An ANTS Heuristic for the Frequency Assignment Problem

Vittorio Maniezzo
Department of Computer Science, University of Bologna
Via Sacchi 3, 47023 Cesena, Italy.
Tel. (+39) 347 6883099, Fax. (+39) 0547 610054
Email: maniezzo@csr.unibo.it, Web: http://www.csr.unibo.it/~maniezzo

Antonella Carbonaro
Department of Computer Science, University of Bologna
Via Sacchi 3, 47023 Cesena, Italy.
Tel. (+39) 0547 642830, Fax. (+39) 0547 610054
Email: carbonar@csr.unibo.it, Web: http://www.csr.unibo.it/~carbonar

Abstract
The problem considered in this paper consists in defining an assignment of frequencies to radio links, to be established between base stations and mobile transmitters, which minimizes the global interference over a given region. This problem is NP-hard and few results have been reported on techniques for solving it to optimality. We have applied to this version of the frequency assignment problem an ANTS metaheuristic, that is an approach following the ACO optimization paradigm. Computational results, obtained on a number of standard problem instances, testify the effectiveness of the proposed approach.

1. Introduction

The introduction of mobile communication, such as portable phones, has a tremendous impact on everyday life. Mobility raises a number of research questions: for many of them discrete models and algorithms are required in order to solve the underlying mathematical problem.

The Ant Colony Optimization paradigm (ACO) [Dorigo and Di Caro, 1999], [Maniezzo and Carbonaro, 1999] has proved successful in dealing with discrete optimization problems, for example with the travelling salesman, the quadratic assignment, the graph coloring and other problems. This paper presents its application to one of the main problems arising in mobile telecommunication, namely the *frequency assignment problem* (FAP, also called *channel assignment problem*), in the specific case of the *radio link frequency assignment problem* [Hale, 1980].

Frequency assignment problems arise when a network of radio links is to be established, and a frequency has to be assigned to each link. This is the case – for example – when cellular mobile phoning is to be introduced in an area, where the connection between the cellular phones and the transmission network is supported by radio links. Specifically, the transmitter (i.e., the phone) establishes a radio link with a receiver (an antenna of a *base-station*), on one of the frequencies that the receiver supports. Interference may arise among different communications, but its level has to be acceptable, or communication will be distorted. Acceptability is usually specified by means of a threshold, called *separation*, on the distance between frequencies which can be operated concurrently by the same receiver or which can be used in areas (the *cells*) where the transmitter interacts with more than one receiver.

Each receiver can operate on a given spectrum of frequencies, which is usually partitioned into *channels*. The problem arising is to define which among the available channels are to be used by each receiver for servicing the radio links minimizing the resulting interference.

The FAP can be formulated as a generalized graph coloring problem, that is the problem of finding a coloring of a graph so that the number of used colors is minimum, subject to the constraint that any two adjacent vertices have two different colors: as such, FAP is a NP-hard problem.

Different lower bounds on the optimal solution value for the frequency assignment problem have been proposed, which are useful both in assessing the quality of approximate solutions and in limiting the search for optimal assignments. Among the others, lower bounds have been proposed by [Smith and Hurley, 1997], [Adjakplé and Jaumard, to appear], [Tcha et al., 1997], [Gamst, 1986] and [Warners et al., 1997]. They are usually derived from graph-theoretic approaches, which adapt techniques originally developed for the coloring problem.

Exact algorithms have been proposed by [Aardal et al., 1995], [Giortzis and Turner, 1996], [Fischetti et al., 1996], [Koster et al., 1997] and [Mannino and Sassano, 1998].

Due to the NP-hardness of the problem, any exact optimization algorithm requires in the worst case an amount of time exponentially growing with the size of the instance. In order to obtain good solutions in a reasonable amount of time and due to the relevant actual importance of the FAP, much effort has been spent in studying heuristic algorithms. Different approaches have been used, including greedy heuristics [Zoellner and Beall, 1977], artificial neural networks

[Funabiki and Takefuji, 1992], simulated annealing [Duque-Anton et al., 1993], constraint programming [Caminada, 1995], tabu search [Adjakplé and Jaumard, 1997], and adapted Dsatur techniques [Bornörfer et al., 1998a]. Many of these techniques have been included in the FASoft algorithm suite [Hurley et al., 1997]. For a detailed overview on heuristics and exact methods for FAP, see [Tiourine et al.,1995].

This work reports about the results obtained applying to FAP a particular instance of the ACO class, namely the ANTS metaheuristic [Maniezzo, 1998]. ANTS is a general combinatorial optimization metaheuristic, which can be tuned to solve the specific problem by including, as component modules, lower bounds and local optimization procedures. The possibility of utilizing strong results from mathematical programming, such as lower bounds, dual analysis, dominances and branching schemes etc., makes the approach appealing for difficult problems for which substantial mathematical results exist.

The paper is structured as follows. In Section 2 we describe the mathematical formulation used to represent the FAP; in Section 3 the general structure and the FAP-specific components of the ANTS algorithm are introduced, while in Section 4 two heuristics with which we compared ANTS results are summarized. In Section 5 we discuss the computational results obtained on different test databases and, finally, in Section 6 we report about the conclusions drawn from our work.

2. Mathematical formulation

The evolution of the telecommunication technology underlying the FAP reflects on the details of the mathematical formulation of the problem, specifically, on the objective function to optimize. While in fact, during the early 80's, the primary concern was that of minimizing frequency rent costs, subject to the constraint of satisfying all requests, we now moved into a situation where the increased service requests force the operators to use all frequencies they can rent, servicing the requests while trying to minimize the arising interferences. In the literature different problems can be found under the common heading of FAP. They can all be formulated by means of the following elements.

