An Improved Tabu Search Algorithm for the Fixed-Spectrum Frequency-Assignment Problem

Roberto Montemanni, Jim N. J. Moon, and Derek H. Smith

***Abstract*—A tabu search algorithm with a dynamic tabu list for the fixed-spectrum frequency-assignment problem is presented. For cellular problems, the algorithm can be combined with an efficient cell reoptimization step. The algorithm is tested on several sets of test problems and compared with existing algorithms of established performance. In particular, it is used to improve some of the best existing assignments for COST 259 benchmarks. These results add support to the claim that the algorithm is the most effective available, at least when solution quality is a more important criterion than solution speed. The algorithm is robust and easy to tune.**

***Index Terms*—Fixed-spectrum problems, radio-frequency assignment, tabu search algorithm.**

I. Introduction

THE importance of the frequency-assignment problem (FAP) for efficient use of the radio spectrum and minimization of interference is now well recognized. The majority of the early work on this problem (see, for example, [1] and [2]) concentrated on the minimum-span problem. In this variation of the FAP, constraints are specified, which, if satisfied, should lead to acceptable interference. It is then necessary to minimize the spectrum used, measured as the difference between the greatest and smallest assigned frequencies. It is now increasingly recognized that the fixed-spectrum frequency-assignment problem (FS-FAP) is of greater importance to network operators. In this variation of the FAP, the spectrum available to the operator is known in advance and is necessary to minimize some measure of interference.

From an academic viewpoint, the availability of good lower bounds for the span [3]–[5] has aided the theoretical study of the minimum-span FAP. These lower bounds typically exceed 90% of the span of the best-known assignment and are frequently tight. The recent availability [6]–[8] of reasonably good lower bounds for fixed-spectrum problems, at least for problems of a moderate size, has now put fixed-spectrum problems in a theoretical position similar to minimum-span problems.

In fixed-spectrum problems, it is not usually possible for all constraints to be satisfied. Early work tended to concentrate on minimizing the number of constraint violations [9]. Normally, this meant that several or many pairs of transmitters had inadequate frequency separation. It is now recognized that the pattern of interference obtained when this approach is used may be quite unsatisfactory. One way to overcome the problem is to minimize a measure of interference based on the signal-to-interference ratio (SIR) at specified reception points [10], [11]. This approach is conceptually attractive as interference is directly minimized and it can take account of multiple interference. However, it is computationally demanding and may restrict the size of the problems that can be handled. Our recent experiences with this approach suggest that, if data structures of similar efficiency are used, run times are increased by a factor of about 20. Thus, for the largest problems presented here (which have up to approximately 2800 transmitters) run times might be measured in weeks. As a consequence, network operators have persevered with binary constraints.

A way of controlling the pattern of interference resulting from residual constraint violations is to attach a *weight* or *penalty* to each constraint [12], [13]. The objective then is to minimize the sum of the weights associated with violated constraints. In this way, the number of constraints violated may be greater, but they tend to be less critical. Weights can be defined at two (or conceivably more) levels, depending on whether a constraint is broken by one channel or more. A formulation with multiple levels is described in [14]. The determination of appropriate weights to use is sometimes somewhat arbitrary. A rational method of determining weights in terms of the expected areas of cells that would have unsatisfactory SIRs, which has been used in the mobile telephone industry, is described in [14].

In the opinion of the authors, there is ample evidence to support the claim that tabu search is the most effective of the algorithms that have been proposed for the frequency-assignment problem, at least when solution quality is a more important criterion than solution speed. This does not mean that there are no problems for which another metaheuristic, such as simulated annealing [15], [2], will not offer a slightly better quality solution than tabu search. It simply means that, taken over a wide range of problem types and problem instances, tabu search most often offers the best-quality solution and always offers a high-quality solution.

In this paper, it will be shown that tabu search can be further improved by a simple scheme for reducing the size of the short-term memory as the algorithm proceeds. The improved quality of the solutions obtained will be demonstrated using several sets of benchmark problems that have appeared in the literature. A further important advantage is that the scheme makes the algorithm more robust by easing the task of parameter tuning when a new class of problems must be solved. This

Manuscript received May 8, 2002; revised October 4, 2002.

R. Montemanni was with University of Glamorgan, Pontypridd, Mid Glamorgan, CF37 1DL, U.K. He is now with Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland. (e-mail: roberto@idsia.ch).

J. N. J. Moon is with the School of Computing, University of Glamorgan, Pontypridd, Mid Glamorgan CF37 1DL, U.K. (e-mail: jnjmoon@glam.ac.uk).

D. H. Smith is with the Division of Mathematics, University of Glamorgan, Pontypridd, Mid Glamorgan CF37 1DL, U.K. (e-mail: dhsmith@glam.ac.uk).

Digital Object Identifier 10.1109/TVT.2003.810976

will be illustrated in this paper. Attention is also given to the data structure and implementation details, which allow the number of iterations that can be completed in a given time to be maximized.

The effectiveness of the algorithms is further demonstrated by applying them to some of the COST 259 benchmarks [13], [16]. This is a major set of real global system for mobile communications (GSM) mobile telephone problems. They use a two-level weighted model that is currently very popular in the industry. The largest problem considered has nearly 2800 transmitters; it should be noted that these weighted problems are more challenging than unweighted problems of a similar size. Several groups of researchers have applied a variety of algorithms to the COST 259 benchmarks [16]. The most successful previous algorithm applied to these problems combines threshold accepting with a cell reoptimization step [13]. The present paper shows that this cell-reoptimization step can be efficiently combined with the dynamic tabu search algorithm. The resulting algorithm offers significantly improved solutions for four of the five COST 259 benchmarks attempted. It is also shown that threshold accepting without cell reoptimization is relatively ineffective on these problems. Thus, it can be inferred that the merits of the algorithm presented here are even more marked for noncellular problems.

