FDMA, TDMA, and code division multiple access (CDMA). In FDMA systems, the spectrum is divided into nonoverlapping frequency bands. Each user is allocated a dedicated frequency band for information transmission. In TDMA systems, each user is allocated a dedicated time slot for transmission, and different users may share the same frequency band. In CDMA systems, each user is assigned a well-designed code such that the interference among users is minimized. The entire time frame as well as the entire spectrum can be used for transmission. For simplicity, in this paper we focus on channel assignments in FDMA systems. In this context, a channel is

Manuscript received January 27, 1995; revised April 20, 1996. This work was supported by a grant from the Hong Kong Research Grants Council.

The authors are with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: cwsung3@ie.cuhk.hk; wswong@ie.cuhk.hk).

Publisher Item Identifier S 0018-9545(97)04621-5.

referred to as a frequency channel. However, the idea may as well be applied to TDMA systems, provided that we refer to a channel as a time slot and define the compatibility matrix mentioned below in an appropriate way.

We assume that the channels are equally spaced in the frequency domain and are ordered from the low-frequency band to the high-frequency band with numbers 1, 2, 3, etc. A system of  $n$  cells is represented by an  $n$  vector  $X = [x_1, x_2, \dots, x_n]$ . Each cell  $x_i$  requires  $m_i$  channels ( $m_i \geq 0$ ). This forms a requirement vector  $M = [m_1, m_2, \dots, m_n]$ . The assignment of the channels to the cells subjects to three different types of constraints.

- 1) Cochannel constraint (c.c.c.): the same channel is not allowed to be assigned to certain pairs of cells simultaneously.
- 2) Adjacent-channel constraint (a.c.c.): channels adjacent in number are not allowed to be assigned to certain pairs of cells (typically, adjacent cells) simultaneously.
- 3) Cosite constraint (c.s.c.): any pair of channels assigned to the same cell must be separated by a certain number.

The above constraints can be represented by an  $n \times n$  nonnegative symmetric matrix  $C$ , the so-called *compatibility matrix*. The  $ij$ th element  $c_{ij}$  represents the minimum difference between channels assigned to cell  $x_i$  and that assigned to cell  $x_j$ . If any pair of cells  $x_i$  and  $x_j$  is subjected to the cochannel constraint or adjacent-channel constraint, we have  $c_{ij} = 1$  or  $2$ , respectively. The cosite constraint is represented by the diagonal elements  $c_{ii}$ 's. Typically,  $c_{ii}$  is greater than or equal to five.

The CAP is specified by the triple  $P = (X, M, C)$  [7]. Let  $\{1, 2, \dots, N\}$  be a set of channels and  $H_i$  the set of channels assigned to cell  $x_i$ . The objective of the problem is to find the minimum value of  $N$  such that there exists an assignment pattern  $H = \{H_1, H_2, \dots, H_n\}$ , which satisfies the following conditions:

$$|H_i| = m_i, \quad \text{for all } i$$

and

$$|h - h'| \geq c_{ij}, \quad \text{for all } i, j$$

where  $|H_i|$  denotes the number of channels in the set of  $H_i$ , and  $h, h'$  denote an arbitrary channel in  $H_i$  and  $H_j$ , respectively.

This problem is equivalent to a generalized graph-coloring problem [16]. We represent each cell by a vertex with weight  $w_i = m_i$ . If  $c_{ij} > 0$ , the vertices  $v_i$  and  $v_j$  are joined together by an edge with label  $c_{ij}$ . The resulting graph is called an *interference graph*. The CAP is equivalent to assigning positive integers  $\{1, 2, \dots, M\}$  to the vertices such that each vertex has  $w_i$  integers assigned. The difference between the integers assigned to two adjacent vertices must not be less than the edge label. The objective is to minimize the maximum integer used. In the special case, where only cochannel interference is considered, the  $c_{ij}$ 's are either zero's or one's. This problem can then be transformed easily to the classical graph-coloring problem. In the next section, we will consider this special case first.

### III. PURE COCHANNEL INTERFERENCE CASE

The pure cochannel-interference problem can be defined by a *topology graph*  $G$  with  $n$  vertices representing the  $n$  cells: each vertex has a weight  $w_i$  ( $1 \leq i \leq n$ ). A feasible coloring solution assigns colors to the vertices with the constraint that no two adjacent vertices have the same color. Moreover, a vertex  $v_i$  with a weight  $w_i$  needs to be assigned  $w_i$  colors. The objective of the problem is to find a solution with the minimum number of colors. The optimal policies are termed the maximum packing (MP) assignments [2].

Unfortunately, for an arbitrary graph, the problem of determining an MP assignment is NP complete. Hence, MP is an ideal concept rather than a practical solution. However, for graphs of special structures, efficient algorithms to compute MP assignments may exist, and we call them *MP algorithms*.

In this paper, a heuristic algorithm is proposed. It has the property of yielding solutions with performance close to the MP assignments. Moreover, for a special class of network topology, it can be proved that this heuristic method is an MP algorithm.

Before we proceed, we have to define some terms. First of all, we define the neighborhood of  $v$ ,  $N(v)$  as the set of  $v$ 's adjacent vertices. A set of vertices in a graph, which are interconnected, is called a *clique*. For every clique, we define its *clique weight* as the sum of weights of all the vertices inside it. A vertex typically belongs to more than one clique. We denote  $C_{\max}(v)$  as the clique, which contains  $v$  and has maximum weight. (Ties are resolved randomly.) The *maximum clique weight* of  $v$ , denoted as  $W(v)$ , is defined as the clique weight of  $C_{\max}(v)$ . When it is necessary to make the corresponding graph  $G$  explicit, we write it as  $W(v|G)$ .

Basically, our algorithm uses the *requirement exhaustive strategy* [5]. We pick up a color  $c_i$  and assign it to the vertices one by one until no further assignment of that color is possible. Then, the next color  $c_{i+1}$  is used, and the procedure is started over again. The question is how to determine which vertices should be colored by  $c_i$ . We choose the vertex with the greatest weight as the first vertex. To choose the subsequent vertices, the principle of the *maximum overlap of denial areas* as defined in the third method in [5] is used. This principle states that a channel should be assigned to the cell whose denial area has maximum overlap with the already existing denial area of that channel. (The denial area for a cell  $c$  is the set of neighboring cells, which cannot share the same frequency with  $c$  due to cochannel interference. The denial area of a channel is the set of vertices, which cannot be assigned with that channel.) Our algorithm differs from the algorithm proposed in [5] in the way the overlap is defined. In our algorithm, we define the overlap as the number of cells within the intersection of the two denial areas. In [5], overlap is defined as the sum of the requirements of the cells within the intersecting areas. Our definition ensures that the cells to which a channel is assigned can be packed as close to each other as possible. When there is a tie, we break it by choosing the vertex with the largest maximum clique weight with respect to the topology graph induced by the intersection of the denial areas. The rationale of this rule is that the larger the maximum clique weight is,

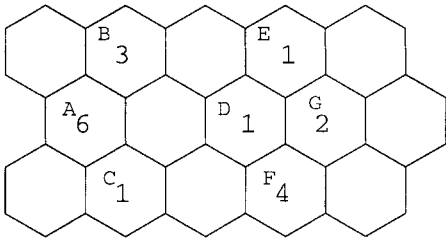


Fig. 1. A seven-cell system for the demonstration of the SP algorithm.