Let us define an index set of links $\mathfrak{L} = \{1, \dots, n\}$, a set $\mathfrak{F}_i = \{1, \dots, F_i\}$, $i=1, \dots, n$ of available frequencies for each link and a Channel Separation Matrix (CSM) = $[d_{ij}]$ $i,j = 1, \dots, n$, where d_{ij} defines the minimal distance between frequencies assigned to links i and j.

Let $f_i \in \mathfrak{I}_i$ be a positive integer variable specifying the frequency assigned to link i, Φ be a variable representing the highest used frequency. Three different problems can be defined.

The first problem (FAP1) asks to minimize the number of frequencies used - that is, the channel *spectrum* - while satisfying all requests with no interference. Formally:

$$\text{(FAP1)} \quad min \quad \Phi \tag{1}$$

$$\text{s.t.} \quad \Phi \geq f_i \qquad i = 1, \dots, n \tag{2}$$

$$|f_i - f_j| \geq d_{ij} \qquad i,j = 1, \dots, n \tag{3}$$

$$f_i \in \mathfrak{I}_i \qquad i = 1, \dots, n \tag{4}$$

The second problem (FAP2) still does not accept interferences, but operates on a fixed frequency spectrum, thus Φ becomes a parameter: $\Phi = max\ \{F_i, i=1, \dots, n\}$. The objective is now to accept as many connection requests as possible. Formally, let y_i be a binary variable which is equal to 1 if and only if the connection associated with link i cannot be accepted. The formulation is:

$$\text{(FAP2)} \quad min \quad \sum_{i=1}^{n} y_i \tag{5}$$

$$\text{s.t.} \quad \Phi(y_i + y_j) \geq d_{ij} - |f_i - f_j| \qquad i = 1, \dots, n,\ j \in \Gamma_i \tag{6}$$

$$f_i \in \mathfrak{I}_i \qquad i = 1, \dots, n \tag{4}$$

$$y_i \in \{0,1\} \qquad i = 1, \dots, n \tag{7}$$

Finally, the third version of the problem, still operating on a fixed frequency spectrum, asks to find an assignment of frequencies to the links, so that the cost incurred for violating constraints from the CSM is minimized. Formally, let ζ_{ij} be a binary variable which is equal to 1 if and only if the minimal required separation between frequencies f_i and f_j is not guaranteed and c_{ij} be the cost for violating a CSM constraint (ij). The formulation is:

$$\text{(FAP3)} \quad min \quad \sum_{i,j=1}^{n} c_{ij}\, \zeta_{ij} \tag{8}$$

$$\text{s.t.} \quad \Phi\zeta_{ij} \geq d_{ij} - |f_i - f_j| \qquad i,j = 1, \dots, n \tag{9}$$

$$f_i \in \mathfrak{I}_i \qquad i = 1, \dots, n \tag{4}$$

$$\zeta_{ij} \in \{0,1\} \qquad i,j = 1, \dots, n \tag{10}$$

In all cases, the problem may be viewed as a generalization of the graph coloring problem, and as such it is NP-complete. In fact, to each instance of the problem it is possible to associate a weighted interference graph G = (V,E,W), where V is the set of vertices, E is the set of edges and

W is a weight vector. To each frequency request there corresponds a vertex v from V. There is an edge (ij) between vertices i and j if and only if there is a non zero entry in the CSM for the corresponding request. The edges are weighted by means of the corresponding values from the CSM. Clearly, no two connected vertices can be labeled with the same frequency.

A further element of some instances of the problem is that some links may have a pre-assigned frequency (they have reduced *mobility*), and it is expensive to modify this assignment [Warners et al., 1997]. The full formulation for these instances becomes the following. Besides the already introduced variables, let β_i be a binary variable which is equal to 1 iff link i was pre-assigned to a frequency $\bar{f}_i$, but it has to operate on a frequency different from $\bar{f}_i$, and p_i be the cost for not meeting this mobility constraint. The problem becomes:

$$\text{(FAP)} \quad \min \sum_{(ij)\in E} c_{ij}\,\zeta_{ij} + \sum_{i\in V} p_i\,\beta_i \tag{11}$$

$$\text{s.t.} \quad \Phi\beta_i \geq |f_i - \bar{f}_i| \qquad i \in V \tag{12}$$

$$\Phi\zeta_{ij} \geq d_{ij} - |f_i - f_j| \qquad (ij) \in E \tag{9}$$

$$f_i \in \Im_i \qquad i \in V \tag{4}$$

$$\zeta_{ij}, \beta_i \in \{0,1\} \qquad (ij) \in E,\ i \in V \tag{10'}$$

Constraints (12) are the mobility constraints, constraints (9) are the separation constraints, while constraints (4) and (10') represent restricted integer domain conditions. The following of the paper will make reference to problem FAP, that is a specialization of problem FAP3.

3. The ANTS algorithm

Being FAP NP-hard, heuristics are in order, and so-called metaheuristic algorithms are a possibility for obtaining good, sub-optimal solutions in a reasonable amount of time. The solution technique under investigation in this research is an effective metaheuristic of the ACO class.

The first ACO metaheuristic has been the Ant System, proposed by Colorni, Dorigo and Maniezzo [Colorni et al., 1991], [Dorigo et al., 1991], [Dorigo, 1992]. The main underlying idea was that of parallelizing search over several constructive computational threads, all based on a dynamic memory structure incorporating information on the effectiveness of previously obtained results and in which the behavior of each single agent is inspired by the behavior of real ants. The work presented in this paper is based on an adaptation [Maniezzo, 1998] of the original Ant System, designed to make it more effective on combinatorial problems. This new method has been given the name ANTS, to reflect both the Ant System underlying approach and the possibility of viewing it, as explained in Section 3.2, as an ***A***pproximate ***N***ondeterministic ***T***ree-***S***earch procedure.

