A Multi-agent System for FrequencyAssignment in Cellular Radio Networks

Jaime Abril, Francesc Comellas, Alejandro Cortés, Javier Ozón and Miguel Vaquer

Abstract-- **In this paper we present a multi-agent algorithm for the frequency assignment problem in cellular radio networks. The algorithm, that has been successfully applied to GSM networks, efficiently assigns frequencies to each radio cell satisfying the constraints given by a compatibility matrix.**

***Index terms*—frequency assignment, channel assignment, multi-agent algorithm, GSM networks, cellular networks, graph coloring**

I. Introduction

The frequency assignment problem for a cellular network has been thoroughly discussed in the literature owing, in part, to the important economic repercussions that it has for network operators. The largest outlay to be confronted by an operator is the license fee for the allocation of the radio frequency spectrum. As this spectrum is limited, the profitability of the network will depend on the number of stations required to satisfy the user demand.

In the design of a cellular mobile communication system, a region is divided into a series of cells, each one covered by a base station. Each station can also have more than one transmitter-receiver (also called a transceiver or simply transmitter). Each transmitter works at a certain frequency and the same frequency can be used in different cells (allowing a reduction in the number of base stations required and in the overall cost of the installation). However, the reuse of the same frequency in different cells is limited by the maximum tolerable interference between cells. The frequency assignment problem consists of finding an assignment (for each transmitter) that satisfies certain traffic requirements, using the smallest possible number of frequencies and considering the interference constraints (a frequency has one or more channels, depending on whether the system is FDMA or FD/TDMA). The problem discussed here is a problem of optimization, and has evident economic repercussions.

This paper presents an algorithm, *ants*, for the assignment frequency problem. The algorithm has been shown [1] to be more effective than simulated annealing or genetic algorithms for the bipartite subgraph problem (an instance of graph coloring). We compare its performance with simulated annealing using data from GSM networks in operation in Spain.

This paper is organized as follows: The frequency assignment problem is approached in the next section in terms of a graph coloring problem. In Section III, we describe a method for solving this problem based on a multi-agent system (*ants* algorithm). In Sections IV and V we delimit the solution to the problem by generating upper bounds through sequential assignment methods, and introducing a new algorithm (*worms*) which gives lower bounds through the search of cliques. Finally, in Sections VI and VII we discuss the implementation of the algorithms and we present the results obtained for real GSM networks.

II. The frequency assignment problem

A cellular network can be described by a weighted graph where the nodes correspond to the cells or the transmitters (if each cell has more than one transmitter) and the edges join nodes corresponding to adjacent cells or transmitters in the network. The weight of the edges (0, 1, 2, etc.) represents the separation that the frequencies corresponding to the cells or transmitters should keep between each other to prevent interference. In this context, the frequency assignment problem may be seen as a graph coloring problem, which consists in assigning colors (i.e. frequencies) to the nodes (0, 1, 2, etc.) so that the separation between the colors of any pair of nodes (defined as the absolute value of their difference) is at least the weight of the edge joining them.

The frequency assignment problem is characterized by a compatibility matrix, a requirement vector and a fixed frequency plan.

The **compatibility** or **constraints matrix** is a matrix whose elements give the separation that should exist between the frequencies corresponding to the cell row and the cell column. This separation is represented by a natural number with values 0, 1, 2, etc. A 0 means that the two cells hardly interfere and therefore the same frequency may be reused. In this case, transmitters located in each cell can share the same frequency. A 1 implies that the transmitters located in these cells must use frequencies that maintain a minimum separation of one unit. That is, co-channel interference between the two transmitters is unacceptable

J.A (jabril@airtel.es), A.C. (cortesa@en.ono.es) and M.V. (Amena16@retemail.es) are with Airtel Móvil, S.A., Servicio de Radiofrecuencia, Porto Pi 8, 07015 Palma de Mallorca, Spain. F.C. (comellas@mat.upc.es) and J.O. (ozzy@mat.upc.es) are with the Universitat Politècnica de Catalunya, Dept. Matemàtica Aplicada i Telemàtica, Campus Nord, J.Girona Salgado 1-3, 08034 Barcelona, Spain.

but interference of adjacent channels is allowed. This situation corresponds to neighboring or border cells in radio terms. A 2 or a higher number means that these cells must use frequencies separated by at least two units. This is usually required for frequencies in the same cell, depending of the base station equipment. Two cells or transmitters are adjacent (neighbor nodes) if their frequency separation is greater or equal than one.