In Section II, the role and nature of benchmark problems for the frequency-assignment problem are discussed. Section III outlines the formulation used and Section IV briefly describes some existing solution approaches. In Section V, the tabu search algorithm is described. In Section VI, threshold accepting is described and briefly compared with simulated annealing. An outline of cell reoptimization and its implementation in conjunction with tabu search is given in Section VII. Results comparing the improved tabu search with the original and the tabu search with threshold accepting are given in Section VIII. In Section IX, the relationship between the cost function and the required SIR over the service area is outlined.

II. Benchmark Problems in the Literature

In recent years, the number of papers published that present algorithms for frequency-assignment problems has increased markedly. At the same time, the difficulty of evaluating the claims made for these algorithms has increased. Many are tested on data that is not available to others or use formulations of the problem that are less satisfactory than the best of those now in commercial use. In some cases, the data is not particularly challenging. This has sometimes made the (fairly common) claim that the algorithm presented is “better than all existing algorithms” at best misleading. The application of frequency-assignment algorithms is wider than is sometimes appreciated. The third author of this paper has been involved in work not only in mobile telephone systems, but also in commercial satellite systems and private business-radio assignment, as well as in military trunk radio and combat net-radio systems. Ideally, an algorithm for which general application is claimed should be tested on each of these problem types. Many groups working on frequency assignment have adequate sets of data available, but often the data is supplied confidentially and cannot be made available to others.

There are three main sets of data that have achieved widespread acceptance, the *CELAR* data, the *Philadelphia* problems, and, more recently, the *COST 259* data. All of these benchmarks can be found in [16]. Other problems, in the form of challenges, have been presented from time to time (see, for example, [17]). The CELAR problems (see, for example, [18]) were very widely studied, but most problems have been solved to optimality and can no longer be regarded as sufficiently challenging. They represent a particular military trunk radio-link system, which contained features not present in all such systems. The Philadelphia problems date back to approximately 1973 [19]. The main (minimum-span) versions of these problems were solved to optimality in 1996 (see [2], [20], and [4]). Fixed-spectrum versions are easily devised and many authors continue to use them. However, the challenge they represented is now largely superseded by the availability of weighted fixed-spectrum data for GSM networks made available under the COST 259 project [13], [16]. The COST 259 data is realistic practical data for GSM networks and is typical of the approach used by many operators. It is possible to state some reservations about these benchmarks. The authors are aware of commercial data that is more flexible in the treatment of the difference between broadcast control channel (BCCH) and traffic channel (TCH) carriers, which also allows a penalty for the use of certain channels in place of blocking the channels, as in the COST 259 data. It can be argued that the task of converting the COST 259 data to a form acceptable to most algorithms is more complex than necessary. It should also be noted that these problems do not address the issue of multiple interference or of direct use of C/I calculations in assignment [10] and [11]. However, in general, the wide availability of these benchmark problems represents a major contribution that should not be ignored by those proposing new algorithms. The need for further benchmarks for some of the applications other than cellular telephone networks remains.

In this paper, we will use a variety of data sets that have previously appeared in the literature to assess our proposed revisions to the tabu search algorithm. The final modified algorithm will then be evaluated on certain COST 259 benchmark problems. The emphasis will be on solution quality rather than speed, but some consideration will be given to the tradeoff between quality and speed.

III. The Fixed-Spectrum Frequency-Assignment Problem

The FS-FAP can be represented by a weighted, undirected graph. Formally, it is a five-tuple FS-FAP $= \{V, E, D, P, F\}$ with

- V, vertex set of an undirected graph G. Every vertex represents a transmitter of the original frequency-assignment problem.
- E, set of edges of the undirected graph G. Edges represent those transmitters that are constrained, i.e., pairs of transmitters for which one may potentially interfere with receivers of the other. Edges will be written as $\{v, w\}$ with $v < w$.

- D, set of labels. There is a mapping $E \rightarrow D$ such that $\{v, w\}$ is mapped to $d_{vw} \in \mathcal{N}_0$. d_{vw} is the highest separation between the frequency assigned to the transmitter v and the one assigned to w that may cause the generation of unacceptable interference. If we denote by $f(v)$ the frequency assigned to transmitter v, then if $|f(v) - f(w)| > d_{vw}$, the interference involving the two transmitters is acceptable.
- P, set of labels. There is a mapping $E \rightarrow P$ such that $\{v, w\}$ is mapped to $p_{vw} \in \mathcal{N}_0$. p_{vw} is a cost to be paid if the separation between the frequencies assigned to transmitters v and w is less than or equal to d_{vw}.
- F, set of consecutive frequencies available for every vertex (transmitter) in V (assumed the same $\forall v \in V$).

The objective of the FS-FAP is to find an assignment that minimizes the sum of p_{vw} over all pairs $\{v, w\} \in E$ for which $|f(v) - f(w)| \leq d_{vw}$.

An example of a graph associated with a problem is given in Fig. 1. The model above is easily extended to include the COST 259 problems (see Section VIII-F).