the more difficult it is for that vertex to be colored. We call our algorithm the sequential packing (SP) algorithm, whose pseudocode is shown as follows. Note that  $A$  denotes the set of vertices being colored by the current color  $c$ , and  $F$  denotes the vertices, which are forbidden to be colored by  $c$ :

```

procedure SP [ $G(V, E)$ : graph]
   $c := 1$ ;
  while  $G$  has 1 or more vertices do
    let  $A$  and  $F$  be empty sets;
    let  $v$  be a vertex in  $\{v_i : v_i \in V \text{ and } w_i = \max_{v_j \in V} w_j\}$ ;
    repeat
      put  $v$  into  $A$ ;
       $F := F \cup \{v\} \cup N(v)$ ;
       $m := \max_{v_i \in V \setminus F} |N(v_i) \cap F|$ ;
      if  $m = 0$ 
        let  $v$  be a vertex in  $\{v_i : v_i \in V \setminus F \text{ and } w_i = \max_{v_j \in V \setminus F} w_j\}$ ;
      else
         $K := \{v_i : v_i \in V \setminus F \text{ and } |N(v_i) \cap F| = m\}$ ;
        if  $|K| = 1$ 
          then let  $v$  be the only vertex in  $K$ ;
        else
          for each  $v_i \in K$  do
            construct a subgraph  $S_{v_i}$  induced by the vertex set  $(N(v_i) \cap F) \cup \{v_i\}$ ;
            let  $v$  be a vertex in  $\{v_i : v_i \in K \text{ and } W(v_i|S_{v_i}) = \max_{v_j \in K} W(v_j|S_{v_j})\}$ ;
          end{else}
        end{else}
      until  $F = V$ ;
      for each  $v_i \in A$  do
        color it with  $c$ ;
        increase  $c$  by 1;
        decrease  $w_i$  by 1;
        if  $w_i = 0$ 
          then delete from  $G$  the vertex  $v_i$  and all the edges connecting to  $v_i$ ;
        end{for}
      end{while}
    end.
  
```

*Example:* Fig. 1 shows a system of seven cells  $\{A, B, \dots, G\}$ . The number of required channels or weight of each cell is specified inside the corresponding hexagon. Only cochannel constraint is considered. A cluster size [14]  $N_c$  of seven is assumed. In a two-dimensional (2-D) hexagonal

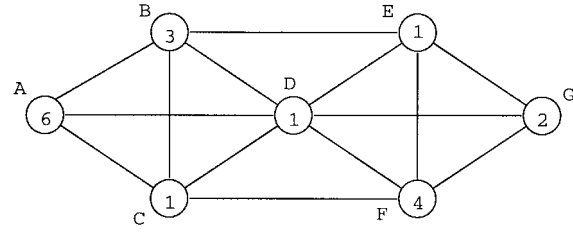


Fig. 2. The corresponding graph of the seven-cell system.

system, it corresponds to a two-cell buffering scheme, which means that a channel used by a particular cell cannot be shared by any other cell whose distance from the original one is less than or equal to two “cells.” For example, in Fig. 1, a channel used by cell  $A$  can only be reused by cells  $E$ ,  $F$ , or  $G$ . This system can be represented by a topology graph (shown in Fig. 2).

- 1) First of all, we use  $c_1$  to color vertex  $A$ , which has the greatest weight among all the vertices. The denial area, denoted by  $S$ , becomes  $\{B, C, D\}$ .
- 2) Next, we choose the vertex whose denial area has maximum overlap with the current denial area. Totally, there are three candidates, namely, vertices  $E$ ,  $F$ , and  $G$ :
  - a) (denial area of  $E$ )  $\cap S = \{B, D\}$ ;
  - b) (denial area of  $F$ )  $\cap S = \{C, D\}$ ;
  - c) (denial area of  $G$ )  $\cap S = \{D\}$ .
- 3) We choose the one with maximum cardinality in the intersecting region. In this case, there is a tie between vertices  $E$  and  $F$ . To break the tie, we first form a subgraph for vertices  $E$  and  $F$ . The subgraph is induced by the candidate vertex and the vertices in the overlapping of the denial areas. Then, we choose the vertex that has a larger maximum clique weight in the corresponding subgraph.
  - a) For vertex  $E$ , consider the subgraph induced by  $\{B, D, E\}$ . The maximum clique weight of  $E$  is five.
  - b) For vertex  $F$ , consider the subgraph induced by  $\{C, D, F\}$ . The maximum clique weight of  $F$  is six.

Therefore, vertex  $F$  is chosen and colored by  $c_1$ . The new denial area  $S$  becomes  $\{B, C, D, E, G\}$ .

- 4) No more vertices can be colored by  $c_1$ . Then, the weight of  $A$  and  $F$  are both decreased by one. The next color,  $c_2$ , is used, and the procedure is repeated.

As will be discussed later, Theorem 1 shows that this algorithm yields an optimal solution for this simple example. Besides, it is worth noting that the most time-consuming task in this algorithm is the calculation of clique weight. The calculation of clique weight is needed only if there is a tie in the maximum overlap criterion. The number of candidate vertices involved in the tie must be less than  $n$ . A vertex typically belongs to more than one clique. However, due to the cellular structure, a vertex cannot belong to more than  $k$  cliques, where  $k$  is a constant. Therefore, in choosing a vertex to be colored, the clique-weight calculation is less than  $kn$ .

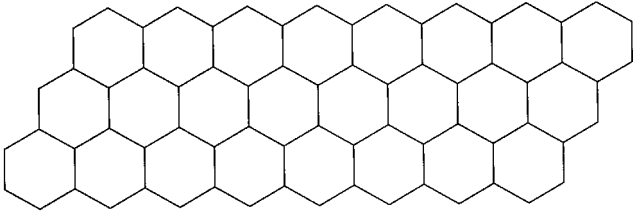


Fig. 3. Example of a three-stripe system.

Since there are only  $n$  vertices, the number of vertices being assigned the same color must be less than  $n$ . As a result, the number of calculations is  $O(Nn^2)$ , where  $N$  is the number of colors used and  $n$  is the number of vertices.

Although the CAP  $P(X, M, C)$  is NP complete in general, for the pure cochannel case, it turns out that the SP algorithm allocates channels optimally for networks with certain special structure.

From now on, to facilitate the discussion, we will make the standard technical assumption that cells are laid out in the regular hexagonal tiling pattern. We define a system consisting of  $i$  rows of cells an *i-stripe cellular system*. Notice that a linear cellular system is a one-stripe system. A three-stripe system is shown in Fig. 3. Furthermore, we assume that the cochannel constraint is equivalent to a cluster size  $N_c$  of seven.

*Theorem 1:* For a graph arising from an *i-stripe cellular system* with  $i$  less than or equal to three and a cluster size equal to seven, the SP algorithm always yields a solution, which has the smallest possible number of colors used. In other words, it is an MP algorithm for this special class of graphs.

The proof of Theorem 1 is given in Appendix A.

#### IV. GENERALIZED SEQUENTIAL PACKING

The SP algorithm stated in the previous section can only be used in the pure cochannel case. In this Section, we will generalize it to include the adjacent channel ( $c_{ij} = 2$ ) and cosite constraints.

