Minimization of Frequency Assignment Span in Cellular Networks

Angelos N. Rouskas, *Member, IEEE*, Michael G. Kazantzakis, and Miltiades E. Anagnostou, *Member, IEEE*

***Abstract*— We consider the problem of minimizing the span of frequencies required to satisfy a certain demand in a cellular network under certain interference constraints. A new iterative algorithm exploiting the special nature of such systems is presented. The general procedure has the ability to react to variations of the traffic demand as more and more channels are being assigned to cell requirements. Allocations of channels to cells are made with a method that borrows insight from the theory of convex maximization. This method is, however, equivalent to simple and fast heuristics when selecting proper values for its parameters. Our technique yields quite encouraging results, showing that it represents an efficient alternative to attacking this type of problems.**

***Index Terms*—Cellular networks, frequency assignment, minimum span.**

I. Introduction

TRAFFIC demand in cellular networks is growing very rapidly over the past years for both voice and data services, whereas the capacity of these systems is expanding at a much slower rate. The cellular concept [1] has significantly increased the efficiency of mobile networks. However, considering the tremendous fall of the computational cost, the use of sophisticated algorithms seems to be one of the major factors for improved exploitation of the available spectrum.

Frequency assignment has been investigated in many contexts and several efforts have been made toward its solution. A formal and thorough description of several instances of this problem, including the *minimum span channel assignment*, is conducted in [2], where its relation to the graph coloring problem is also examined. Many algorithms for this problem are based on procedures that have their grounds on graph theory [3]–[6]. According to the majority of schemes, the cell requirements are ranked on the basis of some measure that counts their difficulty of assignment. Allocations are afterwards performed using some heuristic rules. Nevertheless, there have been several other proposals based on different heuristic techniques [7]–[9], simulated annealing [10], [11], and neural networks [12], [13].

In [14], an extensive study is carried out, concerning the impact that cochannel, adjacent channel, and cosite constraints have on the value of the minimum required span of frequencies. Drawing upon the insight provided there, we introduce here a new iterative algorithm which is able to adapt to the existing requirements and "tune" the minimum allowed separation between channels of the same cell to a smaller value according to a simple rule. At each iteration, the number of channels allocated is always an integer multiple of the aforementioned minimum permitted spectrum width and two different policies are presented for determining this integer multiple. For the allocation of channels, we develop a general procedure based on convex maximization. Yet, this procedure is equivalent to simple heuristic strategies for certain choices for the parameters of the optimization function. In many cases, the results obtained by this algorithm on several benchmark problems are quite encouraging, suggesting that our algorithm is worthwhile to be considered.

The paper is organized as follows. The cellular network model is presented in Section II, and the minimum channel assignment problem is formulated in Section III. Section IV describes the steps of the iterative channel assignment algorithm, and in Section V the method of assigning channels during each iteration of the algorithm is developed. Section VI presents some numerical results, and the paper is concluded in Section VII.

II. Cellular Network Model

Bandwidth in a cellular communication network is divided into a number of disjoint carriers, each one being able to accommodate one connection or multiple connections when some multiple-access technique is employed. Considering a continuous spectrum allocation for the system demands, as well as a smooth spacing between the consecutive frequency bands, the available frequencies can be represented by the set of integers $\{1, \cdots, F\}$, $1(F)$ being the lowest (highest) available frequency of the system. The area covered by the network is separated into a number of cells, denoted by the set of integers $\{1, \cdots, M\}$. Every cell is being served by a different base station, and each mobile user within the coverage area of a cell can have access to every channel, provided that certain interference limitations are not violated. These constraints are developed for each pair of cells by measurements conducted throughout the geographical area of the cellular system.

A fairly general representation of these constraints can be made by means of an $M \times M$ *symmetric compatibility matrix* $\boldsymbol{C} = [c_{ij}]$, introduced in [4]. Each entry c_{ij} of this matrix is a nonnegative integer number denoting the minimum permissible value of separation between a frequency f_i assigned to cell i and a frequency f_j assigned to cell j, that

Manuscript received June 19, 1996; revised June 1, 1998.

The authors are with the Department of Electrical and Computer Engineering, National Technical University of Athens, Athens 15780, Greece.

Publisher Item Identifier S 0018-9545(99)01052-X.

is,

$$\text{if } a_{if_i} = a_{jf_j} = 1 \text{ then } |f_i - f_j| \geq c_{ij} \qquad i, j = 1, \cdots, M, \quad f_i, f_j = 1, \cdots, F \tag{1}$$

where $a_{if}, i = 1, \cdots, M, f = 1, \cdots, F$ are variables which can only take the values zero or one, defined as

$$a_{if} = \begin{cases} 1, & \text{if channel } f \text{ is assigned to cell } i \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Based on the assumptions adopted by other authors as well, e.g., [8] and [14], about the range of values of c_{ij}'s encountered in most real problems, we also suppose that:

1) c_{ij} can take *small* nonnegative values, $i, j = 1, \cdots, M$;
2) $c_{ij} \geq c_{ik}$ if the distance between cells i and j is smaller than that between cells i and $k, i, j, k = 1, \cdots, M$;
3) $c_{ii} = \alpha$, some small positive integer, $i = 1, \cdots, M$.

Assumption 3) suggests that the cosite constraints are fixed at the same value for every cell of the system, while assumption 2) reflects the natural meaning of the compatibility matrix. Namely, remote cells, in contrast to neighboring ones, suffer less interference between each other. Obviously, these assumptions are fairly reasonable and do not limit the real network structures. However, in a later section we will see how the general case $c_{ii} = \alpha_i$ can be treated. Finally, we let $R = [r_i]$ be an M-dimensional positive integer vector, each component of which stands for the number of requests for channels placed on the system by cell i.

III. Formulation of the Channel Allocation Problem

We can now formulate the channel assignment problem as a minimum span channel assignment problem (MSCAP), as follows.

Problem MSCAP

$$\min\{\max\{f | a_{if} = 1, i = 1, \cdots, M, f = 1, \cdots, F\}\}$$

subject to

$$\sum_{f=1}^{F} a_{if} = r_i, \qquad i = 1, \cdots, M \tag{3}$$

$$\sum_{f=f'}^{f'+c_{ij}-1} (a_{if} + a_{jf}) \leq 1, \qquad i, j = 1, \cdots, M \qquad f' = 1, \cdots, F - c_{ij} + 1 \tag{4}$$

$$a_{if} = 0 \text{ or } 1 \text{ only } i = 1, \cdots, M, f = 1, \cdots, F. \tag{5}$$

Formulating the channel assignment problem in this way, we have assumed that the available frequency span of the system is large enough to meet the demands of any of the problems we have to deal with. Clearly, constraints (3) form the channel requirements constraints, while it is easy to show that the *serial* compatibility constraints (4), together with the *integral* constraints (5), enforce the requirement (1) imposed by the compatibility matrix entries.