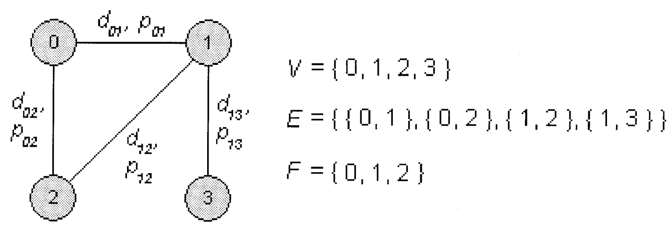


Fig. 1. An example of the FS-FAP.

IV. Algorithms for the Frequency-Assignment Problem

It is not the objective of this paper to give a survey of all the algorithms that have been proposed for the frequency-assignment problem. A recent survey and many additional references can be found in Aardal *et al.* [21]. As a result of the known difficulty of frequency-assignment problems, the majority of the approaches have been heuristic. Greedy and hillclimbing approaches are relatively ineffective.

Simulated annealing is often used as a benchmark against which other algorithms can be compared. However, with a careful implementation and correct choice of parameters, it is already a good algorithm in its own right. Genetic algorithms do not appear to be ideally suited to the frequency-assignment problem. However, they can be made to work well. Similarly, satisfactory results have been presented for artificial neural networks, although they do not appear to have been tested on the most challenging modern data. Ant-colony optimization algorithms have also been demonstrated.

However, taken over a variety of cost functions to be optimized and judged in terms of solution quality alone, it is rare for tabu search to fail to match or improve the quality of assignments. This of course requires that authors have given sufficient information for a practical comparison to be possible. An exception to the above statement has proved to be the results presented for the COST 259 problems by Hellebrandt and Heller (see [13]). In order to match and then improve these results, it has been necessary to enhance tabu search by the use of a dynamic tabu list and to find a way to efficiently combine tabu search with Hellebrandt and Heller's cell-reoptimization step.

It has never been a requirement of tabu search algorithms that the length of the tabu list remains constant. For example, Fleurant and Ferland [22] present a graph-coloring algorithm in which the length of the tabu list varies randomly to diversify the search. However, in almost all tabu search algorithms for the frequency-assignment problem, constant list lengths have been used. An exception is the algorithm of Hao *et al.* [23] for minimum-span problems, where the length of the list is related to the size of the current violating neighborhood. For minimum-span problems, the size of the violating neighborhood reduces until zero cost is achieved and the span then reduces. For fixed-spectrum problems, the behavior is different. In particular, for weighted problems, constraint violations with a high penalty may be replaced by several violations with lower penalties and a small violating neighborhood may never be obtained. Thus, a different approach is necessary. The technique used here ensures that constraints associated with the full range of penalties are addressed by the algorithm and is also effective for unweighted problems.

V. The Tabu Search Algorithm

A. General Description of Tabu Search Algorithms

The Tabu search metaheuristic is a local search algorithm. It was first suggested in Glover [24] (also see Glover *et al.* [25]). The basic idea of the method is to partially explore the search space of all feasible solutions by a sequence of moves. At each iteration, the move carried out is the most promising among those available. A mechanism that forbids a set of moves at each iteration is present, aiming to help the algorithm to escape from local (but not global) minima.

Formally, the main elements of the algorithm are

- Solution representation: each feasible solution to the optimization problem must have a unique representation within the search space;
- Cost function: a function Cost mapping each feasible solution into a value representing its optimization cost. The goal of the algorithm is to find a solution that minimizes this value;
- Neighborhood: a function mapping each feasible solution S into a set of other solutions. Each time the algorithm has to consider a new solution, it is chosen from the neighborhood of the current solution;
- Tabu list: a list containing the last T moves carried out, which, for this reason, are forbidden. A solution obtained from the current solution S with a move contained in the tabu list cannot (in general) be a member of the neighborhood of S. This list is also referred to as the short-term memory or recency.
- Aspiration criterion: if a tabu move (a move that is contained in the tabu list) satisfies this criterion, then the solution obtained by applying it to the current solution S can be considered to be in the neighborhood of S. The usual

```
TabuSearch(Pr) // Pr is an optimisation problem

S := randomly generated solution of Pr;
Best := S;
While(termination criterion not met)
    S := best solution in the neighbourhood of S*;
    If(Cost(S) < Cost(Best))
        Best := S;
    EndIf
    update tabu list;
EndWhile
Return Best;

*the neighbourhood of S does not include solutions obtained using
those moves which are contained in the tabu list and do not sati-
sfy the aspiration criterion.
```

Fig. 2. Tabu search algorithm.

criterion is that the move produces the best solution obtained so far.

- Termination criterion: the algorithm stops when the termination criterion is satisfied.

At each iteration, the algorithm calculates the neighborhood of the current assignment. Solutions generated by using a move contained in the tabu list cannot be in the neighborhood set, unless the respective move satisfies the aspiration criterion. Such aspirate moves are allowed, as they may give a very promising new solution. The solution with the minimum cost among those in the neighborhood becomes the new current solution.

In Fig. 2, pseudocode of the general tabu search algorithm is presented.

B. A Tabu Search Algorithm for the FS-FAP

Tabu search has been used successfully to solve frequency-assignment problems in several papers, including [26], [2], [20], [23], [10], and [14]. It has also been used in several commercial systems and has a strong claim to be the most successful technique for frequency assignment. In this section, we present a tabu search algorithm specifically for the FS-FAP.