In the pure cochannel case, cells using the same channel are packed closely to each other such that the utilization of each channel is maximized. However, this may not be a good policy if we have to take account of the adjacent-channel constraint for the reason that a closely packed channel will leave little room for its adjacent channels. Because of this mutual intervention, we pack the channels on a two-by-two basis.

The generalized sequential packing (GSP) algorithm can be described as follows. Suppose the round to color with  $c_i$  and  $c_{i+1}$  has just been initiated. We first find the vertex with the greatest weight. The tie is broken arbitrarily. If this vertex can be colored by  $c_i$ , we color it using  $c_i$ . Otherwise, we use  $c_{i+1}$ . If both colors cannot be assigned, we choose the vertex with the next greatest weight. This process repeats until a vertex that can be colored by either  $c_i$  or  $c_{i+1}$  is found. We call this the maximum weight criterion. Then, we try to color the remaining vertices in a round-robin fashion. We first find the set of vertices that are allowed to be colored by  $c_i$ , but not  $c_{i+1}$ . Call this set  $P_i$ . If  $P_i$  is empty, it becomes  $c_{i+1}$ 's turn. If not, we choose a vertex from  $P_i$  using the same criterion of

SP, i.e., first by the principle of maximum overlap of denial area and then by the maximal clique weight if there is a tie. This vertex is then colored by  $c_i$ . This process repeats until  $P_i$  is empty. Afterwards, we continue the coloring using  $c_{i+1}$ . If both  $P_i$  and  $P_{i+1}$  are empty, we start the process again using the maximum weight criterion. The whole procedure repeats until no more vertices can be colored by either  $c_i$  or  $c_{i+1}$ .

The above procedure can be more succinctly described by the following pseudocode. We use  $F_n$  to denote the set of vertices that are forbidden to be colored by  $c_n$  due to the interference constraints

- 1) if  $F_n = V$  and  $F_{n+1} = V$   
return
- if  $F_n \neq V$   
let  $v_i \in (V \setminus F_n)$  which has maximum weight.  
let  $w_i$  be the weight of  $v_i$
- else  
let  $w_i := 0$
- if  $F_{n+1} \neq V$   
let  $v_j \in (V \setminus F_{n+1})$  which has maximum weight  
let  $w_j$  be the weight of  $v_j$
- else  
let  $w_j := 0$
- if  $w_i \geq w_j$   
color  $v_i$  by  $c_n$
- else  
color  $v_j$  by  $c_{n+1}$
- update  $F_n$  and  $F_{n+1}$ ;
- 2) let  $P_n := (V \setminus F_n) \cap F_{n+1}$  and  $P_{n+1} := F_n \cap (V \setminus F_{n+1})$   
if  $P_n = \phi$  and  $P_{n+1} = \phi$   
goto 1;
- 3) if  $P_n = \phi$   
goto 4
- else if  $N(v_i) \cap F_n = \phi$  for all  $v_i \in P_n$   
choose  $v_i \in P_n$  by the maximum weight criterion
- else  
choose  $v_i \in P_n$  by the principle of maximum overlapping  
(tie is broken by the condition of maximum clique weight)
- color  $v_i$  by  $c_n$
- update  $F_n, F_{n+1}, P_n$  and  $P_{n+1}$
- repeat 3;
- 4) if  $P_{n+1} = \phi$   
goto 2
- else if  $N(v_i) \cap F_{n+1} = \phi$  for all  $v_i \in P_{n+1}$   
choose  $v_i \in P_{n+1}$  by the maximum weight criterion
- else  
choose  $v_i \in P_{n+1}$  by the principle of maximum overlapping  
(tie is broken by the condition of maximum clique weight)
- color  $v_i$  by  $c_{n+1}$
- update  $F_n, F_{n+1}, P_n$  and  $P_{n+1}$
- repeat 4.

We have two variations of GSP. In the first variation, when no further coloring by  $c_i$  or  $c_{i+1}$  is possible, we use the next



Fig. 4. A seven-cell linear system for the demonstration of the GSP algorithm.

two colors  $c_{i+2}$  and  $c_{i+3}$ . The second one, however, will “uncolor” the vertices already colored by  $c_{i+1}$ . The procedure is started again using  $c_{i+1}$  and  $c_{i+2}$  instead. We call these two variations GSP1 and GSP2, respectively.

*Example:* In Fig. 4, a linear network with seven cells is shown. The channel requirement of each cell is shown inside the cell. We assume that the cochannel constraint is equivalent to a cluster size of three, i.e., two-cell buffering, and adjacent channels cannot be used by adjacent cells. No cosite constraint is imposed.

Let  $c_i^{(j)}$  be the  $j$ th assignment, where channel  $i$  is allocated. In other words, the superscript  $j$  specifies the order of the assignment. First of all, we consider the assignment if SP is used, i.e., channels are assigned one by one instead of two by two (cf. GSP). In case of a tie, the cell in the left-most position is chosen.

The assignment pattern of SP is

$$\left\{ \left\{ c_1^{(2)} \right\}, \left\{ c_3^{(4)} \right\}, \left\{ c_5^{(6)} \right\}, \left\{ c_1^{(1)}, c_7^{(8)} \right\}, \right. \\ \left. \left\{ c_3^{(5)} \right\}, \left\{ c_5^{(7)} \right\}, \left\{ c_1^{(3)} \right\} \right\}.$$

Although  $c_1$  is packed in the most compact way, there is no room for  $c_2$  due to the adjacent-channel constraint. This is the motivation for the design of GSP. Now, we consider GSP1. The assignment pattern is

$$\left\{ \left\{ c_4^{(5)} \right\}, \left\{ c_2^{(2)} \right\}, \left\{ c_5^{(7)} \right\}, \left\{ c_1^{(1)}, c_3^{(4)} \right\}, \right. \\ \left. \left\{ c_6^{(8)} \right\}, \left\{ c_2^{(3)} \right\}, \left\{ c_4^{(6)} \right\} \right\}.$$

At first,  $c_1$  is assigned to the fourth cell (vertex), which has the greatest weight. Now, since we cannot find a vertex that can be colored by  $c_1$ , but not  $c_2$ , we stop using  $c_1$  and try  $c_2$ . The second and the sixth vertex can be colored by  $c_2$ , but not  $c_1$ . So, we color them using  $c_2$ . No more vertices can be colored by either  $c_1$  or  $c_2$ . Therefore, we start the procedure again using  $c_3$  and  $c_4$ . Finally, the assignment pattern shown above can be obtained.

Next, we show the assignment procedure of GSP2, shown at the bottom of the page.

The first step is simply the same as GSP1. However, in the second step, we “undo” the coloring of  $c_2$ . The procedure then continues using  $c_2$  and  $c_3$ . Finally, a feasible assignment is derived.

In this example, SP uses seven channels, while both GSP1 and GSP2 use six. It demonstrates the effectiveness of packing channels on a two-by-two basis. In fact, the solution obtained by GSP1 or GSP2 is optimal for this problem.

#### V. A LOWER BOUND FOR THE GENERAL CASE

Before evaluating the performance of the algorithm GSP, it is useful to first derive a lower bound for the evaluation of different channel-assignment algorithms.