Problem MSCAP is a hard allocation problem which is shown to be $\mathcal{NP}$ hard, by reduction to the graph coloring problem [2], [4], [5], and is mentioned as the *generalized graph coloring problem*. Our intention is to decompose it into a series of smaller and manageable subproblems using an iterative procedure. Even though the iterative procedure cannot always guarantee optimum solutions, results show that it represents an improvement over previously published approaches.

IV. Iterative Channel Assignment Algorithm for MSCAP

A set of lower bounds for the minimum required span of frequencies are derived for certain classes of assignment problems in a thorough study of [14]. At some stage of the proof of Lemma 10 in [14], an increase on the value of separation between the channels allocated to cells with the greatest number of requirements is taking place so that space is created to most economically accommodate requirements of other cells. Incited by this observation, we allow the minimum distance between consecutive channels of the same cell to vary between different spectrum points.

A. Description of the General Procedure

We define the allocation width w_l as the spectrum width between consecutive channels of the same cell during iteration l. This value is considered as fixed for each iteration of the algorithm and is greater than or equal to α so that the cosite constraints are always met. Starting from the lowest available frequency, the allocation width takes its maximum initial value w_1. After successive iterations with this value, and as subsequent channels are being assigned to cell requirements, which are consequently reduced, this value can be lowered by one, depending on whether some *criterion for reduction* is satisfied. This procedure can be repeated until the allocation width reaches its minimum value α, and from that point on all the following iterations must use this value, without considering the criterion for reduction.

At the beginning of iteration l, the set of channels $\{1, \cdots, F_{l-1}\}$ have already been assigned and cannot be reused, the vector of the existing requirements has become $R_l = [r_i^l] \leq R$, and the set of cells with nonzero requirements is $V_l = \{i | i \in \{1, \cdots, M\} \text{ and } r_i^l > 0\}$. Iteration l is then performed on the subsequent set of channels $W_l = \{F_{l-1} + 1, \cdots, F_l\}$, where $F_l = F_{l-1} + k_l \times w_l$, namely, the number of channels assigned by the end of this iteration will have been increased by some integer multiple k_l of the allocation width. We will see shortly how we determine the value of k_l.

We define the *dominant* cell $c_d^l \in V_l$ as the cell whose number of requirements is greatest and the first available free channel that can be assigned to it, greater than F_{l-1}, is smallest. We will refer to this channel as the *dominant* channel f_d^l. If more than one cell are candidates for the dominant cell we can choose randomly one of them, provided that $c_d^{(l-1)}$ is not also a candidate in this iteration. In such a case, we choose $c_d^l = c_d^{(l-1)}$.

1. Set $l = 1, F_{l-1} = 0, V_l = \{1, \ldots, M\}, R_l = R$. Set also w_l to a value $\geq \alpha$.
2. Solve problem APP to allocate the channels of W_l^i to cells of V_l and obtain p_l^i.
3. If $w_l > \alpha$ and the criterion for reduction of w_l is satisfied, set $w_l = w_l - 1$ and goto step 2.
4. Determine the number of repetitions k_l of p_l^i and perform the same mapping to the channels $W_l - W_l^i$, shifting all allocations in p_l^i, by w_l, $k_l - 1$ times.
5. Update R_{l+1} : if $(f_j, j) \in p_l^i$, set $r_j^{(l+1)} = r_j^l - k_l$, else $r_j^{(l+1)} = r_j^l$. Update V_{l+1}. If $V_{l+1} = \emptyset$ stop.
6. Set $l = l + 1$, $F_{l-1} = F_{l-2} + k_{l-1} \times w_{l-1}$, $w_l = w_{l-1}$.

Fig. 1. General procedure for MSCAP-GPMSCAP.

Let $W_l^i = \{F_{l-1}+1, \cdots, F_{l-1}+w_l\}$ be the set of the lower w_l channels not assigned to cells yet. We define the *initial allocation pattern* as the one-to-many mapping p_l^i: $W_l^i \rightarrow V_l$, so that some *benefit function* $g(\cdot)$ is optimized. In other words, each cell in V_l is restricted to require only one channel from W_l^i. Note that at this point of spectrum the minimum frequency separation in each cell is w_l. Previously existing allocations affect the construction of p_l^i through the following *boundary* compatibility constraints:

$$\begin{aligned} &\text{if } a_{jf'} = 1, F_{l-1} - c_{ij} + 2 \leq f' \leq F_{l-1} \text{ then } a_{if} = 0 \\ &\qquad i \in V_l, f = F_{l-1} + 1, \cdots, f' + c_{ij} - 1 \end{aligned} \tag{6}$$

while enforcing requirement (1) in p_l^i, requires the use of serial constraints similar to (4)

$$\begin{aligned} &\sum_{f=F_{l-1}+f'}^{F_{l-1}+f'+c_{ij}-1} (a_{if} + a_{jf}) \leq 1 \\ &i \neq j, \quad i, j \in V_l, \quad c_{ij} \geq 1 \\ &f' = 1, \cdots, w_l - c_{ij} + 1. \end{aligned} \tag{7}$$

In addition to the above serial constraints, we introduce the following *circular* compatibility constraints to also provide for a repetitive application of a similar pattern to consecutive channels:

$$\begin{aligned} &\sum_{f=F_{l-1}+f'}^{F_{l-1}+w_l} (a_{if} + a_{jf}) + \sum_{f=F_{l-1}+1}^{F_{l-1}+f'+c_{ij}-1-w_l} (a_{if} + a_{jf}) \leq 1, \\ &\qquad i \neq j, i, j \in V_l, c_{ij} \geq 2, f' = w_l - c_{ij} + 2, \cdots, w_l. \end{aligned} \tag{8}$$

Note that assumptions 2) and 3) of Section II imply that $c_{ij} \leq \alpha \leq w_l, i \neq j$.

We can now formulate the problem of finding an optimal initial allocation pattern as the allocation pattern problem (APP) as follows.

Problem APP: Find an initial allocation pattern p_l^i such that

$$\begin{aligned} &g(a_{1,F_{l-1}+1}, \cdots, a_{1,F_{l-1}+w_l}, \cdots \\ &a_{M,F_{l-1}+1}, \cdots, a_{M,F_{l-1}+w_l}) \text{ is optimum} \end{aligned} \tag{9}$$

subject to (6)–(8) and

$$a_{c_d^l, f_d^l} = 1 \tag{10}$$

$$a_{if} = 0, \qquad i \in \{1, \cdots, M\} - V_l, f \in W_l^i \tag{11}$$

$$\sum_{f=F_{l-1}+1}^{F_{l-1}+w_l} a_{if} \leq 1, \qquad i \in V_l \tag{12}$$

$$a_{if} = 0 \text{ or } 1 \text{ only}, \ i \in V_l, \quad f \in W_l^i. \tag{13}$$

The reader may check that constraints (12) enforce the mapping to be one-to-many and are more strict than the cosite ones. Moreover, since cell c_d^l greatly affects the minimum number of required frequencies for the total remaining requirements [14], we prefer to fix the allocation to this cell at channel f_d^l in p_l^i using constraint (10). However, the most important point in the previous formulation is the use of (8). Let $V_l^i = \{j | (f_j, j) \in p_l^i\}$ be the set of cells that have been granted a channel in p_l^i. These constraints allow us to also allocate channels $f_j + w_l, f_j + 2 \times w_l, \cdots, f_j + (k_l - 1) \times w_l$ to every cell $j \in V_l^i$.