The compatibility matrix must be constructed with extreme precision so that it reflects the real network as closely as possible. The quality of the service will depend greatly on the precision of this matrix, since it controls the frequencies that are assigned. A badly estimated constraint (0 instead of 1) may cause interference if the solution involves the reuse of the same frequency in affected cells, causing an obvious degradation of the service. The compatibility matrix is therefore the most critical parameter for solving the problem.

The criteria used to obtain the compatibility matrix may vary according to the use of certain features of the system such as dynamic power control, discontinuous transmission and frequency hopping, which are characteristic of GSM networks. Different matrices can also be considered, one for broadcasting frequencies or beacons (radiated continually at maximum power from the base station) and a less intense one for traffic frequencies (only radiated when there is a conversation underway at medium power). It is advisable to be conservative in applying contraints.

The **requirement vector** indicates the number of frequencies to be used in each cell. This variable will depend on the population index and the total number of subscribers, the average traffic generated at peak time and the grade of service of the network.

Finally, the **fixed frequency plan** consists of a set of predetermined frequencies that cannot be modified in the frequency assignment when the network is extended with new stations or transmitters. Sometimes it is not possible to establish a new design under the constraints of the fixed frequency plan. In these cases the network is redesigned after freeing some frequencies until a correct assignment is obtained.

Tables 1 and 2 correspond to a compatibility matrix, a requirement vector and a fixed frequency plan of a GSM network in operation at the Balearic Islands at the time of this study. Note that the upper triangle of the compatibility matrix of Table 1 only contains zeros, since it is assumed that the matrix is symmetrical (thanks to a previous symmetrization: constraint(i,j) = max(constraint(i,j), constraint(j,i)). Also, all the elements of the main diagonal have a value of 2. This means that the separation between the frequencies of the different transmitters in the same cell must be at least 2. A -99 in the fixed frequency plan vector means that the frequency is free, and a value between 0 and *S-1* should be understood as the fixed value for this frequency.

The frequency assignment problem *P* considered here can be formulated in the following way: given a set *X* of *n* cells, a compatibility $n \times n$-matrix $C=(c_{ij})$, a requirement *n*-vector $R=(r_i)$ and a fixed frequency plan *q*-vector $F'=(f'_i)$, where $q = S_i r_i$ is the total number of transmitters, find the frequency plan *q*-vector $F=(f_i)$ compatible with *C, R* and *F'*, using the minimum range of frequencies *S(P),* see [3].

The above problem can be considered in terms of another approach *P'* which is even wider in scope: given a set *X* of cells, a compatibility matrix *C*, a requirement vector *R*, a fixed frequency plan *F'* and a specified range of frequencies *S*, find the frequency plan *F* that minimizes the number of unfulfilled constraints, quantified according to the following cost function: $k=S_{i,j}\ d_{ij},$ where $d_{ij}=0$ if $|f_i-f_j|$ ³ $c_{i'j'}$, or $d_{ij}=1$ if $|f_i-f_j| < c_{i'j'}$. where *i'* and *j'* are respectively the cells where the transmitter *i* and the transmitter *j* are located. That is, a node (or transmitter) *i* does not comply with a constraint when connected to a node *j* by an edge with weight $c_{i'j'}$ it is assigned a frequency at a separation of less than $c_{i'j'}$ of the frequency of node *j*. The cost function *k* is the sum of all the unfulfilled constraints for each of the nodes or transmitters in the network. *P'* can be defined as the quintuple *P' = (X, R, C, F'*, *S).*

III. THE *ANTS* ALGORITHM

The frequency assignment problem described in the previous section is an NP-complete problem [4]: when the size of the graph increases, the execution time of an algorithm capable of solving the problem can be assumed to grow exponentially. This means that the problem is practically unsolvable for real networks and that approximate algorithmic methods (able to obtain solutions close to the absolute minimum in a reasonable calculation time) must be used in practical applications. The research carried out include[1] sequential assignment methods, exhaustive search, simulated annealing [2,9], tabu search, genetic algorithms [8], greedy algorithms and hybrid methods [6,10]. The only way to guarantee the optimum solution is an exhaustive search. Nevertheless, this is only applicable to very simple problems in which the number of nodes is small.

The *ants* algorithm is a multi-agent system based on the idea of parallel search. A generic version of the algorithm was proposed in [1]. Here, the algorithm is applied to the frequency assignment problem in cellular mobile telephone networks.