In [7], several lower bounds are given. However, we find that in some cases, the result obtained by our proposed GSP is still quite far away from the tightest lower bound given. This motivates us to improve the bound. Here, we will derive a lower bound, which, in some cases, is tighter than those given in [7].

As in [7], we use  $S_0(P)$  to denote the minimum number of channels used for problem  $P$ , and we call a subset  $Q$  of  $X$   $v$ -complete if

$$c_{ij} \geq v, \quad \text{for all } x_i, x_j \in Q.$$

Note that a 1-complete subset is equivalent to a clique. The concept of a  $v$ -complete subset is just a generalization of a clique.

*Theorem 2:* Let  $P = (X, M, C)$  be a CAP and  $Q$  be a 1-complete subset of  $X$ . Let  $x_i \in Q$ . Assume  $c_{ii} = k > u > 1$  and there exists a subset of  $Q, R$  such that

$$x_i \notin R$$

and

$$c_{ij} \geq u, \quad \text{for all } x_j \in R.$$

Furthermore, let  $m_R = \sum_{j \in R} m_j$ . If  $k - 2u + 1 \leq 0$

$$S_0(P) \geq (m_i - 1)k + 1 + m_R \quad (1)$$

else

$$S_0(P) \geq (m_i - 1)k + 1 \\ + \max(m_R - (m_i - 1)(k - 2u + 1), 0). \quad (2)$$

*Proof:* Define  $P' = (X, M', C')$  with  $M'$  having only two nonzero components

$$m'_i = m_i(x_i \in Q)$$

and

$$m'_j = m_R(x_j \in R).$$

The entries of the compatibility matrix  $C'$  are

$$c'_{ii} = k, \quad c'_{ij} = c'_{ji} = u, \quad \text{and} \quad c'_{jj} = 1.$$

$$\begin{array}{l} 1) \left\{ \phi, \quad \left\{ c_2^{(2)} \right\}, \quad \phi, \quad \left\{ c_1^{(1)} \right\}, \quad \phi, \quad \left\{ c_2^{(3)} \right\}, \quad \phi \right\} \\ 2) \left\{ \left\{ c_2^{(1)} \right\}, \quad \phi, \quad \left\{ c_3^{(2)} \right\}, \quad \left\{ c_1 \right\}, \quad \phi, \quad \left\{ c_2^{(3)} \right\}, \quad \phi \right\} \\ 3) \left\{ \left\{ c_2 \right\}, \quad \phi, \quad \left\{ c_3^{(1)} \right\}, \quad \left\{ c_1 \right\}, \quad \left\{ c_4^{(2)} \right\}, \quad \left\{ c_2 \right\}, \quad \phi \right\} \\ 4) \left\{ \left\{ c_2 \right\}, \quad \left\{ c_5^{(2)} \right\}, \quad \left\{ c_3 \right\}, \quad \left\{ c_1 \right\}, \quad \left\{ c_4^{(1)} \right\}, \quad \left\{ c_2 \right\}, \quad \left\{ c_5^{(3)} \right\} \right\} \\ 5) \left\{ \left\{ c_2 \right\}, \quad \left\{ c_5^{(1)} \right\}, \quad \left\{ c_3 \right\}, \quad \left\{ c_1, c_6^{(2)} \right\}, \quad \left\{ c_4 \right\}, \quad \left\{ c_2 \right\}, \quad \left\{ c_5^{(3)} \right\} \right\}. \end{array}$$

TABLE I  
PERFORMANCE OF SP UNDER DIFFERENT SYSTEM LAYOUTS  
(COCHANNEL CASE)

| dimension of system layout | #. of times optimal solution found (out of 30) | worst case performance | average performance |
|----------------------------|--|------------------------|---------------------|
| $10 \times 10$             | 16   | 7.40%                  | 1.22%               |
| $5 \times 20$              | 22   | 3.59%                  | 0.30%               |

By Lemmas 4 and 5 in [7],  $S_0(P) \geq S_0(P')$ . Now, we want to find  $S_0(P')$ . It is obvious that

$$S_0(P') \geq (m'_i - 1)k + 1$$

since the channels used in  $x_i$  must be separated with minimum distance  $k$ .

For any two channels in  $x_i$  spaced with distance  $k$ , the number of channels that can be used by cells in  $R$  between the gaps is

$$n = (k - 1) - 2(u - 1) = k - 2u + 1.$$

If  $n \leq 0$ , no gap exists and  $m'_i$  more channels are needed. Hence, (1) is obtained.

On the contrary, if  $n > 0$ , there are  $(m'_i - 1)n$  usable channels left inside all the gaps. If  $m'_j \leq (m'_i - 1)n$ , no additional channel is needed. Otherwise, we need  $m'_j - (m'_i - 1)n$  more channels. Hence, (2) is obtained.  $\square$

An example illustrating Theorem 2 is shown in the next section.

## VI. NUMERICAL EXAMPLES

### A. Example 1: Pure Cochannel Case

In the pure cochannel case, it is well-known that the *clique number*

$$\rho = \max_{Q:\text{clique}} \sum_{x_i \in Q} m_i$$

provides a lower bound for the number of channels needed since the channels assigned to the same clique must all be different. Hence, we will use this bound to judge the performance of SP under different system topologies.

We compare two different layouts of hexagonal cells. The first one we considered is a  $10 \times 10$  system, and the second one is  $5 \times 20$ . We assume a cluster size  $N_c$  of seven. The channel requirement in each cell is generated randomly, ranging from 1 to 100. Thirty instances of each system are obtained by varying the seed of our random-number generator. Generally, we do not know the optimal solution, except when the solution of our algorithm is the same as the lower bound. So, we use the percentage of additional channels required relative to the lower bound as the performance measure.

The result is shown in Table I. It can be seen that SP performs better when applying to the  $5 \times 20$  system in both the worst and average case. An optimal solution is found 22 out of 30 times. It is reasonable to expect that the “narrower” the network structure, the better the performance of SP.

In general, the performance of SP is acceptable in light of the fact that the problem to find an algorithm, which can

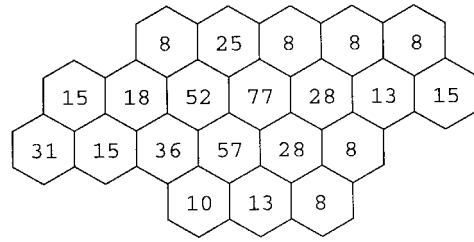


Fig. 5. Example 2 and case 1 of example 3. The numbers in the cells represent the corresponding  $m_i$ .

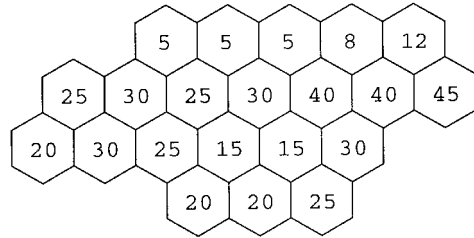


Fig. 6. Case 2 of example 3. The numbers in the cells represent the corresponding  $m_i$ .

guarantee the obtained solution does not exceed the optimal value by more than 100%, is NP complete [9].

### B. Example 2: General Case (Lower Bound)

We take an example from [7] to demonstrate that the lower bound presented in the previous section can be tighter than that in [7]. The cellular layout is shown in Fig. 5. The numbers in the cells represent the corresponding channel requirements. As in [7], we assume cochannel constraints equivalent to a 12-cell cluster, adjacent-channel constraints for adjacent cells, and the cosite constraint  $c_{ii} = 5$ .