The general procedure for MSCAP (hereafter referred to as GPMSCAP) is shown in Fig. 1. We see that the computational burden of GPMSCAP is determined by the solution to APP (step 2) in combination with the value of k_l (step 4). The worst, but most unlikely, case appears when $k_l = 1$ for every iteration l. In such a case, a new APP is constructed and solved to allocate only w_l channels at each iteration l.

B. Some Important Special Cases

There may exist some cases where the repetitive application of p_l^i does not fully exploit spectrum $W_l - W_l^i$. On one hand, the initial boundary constraints may be more strict than the circular ones because cells whose allocations affect allocations in p_l^i, may not appear in V_l^i. On the other hand, at the k_lth repetition of this pattern there is no need for the whole set of circular constraints to hold, since a new initial pattern will be constructed from scratch in iteration $l + 1$.

For these reasons we can divide $W_l - W_l^i$ into $k_l - 1$ consecutive groups of w_l channels and define and construct the *intermediate* and *final allocation patterns* p_l^m and p_l^f. These mappings are essentially similar to p_l^i with different definitions sets and slightly different constraints. Pattern p_l^m is defined in $W_l^m = \{F_{l-1} + w_l + 1, \cdots, F_{l-1} + 2 \times w_l\}$ and after its construction can be repeated $k_l - 3$ times, while p_l^f is defined in $W_l^f = \{F_{l-1} + (k_l - 1) \times w_l + 1, \cdots, F_{l-1} + k_l \times w_l\}$. All allocations in p_l^i are included into the set of constraints of p_l^m and p_l^f with the form of (10). That is, $\forall j$ such that

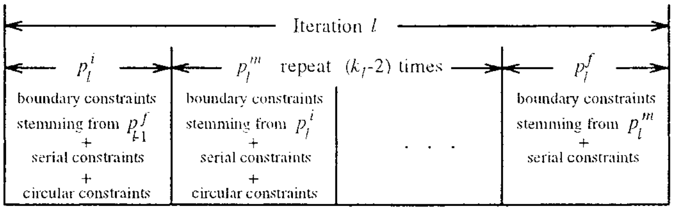


Fig. 2. Application of the allocation patterns during iteration l.

$(f_j, j) \in p_l^i$, we add equation $a_{j,f_j+w_l} = 1$ in constraints of p_l^m. Thus, the majority of allocations in these mappings are already constructed and new allocations may happen only on the boundary channels of their definitions sets. These new assignments are determined and limited by the values of c_{ij}'s, $i \neq j$. Obviously, if $\max\{c_{ij} | i, j \in V_l, i \neq j\}$ is equal to one, mappings p_l^i, p_l^m, and p_l^f coincide in any case. If, however, this value is equal to two, there may exist new allocations of only the first and last channels of the definition sets of p_l^m and p_l^f.

Depending on the value of k_l there may not exist an intermediate pattern $(k_l = 2)$ or all patterns may collapse to only one pattern $(k_l = 1)$, the general case being the one depicted in Fig. 2. In this figure, we see that allocations in p_l^f do not take into account circular constraints but *serial* constraints similar to (7) only. However, there is an important exception which involves a special treatment of the dominant cell which is not illustrated. If after iteration l the dominant cell c_d^l still retains the greatest number of unserviced requirements, we choose to apply all those circular constraints in the construction of p_l^f so that the first allocation of c_d^l in the next iteration will not be affected. In this way, we provide for $c_d^{(l+1)} = c_d^l$ and $f_d^{(l+1)} = f_d^l + k_l \times w_l$.

C. Number of Repetitions of the Allocation Pattern

Let us turn now our attention to the problem of determining the value of k_l, namely, the number of repetitions of the allocation pattern in each iteration. This value is very important for the overall performance of GPMSCAP, as already stated in Section IV-A. We can identify two major classes of problems. The distinction between these two classes, can be made on the basis of the ability of the allocation patterns to assign one channel to every heavily loaded cell. The first one encompasses those systems in which all hot spots of the network can be serviced by the allocation pattern. In the latter class of systems, we encounter a larger number of such cells that cannot be simultaneously serviced by the allocation pattern. We propose two different heuristic rules for k_l.

1) Simple/Fast—S/F. Let $j \in V_l^i$ such that r_j^l is minimum. Repeat the allocation pattern $k_l = r_j^l$ times.
2) Complex/Slow—C/S. Let $j_1 \in V_l - V_l^i$ such that $r_{j_1}^l$ is maximum. Let also $j_2, j_3 \in V_l^i$ such that $r_{j_2}^l$ is minimum and $r_{j_3}^l$, $r_{j_3}^l \geq r_{j_1}^l$, is smallest. Repeat the allocation pattern $k_l = \min(r_{j_2}^l, r_{j_3}^l - r_{j_1}^l + 1)$ times.

In both cases, if $k_l > 2$ and some new cells appear with an allocation in p_l^m, k_l should be updated according to these rules once more.

The first heuristic repeats the allocation pattern until some cell that has been granted a channel in p_l^i has all its requirements satisfied. It does not take into account the requirements of the cells that have no assignment in the allocation pattern and thus is more appropriate for the first class of problems. Application of this rule is more probable to yield large numbers for k_l at least during the first iterations of GPMSCAP, and as a result small number of iterations in total. On the other hand, the effect of the second rule is expected to slow down the overall procedure because it causes less repetitions inside each iteration. To justify this argument, note that C/S takes also into account the difference of the requirements between two heavily loaded cells. One of them has been granted a channel in p_l^i (j_3), while the other one has no allocation in p_l^i (j_1). Consequently, the values of k_l are expected to be smaller than those of S/F. However, C/S results in a frequent multiplexing of allocations to the requirements of the hot spots of the network and as a result fair treatment of all these cells, and better final span of frequencies.

D. Criterion for Reduction of the Allocation Width

We have identified one simple criterion for the reduction of the allocation width. This criterion reflects the ability of our procedure to identify that significant changes have occurred to the current requirements and there is a need to adapt to these new conditions. In other words, channels will be wasted unless the allocation width is reduced. It is always checked after the construction of the initial allocation pattern, and if satisfied, our procedure ignores this initial pattern, reduces by one the allocation width, and backtracks at the end of the previous iteration. The criterion examines the level of utilization of every channel in the allocation pattern and reacts in case of zero utilization.

