Future Generation Computer Systems 16 (2000) 927–935

An ANTS heuristic for the frequency assignment problem

Vittorio Maniezzo*,[1], Antonella Carbonaro[2]

Department of Computer Science, University of Bologna, Via Sacchi 3, 47023 Cesena, Italy

Abstract

The problem considered in this paper consists in assigning frequencies to radio links between base stations and mobile transmitters in order to minimize 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 problem an ANTS metaheuristic, that is, an approach following the ant colony optimization paradigm. Computational results, obtained on a number of standard problem instances, testify the effectiveness of the proposed approach. © 2000 Elsevier Science B.V.

Keywords: Frequency assignment problem; Ant colony optimization; Metaheuristic algorithms

1. Introduction

The introduction of mobile communication, such as portable phones, has had 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.

Ant colony optimization (ACO) is a class of constructive metaheuristic algorithms that take inspiration from real ants behavior. These algorithms have proved successful in dealing with discrete optimization problems, for example with the traveling salesman, the quadratic assignment, the graph coloring and other problems [11,22]. This paper presents its application to one of the main problems arising in mobile telecommunication, namely the *frequency* (*or channel*) *assignment problem* (FAP) [17].

The frequency assignment problem we consider arises 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. Because of the limited frequency spectrum which can be operated by the receivers, overlapping frequencies could be assigned simultaneously to different transmissions, possibly causing interference. The interference 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.

* Corresponding author. Fax: +39-0547-610054.
E-mail addresses: maniezzo@csr.unibo.it (V. Maniezzo), carbonar@csr.unibo.it (A. Carbonaro)
[1] http://www.csr.unibo.it/∼maniezzo.
[2] http://www.csr.unibo.it/∼carbonar.

PII: S0167-739X(00)00046-7

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

The evolution of the telecommunication technology underlying the FAP is reflected in the objective function to optimize. While, during the early 1980s, 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 and to service the requests while trying to minimize the arising interference.

The FAP is a generalization of the graph coloring problem, which 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. In fact, to each instance of the FAP it is possible to associate a weighted interference graph $G=(V,E,W)$, where V is the set of vertices, E the set of edges and W is a weight vector. To each frequency request corresponds a vertex v from V. There is an edge (i, j) between vertices i and j if and only if there exists a minimal distance requirement between frequencies assigned to links corresponding to vertices i and j, and the edges are weighted by means of the required distance. Clearly, no two connected vertices should be labeled with the same frequency.

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 others, lower bounds have been proposed by Adjakplé and Jaumard [3], Gamst [15], Smith and Hurley [24], Tcha et al. [26] and Warners et al. [29]. They are usually derived from graph-theoretic approaches, which adapt techniques originally developed for the coloring problem.

Exact algorithms for solving frequency assignment problems have been proposed by Aardal et al. [1], Fischetti et al. [13], Giortzis and Turner [16], Koster et al. [20], and Mannino and Sassano [23].

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 importance of the FAP, much effort has been spent in studying heuristic algorithms. Different approaches have been used, including greedy heuristics [30], artificial neural networks [14], simulated annealing [12], constraint programming [7], tabu search [2], and adapted Dsatur techniques [5]. Many of these techniques have been included in the FASoft algorithm suite [19]. For a detailed overview on heuristics and exact methods for FAP we refer to [27].

This work reports about the results obtained applying a particular instance of the ACO class to the FAP namely the ANTS metaheuristic [21]. 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 using strong results from mathematical programming, such as lower bounds, dual analysis, dominances and branching schemes, and so on, makes the approach appealing for difficult problems for which substantial mathematical results exist.

The paper is organized as follows. In Section 2, the general structure and the FAP-specific components of the ANTS algorithm are introduced. In Section 3, we discuss the computational results obtained on different test datasets, and finally, in Section 4, we report about the conclusions drawn from our work.

2. 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.