The mechanism of the algorithm is as follows: Initially the graph is colored at random and a given number of agents (ants) is placed on its nodes, also at random. Then the ants move around the nodes and change the coloring according to a local optimization criterion. At a given iteration each ant moves from the current node to the adjacent node with the greatest number of constraints, and replaces its color with a new color which lowers this

[1] S.U. Thiel, S. Hurley and D.H. Smith, "Frequency Assignment Algorithms," Radiocommunications Agency Agreement. Ref. RCCM070. Final Report Year 2. 1996/97. http://www.cs.cf.ac.uk/User/Steve.Hurley/Ra/year2/

v l C >	CELL01	CELL02	CELL03	CELL04	CELL05	CELL06	CELL07	CELL08	CELL09	CELL10	CELL11	CELL12	CELL13	CELL14	CELL15	CELL16	CELL17	CELL18	CELL19	CELL20	CELL21	CELL22	CELL23	CELL24	CELL25	CELL26	CELL27	CELL28	CELL29	CELL30	CELL31	CELL32	CELL33	CELL34	CELL35
CELL01	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL02	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL03	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL04	0	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL05	1	1	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL06	1	0	1	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL07	1	1	1	2	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL08	0	0	1	1	1	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL09	1	1	1	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL10	1	1	2	1	1	1	1	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL11	1	1	1	0	0	0	0	0	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL12	1	1	1	1	1	1	1	1	1	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL13	2	2	1	1	1	1	1	0	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL14	1	2	2	1	1	1	1	0	1	2	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL15	1	1	1	2	2	1	1	1	1	2	1	1	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL16	1	1	0	0	0	0	1	0	1	1	1	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL17	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL18	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL19	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL21	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL22	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
CELL23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0
CELL24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2	2	2	0	0	0	0	0	0	0	0	0	0	0
CELL25	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0
CELL26	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	0	0	0	0	0	0	0	0	0
CELL27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	2	0	0	0	0	0	0	0	0
CELL28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	2	0	0	0	0	0	0	0
CELL29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2	0	0	0	0	0	0
CELL30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	2	2	2	0	0	0	0	0
CELL31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	2	2	2	0	0	0	0
CELL32	1	1	1	1	1	2	1	2	0	1	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	2	2	1	1	1	1	2	0	0	0
CELL33	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	2	2	2	2	2	2	2	0	0
CELL34	0	1	0	1	1	1	2	2	0	1	1	1	1	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	2	0
CELL35	0	1	0	1	1	1	2	1	1	1	1	1	1	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	2	2

Table 1. The compatibility matrix for GSM network 3b.

	# Transmitters		1st fixed frequency	2nd fixed frequency
CELL01	2	CELL01	-99	-99
CELL02	2	CELL02	-99	-99
CELL03	2	CELL03	-99	-99
CELL04	2	CELL04	-99	-99
CELL05	2	CELL05	-99	-99
CELL06	2	CELL06	-99	-99
CELL07	2	CELL07	-99	-99
CELL08	2	CELL08	-99	-99
CELL09	1	CELL09	-99	
CELL10	2	CELL10	-99	-99
CELL11	2	CELL11	-99	-99
CELL12	1	CELL12	-99	
CELL13	2	CELL13	-99	-99
CELL14	2	CELL14	-99	-99
CELL15	2	CELL15	-99	-99
CELL16	2	CELL16	-99	-99
CELL17	2	CELL17	-99	-99
CELL18	2	CELL18	-99	-99
CELL19	1	CELL19	-99	
CELL20	1	CELL20	-99	
CELL21	1	CELL21	-99	
CELL22	2	CELL22	-99	-99
CELL23	2	CELL23	-99	-99
CELL24	2	CELL24	-99	-99
CELL25	2	CELL25	-99	-99
CELL26	1	CELL26	-99	
CELL27	2	CELL27	-99	-99
CELL28	2	CELL28	-99	-99
CELL29	1	CELL29	-99	
CELL30	1	CELL30	-99	
CELL31	2	CELL31	-99	-99
CELL32	1	CELL32	-99	
CELL33	1	CELL33	-99	
CELL34	2	CELL34	-99	-99
CELL35	2	CELL35	-99	-99

Table 2. Requirement vector and fixed frequency plan for network 3b. A –99 means that the frequency is not fixed.

number. Associated to each node there is a cost function which gives its number of unfulfilled constraints. This local cost function must be updated (for the node and its adjacent nodes) after a change of color.