- Zero channel utilization criterion (ZCUC): If at the initial allocation pattern p_l^i there exists some channel that is not assigned to any cell from V_l, then backtrack and set $w_l = w_{l-1} - 1$, unless $w_l = \alpha$.

Let us now illustrate the expected behavior of our algorithm with the following example. So far, we have not discussed the problem of allocating channels in each pattern. However, our intention here is to illustrate the intuition behind our concept of varying the value of the allocation width. We consider the cellular network depicted in Fig. 3, where its compatibility matrix and requirement vector are shown. Each cell is denoted by a different symbol, shown in Fig. 3(c). Using this symbolism, the appearance of a number inside a "triangle" implies that the frequency corresponding to this number is assigned to cell 2. Fig. 4 shows the output of our algorithm for different starting values of the allocation width, considering that the number of repetitions of an allocation pattern inside an iteration is determined by the S/F rule. In this example, no reuse is possible since there is no zero entry in the compatibility matrix, and the maximum lower bound obtained from [14] is 14.

If we set $w_1 = 3$, a possible allocation using 16 channels in total is shown in Fig. 4(a). Using this value for w_1 and due to the special treatment of the dominant cell 1, this cell

$$\begin{bmatrix} 3 & 1 & 2 & 1 \\ 1 & 3 & 1 & 1 \\ 2 & 1 & 3 & 1 \\ 1 & 1 & 1 & 3 \end{bmatrix} \qquad \begin{bmatrix} 5 \\ 4 \\ 2 \\ 3 \end{bmatrix}$$

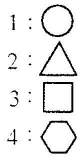


Fig. 3. Example cellular network.

will acquire the first frequency in p_1^i. Furthermore, owing to the serial and circular constraints, none of the other two frequencies of p_1^i can be assigned to cell 3 since $c_{13} = c_{31} = 2$. The S/F rule yields $k_1 = 3$, which is equal to the number of requirements of cell 4, and pattern p_1^i is repeated two more times during iteration 1, without any changes. Obviously, the main problem with the allocation of Fig. 4(a) is that it is impossible to multiplex allocations to cells 1 and 3 at early stages of the algorithm, that is, during iteration 1.

However, this can be achieved by increasing w_1 to four. Such an allocation is shown in Fig. 4(b). In this figure, all four cells have one request satisfied in p_1^i and this pattern can be shifted upwards in spectrum by $w_1 = 4$ channels once more, due to the circular constraints. At that point cell 3 has all its requirements satisfied (S/F results in $k_1 = 2$). Note that there is no p_1^m and that p_1^i and p_1^f coincide in this figure. During iteration 2, criterion ZCUC is satisfied in p_2^i, w_2 is reduced, and backtracking is performed (step 3 of GPMSCAP). The resulting allocation uses only 15 channels, and, as can be verified by using exhaustive search techniques, it is an optimum solution for this small problem. If we had not reduced w_l, channel 12, as well as more channels in subsequent iterations, would be wasted. From this example, we see that our algorithm is expected to behave similarly to the statement in Lemma 10 of [14], namely, that the distance of channels allocated to cells with the greatest demand can be increased by one (first three allocations to cell 1) to accommodate requests from other cells (cell 3) in the most efficient manner.

V. Solution to the Allocation Pattern Problem

Our objective is to allocate the frequencies of W_l^i to the cells of V_l, so that the number of potentially serviced requirements will be maximum, by defining an appropriate benefit function g. However, even if such an optimal assignment is obtained during all iterations of GPMSCAP, there is no guarantee that the overall algorithm will yield an optimal span of frequencies. The location of cells is well represented by the number of nonzero entries c_{ij}'s of the corresponding rows i of the compatibility matrix, since these entries have a natural geometric reflection, as we have already stated in Section II. It is well known that cells lying deep inside the geographical area of the network experience great difficulty in assigning channels to them than those cells lying in the periphery of the network [15]. Thus, if we focus only on the aforementioned objective, it is quite certain that channels will start to be assigned to the former cells when the majority of their neighbors will have their requirements satisfied. It is also known that vicinities of cells or single cells with great requirements influence mostly the necessary span of frequencies [14]. As a consequence, it is necessary to favor allocations to cells with great requirements, irrespectively of their location.

A. Relaxed Allocation Pattern Problem

Let us introduce the following benefit function g to be maximized for APP:

$$g(a_l) = a_l^T \cdot a_l + \Delta_d D_l^T \cdot a_l + \Delta_c C_l^T \cdot a_l \tag{14}$$

where $(^T)$ denotes the transpose of a matrix

$$\begin{aligned} a_l^T = &[a_{1,F_{l-1}+1}, \cdots, a_{1,F_{l-1}+w_l}, \cdots \\ &\cdot a_{M,F_{l-1}+1}, \cdots a_{M,F_{l-1}+w_l}] \end{aligned}$$

is the $M \times w_l$ unknown vector

$$\begin{aligned} D_l^T = &[r_1^l/r_{\max}^l, \cdots, r_1^l/r_{\max}^l, \cdots \\ &\cdot r_M^l/r_{\max}^l, \cdots, r_M^l/r_{\max}^l] \end{aligned}$$

is the $M \times w_l$ normalized demand vector with $r_{\max}^l = \max\{r_i^l | i \in V_l\}$

$$\begin{aligned} C_l^T = &[1, (w_l - 1)/w_l, \cdots, 1/w_l, \cdots \\ &\cdot 1, (w_l - 1)/w_l, \cdots, 1/w_l] \end{aligned}$$

is the $M \times w_l$ normalized channel vector, and Δ_d and Δ_c are factors of the relative importance of the last two terms, called *demand* and *channel* factors, respectively.

The first term of g is quadratic, whereas the second and third ones are linear. The second term favors allocations to cells with great demand, whereas the third one favors allocations of lowest channels, namely, channels closer to F_{l-1} than to $F_{l-1} + w_l$. By choosing values sufficiently smaller than unity for the factors Δ_d and Δ_c, we can force the first term to be the dominant one in g.

If we had not included the quadratic term in g, APP would be a 0–1 integer linear programming problem maximizing g. The general form of these problems is $\mathcal{NP}$ hard [16], and instead of trying to solve this problem we relax the integral constraints (13) using the following ones instead:

$$0 \leq a_{if} \leq 1, i \in V_l, f \in W_l^i \tag{15}$$

and introduce to the constraints of APP the next equality

$$\sum_{i=1}^{M} \sum_{f=F_{l-1}+1}^{F_{l-1}+w_l} a_{if} = b, b > 1, \text{ some small integer.} \tag{16}$$

In this way, we can formulate the relaxed allocation pattern problem (RAPP) as follows.

Problem RAPP: $\max g(a_l)$ subject to (6)–(8), (10)–(12), (15), and (16).