1) Solution Representation: The representation of a frequency assignment S is obtained by using a list

$$\langle f_S(0), f_S(1), \ldots, f_S(|V|-1)\rangle$$

where the $v\text{th}$ element $\left(f_S(v)\right)$ contains the frequency assigned to transmitter v.

2) Cost Function: The cost function Cost maps an assignment into the sum of the penalties paid in it. Formally we have

$$\mathrm{Cost}(S) = \sum_{\substack{\{v,w\}\in E; \\ |f_S(v)-f_S(w)|\le d_{vw}}} p_{vw}. \tag{1}$$

3) Neighborhood: Let S_O be the current assignment. A *violating transmitter* is then defined as a transmitter involved in at least one constraint violated in S_O. A new assignment S_N is in the neighborhood of the current solution S_O if S_N differs from S_O in the frequency assigned to exactly one violating transmitter and the move that produces S_N from S_O is not in the tabu list (no aspiration criterion is used; see Section V-B-V). Defining $V_{S_O}^V$ as the set of violating transmitters in the assignment S_O, S_N is a neighbor of S_O if $\exists v \in V_{S_O}^V | f_{S_O}(v) \neq f_{S_N}(v)$ and $f_{S_O}(w) = f_{S_N}(w)\ \forall w \in V, w \neq v$ and the move $(v, f_{S_N}(v))$ is not in the tabu list.

4) Tabu List: In this algorithm, the tabu list contains pairs (v, f) where v is a transmitter and f is a frequency. Each time a move involving the assignment of frequency f to transmitter v is carried out, (v, f) is inserted into the tabu list, where it will remain for approximately T iterations.

Usually, the tabu list has a fixed length T. In this algorithm, the value of T is varied dynamically during the running of the algorithm. This choice has been suggested by some tests that indicated the superiority of the dynamic tabu list over the static one (see Section VIII-D). The best results are achieved by reducing the length of the tabu list in the same way as is done for the temperature parameter in a simulated annealing algorithm. Every I_{ts} iterations the length T of the tabu list is reduced using the assignment

$$T := \beta T \tag{2}$$

where $0 < \beta \leq 1$ is a user-defined parameter. When T is reduced, the oldest moves, which exceed the new length of the list, become feasible. The initial value of T, which we will refer to as T_{init}, is defined by the user. With this dynamic-length tabu list, the algorithm will be referred to as the dynamic tabu search algorithm.

5) Aspiration Criterion: No aspiration criterion is used in the dynamic tabu search algorithm. The use of an aspiration criterion slows down the implementation of the algorithm without improving the results. Experiments supporting this conclusion will be presented in Section VIII-C.

6) Termination Criterion: The algorithm stops when T, the length of the tabu list, becomes smaller than a threshold value $T_{\min}$, which is specified by the user. This termination criterion is quite uncommon for a tabu search algorithm, but here its use is indicated by our strategy of dynamically modifying the parameter T.

7) Implementation Details: The implementation technique adopted is inspired by those described in Fleurant and Ferland [22] and Hao *et al.* [23]. A table of dimension $|V| \times |F|$ is maintained, called the *cost-change table*, where position (v, f) contains the cost of the solution obtained by changing the frequency currently assigned to transmitter v to f (if f is the frequency currently assigned to v, then the value contained in the entry (v, f) is the cost of the current solution). Each time a move is carried out, the elements of the table affected by the move are updated accordingly. For each transmitter v, a list containing its adjacent transmitters (transmitters involved in at least one constraint with v) is adopted. This list is used to speed up the table-updating process. Only the transmitters that are adjacent to the one modified are involved in the updating process. Specifically, only the positions of the table corresponding to the frequencies that interfere with the old

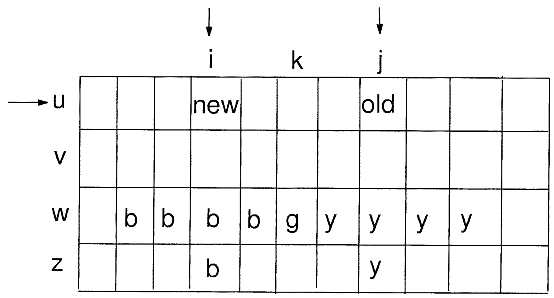


Fig. 3. Updating the cost-change table.

or new frequency of the reassigned transmitter are modified. An example of the use of the cost-change table is given in Fig. 3. An interval of columns of the rows corresponding to four transmitters, u, v, w and z, appears in the figure. In the graph representing the problem, u and v are not connected, u and w are connected by an edge with separation $d_{uw} = 2$ and u, and z are connected by an edge with separation $d_{uz} = 0$. The figure depicts the effects that derive from the modification of the frequency assigned to transmitter u from j (old) to i (new). In the figure, the table entries that are modified due to the frequency reassignment have been marked. The entries that increase their values (of p_{uw} and p_{uz} for w and z, respectively) have been marked "b." The entries that have their values decreased have been marked "y."Notice that the entry (w, k) (marked "g") is not modified. As transmitter v is not connected with transmitter u, its row is not affected by the frequency reassignment.

There is also a second table of dimension $|V| \times |F|$, which contains in each position (v, f), the last iteration number in which the respective move (assignment of frequency f to transmitter v) has been carried out. This table is used to quickly check whether or not a candidate move is tabu: if the iteration number stored in the cell is greater than or equal to the current iteration number minus T, then the move is tabu. A list is also maintained, which indicates, for each transmitter, the frequency (different from the one currently assigned) that, without generating a tabu move, would produce the lowest cost if assigned to it. This list is used at each iteration to quickly select the best move in the active neighborhood. After each iteration, this list is updated efficiently by modifying only the entries affected by the last move. A one-dimensional array of length $|T_{\text{init}}|$ that contains the conventional tabu list is also stored. It is used at each iteration to quickly identify the move that exits from the list. (It is used to efficiently update other data structures.)