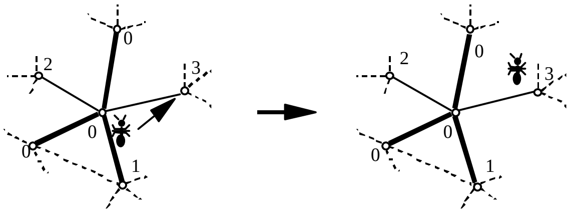


Fig. 1. Movement of an ant towards the worst node (the values indicate the number of constraints that each node does not fulfill). The ant changes the color of this node and updates the cost functions of neighboring nodes. The edges in bold have weight 2; the others are of weight 1.

This action is randomly repeated for each ant: the agent or ant moves to the worst adjacent node with a probability of p_n (otherwise it moves to any other adjacent node), and assigns the best color, under probability p_c (otherwise it assigns any color at random). Both probabilities are adjustable parameters. The probabilistic nature of the algorithm allows it to escape from local minima and obtain colorings close to the absolute minimum. The process is repeated until a solution fulfilling all the constraints is found or the algorithm converges. The number of ants in the algorithm is also an adjustable parameter that should increase with the diameter of the graph (maximum of the distances between pairs of nodes) .

In the same way as in an anthill, the co-ordination between different agents with simple behavior gives rise to a social structure capable of carrying out complicated tasks (such as finding the shortest route between two points), the algorithm presented here, which is based on a series of simple local actions that might be carried out in parallel, obtains restrictive graph colorings. An outline of the *ants* algorithm is shown in algorithmic notation.

```
ANTS Algorithm

initialize
   put each ant on a randomly chosen node
   color each node at random
   for all nodes
     initialize local_cost_function
   end for
   initialize global_cost_function
   best_cost = global_cost_function

while (best_cost > 0) do
   for all ants
     if (random < $p_c$)
       move to the worst adjacent node
     else
       move randomly to an adjacent node
     end if
     if (random < $p_n$)
       change node color to best color
     else
       change to a randomly chosen color
     end if
     for the chosen node and adjacent nodes
       update local_cost_function
       update global_cost_function
     end for
     if (global_cost_function < best_cost )
       best_cost = global_cost_function
   end for
end while
```

IV. UPPER BOUNDS: SEQUENTIAL ASSIGNMENT METHODS

Sequential methods [5] are a simple way for solving frequency assignment problems. Nevertheless, the solutions obtained by these methods are not optimal and, as we will see later on, it is not difficult to obtain better solutions using other combinatorial optimization methods such as simulated annealing or the *ants* algorithm. For this reason, and also because they are quick to execute, sequential assignment methods are considered here as upper bounds for the solution of the problem, and are used as a reference for the solutions obtained by the *ants* algorithm.

A sequential assignment method has essentially two stages: ordering of the nodes on the graph and channel assignment.

There are several approaches to the ordering of the nodes. Among the most usual are ordering by *degree* and by *degree reduction.* The *degree* of the *i*-th cell, *1£ i£n,* is defined *as* d_i=S_j r_jc_{ij}-c_{ii} which is an approximate measure of the difficulty of assigning a frequency to cell *i.* In the ordering by degree, cells are arranged in descending order of degree and the result is stored in an auxiliary matrix (one row of the matrix corresponds to one cell) and each row contains the nodes or transmitters corresponding to each cell. Finally, the matrix is read by rows or by columns, resulting in two orderings of the nodes for each degree.

The ordering by degree reduction is obtained by iteration of the following steps: selection of the cell with the smallest degree which is then placed in the last position of a temporary list, removal of this cell from the graph and recalculation of the degrees for the cells of the resulting graph. As before, the worst cell in the resulting graph is placed in the penultimate position of the temporary list and is removed from the graph; the process is repeated until the removal of the last cell of the graph. As before, the classification is stored in an auxiliary matrix, yielding two classifications per color, one by rows and the other by columns.

There are also a number of **channel assignment strategies**. The most usual strategies are *exhaustive in frequency* and *exhaustive in requirement.* The *exhaustive in frequency* strategy consists of taking the first node of the list and assigning it frequency 1; taking the second node and assigning it the minimum frequency which is still compatible with the previous assignment; taking the third node and assigning it the minimum possible frequency compatible with the previous assignments; and so forth.

The *exhaustive in requirement* strategy consists of taking the frequency 1 and assigning it to the first node of the list and the rest of nodes of the compatible list that can also use this frequency; taking frequency 2 and assigning it to all the nodes possible of the list, beginning with the first one that is free; and so forth.