2.1. General framework of the ANTS algorithm

The first ACO metaheuristic has been Ant System, proposed by Colorni et al. [8] and Dorigo et al. [9,10]. The main underlying idea was that of parallelizing search over several constructive computational threads. Each thread is based on a dynamic memory structure incorporating information on the effectiveness of previously obtained results. The work presented in this paper is based on an adaptation [21] 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 similarity to the Ant System approach and the possibility of viewing it, as explained in the following, as an Approximate Nondeterministic Tree-Search procedure.

In general, in ACO algorithms, and in ANTS in particular, 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 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:

1. the *attractiveness* $\eta_{\iota\psi}$ of the move, as computed by some heuristic indicating the a priori desirability of that move;
2. the *pheromone trail level* $\tau_{\iota\psi}$ of the move, indicating how profitable it has been in the past to make that particular move: it represents therefore an a posteriori indication of the desirability of that move.

Pheromone trails are *updated* when all ants have completed the construction of their solution, increasing (decreasing) the level of pheromone trails corresponding to moves that were part of "good" ("bad") solutions.

The previous overview applies to all ACO algorithms. In the following we give a more detailed description for the particular choices made in ANTS.

2.1.1. Computation of attractiveness

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 (ι, ψ), it is possible to compute a 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 bounds for the different problems of interest, good lower bounds are usually available. 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 be completed into a solution better than the current best one. The move can therefore be discarded from further analysis.

A further advantage of lower bounds is that in many cases the values of the decision variables, as appearing 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. For more details see [21].

2.1.2. Move probabilities

One of the most difficult aspects to be considered in metaheuristic algorithms is the trade-off between exploration and exploitation. To obtain good results, a system should prefer actions that it has tried in the past and found to be effective in producing desirable solutions (exploitation); but to discover such actions, it has to try actions that it has not selected before (exploration). Neither exploration nor exploitation can be pursued exclusively without failing at the task: for this reason, the ANTS system integrates a stagnation avoidance procedure to facilitate exploration and a move probability definition mechanism to determine the desirability of different moves.

The specific formula for defining the probability distribution for 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 [21] for a discussion):

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

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:

$$\tau_{\iota\psi}(t) = \tau_{\iota\psi}(t-1) + \Delta\tau_{\iota\psi}, \tag{2}$$

where $\Delta\tau_{\iota\psi}$ represents the sum of the contributions of the ants that used move (ι, ψ) to construct their solution. The ants' contributions are proportional to the quality of the achieved solution, that is, the better an ant's solution, the higher will be the trail contribution added to the moves it used.

2.1.3. Trail update and stagnation avoidance

Stagnation denotes the undesirable situation in which all ants repeatedly construct the same solution, 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.

The stagnation avoidance procedure compares each solution against the k most recently constructed ones. As soon as k solutions are available, we compute in fact the moving average $\bar{z}$ of their costs. When a new solution is constructed, its cost 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. Formula (2) specifies how this is implemented, where

$$\Delta\tau_{\iota\psi} = \tau_0\left(1 - \frac{z_{\text{curr}} - \text{LB}}{\bar{z} - \text{LB}}\right). \tag{3}$$

LB is a lower bound to the optimal problem solution cost (see [21] for more details) and τ_0 is the initial trail level. Fig. 1 depicts the moving average linear scaling function we used for trail updating.

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

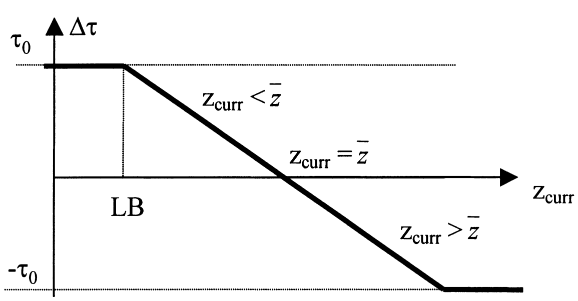

Fig. 1. Linear dynamic scaling.

Based on the described elements, the ANTS metaheuristic is the algorithm presented in Fig. 2.