Initializing the structures has a computational complexity of $O(|V|^2|F|)$. After each move, the tables can be updated with a theoretical complexity of $O(|V||F|)$. In practice, the updating process is extremely fast.

It should be noted that neither the cost-change table nor the table implementation of the tabu list is universally applicable. For example, in certain frequency-assignment problems for satellites, the frequencies can be tuned very finely. Thus, the number of channels is very large in comparison to a mobile telephone problem. The memory requirements become prohibitive and it is necessary to revert to simpler techniques using random violating neighborhoods [2].

```
ThreshAccept(Pr) // Pr is a freq. assignment
problem

Initialise threshold
S := randomly generated solution of Pr;
While(termination criterion not met)
    For I_ta iterations do;
        Select random move M
        Apply M to S giving assignment S_M;
        If(Cost(S_M) - Cost(S) < threshold)
            S := S_M;
        EndIf
    Reduce threshold;
EndWhile
Return S;
```

Fig. 4. Threshold-accepting algorithm.

C. Advantages of Recency Reduction

The improved solution quality due to the recency-reduction technique (i.e., reducing the length of the tabu list as described in Section V-B-IV) will be seen in Section VIII-D. However, there is a further advantage. When used in conjunction with a full-violating neighborhood in place of a random-violating neighborhood (see [14]), there are essentially only two parameters, T_{init} and β, to set. (Note that the value of I_{ts} does not appear to be critical; values between 10^3 and $5 * 10^4$ have been used. The choice of some small value for $T_{\min}$ is also not critical.) The initial length T_{init} of the tabu length can be set conservatively large if a random starting assignment is used. The recency-reduction parameter β can be adjusted to allow a tradeoff between speed and solution quality. Values of β with $0.999 < \beta < 0.9999$, for example, may give reasonable-quality solutions quickly. If β is closer to 1, better quality solutions may be obtained, but the required run time becomes longer. Thus, the tabu search algorithm becomes very easy to tune for a new class of data and is easily adjusted for ultimate quality or to suit the available run time. This contrasts with some previous implementations where correct tuning of several parameters was necessary for good results.

Note that if tabu search is used to improve an already-good assignment, more care is necessary to avoid setting T_{init} too large, which would lose the quality of the initial assignment very quickly.

VI. The Threshold-Accepting Algorithm

The best results previously available for the COST 259 benchmarks used a threshold-accepting algorithm [16], [13]. For this reason, the algorithm is described here and has been implemented in the current work for comparison. In Fig. 4, pseudocode of the threshold-accepting algorithm is presented. Moves are generated randomly by selecting a transmitter and then selecting an available frequency. If the move reduces cost or increases cost by less than the current threshold, the move is accepted. After I_{ta} iterations, the threshold is reduced using an assignment of the form

$$\text{threshold} := \beta_{ta} \times \text{threshold} \tag{3}$$

similar to that used for recency reduction in our tabu search algorithm. Clearly, threshold accepting is a simple algorithm and has advantages similar to those described for tabu search in Section V-C.

Threshold accepting can be considered as a variant of simulated annealing with a deterministic acceptance criterion. In simulated annealing, the acceptance criterion for a random move is to accept it with probability $e^{-\Delta/t}$ where $\Delta = \text{Cost}(S_M) - \text{Cost}(S)$ and t is a temperature parameter that reduces as the run proceeds in a way similar to the recency in tabu search and the threshold in threshold accepting.

Taken over a wide variety of frequency-assignment problems, it appears that simulated annealing outperforms tabu search in a small minority of instances, whereas, for most instances, tabu search gives the best results. The results in [2] for minimum-span problems are typical. It appears that a similar statement holds if "simulated annealing" is replaced by "threshold accepting." However, Hellebrandt and Heller (see [13]) have shown that, for cellular problems of a GSM type, threshold accepting can be interleaved with a cell-reoptimization step to obtain significantly better results. In the next section, it will be shown how this can be done for tabu search in a somewhat different way, using the tabu search data structures for efficient optimization. The result will be that tabu search is significantly better than threshold accepting when cell reoptimization is not used (for example, in noncellular problems), but somewhat better than the Hellebrandt and Heller results for COST 259 data. These statements are made on the assumption that, when necessary, run times of hours or days are available in order to obtain the best possible assignment quality. If very short run times are used for larger problems, metaheuristic algorithms are best treated as an improvement phase to a fast sequential algorithm. The authors have no evidence to suggest that similar statements do not then apply.

VII. Cell Reoptimization

Cell reoptimization is an appropriate technique only in cellular networks. In such a network, there are a small number of transmitters (say, 1–5) serving each cell. The constraints involving the frequency assigned to each transmitter in the cell may be identical, with the exception of the so-called BCCH carrier. This control carrier is of critical importance and the frequency separations or penalties involving this carrier may sometimes be higher.