3.1 General framework

An *ant* is defined to be a simple computational agent, which iteratively constructs a solution for the problem to solve. Partial problem solutions are seen as *states*; each ant *moves* from a state ι to another one ψ, corresponding to a more complete partial solution. At each step σ, each ant *k* computes a set $A_{\sigma}^{k}(\iota)$ of feasible expansions to its current state, and moves to one of these in probability, according to a probability distribution specified as follows.

For ant k, the probability $p_{\iota\psi}^{k}$ of moving from state ι to state ψ depends on the combination of two values:

i) the *attractiveness* η of the move, as computed by some heuristic indicating the *a priori* desirability of that move;

ii) the *trail level* τ of the move, indicating how good it has been in the past to make that particular move: it represents therefore an *a posteriori* indication of the desirability of that move.

Trails are *updated* at each iteration, increasing the level of those that facilitate moves that were part of "good" solutions, while decreasing all others.

The specific formula for defining the probability distribution at each move makes use of a set tabu$_k$, which indicates the problem-dependent set of infeasible moves for ant *k*. Probabilities are computed as follows (see [Maniezzo, 1998] for a discussion):

$$p_{\iota\psi}^{k} = \begin{cases} \dfrac{\alpha \cdot \tau_{\iota\psi} + (1-\alpha)\cdot \eta_{\iota\psi}}{\sum_{(\iota\nu)\notin tabu_k} \left(\alpha \cdot \tau_{\iota\nu} + (1-\alpha)\cdot \eta_{\iota\nu}\right)} & \text{if } (\iota\psi) \notin \text{tabu}_k \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Parameter α defines the relative importance of trail with respect to attractiveness.

After each iteration *t* of the algorithm, that is when all ants have completed a solution, trails are updated in the following way:

$$\tau_{\iota\psi}(t) = \rho\tau_{\iota\psi}(t-1) + \Delta\tau_{\iota\psi} \tag{14}$$

where ρ is a user-defined coefficient and $\Delta\tau_{\iota\psi}$ represents the sum of the contributions of all ants that used move ($\iota\psi$) to construct their solution. The ants' contributions are proportional to the quality of the achieved solution, i.e., the better an ant solution, the higher will be the trail contribution added to the moves it used.

3.2 The ANTS algorithm

Several modifications of the Ant System are incorporated in the ANTS algorithm. Besides formula (13), which already differs from the original proposal, other modifications include:

Use of lower bounds (min problems)

The attractiveness of a move can be effectively estimated by means of lower bounds (upper bounds in case of maximization problems) to the cost of the completion of a partial solution. In fact, if a state ι corresponds to a partial problem solution it is possible to compute a lower bound to the cost of a complete solution containing ι. Therefore, for each feasible move ($\iota\psi$), it is possible to compute the lower bound to the cost of a complete solution containing ψ: the lower the bound the better the move.

Since large part of research in combinatorial optimization is devoted to the identification of tight lower bound for the different problems of interest, good lower bounds are usually available. Their use has several advantages, some of which are listed in the following.

- A tight bound gives strong indications on the opportunity of a move.
- When the bound value becomes greater than the current upper bound, it is obvious that the considered move leads to a partial solution which cannot possibly be completed in a solution better than the current best one. The move can therefore be discarded from further analysis.
- If the bound is derived from linear programming (LP) it is possible to compute "reduced costs" for the problem decision variables, which in turn permit – when compared with an upper bound to the optimal problem solution cost – to a priori eliminate some variables [Dantzig, Thapa, 1997]. This results in a reduction of the number of possible moves, therefore to a reduction of the search space.
- A further advantage of LP lower bound is that the values of the decision variables, as computed in the bound solution, can be used as an indication of whether each variable will appear in good solutions. This provides an effective way for initializing the trail values, thus eliminating the need for the user-defined parameter $\tau_{\iota\psi}(0)$.

While the use of LP bounds is a very effective and straightforward general policy, this turns out to be complicated in the case of FAP, specifically of FAP3. In fact, despite the number of efforts devoted to finding tight bounds little success has been achieved. The identification of an effective lower bound for the FAP, let alone an LP-based one, is still a research topic of its own. We believe that one of the most promising bounds, so far presented with no computational tests reported, is the orientation model of [Borndörfer et al., 1998b]. While the formulation presented by Borndörfer et al. is not of direct interest for us, it can be easily adapted to produce a linearization of FAP3, in the following denoted OP.

A new set of binary variables o_{ij}, $i,j=1, \dots, n$, is introduced in order to specify the *orientation* of edge $(ij) \in E$. When variable o_{ij} has value 1, this means that edge (ij) is to be oriented from i to j, thus that the frequency assigned to link j has to be greater than that assigned to link i at least of d_{ij},. When variable o_{ij} has value 0, this means that edge (ij) is to be oriented from j to i, thus that the frequency assigned to link j has to be smaller of that assigned to link i at least of d_{ij}. The formulation is as follows:

$$\text{(OP)} \quad \min \sum_{(ij)\in E} c_{ij}\,\zeta_{ij} \tag{8}$$

$$\text{s.t.} \quad f_j - f_i + \Phi\zeta_{ij} \geq d_{ij}\, o_{ij} - M(1 - o_{ij}) \qquad (ij) \in E \tag{9'}$$

$$f_i - f_j + \Phi\zeta_{ij} \geq d_{ij}\,(1 - o_{ij}) - M\, o_{ij} \qquad (ij) \in E \tag{9''}$$

$$f_i \in \mathfrak{F}_i \qquad i \in V \tag{4}$$

$$o_{ij}, \zeta_{ij} \in \{0,1\} \qquad (ij) \in E \tag{10''}$$

Formulation OP makes use of M, an arbitrary large constant, and transforms nonlinear constraints (9) into constraints (9') and (9''). Constraints (9') are active in case of orientation of edge (ij) from i to j, constraints (9'') in case the edge is oriented the opposite direction. Formulation OP can in turn be relaxed assuming a contiguity of the frequencies in sets $\mathfrak{F}_i$, obtaining formulation LP1.