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 those associated to the surviving offspring on the basis of lower bound considerations. By simply adding backtracking and choosing deterministically the best one as the node to move to, we revert to a standard branch and bound procedure (this is in fact the reason behind the use of the name ANTS, for approximate nondeterministic tree-search). An ANTS code can therefore be easily turned into an exact procedure: this possibility falls, however, outside the scope of the present paper.

2.2. ANTS applied to FAP

The two most important elements to be specified when applying the general ANTS algorithm to a specific combinatorial optimization problem, such as the FAP, are the lower bound to be used to compute the η values and the local search to be used at the end of the loop at step 2.

In the case of the application described in this paper, they were implemented as follows.

The lower bound used is the linear relaxation of the *Orienteering Model* introduced by Borndörfer et al. [6]. The bound so obtained is reportedly very weak, but it is useful to define promising primal frequency assignment values. These, in turn, can be used to initialize pheromone trails, as required at the first step of the ANTS algorithm, simply by defining the trail levels as a function of the optimal primal variable values (see [21] for a more detailed discussion). In our case, we computed the average of the optimal values of all variables appearing in the bound solutions and we initialized all pheromone trails to this value. Moreover, we added to the pheromone trail levels of the variables appearing in the bound solution the value with which they appeared in it, in order to bias search toward solutions compatible with the bound suggestions.

```
ANTS algorithm

1. (Initialization)
   Compute a (linear) lower bound LB to the problem
   Initialize τ_ιψ, ∀ι,ψ (with the bound 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 to, with probability given by equation (1)
        add the chosen move to set tabu_k of the k-th ant
     until ant k has completed its solution
     apply a local optimizaton procedure to the solution found
   endfor

3. (Trail update)
   For each ant move (ι ψ) do
      compute Δτ_ιψ
      update the trail values by means of equations (2) and (3)
   endfor

4. (Terminating condition)
   If not(end_test) goto step 2.
```

Fig. 2. Pseudo-code for the ANTS algorithm.

The local search procedure is a straightforward greedy procedure based on the marginal cost of adding a particular link/frequency assignment to a partial solution. More in detail, the procedure is as follows. First of all, before starting the construction of a solution (i.e., before the “repeat” statement at step 2 of Fig. 2) links are randomly ordered. When a solution is completed and we want to find the corresponding local optimum, we proceed as follows. We scan the links following the imposed order and try to find a different frequency assignment for the current link. For doing that, we test all feasible assignments for the current link and compute the resulting variation of the objective function value. As soon as an improving reassignment is found, the solution is accordingly updated and the process is restarted from the first link of the ordering. The procedure terminates when all the links are checked without finding any improving assignment.

3. 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 Pentium II 233 MHz machine equipped with 64 Mb of RAM.

The CELAR dataset consists of 11 problems made available as part of the international CALMA project.[3] They vary in size between 200 and 916 links. The dataset contains problems with soft and hard constraints, in which the distance which should be respected between frequencies can be a minimal or an exact value, respectively. This particular structure of the instances follows the pattern of a specialization of the FAP named Radio Link FAP (RLFAP) [27]. Six of these instances are interference-free, that is, they have a frequency assignment that satisfies all constraints, so the best objective function value is equal to zero.

The GRAPH test problems [28] comprise 14 instances with the same structure of the CELAR problems. 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 [4], are among the most studied FAP instances. The problems consist of cells located in a hexagonal grid, and have only soft 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. We have adapted them to our objective function by using the best known solution as frequency spectrum, thereby defining zero-interference instances.

[3] http://www.win.tue.nl/math/bs/comb_opt/hurkens/calma.html.

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 *ceteris paribus* fashion the parameters, on the basis of the following values: number of ants $m \in \{n/10, n/40, n/100, n/160, n/200\}$, width of the moving average window $k \in \{n/5, n/8, n/10, n/14, n/20\}$, and importance of trail versus visibility $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The best setting turned out to be $m = n/100$, $k = n/10$ and $\alpha = 0.3$.