At the end, a total of 8 assignments of frequencies are obtained: DWF, DWE, DCF, DCE, RWF, RWE, RCF and RCE, depending on whether the classification is by degree or degree reduction (D/R), rows or columns (W/C), and whether the assignment strategy has been exhaustive in frequency or exhaustive in requirement (F/E). Finally, the most restrictive of these 8 assignments is chosen.

V. Lower bounds: *Worms* algorithm

The following are some lower bounds for the problem of coloring graphs proposed by Gamst [3].

Let $P=(X, R, C, F')$ be a frequency assignment problem, v a positive integer and Q a v-complete subset of X, i.e. a subset such that $c_{ij} \geq v$ for all $i, j \in Q$. Assume that Q' Ì Q (Q' ¹ Q) exists so that c_{ij} ³ $v' > v$ for all $i \in Q'$ and $j \in Q$. Notice that j can also belong to Q'. Then, if $\sum_i r_i > 0$, $i \in Q \backslash Q'$, i.e. if $Q \backslash Q'$ is not an empty set, the following bound is proven for the problem P (where $S(P)$ is the minimum number of frequencies required for P):

$$S(P) \text{ ³ } v'\sum_{i \in Q'} r_i + v\Big(\sum_{j \in Q\backslash Q'} r_j - 1\Big) + 1 \qquad [1]$$

In the event that there is no subset Q' of Q that complies with the previous condition, the bound for the problem frequency range P' is given by the formula:

$$S(P) \text{ ³ } v\Big(\sum_{i \in Q} r_i - 1\Big) + 1 \qquad [2]$$

Worms is a non-deterministic iterative algorithm that uses the previous results to provide a lower bound for the coloring problem. This is performed by agents (here called worms) that move around the nodes of the graph. A worm is a sequence of nodes on the graph that are all adjacent to each other (that is, with constraints greater than 0 between each other). Starting from one node of the graph, the worm goes toward the adjacent node of highest degree (i.e., toward the neighboring node that has the largest number of neighbors) with a given probability (otherwise it would move to any other node, selected at random). It then verifies whether or not the new node is adjacent to the other nodes associated with the worm. If so, the worm incorporates the new node into its chain and the search continues. If not, the worm rejects the node and goes toward another adjacent node. If, finally, it does not find any suitable adjacent node, the worm leaves all the nodes and begins a new cycle from its current location. At each iteration the algorithm updates the best bound found by applying formula [1] or [2] (with $v = 1$ and $v' = 2$) to the appropriate worms. The best bound is not always given by the largest subgraph.

The time taken by the algorithm to find the maximum bound depends greatly on the initial situation; the best strategy consists of starting as many worms as nodes in the network. Each worm would also begin its journey at a different node. Finally, the stop condition for the algorithm is defined by imposing a limited number of iterations after the point at which the most restrictive bound is found, or by establishing an initial limit on the iterations. Note that lower bounds depend on the local characteristics of the graph and therefore it is difficult to predict the general behavior of the algorithm.

From a mathematical point of view, the worm represents a clique (an 1-complete subgraph). The search for the maximum clique of a graph is, like the problem of coloring, NP-complete. In fact, the *worms* algorithm will not always obtain the best clique, though in practice good bounds have been found. It has been demonstrated [10,11] that exact maximum clique algorithms can be used for problems with up to 800 cells. The following is the core of *worms* in algorithmic notation.

```
WORMS Algorithm
initialize
  put one worm over each node
while ( time < maxtime ) do
  for all worms
    if (random < $p_n$)
      move to the adjacent node with max degree
    else
      move randomly to an adjacent node
    end if
    if (new_node forms clique)
      add to worm
      calculate bound & actualize if better
    else
```

```
      initialize worm
       remove all nodes
       choose a new node at random
    end if
  end for
end while
```

Network	#cells	#TRX	neigh/ TRX	Low/upper Bounds	ANTX	ANTX (# viol.)	SA	SA (# viol.)
Red 1	43	43	21.3	14-18	17	16(1)	18	17(1)
Red 2	314	314	26.1	13-19	16	15(6)	16	15(10)
Red 3	45	45	20.2	17-21	18	17(1)	19	18(1)
Red 3b	35	60	27.1	21-23	21	20(1)	21	20(1)
Red 4	99	99	24.1	15-19	17	16(1)	18	17(1)
Red 4b	99	198	54.7	37-44	41	40(2)	40	39(4)
Red 5	90	90	32.3	16-19	18	17(1)	18	17(1)
Red 5b	90	163	39.3	19-22	20	19(1)	19	18(6)

Table 3. Results obtained using ants (ANTX) and a simulated annealing algorithm (SA) for several GSM networks.