$$\text{(LP1)} \quad \min \sum_{(ij)\in E} c_{ij}\,\zeta_{ij} \tag{8}$$

$$\text{s.t.} \quad f_j - f_i + \Phi\zeta_{ij} \geq d_{ij}\, o_{ij} - M(1 - o_{ij}) \qquad (ij) \in E \tag{9'}$$

$$f_i - f_j + \Phi\zeta_{ij} \geq d_{ij}\,(1 - o_{ij}) - M\, o_{ij} \qquad (ij) \in E \tag{9''}$$

$$f_i \leq F_i \qquad i \in V \tag{4'}$$

$$0 \leq o_{ij}, \zeta_{ij} \leq 1 \qquad (ij) \in E \tag{10'''}$$

The bound obtained by means of LP1 is very weak, but it is useful to define tentative values of f_i, $i \in V$, and to direct search. A new one is under development ([Maniezzo and Montemanni, 1999]), which is also not very tight, but it is efficient to compute.

Stagnation avoidance

Stagnation denotes the undesirable situation in which all ants repeatedly construct the same solutions, making impossible any further exploration in the search process. This derives from an excessive trail level on the moves of one solution, and it can be observed in advanced phases of the search process, if parameters are not well tuned to the problem.
The stagnation avoidance procedure evaluates each solution against the last *k* ones globally constructed by ANTS. As soon as *k* solutions are available, we compute their moving average $\bar{z}$; each new solution z_{curr} is compared to $\bar{z}$ (and then used to compute the new moving average value). If z_{curr} is lower than $\bar{z}$ the trail level of the last solution's moves is increased, otherwise it is decreased. Formulae (15) and (16) specify how this is implemented.

$$\tau_{ij}(t) = \tau_{ij}(t-1) + \Delta\tau_{ij} \tag{15}$$

where

$$\Delta\tau_{ij} = \tau_0 \cdot \left(1 - \frac{z_{curr} - LB}{\bar{z} - LB}\right) \tag{16}$$

$\bar{z}$ is the average of the last *k* solutions and LB is a lower bound to the optimal problem solution cost. Figure 1 depicts the moving average linear scaling function we used for trail updating.

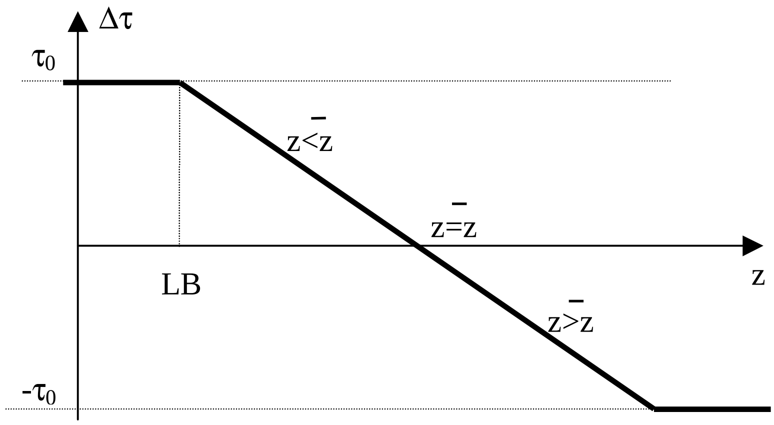

Figure 1. Linear dynamic scaling

The use of a dynamic scaling procedure permits to discriminate small achievement in the latest stage of search, while avoiding to focalize search only around good achievement in the earliest stages.

Based on the described elements, the ANTS metaheuristic is the following.

```
ANTS algorithm
1. (Initialization)
   Compute a (linear) lower bound LB to the problem to solve
   Initialize τ_ιψ, ∀ι,ψ with the primal variable values
2. (Construction)
   For each ant k do
      repeat
         compute η_ιψ ∀ι,ψ, as a lower bound to the cost of completing a solution
         containing ψ.
         choose the state to move into, with probability given by (13)
         append the chosen move to the k-th ant's set tabu_k
      until ant k has completed its solution
      apply a local optimizaton procedure to the solution found
   enddo
3. (Trail update)
   For each ant move (ιψ) do
      compute Δτ_ιψ
      update the trail values by means of (15) and (16)
   enddo
4. (Terminating condition)
   If not(end_test) goto step 2.
```

Figure 2. Pseudo code for the ANTS algorithm

It can be noticed that the general structure of the ANTS algorithm is closely akin to that of a standard tree-search algorithm. At each stage we have in fact a partial solution which is expanded by branching on all possible offspring; a bound is then computed for each offspring, possibly expunging dominated ones, and the current partial solution is selected among that associated to the surviving offspring on the basis of lower bound considerations. By simply adding backtracking and choosing deterministically the node to move to as the best one, we revert to a standard branch and bound procedure (this is in fact the reason behind the use of the denotation ANTS, for Approximate Nondeterministic Tree-Search). An ANTS code can therefore be easily turned into an exact procedure: this possibility falls however outside of the scope of the present paper.

4. Benchmark algorithms

In this Section, we describe two heuristics which have been presented in the literature to compute frequency assignments and that we have used to benchmark the results obtained with the ANTS algorithm. The first is an adaptation to FAP of a well-known heuristic for the coloring problem, called Dsatur, the second is a Tabu Search procedure. Both these algorithms are therefore original proposal, even though they represent simple adaptation of well-known approaches. Moreover, in the computational results section, we will compare the obtained results with those of two further metaheuristics specific for FAP.