Let cell  $A$  be the cell that requires 77 channels. Since  $c_{ii} = 5$ , the most compact way to assign channels to cell  $A$  is

$$1, 6, 11, 16, \dots, 376, 381.$$

Due to the adjacent-channel constraints, channels that can be used by its neighbors are

$$3, 4, 8, 9, 13, 14, \dots, 378, 379.$$

Therefore, there are  $(77 - 1) \times 2 = 152$  channels inside the “gaps.” Since the total requirement of all its six neighbors is 198, an additional  $198 - 152 = 46$  channels are needed. Hence, the lower bound is  $381 + 46 = 427$ . This bound is tighter than the best lower bound given in [7], which is just 414.

The derivation of this lower bound is in the same spirit as the proof of Theorem 2. Actually, we can directly apply Theorem 2 with  $k = 5$  and  $u = 2$ . Let  $x_i$  be cell  $A$  and  $R$  the set containing the six adjacent cells of  $x_i$ . Then,  $m_R = 198$ . Equation (2) gives a lower bound of 427 as above.

### C. Example 3: General Case (Algorithmic Results)

We now compare GSP1 and GSP2 with the algorithms proposed by Box [1] and Sivaraman *et al.* [16]. The examples we use are taken from [7] and [16]. The network structure and channel requirements are shown in Figs. 5 and 6.

TABLE II  
ALGORITHMIC RESULTS CORRESPONDING TO THE PROBLEM IN FIG. 5

| Problem #. | $N_c$ | a.c.c. | $c_{ii}$ | LB  | Sivarajan's | Box's<br>(50 iterations) | Box's<br>(100 iterations) | GSP1 | GSP2 |
|------------|-------|--------|----------|-----|-------------|--------------------------|---------------------------|------|------|
| P1         | 12    | 2      | 5        | 427 | 460         | 449                      | 446                       | 440  | 450  |
| P2         | 7     | 2      | 5        | 427 | 447         | 446                      | 445                       | 436  | 444  |
| P3         | 12    | 2      | 7        | 533 | 536         | 533                      | 533                       | 565  | 533  |
| P4         | 7     | 2      | 7        | 533 | 533         | 535                      | 533                       | 563  | 533  |
| P5         | 12    | 1      | 5        | 381 | 381         | 381                      | 381                       | 381  | 381  |
| P6         | 7     | 1      | 5        | 381 | 381         | 381                      | 381                       | 381  | 381  |
| P7         | 12    | 1      | 7        | 533 | 533         | 533                      | 533                       | 533  | 533  |
| P8         | 7     | 1      | 7        | 533 | 533         | 533                      | 533                       | 533  | 533  |

TABLE III  
ALGORITHMIC RESULTS CORRESPONDING TO THE PROBLEM IN FIG. 6

| Problem #. | $N_c$ | a.c.c. | $c_{ii}$ | LB  | Sivarajan's | Box's<br>(50 iterations) | Box's<br>(100 iterations) | GSP1 | GSP2 |
|------------|-------|--------|----------|-----|-------------|--------------------------|---------------------------|------|------|
| Q1         | 12    | 2      | 5        | 258 | 283         | 274                      | 274                       | 291  | 273  |
| Q2         | 7     | 2      | 5        | 258 | 270         | 273                      | 273                       | 273  | 268  |
| Q3         | 12    | 2      | 7        | 309 | 310         | 309                      | 309                       | 314  | 309  |
| Q4         | 7     | 2      | 7        | 309 | 310         | 309                      | 309                       | 323  | 309  |
| Q5         | 12    | 2      | 12       | 529 | 529         | 529                      | 529                       | 530  | 529  |

The original algorithm proposed by Box attempts to satisfy the requirements using a given number of channels  $N$ . It is an iterative algorithm, which starts with an arbitrary initial order of the requirement list. Each requirement is associated with a real number, which represents the assignment difficulty. Assignment is made according to the order, using the first channel that is compatible with previous assignments. If a requirement cannot be satisfied, the assignment difficulty of that requirement is increased by a random amount drawn uniformly from  $[0.5, 1.5]$ . After an iteration, the requirement list is rearranged in decreasing order of the assignment difficulty. Then, the procedure is repeated.

In order to make a fair comparison, Box's algorithm is slightly modified. We use the tightest lower bound as the input parameter  $N$ . Additional channels will be used if a requirement cannot be satisfied by the first  $N$  channels. The algorithm terminates when all the requirements can be satisfied by  $N$  channels or the number of iterations reaches a prescribed maximum value.

Another algorithm we considered is the one proposed by Sivarajan [16]. Actually, it is not a single algorithm, but a class of algorithms based on different ordering strategies. For a detailed description, see [16].

The results are shown in Tables II and III with different interference constraints. The best result obtained among the whole class of Sivarajan's algorithms are reproduced from [16]. The Box's algorithm is performed twice. One is limited to a maximum of 50 iterations, and the other is limited to 100.

It can be seen that most of the algorithms find the optimal solution in problems P3-8 and Q3-5. In these cases, the lower bounds are obtained by

$$\max_{x_i \in X} \{(m_i - 1)c_{ii}\} + 1$$

which implies that these problems are limited by the cosite constraint. This class of problems can be well solved by GSP2 since it always assigns the smallest possible color to the vertex

with maximum weight. Hence, optimal solutions are found in all these cases.

However, problems P1-2 and Q1-2 are relatively hard to solve. The best results are obtained either by GSP1 or GSP2. Although GSP1 cannot deal with the problem limited by cosite constraints adequately, it does have the best performance in P1 and P2. Problem P2 is just the same problem considered in the previous example. As stated in [7], Box's algorithm gives a solution of 445, which is the best result at that time. However, both GSP1 and GSP2 yield a better solution.

In general, GSP2 gives satisfactory results in all the cases. Its performance is better than Sivarajan's algorithms in the cases we tested. If compared to Box's heuristic, it requires more channels only for problem P1.

#### D. Example 4: Comparison with the Neural-Network Approach

Recently, the neural-network approach is used to solve CAP [4], [12]. In [4], eight problems were used for testing the proposed neural-network parallel algorithm. It was found that the optimal solutions are obtained in all those cases. We have tried to solve those problems using GSP2, and we find that it also yields the optimal solutions.

Taking a closer look at those problems, we find that four of them are identical to problems P4, P6, P8, and Q4, which we have already examined in the previous example. Another two have the same cellular network (as shown in Fig. 6), but with different constraints:  $N_c = 7$ , a.c.c. = 1 and  $c_{ii} = 5$  or 7. It is worth noting that all these six problems are limited by the cosite constraint, which is relatively easy to solve as we have pointed out already. In fact, seven out of the eight problems are cosite-constraint limited.

The remaining problem is taken from [12]. The data is obtained from a real-world network, which consists of 25 cells. Cosite constraint is not considered. In this case, the optimal solution is found by both the neural-network algorithm and GSP2.

Since most of the tested problems are cosite-constraint limited, the ability of the neural-network approach in dealing with problems like P1-2 and Q1-2 requires further investigation.