VI. IMPLEMENTATION

We built an application called ANTX that integrates the *ants* algorithm, sequential assignment methods and the *worms* algorithm. Firstly, the application reads the parameters of the problem: the compatibility matrix, the requirement vector and a fixed frequency plan. Once the graph of the problem has been generated, a lower bound is calculated by means of the *worms* algorithm, and an upper bound is set using the sequential assignment method. Here the program shows the cliques found by worms, and the subgraph that includes the nodes of all the cliques. As this subgraph contains the nodes with the greatest number of constraints, it is a reliable measure of the complexity of the problem.

At this point, based on the bounds provided by the previous algorithms, the user must decide the frequency range (number of colors) used to start the assignment process. This range must have a minimum value similar to the bound provided by the *worms* algorithm. The next step is to adjust the parameters of the algorithm: number of ants, probability p_n for displacement toward the worst node and probability p_c of assignment of the best color. The convergence time of the algorithm is shorter for larger values of p_n and p_c. This is because the index of local improvement at each iteration increases with p_n and p_c. However, as in this case the probability of moving from one local minimum to another is lower, the probability of ending at a local minimum increases. For the networks considered, both p_n and p_c were set to 0.7 while the number of ants used was 3.

The *ants* algorithm then attempts to obtain a coloring within the initially assigned frequency range. If after a given number of iterations it has not solved the problem, it increases the range of frequencies by one unit (one color is added), and the algorithm is executed again. The process is repeated until a coloring that complies with the constraints of the problem is obtained.

The algorithm stores the best coloring obtained in each case, that is the coloring with the lowest number of unfulfilled constraints. This is because on some occasions (according to the structure of the real network) an admissible coloring may be one that does not comply with a certain constraint, provided it is weak (the separation of two stations is 2, and the difference between the assigned frequencies is 1 and not 0). The *ants* algorithm uses two cost functions, the basic function *k* described in Section II, and an additional function *t* that represents the total number of the constraints that are not completed (each unfulfilled constraint between two transmitters *i* and *j* is now calculated as the difference between the allowed separation less the real difference -in absolute values- of the frequencies assigned to each station: $t_{ij}=c_{ij}-|f_i-f_j|$). One assignment F_1 is considered better than another F_2 if $k_{F_1} < k_{F_2}$ or if $t_{F_1} < t_{F_2}$ when $k_{F_1}= k_{F_2}$.

The solution obtained by the *ants* algorithm is usually 10% to 15% below the best solution obtained by sequential assignment methods. However, there is a reasonable increase in the number of calculations of the *ants* algorithm compared to sequential assignment methods. For common problems of 50 to 300 cells and with requirements of 1 to 3 transmitters per cell, the calculation time for the *ants* algorithm is from a few minutes to several hours using a workstation or PC.

A classical simulated annealing, as described in [7] has been considered for the same GSM networks. Simulated annealing also requires careful tuning of its control parameters to achieve good results. Typical values considered in our implementation are: initial temperature, T_0=0.01, 1000 iterations for a given temperature and exponential cooling rate, T_k=0.9 T_{k-1}.

VII. RESULTS

The application ANTX was tested on several real GSM networks and it resulted in solutions that were close to the lower bounds found by the *worms* algorithm. They were always below the upper bounds calculated using sequential assignment methods and were obtained in reasonable execution times. Five geographical areas of the Balearic Islands and mainland Spain were studied, some of them using different requirement vectors. The data used in the tests is available upon request from the authors at UPC. The results are shown in Table 3.

The results show, for each case, the number of cells in the network under study, the total number of transmitters required (one transmitter per cell in cases 1, 2, 3, 4 and 5 and more than one for the others), the average number of neighbors per transmitter (that is, the average number of constraints between one transmitter and the others, different from 0), the lower bound found by the *worms* algorithm (according to formulas [1] or [2]), the best solution obtained by the sequential assignment method, the best solution obtained by the *ants* and *simulated annealing* algorithms that fulfills all constraints (without violation) and a solution with a lower number of frequencies than the previous one but does not fulfill all the constraints (with violation, the number of constraints not fulfilled is shown in brackets). Network 3 is an extension of network 3b. Moreover, even though networks 4b and 5b have the same number of cells as 4 and 5 respectively, they do not have the same requirement vectors.