We applied to all problems the ANTS algorithm and two versions of simulated annealing: SA1 [18] and SA2 [25], which are two of the best approaches so far presented in the literature.

All algorithms have been run five times for 20 min CPU time on each problem instance. Table 1 reports the best results obtained by these runs. Average results are not reported as they differ only slightly from best values.

The computational results show that the ANTS heuristic, even with few ants, is competitive with SA1 and SA2, being able to find the best solution more often than the two other algorithms. In particular, Table 1 shows that the CELAR problems are best solved by the ANTS and the SA1 algorithms, both able to find the best solution (among those produced by the three tested heuristics) on seven instances, while SA2 can find four best values. On the GRAPH problems, ANTS is able to find the best solution for 13 of the 14 problems, while SA1 and SA2 for 12. In the case of these problems, several instances are comparatively easy, thus all heuristics are able to find a zero cost solution very quickly.

Finally, ANTS is able to find the best solution for 6 over 10 PHILADELPHIA problems, while SA1 for 4 and SA2 for 0.

Moreover, we tested the use of the local optimization procedure alone in a multistart approach, starting from randomly generated solutions, running for 20 min. The results confirm the usefulness of the ant pheromone trail updating mechanism. In fact, ANTS found better results than the multistart approach on 18 of the 35 instances, while the reverse was true for only one instance.

Table 1 shows the comparative good performance of the ANTS algorithm. On all test problems, under the imposed computational constraints (CPU time), it

Table 1
Computational results for ANTS and benchmark algorithms SA1 and SA2, italic cells indicate the best row values[a]

Problem	Φ	DIM.		ANTS		SA1		SA2	
		Links	Const.	Value	Time	Value	Time	Value	Time
CELAR01	792	916	5548	*0*	4	*0*	3	*0*	377
CELAR02	792	200	1235	*0*	0	*0*	0	*0*	19
CELAR03	792	400	2760	*0*	1	*0*	0	*0*	133
CELAR04	792	680	3967	8	437	*0*	222	1	324
CELAR05	792	400	2598	32	545	*11*	106	54	82
CELAR06	792	200	1322	*5319*	614	6994	75	30160	18
CELAR07	792	400	2865	8083093	630	11000296	986	*4698907*	412
CELAR08	792	916	5744	709	572	*306*	1087	457	805
CELAR09	792	680	4103	*16732*	1018	30024	77	23634	644
CELAR10	792	680	4103	*31516*	378	31518	54	33557	244
CELAR11	792	680	4103	*0*	628	*0*	83	2	405
GRAPH01	792	200	1134	*0*	2	*0*	0	*0*	21
GRAPH02	792	400	2245	*0*	1	*0*	1	*0*	79
GRAPH03	792	200	1134	*14*	122	*14*	27	96	12
GRAPH04	792	400	2244	*42*	1162	64	332	213	6
GRAPH05	792	200	1134	*0*	1	*0*	0	*0*	0
GRAPH06	792	400	2170	*0*	1	*0*	0	*0*	0
GRAPH07	792	400	2170	*0*	1	*0*	0	*0*	0
GRAPH08	792	680	3757	*0*	15	*0*	4	*0*	539
GRAPH09	792	916	5246	*0*	53	*0*	8	*0*	903
GRAPH10	792	680	3907	127	1052	91	818	*81*	1 128
GRAPH11	792	680	3757	*0*	1	*0*	0	*0*	0
GRAPH12	792	680	4017	*0*	1	*0*	0	*0*	0
GRAPH13	792	916	5273	*0*	2	*0*	0	*0*	0
GRAPH14	792	916	4638	*0*	2	*0*	3	*0*	763
PHIL01	179	420	44790	*0*	29	51	234	263	1129
PHIL02	239	420	65590	*0*	43	17	315	251	1081
PHIL03	252	470	56940	*0*	47	31	382	288	954
PHIL04	257	470	78635	*0*	53	36	336	353	1144
PHIL05	426	481	76979	*30*	1041	31	359	252	1063
PHIL06	426	481	97835	36	177	*30*	359	288	1139
PHIL07	426	481	93288	*29*	1180	30	365	271	955
PHIL08	426	481	97835	31	1046	*22*	472	249	572
PHIL09	855	962	783642	403	830	*366*	1198	934	1160
PHIL06b	532	481	97835	51	373	*18*	352	254	772