Problem RAPP aims at the maximization of a convex nonlinear function with linear inequality constraints and falls in the category of convex maximization problems under a set of constraints that form a closed convex set $\mathcal{D}$. The

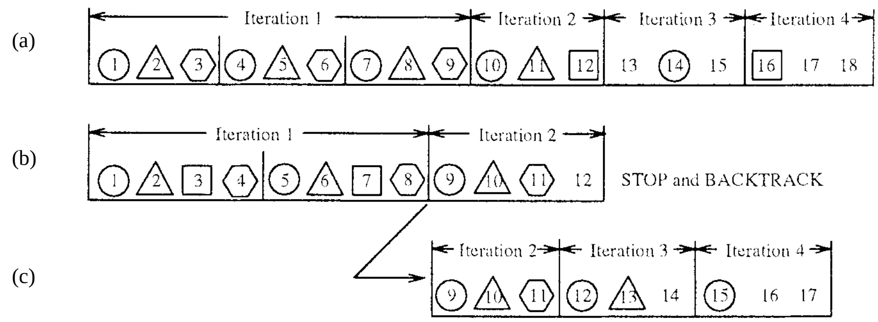


Fig. 4. Algorithm's expected behavior for the example cellular network.

intractability of this type of problems lies on the fact that each local maximum solution is located at one of the nonpolynomial number of extreme points of $\mathcal{D}$ [17]. Even if such a local maximum solution is reached, there is no guarantee that this will also be the global maximum one. Namely, there is no standard procedure to get to the global maximum solution. Moreover, noninteger solutions are of no value to our problem even if they are optimal.

However, RAPP, as formulated, is characterized by the dominating nonlinear term of g and the introduction of equality (16). This term, together with (16) and (15), has the property of favoring integer 0–1 solutions than noninteger ones. To clarify this argument, we rewrite the first dominant term g_d as

$$g_d(a_l) = a_l^T \cdot a_l = \sum_{i=1}^{M} \sum_{f=F_{l-1}+1}^{F_{l-1}+w_l} a_{if}^2. \tag{17}$$

Noting that equality (16) holds at every feasible solution of the convex set $\mathcal{D}$, it is easy to show that if there exists a 0–1 integer solution that satisfies (16), this solution is achieved at the point where g_d takes its greatest possible value b. In other words, maximizing g_d will possibly yield extreme points, with b components equal to one, the rest being zero, under the condition that such points exist. See [15] for a similar argument, where the linear terms are not included in g.

The search for an extreme point of $\mathcal{D}$, which possibly maximizes g, is performed using the gradient projection method [17] slightly modified. Starting from an arbitrary interior point of $\mathcal{D}$, which we will show shortly how can be obtained, the number of independent active constraints is always less than $M \times w_l$. By projecting ∇g onto the intersection of active constraints we obtain a search direction s_g. If $s_g = 0$, then any possible direction on the intersection of active constraints will certainly yield an increase of the objective function g. The one-dimensional (1-D) search from this point toward this direction terminates at another interior point where at least one more constraint becomes active. If this procedure is repeated until the number of independent active constraints is equal to $M \times w_l$, the resulting point is an extreme point of $\mathcal{D}$.

B. Stepwise Procedure for APP

The solution to APP can now be obtained by solving a series of RAPP's using an increasing sequence for the value of b in (16). Namely, at each step of the procedure we attempt to increase the allocations in p_l^i by solving a new RAPP. The additional constraints of the form $a_{if} \leq 1$ that become active at the end of each step, remain active at the next RAPP and as a result the convex set $\mathcal{D}$ is continuously shrinking. This procedure can be repeated until $\mathcal{D}$ becomes a single point and no more allocations are possible.

An initial interior point a_l of $\mathcal{D}$ can be easily obtained in the following way. Suppose that the number of existing allocations in the pattern p_l^i is n_1. Equivalently, n_1 constraints of the form $a_{if} \leq 1$ are active, and the corresponding components of a_l should be set to one. These constraints together with constraints (6)–(8), (11), and (12) enforce n_0 constraints of the form $a_{if} \geq 0$ to be active and the corresponding components of a_l to be set to zero. If $n_1 + n_0$ equals $M \times w_l$, then the whole convex set $\mathcal{D}$ has reduced to a single point. In an opposite situation, let $n_\delta = (M \times w_l) - (n_1 + n_0)$ and fix all n_δ components of a_l to the value $\delta = (b - n_1)/n_\delta$. If some constraint of the forms (7), (8), and (12) is violated, decrease b by one and recompute δ. Repeat this procedure until no constraint from (7), (8), and (12) is violated. The resulting vector is an interior feasible point of $\mathcal{D}$ and the active set at this point is composed by the n_1 constraints of the form $a_{if} \leq 1$, the n_0 constraints of the form $a_{if} \geq 0$ and the equality (16).

The stepwise procedure for the solution of APP (hereafter referred to as SWAPP) is depicted in Fig. 5. At each iteration, SWAPP attempts to allocate Δb more channels to the existing requirements until no more allocations are possible. Consequently, since there will be at most $M \times \Delta b$ searches for extreme points, the worst case complexity of SWAPP is $O(M^5 \times w_l^4 \times \Delta b)$. Two special cases of this procedure, similar to heuristics developed in [4] and [5], can be identified for certain values of Δ_d, Δ_c, and Δb. By letting Δ_d sufficiently greater than Δ_c and $\Delta b = 1$, we get the frequency exhaustive strategy (FES). Alternatively, by letting Δ_d sufficiently smaller than Δ_c and $\Delta b = 1$, we get the requirement exhaustive strategy (RES). These strategies are very similar to those introduced in [4] and [5]. The difference lies on the number of available channels which is restricted to w_l and the ordering of the calls. The latter is obtained by listing the calls according to the number of requirements of their cells. Since at each allocation pattern only one call from each cell is permitted to request a channel, this list contains only one call from each cell. For an equal number of requirements a random choice can take place. These two special strategies for APP have

1. Set $n_1 = 1$ (because of (10)) and compute n_0 from the constraints of $RAPP$.
   Set $b = 1 + \Delta b$, Δb a small positive integer.
2. If $n_1 + n_0$ equals $M \times w_l$ goto 4.
   Make n_1 and n_0 constraints of the forms $a_{if} \leq 1$ and $a_{if} \geq 0$ active and formulate the new $RAPP$ updating (16). Compute an initial interior point a_I of $\mathcal{D}$ as already explained.
3. Search for an extreme point of $\mathcal{D}$ using the gradient projection method on g.
   If the resulting extreme point is not an integer solution set $b = b - 1$ and goto step 2.
   If the extreme point is an integer solution update n_1, n_0, set $b = b + \Delta b$ and goto step 2.
4. Stop. No more allocations are possible.

Fig. 5. Stepwise procedure for APP-SWAPP.

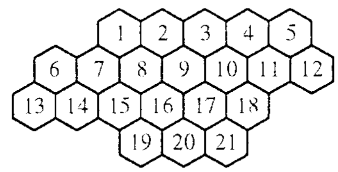
worst case complexity $O(M^2 \times w_l)$, which renders them very attractive for problems that require an upper bound on their response time, namely, in cellular environments with highly varying traffic demand.