The basic idea of cell reoptimization is to fix the frequencies assigned to all cells except for one selected cell. The assignment of this one cell can then be optimized by an exact method. In our implementation (in conjunction with tabu search), each cell is optimized exactly once, in sequential order, after every N_{opt} tabu search iterations. The exact choice of N_{opt} does not appear to be critical; experiments have been carried out with N_{opt} between 100 and 100 000. Some results in Section VIII indicate that it is probably best to avoid smaller values from this range. A constraint of great importance is defined to be *hard*. One way to implement it is to give it a very high weight. Constraints involving transmitters in the same cell or at the same site are normally hard. Cell reoptimization is only carried out when there are currently no violations of hard constraints (see Section III).

```
Optimise_Cells(S) // S is a current assignment

For cell_no.:= 0 to (no. of cells - 1)
    Fix assignment in S of all but current cell;
    For all transmitters in current cell
        Calculate cost of all moves respecting
            external hard constraints;
                //Obtained from cost change table
        Adjust costs for internal hard constraint violations;
    M:= Lowest cost combination of moves
            respecting internal hard constraints;
    If M reduces cost
        For each move in M;
            make move;
            update tabu list;
            update cost change table;
            S:= assignment resulting from move in M;
    EndIf
Return S;
```

Fig. 5. Cell reoptimization.

The cost-change table described in Section V-B-VII can be used to carry out cell reoptimization efficiently. The overall cost change of any transmitter reassignment in the selected cell can be read directly from the cost-change table. From the current assignment of the selected cell, any constraint violations with transmitters in this cell caused by this reassignment can be determined. These two results can then be combined to determine, for each transmitter in the selected cell, the cost of every change of frequency that does not violate any hard constraint with a transmitter in another cell. A tree can be built with each branch corresponding to a reassignment that does not violate any hard constraint (except perhaps with transmitters in the current cell, which are not yet reassigned). The leaf nodes correspond to new assignments of the cell and also store the cost of the assignment.

A recursive depth-first search of this tree (incorporating the above calculations) is used to find the smallest cost reassignment of the selected cell. The moves are then made one at a time, adding them to the tabu list and updating the cost-change table, exactly as is done within the tabu search itself. Note that the non-BCCH transmitters are essentially identical. Speed of computation is increased if an order is chosen for them and they are assigned in ascending order of frequency. Pseudocode for the dynamic tabu search algorithm incorporating cell reoptimization is presented in Figs. 5 and 6.

VIII. Results

In this section, after a brief description of the benchmarks adopted, results obtained by the algorithm described in Section V-B (without cell reoptimization) will be presented. These will illustrate the effectiveness of the dynamic length tabu list within the tabu search algorithm, support the decision not to use the aspiration criterion, and include a comparison with methods proposed by other authors. Results are also presented for a small number of COST 259 problems, with and without cell reoptimization. Most of the tests have been carried out on a computer with an Intel Pentium II 400-MHz processor equipped

```
TS_C_Opt(Pr) // Pr is a FS-FAP

N_it := 0; //iteration number
T := T_init; //initial length of tabu list
N_opt =no. iterations between optimizations;
S := randomly generated solution of Pr;
Best := S;
While(termination criterion not met)
    If N_it mod N_opt = N_opt - 1
        If (all hard constraints satisfied)
            Optimise_Cells(S);
            If(Cost(S) < Cost(Best))
                Best := S;
            EndIf
        EndIf
    Else
        S := best solution in neighbourhood of S*;
        If(Cost(S) < Cost(Best))
            Best := S;
        EndIf
        update tabu list;
        update cost change table;
        EndIf
    EndIf
    N_it := N_it + 1;
    If (I_ts divides N_it)
        T := β × T;
    EndIf
EndWhile
Return Best;
*the neighbourhood of S does not include solutions obtained using
 those moves which are contained in the tabu list.
```

Fig. 6. Tabu search algorithm for the FS-FAP, with violating neighborhood, dynamic tabu list and cell reoptimization incorporated.

with 128 MB of memory, with the exception of Table I and some of the long runs for the COST 259 problems. Results in Table I are obtained on a computer equipped with an Intel Pentium MMX 266-MHz processor and 48 MB of memory. For long runs of COST 259 problems, an Intel Pentium 4 1.5-GHz processor with 512 MB of memory has been used.

The run time of the dynamic-length tabu list algorithm is controlled by the parameter β. When comparing this algorithm with a conventional tabu search algorithm, care is taken in the choice of β to ensure that the run time of the conventional algorithm is not exceeded. Again, when comparing the results with algorithms of other authors, care is taken in the choice of β to ensure that run times are comparable. For the COST 259 problems, the best result obtained is given in the spirit of the results presented in [16]; the best available cost is quoted, independent of runtime. In fact, for most results presented in [16], the run time is not known.

A. Description of the Benchmarks Adopted

There are three different sets of problems. The problems in the first set were originally minimum-span problems. They have been converted into FS-FAP by fixing $p_{vw} = 1\ \forall\{v, w\}$. Specifically

- AC-x-y: scenarios derived from a binary constraint representation of area-coverage problems (see Watkins *et al.* [27]). x is the number of transmitters and y the required SIR;
- GSM-x: realistic GSM scenarios. x is the number of transmitters in the network;
- Test$x$: constraint graphs generated by Cardiff University (see Castellino *et al.* [9] and Smith *et al.* [20]). Again, x is the number of transmitters in the network;
- P06-z: subproblems of the well-known Philadelphia problem, originally proposed in Anderson [19] (see also Smith *et al.* [20]). The generic graph P06-z is obtained by considering for every cell i of the problem a demand of $\lceil m(i)/z \rceil$, where $m(i)$ is the original demand for cell i.
- P06b-z: graphs obtained from the Philadelphia problem with the same method described for P06-z, but with a co-cell separation of three instead of the original five. This has been done to more closely match the characteristics of realistic modern frequency-assignment problems.