Dsatur

Dsatur is a starting heuristic, that is a step-wise procedure extending an initially empty assignment to a complete one. The algorithm we implemented is an adaptation of that proposed by Borndörfer et al. [Borndörfer et al., 1998a].

The original Dsatur proposal is based on a mathematical formulation of the problem slightly different from the one we use, since it considers only interferences arising among links established in the same or in adjacent cells. However, the extension of the approach to the general case we consider is straightforward, and will be presented in the following. For every link $i \in \mathfrak{L}$, a – possibly empty – set $\mathfrak{B}_i \subseteq \mathfrak{F}_i$ of *blocked channels* is specified, where a channel is blocked for link i if it cannot be assigned to link i. The channels in $\mathfrak{F}_i \setminus \mathfrak{B}_i$ are called *available* for link i. A frequency assignment f is *feasible* if every link $i \in \mathfrak{L}$ is assigned to an available channel and all separation requirements are met, i.e., $|f_i - f_j| \geq d_{ij}$ for all (ij). The objective is to determine a feasible assignment that minimizes the total channel interference.

Dsatur makes use of a matrix *cost*, with rows indexed by the links in $\mathfrak{L}$ and columns indexed by the channels in $\mathfrak{F}_i$, to record the costs of the different available combinations. First, the original procedure invalidate all entries corresponding to unavailable combinations of channels by an appropriately chosen entry *BLOCKED*.

A non-blocked channel is *bad* for a link if its matrix entry is at least as large as *BAD*, which is another suitable chosen constant. For every still unassigned link, a heap-entry is maintained, where the key for the heap is the number of blocked or bad channels times *BAD* plus the sum over all non-blocked, non-bad row entries of the matrix cost. That is:

$$key(v) = |B_v| \cdot BAD + \sum_{f \in \mathcal{F}_v \setminus \mathcal{B}_v} h(cost_{v,f}) \quad \text{with } h(c) := \begin{cases} BAD & if \ c \geq \text{BAD} \\ c & \text{otherwise} \end{cases} \tag{17}$$

While the heap is not empty, a link *i* with maximum key is extracted from the heap and it is assigned its least cost available channel, which may also induce separation violations. Next, all rows indexed by links adjacent to *i* and the links' heap keys are updated, considering the costs caused by the frequency assignment.

Our formulation does not force the choice of the frequencies on the set of available channels. As a consequence, the entry *BLOCKED* for a given link and a given frequency of the original procedure is substituted by value 0, and all the other entries in the same row of the *cost* matrix are set equal to the cost produced by the tentatively assigned frequency. In the assignment loop of the algorithm, the actual solution is represented by the channel of least value from each row. The *cost* matrix is updated considering the costs caused by the chosen frequency on the neighbor vertices and the keys of all links in the heap are updated considering the sum over all row entries of the *cost* matrix. The algorithm is as follows.

Adapted DSATUR

```
1.  (Initialization)
    for all i ∈ ℒ do
       if f is in ℱi\ℬi
              set cost[i][f] = 0
       else set cost[i][f] = BLOCKED
       insert i into the heap with key |ℬi|
    enddo

2.  (Assignment)
    extract a link i with maximum key from the heap
    let fi be a non-blocked channel of least value from row cost[i]
    update cost-matrix
    update the keys of all carriers still in the heap

3.  (Terminating condition)
    If the heap is not empty goto 2
```

Figure 3. Pseudo code for Dsatur heuristic procedure

Tabu search

The second heuristic we implemented is a Tabu search (TS) algorithm. TS was first suggested by Glover [Glover, 1989, 1990] and since then it has been successfully applied to obtain high quality solutions for many combinatorial optimization problems, including scheduling, timetabling and travelling salesman.

The general TS framework is as follows. A given instance of an optimization problem, using a cost function *f*, implicitly defines a search space *S* on all feasible solutions. With each solution *s*

$\in S$ we associate a subset *NeighSet(s)* of *S*, called *neighborhood of s*. The neighborhood of *s* contains all solutions which can be obtained with a specified modification of *s*, called a *move*. Different modifications entail different neighborhoods.

The basic idea of the TS method is to explore *S* by a sequence of moves. A move from one solution to another may be the best available, however, to avoid the problem of possible cycling and to allow the search to bypass local optima, TS introduces the notion of *Tabu list*. A tabu list is a special short term memory that maintains a selective history *H*, composed of previously encountered solutions or more generally of pertinent attributes of such solutions. A simple TS consists in preventing solutions of *H* from being reconsidered in the next *k* iterations (length of the tabu list or *tabu tenure*).

The technique may include some *aspiration criteria;* for example, notwithstanding with the fact that a move is tabu, it can be considered if it guides to a solution which is the best one obtained so far.

The Tabu Search we implemented is an adaptation of Hao et al. [Hao et al., 1998]. Considering the FAP problem, for each solution $s \in S$, the cost function *f(s)* corresponds to the total cost of unsatisfied interference constraints. The neighborhood function defines two solutions *s* and *s′* to be neighbors if they are different for the value of a single frequency of a cell. Moreover, it is possible to define a *candidate list strategy* to help search to avoid irrelevant moves and to reduce the number of neighbors to be considered at each iteration. The list is defined as follow:

$$Cset(s) = \{s' \text{ neighbor of } s \text{ and the differing frequency causes an interference in } s\}$$

The length *k* of the tabu list is dynamically adjusted by a function defined over the size of the candidate list *CSet*(*s*). To avoid too large or too small values of *k*, in the case of presence of many conflicting cells and in the late stages of the search, an upper and a lower bounds on its length are imposed, where the two bounds depend on the domain cardinality. Finally, the simple aspiration criterion mentioned is implemented in the procedure.
A description of the Tabu Search we implemented can be found in Figure 4.

```
Tabu Search algorithm

1.  (Initialization)
    Generate an initial solution s∈S

2.  (Main loop)
    identify the candidate list CSet(s) ⊆ NeighSet(s) ⊂ S
    identify the tabu set TSet(s) ⊂ NeighSet(s)
    identify the aspirant set ASet(s) ⊂ Tset(s)
```

```
    choose s' ∈ {(CSet(s) - TSet(s)) ∪ ASet(s)}, for which f(s') is best
    s = s'

3.  (Terminating condition)
    If not (end_test) goto 2
```

Figure 4. Pseudo code for Tabu search heuristic procedure

5. Computational results

In this Section we report the computational results obtained on a number of different test problems drawn from literature: the CELAR, GRAPH and PHILADELPHIA problems. All results have been obtained implementing the algorithms in C and running the codes on a PentiumII 233 MHz machine equipped with 64 Mb of RAM.