In conclusion, the value of Δb plays a significant role on SWAPP. First, it actually determines the number of extreme points that will be computed and as a consequence the total computation burden during each iteration of the algorithm. If it is very large, the first condition of step 3 in SWAPP will be frequently invoked and as a result the number of extreme points examined will be high. Second, in contrast to FES and RES, it allows SWAPP to take decisions about more than one allocations at a time when $\Delta b > 1$. In this way, its decisions are more global and we expect in some cases to be better than those of FES and RES.

VI. Numerical Results and Discussion

We first consider the series of problems in [5] which are based on the 21-cell system introduced in [18]. For convenience, we have reproduced this system as well as the two requirement vectors R_1 and R_2 of [5] in Fig. 6. Note that R_1 has three cells heavily loaded with traffic and R_2 has a great number of cells with similarly heavy traffic demand. The second series of problems are based on a 7 $\times$ 7 array of cells with two different requirement vectors R_3 and R_4 shown in Fig. 7. We note that R_3 has nine central cells with high requirements, while in R_4 the heaviest load is experienced by the 13 cells lying on the two middle (intersected) lines of cells. The interference constraints consist of several combinations of cochannel, adjacent channel and cosite interference. The cluster size N_c determines the set of cells that suffer cochannel interference, while adjacent channel interference is assumed between adjacent cells whenever the adjacent channel constraint (ACC) is set to two.

We first investigate the effectiveness of our algorithm when the simple and fast RES and FES strategies are used for the solution to APP. Our results for the 21-cell system are shown in Table I, for both repetition rules (RR's) S/F and C/S, and different values of w_1. The upper and lower parts of this table, that is, problems 1–8, and 9–13, use as input the requirement vectors R_1 and R_2, respectively. For reasons of comparison with the fast heuristics of [5], we have also tabulated the best and worse spans obtained there. The fifth LB column contains tight lower bounds, namely, optimum spans. Concerning problems 1, 2, 9, and 10, which proved to be more difficult than the rest, the indicated optimum spans have been recently quoted in the literature [19], [20].



R_1^T=[8 25 8 8 8 15 18 52 77 28 13 15 31 15 36 57 28 8 10 13 8]

R_2^T=[5 5 5 8 12 25 30 25 30 40 40 45 20 30 25 15 15 30 20 20 25]

Fig. 6. The 21-cell system and requirements vectors.

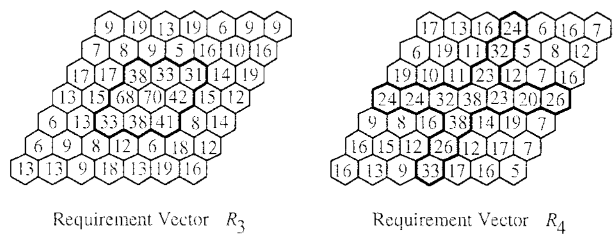


Fig. 7. The 49-cell system and requirements vectors.

Excluding these problems, the optimum solution is obtained in the other cases, using the simple repetition rule and initial allocation width equal to the cosite constraint value α. This is why we have not filled all the other entries of the table. We first observe that in the majority of problems there exists at least one procedure of ours, yielding better results than the best span of [5]. These results are highlighted in the table and the only exception occurs in problem 9 where our best result is only 287, compared to the 283 channels of [5].

The same number and type of problems have been constructed for the 49-cell array of Fig. 7. When RES and FES are used, the spans are also similarly presented in Table II, where the LB column now contains lower bounds obtained with the methods of [14]. The highlighted numbers on this table are the best solutions that could be reached by some of our procedures, and are lower than the best values produced by the algorithms of [5]. Comparing the results of Table II to those of Table I, we observe that our algorithms present a similar behavior in problems of the same interference constraints and

TABLE I
ALGORITHM'S RESULTS WITH STRATEGIES RES AND FES ON THE 21-CELL SYSTEM

Prob No	Problem Type			LB	Results from [5]		*RES*				*FES*			
							RR=S/F		RR=C/S		RR=S/F		RR=C/S	
							w_1		w_1		w_1		w_1	
	N_c	acc	α		Best	Worse	α	α+1	α	α+1	α	α+1	α	α+1
1	12	2	5	427	460	543	487	540	**445**	464	491	**453**	**443**	**440**
2	7	2	5	427	447	543	452	506	455	454	486	449	447	**436**
3	12	2	7	533	536	566	**533**	-	-	-	**533**	-	-	-
4	7	2	7	533	533	566	533	-	-	-	533	-	-	-
5	12	1	5	381	381	381	381	-	-	-	381	-	-	-
6	7	1	5	381	381	381	381	-	-	-	381	-	-	-
7	12	1	7	533	533	533	533	-	-	-	533	-	-	-
8	7	1	7	533	533	533	533	-	-	-	533	-	-	-
9	12	2	5	258	283	360	302	318	287	293	292	328	295	292
10	7	2	5	253	270	347	283	311	**269**	284	273	291	275	**269**
11	12	2	7	309	310	384	**309**	-	-	-	338	344	**309**	-
12	7	2	7	309	310	358	324	337	**309**	-	**309**	-	-	-
13	12	2	12	529	529	534	529	-	-	-	529	-	-	-

TABLE II
ALGORITHM'S RESULTS WITH STRATEGIES RES AND FES ON THE 49-CELL SYSTEM

Prob No	Problem Type			LB	Heuristics of [5]		*RES*				*FES*			
							RR=S/F		RR=C/S		RR=S/F		RR=C/S	
							w_1		w_1		w_1		w_1	
	N_c	acc	α		Best	Worse	α	α+1	α	α+1	α	α+1	α	α+1
14	12	2	5	413	486	585	508	503	481	485	484	481	**470**	474
15	7	2	5	400	481	585	472	486	477	488	487	477	488	**470**
16	12	2	7	484	520	561	533	529	527	527	484	-	-	-
17	7	2	7	484	523	559	513	559	534	551	484	-	-	-
18	12	1	5	413	422	448	**414**	419	419	417	**414**	419	419	417
19	7	1	5	346	349	399	346	-	-	-	346	-	-	-
20	12	1	7	484	484	492	484	-	-	-	484	-	-	-
21	7	1	7	484	484	484	484	-	-	-	484	-	-	-
22	12	2	5	270	307	364	305	308	304	**298**	314	312	309	310
23	7	2	5	219	275	345	298	278	289	281	273	272	273	**266**
24	12	2	7	270	330	418	305	307	302	**301**	319	327	318	317
25	7	2	7	260	297	344	289	290	291	293	280	286	**275**	289
26	12	2	12	445	447	466	452	466	454	467	**447**	463	**447**	467

similar requirement vectors (upper and lower parts of tables), while in the second table more problems prove to be difficult, that is, 14, 15, and 22–25.