## VII. CONCLUSION

The CAP is a well-defined problem and has invoked a lot of interest in the past years. It is known to be NP complete, even if only the cochannel interference is considered. However, this result does not rule out the possibility of an optimal assignment algorithm, which works in polynomial time for some special network structures. In this paper, we propose the SP algorithm, which is optimal for three-stripe cellular systems under the cochannel constraint. However, this optimality is not preserved for other network structures with more general interference considerations.

The SP algorithm is generalized for problems with the adjacent channel and the cosite constraint. Numerical examples from [16] are used to test the algorithm. Comparisons with other existing algorithms are made, and the results are convincing. It is also found that problems, which are limited by the cosite constraint, are relatively easy to solve. Further research on classifying different traffic and network topology instances is interesting and may provide clues on designing algorithms for problems of different classes.

Additionally, we have derived a lower bound for the minimum number of channels required. It is tighter than that proposed by Gamst in some cases. However, in some examples, there is still a gap between the lower bound and the best solution known. Further improvements might be possible.

## APPENDIX

### OPTIMALITY OF SP FOR THREE-STRIPE SYSTEM

In this Appendix, we present the proof of Theorem 1.

Given any graph, we call any feasible coloring of vertices a *realization*. The realization, which requires the minimum number of colors, is called an *MP realization*. Given a realization, it is possible to obtain another realization by simply relabeling some or all of the colors. We call realizations that can be obtained from one another by relabeling colors *equivalent*. For an arbitrary graph, the MP realization is not necessarily unique. This is clear in view of the possibility of equivalent realizations. However, some graphs may allow multiple MP realizations that are not equivalent. For those realizations, which require the same number of colors, we define that they are *similar*. Therefore, if a realization is similar to an MP realization, it is also MP.

One way to construct examples of similar realizations that are not equivalent is to use the following color swapping operation for a three-stripe system with reuse factor of seven. First of all, we introduce the concept of the left- and right-hand side of a cell. In a three-stripe system, there is an obvious left- and right-hand side relation between any two cells with one exceptional case. For example, in Fig. 7,  $v_4, v_7, v_8, v_{11}$ , and  $v_{12}$  are on the right-hand side of  $v_3$ . The left-hand side is defined similarly. The only exception is the vertex  $v_{10}$ , which belongs neither to the left- nor the right-hand side of  $v_3$ . We call it the conjugate cell of  $v_3$ . Now, assume that  $\Psi$  is a

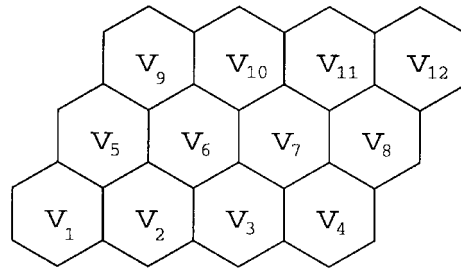


Fig. 7. An example to illustrate the concept of the left- and right-hand side of a cell.

realization in which  $v_j$  is colored by  $c_b$  and  $v_k$  is colored by  $c_a$  and  $v_j \in N(v_k)$ . Define the operation  $\text{Swap}_r(v_j, v_k, c_b, c_a)$  by swapping colors  $c_a$  and  $c_b$  for cells  $v_j$  and  $v_k$  and all the cells on the right-hand side of either of these two cells. Similarly, we can define  $\text{Swap}_l(v_j, v_k, c_b, c_a)$  by swapping  $c_a$  and  $c_b$  for cells  $v_j$  and  $v_k$  and all the cells on the left-hand side of either of these two cells. If we apply the operation  $\text{Swap}_r(v_j, v_k, c_b, c_a)$  to a realization and this swapping operation does not violate the channels assigned for cells on the left-hand side of either  $v_j$  or  $v_k$ , then the realization obtained after the swapping is similar to the original realization. If at least one of the colors has been used on cells on the left-hand side of either  $v_j$  or  $v_k$ , then the two realizations are not equivalent.

*Proof of Theorem 1:* A one- or two-stripe cellular system can be embedded into a three-stripe system. An assignment problem for a one- or two-stripe system can be viewed as a problem on a three-stripe system if cells outside of the original system are considered to have no channel demands. Therefore, it suffices to prove only the three-stripe case.

Let  $\Psi_{\text{SP}}$  be a realization obtained from the SP algorithm. If a color  $c_a$  is used in a realization  $\Psi$ , let  $K_a(\Psi)$  be the set of vertices colored by  $c_a$  in the  $\Psi$ . We claim that there exists an MP realization  $\Psi_{\text{MP}}$  such that

$$K_a(\Psi_{\text{SP}}) = K_a(\Psi_{\text{MP}}).$$

If this claim holds, we can use it to prove the theorem statement by using the following induction argument: if  $\Psi_{\text{SP}}$  has only one color, then it must be an MP realization. Suppose the theorem statement holds for all SP realizations using  $n$  colors. Now, consider a problem  $P = (X, M, C)$ . Suppose that  $\Psi_{\text{MP}}$  and  $\Psi_{\text{SP}}$  uses  $k + 1$  and  $n + 1$  colors ( $k \leq n$ ), respectively. Let  $c_a$  be the first color used in the SP algorithm. For each vertex in  $K_a(\Psi_{\text{SP}})$ , subtract one from the corresponding component of the original requirement  $M$  to obtain  $M'$ . By the definition of the SP algorithm, the realization it yields for  $P' = (X, M', C)$  is equivalent to  $\Psi_{\text{SP}}$  without  $c_a$  and, hence, requires  $n$  colors. If the claim holds, MP will use  $k$  colors for the reduced problem  $P'$ . By the induction assumption,  $n$  is the minimal number of color needed for  $P'$  and, hence,  $n$  must be equal to  $k$ . So,  $\Psi_{\text{MP}}$  uses  $n + 1$  colors, and this shows that  $\Psi_{\text{SP}}$  is also MP.

Before proving the claim, we note that for a three-stripe system and a cluster size of seven, cells (vertices) colored by the same color  $c_a$  can be labeled in a left-right order  $S_a$ , with no ambiguity (see Fig. 8). Let  $R_a(v)$  be the succeeding vertex of  $v$  in  $S_a$ , if it exists. Similarly, define  $R_a^{-1}(v)$  to be the



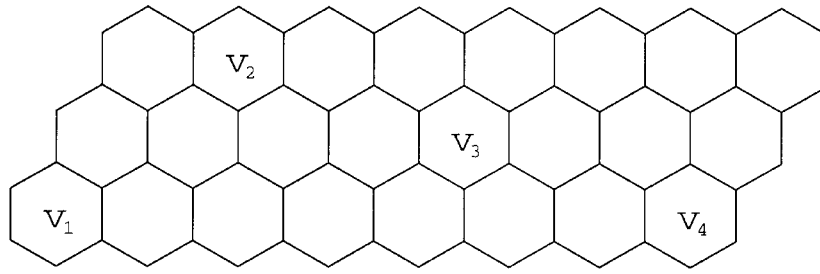


Fig. 8. If  $v_1, v_2, v_3$ , and  $v_4$  are colored by  $c_a$ , then  $S_a = (v_1, v_2, v_3, \text{ and } v_4)$ .