The CELAR dataset consists of 11 problems proposed within the framework of EUCLID (European Cooperation for the Long term in Defence) CALMA (Combinatorial ALgorithms for Military Applications) project by the French "Centre d'Electronique de l'Armement" [Tiourine et al., 1995]. They vary in size between 200 and 916 links. The dataset contains problems with hard and soft constraints of the form $|f_i - f_j| > d_{ij}$ and $|f_i - f_j| = d_{ij}$. Six of these instances are interference-free, i.e., they have a frequency assignment that satisfies all constraints, so the best objective function value is equal to 0.

The GRAPH test problems [van Benthem, 1995] are 14 problems patterned after the CELAR problems which exhibit the same structure. For each instance the following data are specified:

- the number of variables;
- for each variable a set of frequencies which may be assigned to the corresponding variable;
- for each variable its initialization value, if any, and its mobility (which states whether the initial value may be modified or not; if yes, then the cost of the modification is defined);
- a set of constraints which must be satisfied when assigning frequencies.

The Philadelphia problems, originally presented in [Anderson, 1973], are among the most studied FAP instances. The problems are based on the area around Philadelphia and consist of cells located in a hexagonal grid as shown in Figure 5.

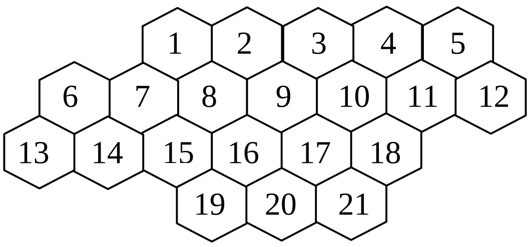

Figure 5. The cellular geometry of Philadelphia problem

The problems have only $|f_i - f_j| > d_{ij}$ constraints. A vector of requirements is used to describe the demand for frequencies in each cell. Transmitters are considered to be located at cell centers and the distance between transmitters in adjacent cells is taken to be 1. Separation distances are specified.

Most of the problems used were originally presented as a minimization of the maximal bandwidth span problem (FAP1). We have adapted it to FAP3 by using an upper bound or the best known solution as Φ, thereby defining a zero-interference instance.

We conducted a number of tests to define which is the best ANTS parameter setting for the FAP. To this end we selected a subset of instances (specifically, problems CELAR 05, 06, 08 and 10 and problem PHILADELPHIA 01) and we tested in a *coeteris paribus* fashion the parameters, on the basis of the predefined ranges $m \in \{n/10, n/40, n/100, n/160, n/200\}$, $k \in \{n/5, n/8, n/10, n/14, n/20\}$, $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The best setting turned out to be $m = n/100$ (number of ants), $k = n/10$ (width of the moving average window) and $\alpha = 0.3$ (importance of trail vs. visibility).

We applied to these problems the ANTS, and our implementation of the Dsatur, the TS algorithms and of two versions of simulated annealing taken from the literature: SA1 [Hurkens and Tiourine, 1995] and SA2 [Smith et al., 1998].

All algorithms have been allowed 20 minutes CPU time on each problem instance, except Dsatur which is a fast constructive heuristic that contains its own termination condition. TS has been applied on the solution initialized by means of the Dsatur procedure.

Table 1 presents the results so obtained, bold cells indicate the best row values. The columns show:

Problem:	problem identifier;
Φ:	maximum span allowed;
Dim. links:	number of variables;
Dim. constraints:	number of constraints;
Dsatur value:	best solution produced by the Dsatur algorithm;
Dsatur time:	CPU seconds used by the Dsatur algorithm to produce its best solution;
TS value:	best solution produced by the TS algorithm;
TS time:	CPU seconds used by the TS algorithm to produce its best solution;
ANTS value:	best solution produced by the ANTS algorithm;
ANTS time:	CPU seconds used by the ANTS algorithm to produce its best solution;

SA1 value: best solution produced by the SA1 algorithm;

SA1 time: CPU seconds used by the SA1 algorithm to produce its best solution;

SA2 value: best solution produced by the SA2 algorithm;

SA2 time: CPU seconds used by the SA2 algorithm to produce its best solution.