[a] Columns show: the problem name, the maximum frequency span (Φ), the number of variables (*DIM.links*), the number of constraints (*DIM.const*), the best solutions produced by the three different algorithms (*ANTS*, *SA*1 and *SA*2 *value*) over the five runs, and the CPU seconds used by the different algorithms to produce their best solution (*ANTS*, *SA*1 and *SA*2 *time*).

found a good solution and exhibited more stable results among those produced by the tested algorithms.

4. Conclusions

This paper presented the application of the ANTS metaheuristic to the 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 [22], it was never tested on problems for which these results are not available. This is the case of the problem examined. The obtained computational results show 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

[1] K.I. Aardal, A. Hipolito, S. van Hoesel, B. Jansen, 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, The Netherlands, 1995.

[2] P.M. Adjakplé, B. Jaumard, Greedy and tabu search heuristics for channel block assignment in cellular systems, Technical Report G-97-45, GERAD, Montreal, Canada, 1997.

[3] P. Adjakplé, B. Jaumard, A new class of lower bounds on the quality of channel assignment using column generation techniques, Technical Report, GERAD, Montreal, Canada, 1998.

[4] L.G. Anderson, A simulation study of some dynamic channel assignment algorithms in a high capacity mobile telecommunications system, IEEE Trans. Commun. 21 (1973) 1294–1301.

[5] R. Borndörfer, A. Eisenblätter, M. Grötschel, A. Martin, Frequency assignment in cellular phone networks, Ann. Oper. Res. 76 (1998) 73–93.

[6] R. Borndörfer, A. Eisenblätter, M. Grötschel, A. Martin, The orientation model for frequency assignment problems, Technical Report TR 98–01, Konrad–Zuse–Zentrum für Informationstechnik, Berlin, Germany, 1998.

[7] A. Caminada, Resolution du probleme de l'affectation des frequences par programmation par contraintes, Technical Report FT.CNET/BEL/POH/CDI/71-95/CA, CNET Belfort, France, 1995.

[8] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, in: Proceedings of the ECAL91 — European Conference on Artificial Life, Elsevier, Amsterdam, 1991, pp. 134–142.

[9] M. Dorigo, A. Colorni, V. Maniezzo, Positive feedback as a search strategy, Technical Report TR91-016, Politecnico di Milano, Italy, 1991.

[10] M. Dorigo, Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, learning and natural algorithms), Ph.D. Thesis, Politecnico di Milano, Italy, 1992 (in Italian).

[11] M. Dorigo, G. Di Caro, The ant colony optimization meta-heuristic, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw-Hill, London, 1999, pp. 11–32.

[12] M. Duque-Anton, D. Kunz, B. Rüber, Channel assignment for cellular radio using simulated annealing, IEEE Trans. Vehicular Technol. 42 (1993) 14–21.

[13] M. Fischetti, C. Lepschy, G. Minerva, G. Romanin Jacur, E. Toto, Frequency assignment in mobile radio system using branch-and-cut techniques, in: Proceedings of the AIRO'96 Meeting, Italy, 1996, pp. 312–314.

[14] N. Funabiki, Y. Takefuji, A neural network parallel algorithm for channel assignment problems in cellular radio network, IEEE Trans. Vehicular Technol. 41 (1992) 430–437.

[15] A. Gamst, Some lower bounds for a class of frequency assignment problems, IEEE Trans. Vehicular Technol. 35 (1986) 8–14.