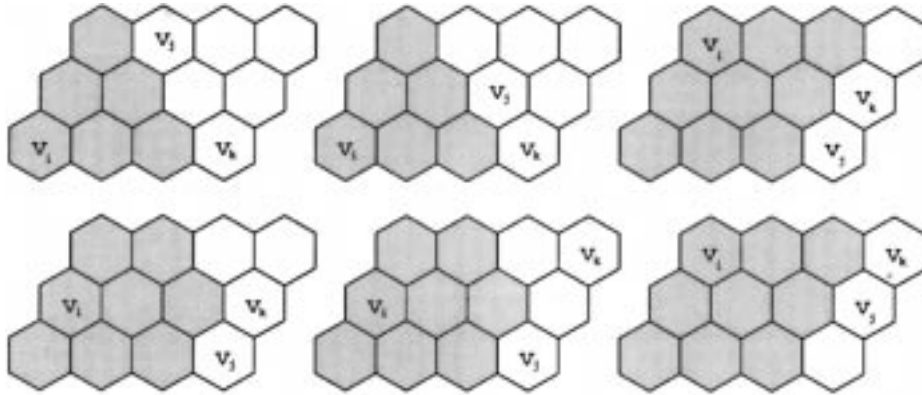


Fig. 9. The cases, where  $|N(v_k) \cap F| < |N(v_j) \cap F|$ . The denial area is shaded. Trivial cases, where  $v_k$  is further away from  $v_i$ , are not shown.

preceding vertex of  $v$  in  $S_a$ , if it exists. Notice that the order induced by  $S_a$  is not necessarily identical to the order how the SP algorithm assigns the vertex to color  $c_a$ . However, for a three-stripe system with cluster size of seven, the following property  $\mathcal{P}$ , follows from the definition of the SP procedure and the geometry of the three-stripe system.

Suppose that vertex  $i$  is assigned before vertex  $j$ . If  $R_a(v_i) = v_j$  and  $N(v_i) \cap N(v_j)$  is not empty, then  $v_j$  is the first node on the right-hand side of  $v_i$  that is assigned after  $v_i$ . Similarly, if  $R_a^{-1}(v_i) = v_j$  and  $N(v_i) \cap N(v_j)$  is not empty, then  $v_j$  is the first node on the left-hand side of  $v_i$  that is assigned after  $v_i$ .

Suppose that, according to SP, vertex  $v_0$  is the first vertex colored by  $c_a$ . Notice that there exists an MP realization  $\Psi_{MP}$  in which  $v_0$  is colored by  $c_a$  since  $v_0$  must be colored by at least one color and one can relabel one of the colors to  $c_a$ . If the vertices referenced by  $R_a^l(v_0)$  for both realizations are identical for all integer  $l$ , then the claim holds. Suppose, on the other hand, that  $l$  is the integer with the smallest absolute value, such that the vertices referenced by  $R_a^l(v_0)$  are not identical for the two realizations. First, assume that  $l$  is positive and  $R_a^l(v_0)$  for  $\Psi_{SP}$  and  $\Psi_{MP}$  refers to  $v_j$  and  $v_k$ , respectively, with  $j \neq k$ . We claim that one can construct another MP realization  $\Psi_{MP'}$  so that all the vertex assignment to the left of and up to  $R_a^{l-1}(v_0)$  are identical for  $\Psi_{MP}$  and  $\Psi_{MP'}$ , and  $R_a^l(v_0)$  for  $\Psi_{MP'}$  refers to  $v_j$ .

For notation simplicity, let us denote  $R_a^{l-1}(v_0)$  by  $v_i$ . This node is colored by  $c_a$  in both realizations. Notice that if  $R_a(v_i)$  is well defined for  $\Psi_{SP}$ , but not for  $\Psi_{MP}$ , then one can pick an arbitrary color used for  $v_j$  in  $\Psi_{MP}$  and replace it with

$c_a$ . This defines the new realization  $\Psi_{MP'}$  as claimed. On the other hand, if  $R_a(v_i)$  is well defined for  $\Psi_{MP}$ , it must be also well defined for  $\Psi_{SP}$  due to the nature of the SP algorithm, which stops the assignment of a color only when there is no candidate cell available. Hence, we may assume  $v_j$  and  $v_k$  are well defined.

If  $v_k$  is also colored by  $c_a$  in the SP realization, then  $v_k$  is not contained in  $N(v_j)$ , and it must be on the right-hand side of  $v_j$ . Hence, one can use  $c_a$  to color  $v_j$  in  $\Psi_{MP}$  without causing any violation with cells on the right-hand side or the conjugate cell of  $v_j$ . There is also no violation on the left-hand side of  $v_j$  because the first cell on the left-hand side of  $v_j$  colored by  $c_a$  in  $\Phi_{MP}$  is  $v_i$ . Hence, we can assume that  $v_k$  is colored by  $c_b$  in  $\Psi_{SP}$  with  $c_a \neq c_b$ . Without loss of generality, we may assume that  $v_j$  is also colored by  $c_b$  in  $\Psi_{MP}$ .

Let  $F$  denote the set of *denial area* just before the assignment to  $v_j$  is made in the SP algorithm. Notice that

$$|N(v_k) \cap F| \leq |N(v_j) \cap F|.$$

If not, then  $v_k$  will be picked by the SP algorithm to be colored by  $c_a$ . There are two possible cases for further consideration.

- Case 1)  $|N(v_k) \cap F| < |N(v_j) \cap F|$ : Due to the special topology of a three-stripe, the condition implies  $|N(v_k) \cap N(v_i)| < |N(v_j) \cap N(v_i)|$ . Recall that both  $v_j$  and  $v_k$  are on the right-hand side of  $v_i$ . It follows that  $(N(v_k) \cap F) \subset (N(v_j) \cap F)$  for all possible  $v_i$  (see Fig. 9). Hence, if an arbitrary color can be used to color  $v_j$  in a realization, it can also be used to color  $v_k$  without causing any violation

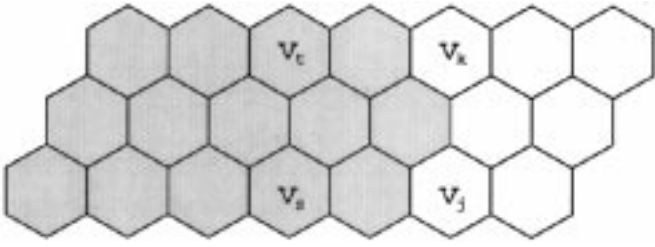


Fig. 10. Let the shaded cells be the denial area  $F$ . The only possible locations for  $v_j$  and  $v_k$  are shown.

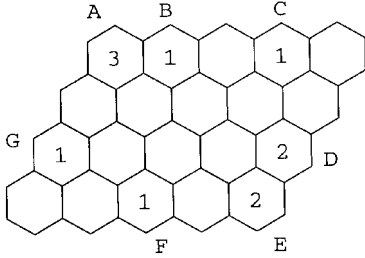


Fig. 11. SP realization may not be optimal in a four-stripe cellular system.