PROBLEM	**Φ**	**DIM.**		**DSATUR**		**TS**		**ANTS**		**SA1**		**SA2**	
		Links	Const.	Value	Time	Value	Time	Value	Time	Value	Time	Value	Time
CELAR01	**792**	916	5 548	625	5	238	37	**0**	4	**0**	3	**0**	377
CELAR02	**792**	200	1 235	428	1	122	60	**0**	0	**0**	0	**0**	19
CELAR03	**792**	400	2 760	798	1	196	347	**0**	1	**0**	0	**0**	133
CELAR04	**792**	680	3 967	1 474	3	1 012	57	8	437	**0**	222	1	324
CELAR05	**792**	400	2 598	1 823	1	688	116	32	545	**11**	106	54	82
CELAR06	**792**	200	1 322	213 866	1	149 607	10	**5319**	614	6 994	75	30 160	18
CELAR07	**792**	400	2 865	$>10^8$	1	$>10^8$	38	8083093	630	11 000 296	986	**4 698 907**	412
CELAR08	**792**	916	5 744	2 468	6	1 364	295	709	572	**306**	1 087	457	805
CELAR09	**792**	680	4 103	79 406	3	45 988	6	**16732**	1018	30 024	77	23 634	644
CELAR10	**792**	680	4 103	107 310	3	58 554	9	**31516**	378	31 518	54	33 557	244
CELAR11	**792**	680	4 103	1 364	4	848	417	**0**	628	**0**	83	2	405
GRAPH01	**792**	200	1 134	160	0	15	8	**0**	2	**0**	0	**0**	21
GRAPH02	**792**	400	2 245	299	1	15	17	**0**	1	**0**	1	**0**	79
GRAPH03	**792**	200	1 134	264	0	35	3	**14**	122	**14**	27	96	12
GRAPH04	**792**	400	2 244	519	2	**33**	3	42	1162	64	332	213	6
GRAPH05	**792**	200	1 134	**0**	0	**0**	0	**0**	1	**0**	0	**0**	0
GRAPH06	**792**	400	2 170	**0**	1	**0**	1	**0**	1	**0**	0	**0**	0
GRAPH07	**792**	400	2 170	**0**	0	**0**	0	**0**	1	**0**	0	**0**	0
GRAPH08	**792**	680	3 757	566	3	18	6	**0**	15	**0**	4	**0**	539
GRAPH09	**792**	916	5 246	665	5	14	14	**0**	53	**0**	8	**0**	903
GRAPH10	**792**	680	3 907	856	3	**37**	10	127	1052	91	818	81	1 128
GRAPH11	**792**	680	3 757	**0**	2	**0**	2	**0**	1	**0**	0	**0**	0
GRAPH12	**792**	680	4 017	**0**	3	**0**	3	**0**	1	**0**	0	**0**	0
GRAPH13	**792**	916	5 273	**0**	5	**0**	5	**0**	2	**0**	0	**0**	0
GRAPH14	**792**	916	4 638	750	5	19	44	**0**	2	**0**	3	**0**	763
PHIL01	**179**	420	44 790	7	2	4	14	**0**	29	51	234	263	1129
PHIL02	**239**	420	65 590	**0**	2	**0**	2	**0**	43	17	315	251	1081
PHIL03	**252**	470	56 940	8	3	8	3	**0**	47	31	382	288	954
PHIL04	**257**	470	78 635	6	3	6	3	**0**	53	36	336	353	1144
PHIL05	**426**	481	76 979	**3**	6	**3**	6	30	1041	31	359	252	1063
PHIL06	**426**	481	97 835	18	7	**17**	92	36	177	30	359	288	1139
PHIL07	**426**	481	93 288	**0**	7	**0**	7	29	1180	30	365	271	955
PHIL08	**426**	481	97 835	17	7	**14**	1 197	31	1046	22	472	249	572
PHIL09	**855**	962	783 642	**0**	35	**0**	35	403	830	366	1198	934	1160
PHIL06b	**532**	481	97 835	**1**	8	**1**	8	51	373	18	352	254	772

Table 1. Computational results for ANTS and benchmark algorithms

The computational results show that the ANTS heuristic is competitive with the best approaches so far presented. In particular, Table 1 shows that, apart from ANTS, different approaches are the best performing ones.

The CELAR problems are best solved by the ANTS and the SA1 algorithms, where ANTS and SA1 are able to find the best solution (among those produced by the five tested heuristics) on 7 instances, and SA2 on 4.

The GRAPH have the same structure as the CELAR problems, and in fact the results obtained are similar among the two groups. ANTS and SA1 are able to find the best solution for 12 over 14 problems, SA2 for 10, TS for 7 and DSATUR for 6. In the case of these problems, several instances are comparatively easy, thus all heuristics are able to find a 0 cost solution very quickly.

The PHIL problems are more suited for the DSATUR and TS approaches. In fact, TS is able to find the best solution for 7 instances, DSATUR for 5 and ANTS for 4. The quality of the solution found by ANTS is actually comparable with that of TS, except for problem PHIL09; we must notice however that this instance is much bigger than the other ones and, given the comparatively long time taken by the local optimization alone, the 1200 seconds allowed are not enough to effectively start the trail updating mechanism, thus turning ANTS into a simple multistart approach.

In summary, Table 1 testifies the comparative good performance of the ANTS algorithm. On all test problems, under the imposed computational constraints (CPU time), it found a good solution and exhibited more stable results among those produced by the tested algorithms.

6. Conclusions

The paper presented the application of the ANTS metaheuristic to the radio link frequency assignment problem, with the objective of minimizing the total interference of an assignment plan.

While ANTS has already proved to be effective on problems for which substantial results on lower bounding techniques are available [Maniezzo and Carbonaro, 1999], it was never tested on problems for which these results are not available. This is the case of the problem examined. Despite the weakness of the search guidance that the procedure can exploit, the computational results report a good global performance, thereby testifying the robustness of the approach. Currently, we are working to develop and include a more effective bound in the system to improve the ANTS performance.

References

Aardal K. I., Hipolito A., van Hoesel S. and Jansen B. (1995). A Branch-and-Cut Algorithm for the Frequency Assignment Problem, *Technical Report Annex T-2.2.1 A, CALMA project*, T.U. Eindhoven and T.U. Delft.

Adjakplé P.M., Jaumard B. (1997). Greedy and Tabu Search Heuristics for Channel Block Assignment in Cellular Systems, *Technical Report G-97-45*, GERAD, Montreal.

Adjakplé P. and Jaumard B. A New Class of Lower Bounds on the Quality of Channel Assignment using Column Generation Techniques, to appear in *Les Cahiers du GERAD*.