Note case 3b, in which the *ants* algorithm has provided an optimal solution, that is, the number of frequencies used is the same as the lower bound, 21. In cases 2, 3 and 4, the solution obtained by the *ants* algorithm is between the bounds obtained by *worms* and by the sequential assignment method. In cases 1, 4b and 5, the solution is closer to the upper bound than to the lower one, and in case 5b the solution is closer to the lower than to the upper bound.

Note also that in spite of the great difference in the number of cells in networks 1, 2, 3, 4 and 5, the number of frequencies required to satisfy the demand hardly varies (in these five networks there is only one transmitter per cell). The solutions all involved between 16 and 19 colors. The same is true for the bounds obtained by the *worms* algorithm and sequential assignment methods. This is because the complexity of the problem is not given by the number of cells but by the average number of neighbors for each transmitter. This number is similar in the first 5 cases, about 25 neighbors per transmitter, so the results are comparable.

We have also compared the *ants* algorithm with our implementation of *simulated annealing*. As is shown in Table 3, *ants* outperforms or is equivalent to this implementation for 6 out of 8 GSM networks. Figure 2 also shows a running time comparison between the *ants* algorithm and *simulated annealing*. As in the *ants* algorithm the number of calculations depends strongly on the average number of neighbors per node, the algorithm obtains best results when this number is low (less than 35). Moreover, the ratio between the computation time for both *ants* algorithm and *simulated annealing* increases with the average number of neighbors per node (see Figure 2).

VIII. CONCLUSIONS

In this paper we have studied the assignment frequency problem for cellular networks, which is of considerable importance for operators of mobile communications services. We have presented the *ants* algorithm for the assignment frequency problem. The effectiveness of the algorithm was proven by applying it to a series of tests on real GSM networks and comparing the results with a standard simulated annealing. The solution to the problem was delimited by devising the *worms* algorithm to calculate lower bounds through the search of cliques. Sequential assignment methods were chosen to obtain upper bounds. Finally, we presented the results of applying ANTX, a tool that combines the previous algorithms, to GSM networks in operation in Spain that were different in size and had various requirements and radio characteristics. We have also shown that in networks with an average number of less than 35 neighbors per node (a common case for GSM networks), the *ants* algorithm works as well or better than our implementation of simulated annealing, and appears to be competitive with the best current algorithms.

Finally, it should be pointed out that the use of the *ants* algorithm, with all the methodology for planning frequencies that it involves (generation of the compatibility matrix, assignment of frequency bands, etc.), simplifies the work of the radio frequency engineer who is regularly faced with this type of problem for a wide variety of reasons: global changes in the frequency plans, integration of new stations, growth of sectors, activation of new functions, external interference, changes in the spectrum, etc.

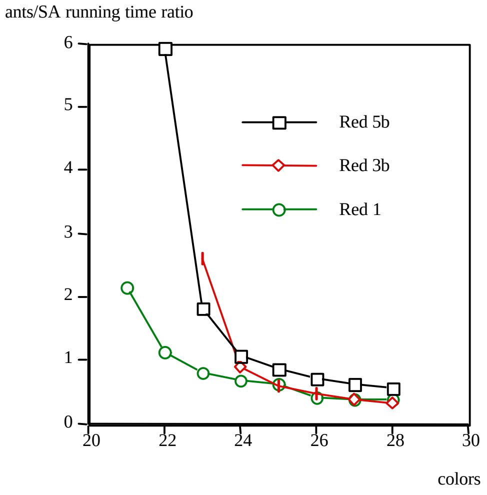


Fig. 2. Ratio between the running time that the ants and the simulated annealing algorithms expend to find a solution for a given number of colors (average on 1000 runs).

IX. ACKNOWLEDGEMENTS

This work is a collaboration between Airtel Movil S.A. (one of the mobile telephone companies in Spain) and the Department of Applied Mathematics and Telematics of the

Technical University of Catalonia (UPC). The authors are grateful to Carlos Rincón, head of the Radio Frequency Service of Airtel for the Balearic Islands and Levante, for promoting this project and providing the necessary support, Gabriel Arjona, Carlos Pulido, Juan Redondo and Germán Rodríguez, radio frequency engineers at Airtel, for the contribution of their ideas and all the other engineers who have used the application to calculate their frequency plans and who, in one way or another, have helped to improve it.
This work was supported in part by the *Comisión Interministerial de Ciencia y Tecnologia*, Spain (ref. TIC-97-0963).