[16] A.I. Giortzis, L.F. Turner, A mathematical programming approach to the channel assignment problem in radio networks, in: Proceedings of the IEEE 46 Vehicular Technology Conference, Vol. 2, 1996, pp. 736–740.

[17] W.K. Hale, Frequency assignment: theory and applications, Proceedings of the IEEE, Vol. 68, 1980, pp. 1497–1514.

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

[19] S. Hurley, D.H. Smith, S.U. Thiel, FASoft: a system for discrete channel frequency assignment, Radio Sci. 32 (5) (1997) 1921–1939.

[20] A. Koster, S. van Hoesel, A. Kolen, The partial constraint satisfaction problem: facets and lifting theorems, Technical Report RM 97/005, University of Maastricht, The Netherlands, 1997.

[21] V. Maniezzo, Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, INFORMS J. Comput. 11 (1999) 358–369.

[22] V. Maniezzo, A. Carbonaro, Ant colony optimization: an overview, in: Proceedings of the Third Metaheuristics International Conference, Brazil, 1999.

[23] C. Mannino, A. Sassano, An enumerative algorithm for the frequency assignment problem, Discrete Appl. Math., submitted.

[24] D.H. Smith, S. Hurley, Bounds for the frequency assignment problem, Discrete Math. 167/168 (1997) 571–582.

[25] D.H. Smith, S. Hurley, S.U. Thiel, Improving heuristics for the frequency assignment problem, European J. Oper. Res. 107 (1998) 76–86.

[26] D.W. Tcha, Y.J. Chung, T.J. Choi, A new lower bound for the frequency assignment problem, IEEE Trans. Networking 5 (1) (1997) 34–39.

[27] S. Tiourine, C. Hurkens, J.K. Lenstra, An overview of algorithmic approaches to frequency assignment problems, Technical Report T.U. Eindhoven, The Netherlands, 1995.

[28] H.P. van Benthem, GRAPH: generating radio link frequency assignment problems heuristically, Master's Thesis, Faculty of Technical Mathematics and Informatics, T.U. Delft, The Netherlands, 1995.

[29] J.P. Warners, T. Terlaky, C. Roos, B.A. Jansen, Potential reduction approach to the frequency assignment problem, Discrete Appl. Math. 78 (1997) 251–282.

[30] J.A. Zoellner, C.L. Beall, A breakthrough in spectrum conserving frequency assignment technology, IEEE Trans. Electromagnetic Compatibility (1977) 313–319.

Vittorio Maniezzo was born in Ferrara, Italy, on February 9, 1962. He graduated in Electrical Engineering at the Polytechnic of Milan in 1986 with full grades. After some experiences as a Software Analyst and Consultant, he did his Ph.D. in Computer Science at the Polytechnic of Milan and obtained the degree in 1993. He won a contest for a postdoctoral position at the Institute for Systems, Informatics and Security (formerly ISEI) at the Joint Research Center of the European Commission, in Ispra, Italy, where he remained until he won a contest for a position as researcher at the University of Bologna, where he is currently also tenural Assistant Professor. The research interest of V. Maniezzo are focused on decision support systems, both from the architectural and the algorithmic viewpoints. He contributed to the design and implementation of several running systems, based on heuristic methods for combinatorial optimization. He authored many scientific publications, mainly on international journals.

Antonella Carbonaro received a degree in Computer Science from the University of Bologna, Italy, in 1992. In 1994 she started her Ph.D. studies working specifically on learning techniques to solve computer vision problems. In 1997, she received a Ph.D. degree for her work on Artificial Intelligent Systems at the University of Ancona, Institute of Informatics. From 1997 to 1999 she received a fellowship on Artificial Intelligence, sponsored by Fondazione Cassa dei Risparmi di Cesena, Faculty of Computer Science, University of Bologna. Now, she is a Research Fellow at the Department of Computer Science, University of Bologna, Cesena. She has authored papers in artificial intelligence and her current research interests concern learning and metaheuristic algorithms.