TABLE IV  
THE REALIZATION OF SP AND MP FOR  
THE EXAMPLE SHOWN IN FIG. 7

| Cell | $\Psi_{SP}$     | $\Psi_{MP}$     |
|------|-----------------|-----------------|
| A    | $c_1, c_2, c_4$ | $c_1, c_2, c_3$ |
| B    | $c_3$           | $c_4$           |
| C    | $c_1$           | $c_3$           |
| D    | $c_2, c_4$      | $c_1, c_2$      |
| E    | $c_3, c_5$      | $c_3, c_4$      |
| F    | $c_1$           | $c_1$           |
| G    | $c_3$           | $c_4$           |

for vertices on the left-hand side of  $v_k$ . Perform a swapping operation  $\text{Swap}_r(v_j, v_k, c_b, c_a)$  on  $\Psi_{MP}$ . This operation does not cause any violation. The resulting realization is  $\Psi_{MP}'$ , which satisfies the claimed property.

Case 2)  $|N(v_k) \cap F| = |N(v_j) \cap F|$ : Let  $G_v$  be the subgraph induced by the vertex set  $(N(v) \cap F) \cup \{v\}$ . In this case,  $W(v_k | G_{v_k}) \leq W(v_j | G_{v_j})$ , otherwise  $v_k$  will be colored by  $c_a$  instead of  $v_j$ . If  $N(v_k) \cap F = N(v_j) \cap F$ , we can perform the operation  $\text{Swap}_r(v_j, v_k, c_b, c_a)$  as in Case 2). So, it is only necessary to consider the complementary situation.

Now, suppose that  $N(v_k) \cap F \neq N(v_j) \cap F$ . The only possible locations for  $v_j$  and  $v_k$  are shown in Fig. 10. From the figure, it is clear that  $v_s$  and  $v_t$  are the only cells on the left-hand side of  $v_j$  and  $v_k$ , which may experience a violation if a swapping operation  $\text{Swap}_r$  is performed.

If  $w_j > w_t$ , then in  $\Psi_{MP}$ ,  $v_j$  is colored by more colors than  $v_t$ . Therefore, we can find a color, say  $c_d$ , that is used to color

$v_j$ , but not  $v_t$ . Then, we can perform  $\text{Swap}_r(v_j, v_k, c_d, c_a)$  as before to obtain  $\Psi_{MP}'$ .

On the other hand, if  $w_j \leq w_t$ , then  $w_s \geq w_k$  due to the fact  $W(v_k | G_{v_k}) \leq W(v_j | G_{v_j})$  and  $W(v_j | G_{v_j}) - W(v_k | G_{v_k}) = w_j + w_s - w_k - w_t$ . In  $\Psi_{MP}$ , one of the colors assigned to  $v_k$  is  $c_a$ , and  $c_a$  cannot be assigned to  $v_s$ . Therefore, we can always find a color, say  $c_d$ , in  $v_s$ , which is not assigned to  $v_k$ . Hence, the operations  $\text{Swap}_r(v_j, v_k, c_b, c_a)$  and  $\text{Swap}_l(v_t, v_s, c_b, c_d)$  can be applied to  $\Psi_{MP}$ . Notice that the operation  $\text{Swap}_l$  does not alter the vertices, which are colored by  $c_a$ , and so will not affect the previous coloring of  $c_a$ . The resulting realization is  $\Psi_{MP}'$ .

Hence, in both cases, we can find  $\Psi_{MP}'$  with the claimed property. Repeating this argument if necessary for the case, where  $l$  is negative, one can then guarantee that there is an MP realization, which has an identical sequence of vertices  $R_a^i(v_0)$  as  $\Psi_{SP}$  up to  $|i| \leq l$ . Hence, there exists an MP realization, which has the same set of vertices colored by  $c_a$  as  $\Psi_{SP}$ , and this proves the claim. As a result, the theorem is proved.  $\square$

This theorem cannot be generalized to an  $i$ -stripe system for  $i > 3$ . This can be seen from the example shown in Fig. 11 (originally due to Keeler [11]). As before, we assume a reuse factor of seven. Table IV shows that SP uses five colors, while MP uses only four. So, the SP realization is not optimal. It has been shown that MP is NP hard. Since SP is a polynomial time algorithm, this should not come as a surprise.

## REFERENCES

- [1] F. Box, "A heuristic technique for assigning frequencies to mobile radio nets," *IEEE Trans. Veh. Technol.*, vol. 27, pp. 57–64, May 1978.
- [2] D. E. Everitt and N. W. Macfadyen, "Analysis of multicellular mobile radiotelephone systems with loss," *Brit. Telecommun. Technol. J.*, vol. 1, no. 2, pp. 37–45, 1983.
- [3] M. Duque-Antón, D. Kunz, and B. Rüber, "Channel assignment for cellular radio using simulated annealing," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 14–21, Feb. 1993.
- [4] N. Funabiki and Y. Takefuji, "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 430–437, Nov. 1992.
- [5] A. Gamst and W. Rave, "On frequency assignment in mobile automatic telephone systems," in *Proc. GLOBECOM '82 IEEE*, pp. 309–315.
- [6] A. Gamst, "Homogeneous distribution of frequencies in a regular hexagonal cell system," *IEEE Trans. Veh. Technol.*, vol. 31, pp. 132–144, Aug. 1982.
- [7] —, "Some lower bounds for a class of frequency assignment problems," *IEEE Trans. Veh. Technol.*, vol. 35, pp. 8–14, Feb. 1986.
- [8] A. Gamst and K. Ralf, "Computational complexity of some interference graph calculations," *IEEE Trans. Veh. Technol.*, vol. 39, pp. 140–149, May 1990.
- [9] M. R. Garey and D. S. Johnson, "The complexity of near-optimal graph coloring," *J. ACM.*, vol. 23, pp. 43–49, Jan. 1976.
- [10] —, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.
- [11] K. Keeler, "On maximum-packing strategy for channel assignment in cellular systems," private communication.
- [12] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 188–193, Feb. 1991.
- [13] —, "Suboptimum solutions obtained by the Hopfield-Tank neural network algorithm," *Biol. Cybern.*, vol. 65, pp. 129–133, 1991.
- [14] V. H. MacDonald, "The cellular concept," *Bell Syst. Tech. J.*, vol. 58, pp. 15–41, Jan. 1979.
- [15] R. Mathar and J. Mattfeldt, "Channel assignment in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 647–656, Nov. 1993.
- [16] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchum, "Channel assignment in cellular radio," in *Proc. 39th IEEE Veh. Technol. Conf.*, 1989, pp. 846–850.
- [17] M. Zhang and T. S. Yum, "The nonuniform compact pattern allocation algorithm for cellular mobile systems," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 387–391, May 1991.



**Chi Wan Sung** (S'94) was born in Hong Kong in 1971. He received the B.Eng. and M.Phil. degrees in information engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 1993 and 1995, respectively. He is currently working toward the Ph.D. degree at the Chinese University of Hong Kong.

His research interests include channel assignment and handoff and power control in mobile communication systems.



**Wing Shing Wong** (M'81–SM'90) received the combined MABA degree in 1976 from Yale University, New Haven, CT, and the M.S. and Ph.D. degrees in 1978 and 1980, respectively, both from Harvard University, Cambridge, MA.

He joined AT&T Bell Laboratories in 1982 and was promoted to Supervisor in 1987. He joined the Information Engineering Department of the Chinese University of Hong Kong, Shatin, Hong Kong, in 1992, where he is now the Chairman and Professor of Information Engineering. His current research interests include mobile communication systems, analog computing, nonlinear filtering, performance analysis, and information issues in estimation and control. He is leading a development effort to prototype mobile integrated services on DECT.

Dr. Wong was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL for four years.