From both tables, we observe that rule S/F is not very effective for difficult problems, irrespectively of the strategy used, and in general performs even worse when $w_1 = \alpha + 1$. The latter is expected since a wrong decision on the number of repetitions wastes more frequencies in this case. When the two strategies RES and FES are compared to each other, it seems that FES is more effective since most of the best solutions have been produced by some algorithm of this kind. Furthermore, when rule C/S is used, FES behaves better than RES on the requirement vectors R_1 and R_3 which both have a vicinity of heavily loaded cells centrally located. This argument can be partly justified by the fact that FES, in conjunction with C/S, always ensures fair multiplexing of those heavily loaded cells. On the other hand, RES cannot perform so well because it is concerned more with allocating lower channels than ensuring that channels are allocated to cells with the great requirements. This effect is somehow alleviated in problems 9, 10, 22, and 24 because in R_2 and R_4 there are many dispersed cells with similar large demand, and RES is more likely to yield allocations to these cells.

The effect of increasing the value of Δb in procedure SWAPP is depicted in Table III, where we have set $w_1 = \alpha + 1$. In general, the most appropriate method for choosing a suitable value for w_1 for the general algorithm, is to adopt that value which produced the best results with FES or RES. Indeed, this is possible considering that all the results of Tables I and II can be obtained within two minutes time. For problems that produce better results with some strategy of the FES (respectively, RES) kind, we have set $\Delta_d \gg \Delta_c$ (respectively, $\Delta_d \ll \Delta_c$). In other words, the results of Tables I and II give us also some indication about what values for the parameters Δ_d and Δ_c are more suitable to certain types of problems. Again, the indicated numbers on this table are the best results obtained for those problems. It seems that increasing Δb performs in some cases better than the corresponding FES or RES strategy. Thus, this table confirms our intuition regarding the performance of the general algorithm where the decisions about the allocations are made when more than one cell requirement is considered at each

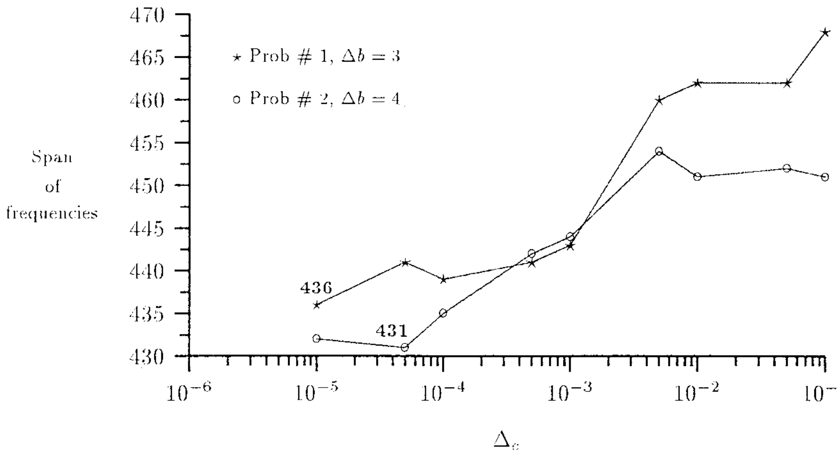


Fig. 8. Effect of ratio Δ_d/Δ_c for problems 1 and 2. RR = C/S, $w_1 = \alpha + 1$, and $\Delta_d = .001$.

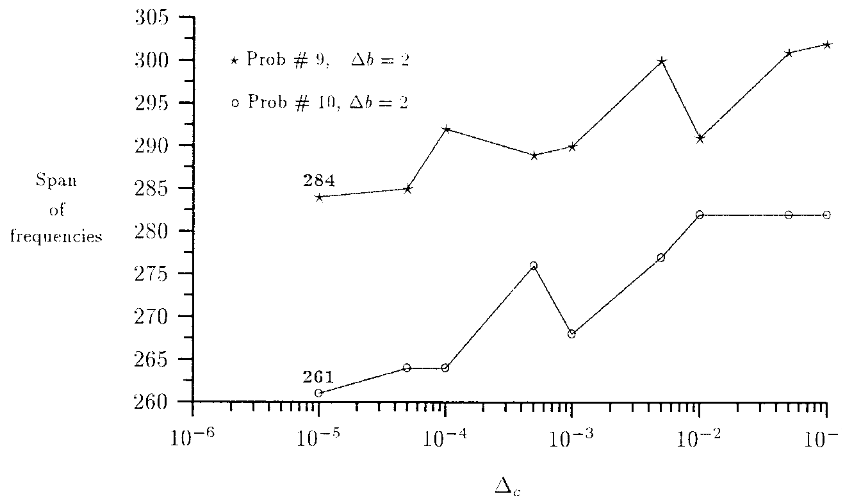


Fig. 9. Effect of ratio Δ_d/Δ_c for problems 9 and 10. RR = $C/S, w_1 = \alpha + 1$, and $\Delta_d = .001$.

TABLE III
EFFECT OF Δb, RR = C/S, AND $w_1 = \alpha + 1$

Prob No	Δ_d	Δ_c	Δb 1	2	3	4
1	.01	.0001	440	440	**436**	437
2	.01	.0001	436	437	435	**432**
9	.01	.0001	292	**284**	298	288
10	.01	.0001	269	**261**	265	265
14	.01	.0001	**474**	479	478	476
15	.01	.0001	**470**	478	480	480
22	.0001	.01	298	303	**294**	**294**
23	.01	.0001	266	261	261	**259**
24	.0001	.01	301	**300**	307	305

step of SWAPP. However, this performance gain is achieved at the expense of higher computational complexity due to the application of the gradient projection method. In order to provide some indication about the response times of these procedures, which were not optimally coded, we note that their running times ranged from 5 to 15 min on a SUN Sparc 2 workstation. These times are of moderate size, but certainly high compared to the running times of the procedures in Tables I and II which never exceeded 2 s.

In Figs. 8 and 9, we demonstrate the algorithm's behavior when the ratio Δ_d/Δ_c is varied. Parameter Δb is always set to the value that produces the best solution in Table III. The bottom scale is logarithmic and the best results are also marked. Note that the results 431 and 261 of problems 2 and 10 are better than those quoted in [9]. These figures justify the conjecture of the previous paragraph regarding the values chosen for the parameters Δ_d and Δ_c in Table III. Namely, it seems that when a problem presents better results with an algorithm of the FES kind, then the search for lower spans with the general procedure should be limited to values greater than one for the ratio Δ_d/Δ_c.

Eventually, as another test system we used the network from Helsinki, introduced in [12]. The results of the simple strategies FES and RES, with $w_1 = \alpha$ and RR $= S/F$, always yielded for the span of frequencies the value 73 which is equal to the lower bound given in [12].