The second family of scenarios contain weighted constraints:

- GSM2-x: adaptation to our model of realistic GSM scenarios. x is the number of transmitters in the network.

The third set is selected from the COST 259 benchmarks in [16].

B. Parameter Settings

The algorithms are naturally fast because of the efficient implementation, so quite-conservative values could be chosen for the parameters, giving priority to a careful search instead of the convergence speed. Execution times are under 45 min. anyway for all of the problems in the first two sets and under 10 min. for most problems in these sets. On the other hand, the larger and more-complex COST 259 problems benefit from much longer run times.

In detail, we have the following settings for the first two sets. For all of the problems $T_{\min} = 10$, $\beta = 0.96$ and $I_{ts} = 5 * 10^4$. $T_{\text{init}} = 500$ for the problems based on *AC-45–17*, *AC-45–25*, *AC-95–9*; $T_{\text{init}} = 1000$ for the problems based on AC-95–17, GSM-93, GSM-246, and Test95 and on the graphs of the second set. Finally, $T_{\text{init}} = 2000$ for the remaining problems (based on Test282, P06–5, P06–3, P06b-5, and P06b-3). Parameter settings for the COST 259 problems are given in Table IV.

C. Ineffectiveness of an Aspiration Criterion

Before presenting our main results, evidence will be presented in Table I to support the decision noted in Section V-B-V to not use an aspiration criterion.

The meanings of the columns of Table I are

- *Problem instance* contains the names of the problems;
- *Best found* contains the best result found by the dynamic tabu search algorithm with and without an aspiration criterion (AC and No AC, respectively) over five runs;
- *Avg. result* contains the average results obtained;
- *Avg. time to best* contains the average time at which the best result was found during the run, taken over five runs;
- *Avg. total time* contains the average of the total computation time, taken over the five runs.

All times are in seconds.

TABLE I
ASPIRATION CRITERION IS NOT USEFUL

Problem instance		Best found		Avg result		Avg. time to best		Avg. total time	
Graph	$\|F\|$	AC	No AC	AC	No AC	AC	No AC	AC	No AC
AC-95-17	15	33	33	33.8	33.8	782.2	589.9	1155.1	873.0
GSM-93	9	32	32	33.4	33.2	437.8	298.1	710.4	474.1
Test282	71	31	31	32.6	32.0	2327.4	1764.1	4682.8	3463.9
P06b-3	31	112	112	112.2	112.8	768.3	582.6	3598.5	2832.0
GSM2-184	39	5760	5547	5814.6	5829.4	1348.1	1138.2	2563.3	1871.1

TABLE II
EFFECTIVENESS OF THE DYNAMIC-LENGTH TABU LIST

Problem instance		DTS	CTS					
Name	$\|F\|$		T_1	Val_1	T_2	Val_2	T_3	Val_3
AC-45-17	7	32	450	36	100	32	30	32
AC-45-17	9	15	450	20	100	15	30	15
AC-45-25	11	33	450	33	140	33	30	34
AC-45-25	19	8	450	8	140	8	30	8
AC-95-9	6	31	450	36	140	31	30	31
AC-95-9	10	3	450	3	140	3	30	3
AC-95-17	15	33	450	40	100	34	30	34
AC-95-17	21	10	450	13	100	10	30	10
GSM-93	9	32	900	44	100	33	50	33
GSM-93	13	7	900	13	100	7	50	7
GSM-246	21	79	900	94	500	87	100	100
GSM-246	31	25	900	36	500	33	100	34
Test95	31	12	900	15	300	12	100	12
Test95	36	8	900	10	300	8	100	8
Test282	61	51	1800	83	400	63	100	68
Test282	71	27	1800	52	400	36	100	40
P06-5	11	133	1800	161	400	133	100	161
P06-5	41	15	1800	15	400	15	100	21
P06-3	31	115	1800	121	400	123	100	144
P06-3	71	26	1800	26	400	29	100	31
P06b-5	21	52	1800	52	400	52	100	61
P06b-5	31	25	1800	25	400	25	100	27
P06b-3	31	112	1800	117	300	117	100	117
P06b-3	71	26	1800	26	300	26	100	26
GSM2-184	39	5521	900	6806	300	5881	100	5896
GSM2-184	49	999	900	1054	300	874	100	999
GSM2-227	39	10979	900	13325	500	11561	100	12996
GSM2-227	49	2459	900	3322	300	2517	100	2961
GSM2-272	39	27416	900	30775	500	29481	100	39506
GSM2-272	49	7785	900	8877	500	8776	100	10165

Table I strongly suggests that it is more convenient to not use the aspiration criterion inside the dynamic tabu search algorithm. The results obtained using the aspiration criterion are almost the same as those obtained without it, but the computation times are much longer when the aspiration criterion is used. This is because some extra data structures are necessary in order to use the aspiration criterion. The overhead of updating them is not negligible, even with a good implementation.

D. *Effectiveness of the Dynamic-Length Tabu List in the Tabu Search Algorithm*

In this section, we compare the results obtained by our tabu search algorithm (which incorporates a dynamic length tabu list) with the results obtained by a tabu search algorithm with a fixed length tabu list (i.e., $\beta = 1$), which we will refer to as the *conventional tabu search algorithm*.