Anderson L.G. (1973). A simulation study of some dynamic channel assignment algorithms in a high capacity mobile telecommunications system, *IEEE Trans. on Communications*, COM-21:1294-1301.

Borndörfer R., Eisenblätter A., Grötschel M., Martin A. (1998a). Frequency Assignment in Cellular Phone Networks, *Annals of Operational. Research*, 76:73-93.

Borndörfer R., Eisenblätter A., Grötschel M., Martin A. (1998b). The Orientation Model for Frequency Assignment Problems, *Technical Report TR 98-01*, Konrad-Zuse-Zentrum für Informationstechnik Berlin.

Caminada A. (1995). Resolution du Probleme de l'Affectation des Frequences Par Programmation Par Contraintes, *Technical Report FT.CNET/BEL/POH/CDI/71-95/CA, CNET* , Belfort.

Colorni A., Dorigo M., Maniezzo V. (1991). Distributed Optimization by Ant Colonies, *Proc. of ECAL91- European Conference on Artificial Life*, Elsevier Publishing:134–142.

Dantzig G.B., Thapa M.N. (1997). *Linear Programming*. Springer Verlag.

Dorigo M., Colorni A., Maniezzo V. (1991). Positive Feedback as a Search Strategy, *Technical Report TR91-016*, Politecnico di Milano.

Dorigo M. (1992). Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, Learning and Natural Algorithms).Ph.D.Thesis, Politecnico di Milano, Italy, in Italian.

Dorigo M. and Di Caro G. (1999). The Ant Colony Optimization Meta-Heuristic. *IRIDIA Technical Report*, University of Bruxelles (to appear in: Corne, Dorigo and Glover (Eds.), New Ideas in Optimization, McGraw-Hill).

Duque-Anton M., Kunz D. and Rüber B. (1993). Channel Assignment for Cellular Radio Using Simulated Annealing, *IEEE Trans. on Vehicular Technology* 42:14-21.

Fischetti M., Lepschy C., Minerva G., Romanin Jacur G. and Toto E. (1996). Frequency Assignment in Mobile Radio System using Branch-and-Cut Techniques, *Proc. of the AIRO'96 meeting*, Italy:312-314.

Funabiki N. and Takefuji Y. (1992). A Neural Network Parallel Algorithm for Channel Assignment Problems in Cellular Radio Network, *IEEE Trans. on Vehicular Technology* 41:430-437.

Gamst A. (1986). Some Lower Bounds for a Class of Frequency Assignment Problems, *IEEE Trans. on Vehicular Technology* 35:8-14.

Giortzis A.I., Turner L.F. (1996). A Mathematical Programming Approach to the Channel Assignment Problem in Radio Networks, *Proc. IEEE 46 Vehicular Technology Conference*, 2:736-740.

Glover F. (1989). Tabu Search - Part I, *ORSA Journal on Computing*, 1.

Glover F. (1990). Tabu Search - Part II, *ORSA Journal on Computing*, 2.

Hale W.K. (1980). Frequency Assignment: Theory and Applications. *Proceedings of the IEEE*, 68:1497-1514.

Hao J-K., Dorne R. and Galinier P. (1998). Tabu Search for Frequency Assignment in Mobile Radio Networks, *Journal of Heuristics* 4:47-62.

Hurkens C.A.J., Tiourine S.R. (1995). Upper and lower bounding techniques for frequency assignment problems, *Technical Report 95-34*, Department of Mathematics and Computing Science, Eindhoven University of Technology.

Hurley S., Smith D.H. and Thiel S.U.(1997). FASoft: A System for Discrete Channel Frequency Assignment, *Radio Science*, 32(5):1921-1939.

Koster A., van Hoesel S. and Kolen A. (1997). The Partial Constraint Satisfaction Problem: Facets and Lifting Theorems, Technical Report RM 97/005, University of Maastricht.

Maniezzo V. (1998). Exact and Approximate Nondeterministic Tree-search Procedures for the Quadratic Assignment Problem, *Report CSR 98-1*, Computer Science, Univ. of Bologna.

Maniezzo V. and Carbonaro A. (1999). Ant Colony Optimization: an Overview. Proceedings of MIC'99, III Metaheuristics International Conference, Brazil.

Maniezzo V., and Montemanni R. (1999). An exact algorithm for the radio link frequency assignment problem, *Report CSR 99-02,* Computer Science, Univ. of Bologna.

Mannino C. and Sassano A. (1998). An Enumerative Algorithm for the Frequency Assignment Problem. to appear in *Discrete Applied Mathematics*.

Smith D. H. and Hurley S. (1997). Bounds for the frequency assignment problem, *Discrete Mathematics* 167/168:571-582.

Smith D. H., Hurley S. and Thiel S. U. (1998). Improving heuristics for the frequency assignment problem, *European Journal of Operational Research* 107:76-86.

Tcha D. W., Chung Y. J. and Choi T. J. (1997). A New Lower Bound for the Frequency Assignment Problem, *IEEE Trans. on Networking* 5(1):34-39.

Tiourine S., Hurkens C. and Lenstra J. K. (1995). An overview of algorithmic approaches to frequency assignment problem, *Technical Report T.U. Eindhoven*.

van Benthem H.P. (1995). GRAPH: Generating Radio Link Frequency Assignment Problems Heuristically, *Master's thesis,* Fac. of Technical Mathematics and Informatics, T.U. Delft, Netherlands.

Warners J. P., Terlaky T., Roos C. and Jansen B. (1997). A Potential Reduction Approach to the Frequency Assignment Problem, *Discrete Applied Mathematics* 78:251-282.

Zoellner J.A., Beall C.L. (1977). A Breakthrough in Spectrum Conserving Frequency Assignment Technology. *IEEE Transactions on Electromagnetic Compatibility*, 313-319.