X. References

[1] F. Comellas and J. Ozón, “Graph coloring algorithms for assignment problems in radio networks,” *Applications of Neural Networks to Telecommunications 2*. J. Alspector, R. Goodman and T.X. Brown (Eds.), Lawrence Erlbaum Ass., Inc., Publis., Hillsdale, NJ, pp. 49-56, 1995.

[2] M. Duque-Anton, D. Kunz, and B. Rüber. “Channel assignment for cellular radio using simulated annealing,” *IEEE Trans. Veh. Technol.*, vol. 42, pp. 14-21, 1993.

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

[4] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W.H. Freeman, 1979.

[5] W.K. Hale, “Frequency assignment: theory and applications”, *Proceedings of the IEEE*, vol. 68, pp. 1497-1514, 1980.

[6] S. Hurley, D.H. Smith and S.U. Thiel, “FASoft: A system for discrete channel frequency assignment,” *Radio Science*, Vol 32, No 5, pp 1921-1939, 1997.

[7] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi. “Optimization by Simulated Annealing,” *Science*, vol. 220, pp. 671-680, 1983.

[8] W.K. Lai and G.G. Coghill, “Channel assignment through evolutionary optimization,” *IEEE Trans. Veh. Technol.*, vol. 45, pp. 91-95, 1996.

[9] R. Mathar and J. Mattfeldt, “Channel assignment in cellular radio networks,” *IEEE Trans. Veh. Technol.*, vol. 42, pp. 647-656, 1993.

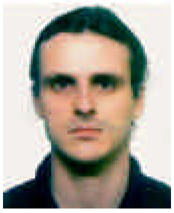
[10] D.H. Smith, S. Hurley and S.M. Allen., “A new lower bound for the channel assignment problem,” *IEEE Trans. Veh. Technol.*, to appear.

[11] D.H. Smith, S. Hurley and S.U. Thiel. “Improving heuristics for the frequency assignment problem,” *European Journal of Operations Research*. Vol 107, No 1, pp 76-86, 1998.

Jaime Abril received a degree in Telecommunication Engineering from the Universitat Politècnica de Catalunya (UPC), Spain, in 1993. In 1996 he obtained a Master Diploma in Mobile Communication from the UPC and Fundación Airtel. Since December 1996 he works as a radio frequency engineer in Airtel Movil S.A. He is currently leading the Cell Planning, Design and Optimization Group of the Airtel GSM network in the Balearic Islands. His current research includes advanced computational techniques and their application to automatically optimize several aspects of radio frequency engineering: frequency planning, antenna direction and downtilt tuning, bestserver optimization, radio interface trouble detection from statistics and drive-tests and other related topics.

Francesc Comellas received a Ph.D. degree (1982) from the Universitat Autònoma de Barcelona. In 1985, he was appointed Professor Titular d'Universitat (Associate Professor) at the Universitat Politècnica de Catalunya. Since then, he has been working with the research group on Graph Theory and Interconnection Networks of the Applied Mathematics and Telematics Department. During the academic years 1988-1989 and 1985-1986 he made research stays at the Informatics Department of the Rutherford-Appleton Lab., S.E.R.C., U.K. and the School of Computing Science of the Simon Fraser University, Burnaby, Canada. His research interests include theoretical aspects and the use of combinatorial optimization algorithm in relation to the design of topologies and communication strategies for interconnection networks.

Alejandro Cortés received a Telecommunications Engineering degree from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1994. After receiving his degree, he spent several years as a radio frequency engineer in Airtel Móvil S.A, a Cellular Operator in Spain. During this period, he used the techniques described in this article. He is currently working as a switching engineer in ONO, a Cable Operator in Spain. His current interests include IP voice and wideband access networks based on cable modem technology.



Javier Ozón was born in Barcelona, Spain, in 1970. He received a Telecommunications Engineering degree from the Technical University of Catalonia (UPC) in 1993. In 1994 he joined the Department of Applied Mathematics and Telematics of the UPC as a Lecturer. He is completing his Ph.D. degree on the algorithmic resolution of some classical graph and combinatorial problems, the analysis and design of small-world networks and the emergence of self-organized criticality in telecommunication networks.

Miguel Vaquer received the degree in Electronics Engineering from the Technical University of Catalonia (UPC), Spain, in 1994. He is currently working as radio frequency engineer at Amena, the third Spanish GSM operator. After receiving his degree, he has been working in the mobile communication industry. His current interests are radio design and optimization, including frequency planning techniques.