Throughout this paper, we are assuming that $c_{ii} = \alpha$. Let us now describe how our procedures can be extended to accommodate the general case $c_{ii} = \alpha_i$. Introducing α_l during each iteration l, we can obtain a similar algorithm where the allocation width w_l of the allocation pattern will depend on the *current* value of α_l. Different ways for calculating α_l can produce several variations of the procedure. For example: 1) $\alpha_l = \min\{\alpha_i | i \in V_l\}$; or 2) $\alpha_l = \alpha_i$ with r_i maximum, $i \in V_l$; or 3) $\alpha_l = \max\{\alpha_i | i \in V_l\}$. In the first two cases, an additional check should be performed before the number k_l of repetitions of the allocation pattern is computed because if a frequency is assigned to some cell i with $\alpha_i > \alpha_l$, then k_l should be one. How well each one of these variations may behave, is again expected to depend on the particular problem instance, and is left for further study.

VII. Concluding Remarks

We have presented a new sequential procedure for channel allocation in cellular networks. It has been successfully applied to a variety of problems, some of which are well known in the literature, and yielded significant results in the majority of cases. The flexibility of our method lies on the possibility of controlling the solutions obtained, by appropriate adjustment of the parameters involved. Thus, for certain values of the parameters, our general procedure is equivalent to very fast sequential heuristics which have computational time, if not better than, at least comparable to that of the graph coloring algorithms. In addition, the spans obtained by our heuristics are in most cases remarkably lower than those produced by the graph coloring heuristics. The general procedure behaves even better and for the majority of the problems studied it yielded a further reduction in the frequency spans. In conclusion, our study shows that the technique developed in this paper is an interesting alternative to existing techniques. Future research will explore possible improvements to this algorithm.

References

[1] V. H. MacDonald, "Advanced mobile phone service: The cellular concept," *Bell Syst. Tech. J.*, vol. 58, pp. 15–41, Jan. 1979.

[2] W. K. Hale, "Frequency assignment: Theory and applications," *Proc. IEEE*, vol. 68, pp. 1497–1514, Dec. 1980.

[3] J. A. Zoellner and C. L. Beall, "A breakthrough in spectrum conserving frequency assignment technology," *IEEE Trans. Electromag. Compat.*, vol. EMC-19, pp. 313–319, Aug. 1977.

[4] A. Gamst and W. Rave, "On frequency assignment in mobile automatic telephone systems," in *Proc. GLOBECOM '82*, pp. 309–315.

[5] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchum, "Channel assignment in cellular radio," in *Proc. 39th IEEE Veh. Technol. Soc. Conf.*, May 1989, pp. 846–850.

[6] M. Sengoku, H. Tamura, S. Shinoda, T. Abe, and Y. Kajitani, "Graph theoretical considerations of channel offsets systems in a cellular mobile system," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 405–411, May 1991.

[7] F. Box, "A heuristic technique for assigning frequencies to mobile radio nets," *IEEE Trans. Veh. Technol.*, vol. VT-27, pp. 57–64, May 1978.

[8] S. Kim and S.-L. Kim, "A two-phase algorithm for frequency assignment in cellular mobile systems," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 542–548, Aug. 1994.

[9] W. Wang and C. K. Rushforth, "An adaptive local-search algorithm for the channel-assignment problem (cap)," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 459–466, Aug. 1996.

[10] R. Mathar and J. Mattfeldt, "Channel assignment in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 647–656, Nov. 1993.

[11] M. Duque-Anton, D. Kunz, and B. Ruber, "Channel assignment for cellular radio using simulated annealing," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 14–21, Feb. 1993.

[12] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 188–193, Feb. 1991.

[13] 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.

[14] A. Gamst, "Some lower bounds for a class of frequency assignment problems," *IEEE Trans. Veh. Technol.*, vol. VT-35, pp. 8–14, Feb. 1986.

[15] A. N. Rouskas, M. G. Kazantzakis, and M. E. Anagnostou, "Optimal channel assignment in cellular networks," *Int. J. Commun. Syst.*, vol. 8, pp. 359–364, 1995.

[16] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[17] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1989.

[18] L. G. Anderson, "A simulation study of some dynamic channel assignment algorithms in a high capacity mobile telecommunications system," *IEEE Trans. Commun.*, vol. COM-21, pp. 1294–1301, Nov. 1973.

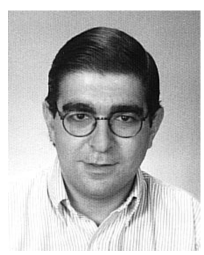
[19] J. Janssen and K. Kilakos, "An optimal solution to the 'Philadephia' channel assignment problem," Tech. Rep., Centre for Discrete and Applicable Mathematics, London School of Economics, London, U.K., 1996.

[20] D. H. Smith, S. Hurley, and S. U. Thiel, "Improving heuristics for the frequency assignment problem," *Eur. J. Oper. Res.*, 1997.

Angelos N. Rouskas (S'94–M'97) was born in Athens, Greece, in 1968. He received the Diploma in electrical engineering from the National Technical University of Athens (NTUA), Athens, in 1991, the M.Sc. degree in communications and signal processing from Imperial College, London, U.K., in 1993, and the Ph.D. degree in electrical and computer engineering from NTUA in 1996.

During 1991, and since 1994, he has been a Research Associate at the telecommunications laboratory of NTUA under the framework of several national and European research projects related to aircraft telecommunications, ATM networks, and wireless communications. His research interests include complexity theory, ATM and wireless networks, and performance evaluation.

Dr. Rouskas is a member of the Technical Chamber of Greece.



Michael G. Kazantzakis was born in Athens, Greece, in 1962. He received the Diploma and Ph.D. degree from the National Technical University of Athens (NTUA), Athens, in 1986 and 1992, respectively.

From 1992 to 1993, he was a Member of the Staff of the Hellenic Navy Research Centre working in the area of underwater acoustics. Since 1994, he has been a Researcher at the Department of Electrical and Computer Engineering, Telecommunications Laboratory, NTUA. His interests include data, mobile, and integrated services networks.

Miltiades E. Anagnostou (M'81) was born in Athens, Greece, in 1958. He received the Electrical Engineers Diploma from the National Technical University of Athens (NTUA), Athens, in 1981 and the Ph.D. degree in computer networks in 1987.

From 1981 to 1987, he was a Teaching Assistant at NTUA. From 1989 to 1991, he was a Lecturer, from 1991 to 1996 an Assistant Professor, and since August 1996 an Associate Professor in the Computer Science Division, Department of Electrical and Computer Engineering, NTUA. He teaches courses on modern telecommunications, formal specification of protocols, stochastic processes, and queueing theory. Since 1981, he has been involved in many research programs—both national and international—in the areas of ATM networks, mobile and personal communications, aircraft communications and air traffic control, and service engineering. He is the author of about 60 scientific papers in the areas of computer networks, protocols, queueing theory, algorithms, mobile and personal communications, and B-ISDN.

Dr. Anagnostou is a member of the Technical Chamber of Greece.