The implementation of the conventional algorithm is the same as described in Section V-B-VII except for the exit criterion, which, in this case, is a maximum computation time of 45 min. For each problem considered, this time is longer than the time required by the dynamic tabu search algorithm.

In Table II, the results achieved by the conventional tabu search are compared with those obtained by the dynamic tabu search. Three different values for the length of the conventional tabu list have been considered for each problem; for each of these values, the best result achieved in five runs is presented in the table. The columns of the table have the following meaning:

- Problem instance: each instance is composed of the following two elements:
  — name of the problem,
  — $|F|$: number of channels available;
- DTS: best results obtained by the dynamic tabu search algorithm;
- CTS: best results obtained by the conventional tabu search algorithm. Subcolumns have the following meaning:
  — T_i: length of the tabu list,
  — Val_i: best upper bound obtained with a tabu list of length T_i.

The advantage arising from the use of the dynamic-length tabu list is clear from Table II. The dynamic tabu search algorithm obtains a worse result than the conventional tabu search algorithm in only one case (problem GSM2–184 with $|F| = 49$).

E. *Comparison With Algorithms of Other Authors*

In this section, we compare our tabu search algorithm with two programs developed by other authors.

The first program considered is FASOFT (Hurley *et al.* [2]). It treats only problems in which the goal is to minimize the number of constraint violations, so we cannot run it on the second family of benchmarks. As stated in Section I, FASOFT contains more than one algorithm. For the tests reported here, the tabu search algorithm is used, which seems to be the best of the collection. It was used within a minimum-span algorithm to solve the Philadelphia benchmarks to optimality (see [2]).

The second algorithm considered here is the tabu search algorithm NBS, developed by Whitaker *et al.* [28]. The algorithm

TABLE III
COMPARISON WITH RESULTS OF OTHER AUTHORS

Problem instance Graph	$\|F\|$	Tabu search	FASOFT	NBS
AC-45-17	7	32	33	32
AC-45-17	9	15	16	16
AC-45-25	11	33	34	33
AC-45-25	19	8	9	8
AC-95-9	6	31	31	31
AC-95-9	10	3	3	3
AC-95-17	15	33	36	36
AC-95-17	21	10	11	10
GSM-93	9	32	34	39
GSM-93	13	7	9	9
GSM-246	21	79	89	93
GSM-246	31	25	35	36
Test95	31	12	15	13
Test95	36	8	10	9
Test282	61	51	75	70
Test282	71	27	45	42
P06-5	11	133	137	144
P06-5	41	15	16	15
P06-3	31	115	123	132
P06-3	71	26	28	29
P06b-5	21	52	52	53
P06b-5	31	25	25	25
P06b-3	31	112	113	116
P06b-3	71	26	26	26

was developed to deal with binary and nonbinary constraints, although this last type of constraint is not present in the current problems. Thus, its functionality is not fully utilized here. Unfortunately, the method is unable to handle problems of the second benchmarks family because their penalties are too high. Consequently, NBS has been run only on the problems of the first family.

The methods have been tuned in such a way as to have computation times similar to those of the algorithms described in this paper.

In Table III, the upper bounds produced by our algorithms are compared with those provided by FASOFT and NBS. For each of the problems, the three methods are run at least five times and the best result for each method is taken. The columns of the tables have the following meaning:

- Problem instance: each instance is composed of the following two elements:
  — Name of the problem,
  — $|F|$: number of channels available;
- Tabu search: best results obtained by the tabu search algorithm described in Section V-B;
- FASOFT: best results obtained by the tabu search algorithm contained in the system FASOFT;
- NBS: best results obtained by the tabu search algorithm NBS.

On the benchmarks analyzed in Table III, both FASOFT and NBS are clearly dominated by our algorithm.

F. Tests on COST 259 Problems

In this section, our algorithm is evaluated on the widely available COST 259 benchmarks. The formulation is typical of that now used in the mobile telephone industry. The dynamic tabu search algorithm, with and without cell reoptimization, is compared with threshold accepting (without cell reoptimization) and the best results previously known (which use threshold accepting with cell reoptimization). The modifications necessary to handle the COST 259 problems will now be indicated. Some constraints are hard and must be satisfied. This is handled by giving them a large penalty, much larger than the likely final solution (10 000 is used here). Other constraints with $d_{vw} = 1$ have two different levels of penalty, with the higher penalty being applied if v and w are assigned the same frequency. Penalties need not be integers. Finally, some frequencies may not be used and the unusable frequencies may vary from transmitter to transmitter. Thus, the set F of consecutive frequencies available for every vertex (transmitter) is replaced by a set $F = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{|V|}\}$ of frequency domains. Vertex i must be assigned a frequency in $\mathcal{F}_i$.

The problems Siemens 2, Siemens 3, Siemens 4, K, and Swisscom are all cellular GSM problems and have 977, 1623, 2785, 267, and 310 transmitters, respectively. It should be noted that K only has three cells with more than one transmitter, so cell reoptimization cannot be expected to improve the dynamic tabu search for this problem. Results are given in Table IV. The emphasis is on ultimate quality, so all runs take several days. However, it must be emphasized that results almost as good can be achieved in a few hours if slightly smaller values of β are chosen. Thus, for example, a cost of below 97 can be achieved for Siemens 4 in less than one hour. It should also be noted that a relatively poor algorithm will not find results of this quality in any realistic run time.

The columns of the table have the following meaning:

- Problem name;
- Threshold accepting: A single result using threshold accepting without cell reoptimization using the parameters given in the row below;
- Tabu search: A single result using tabu search without cell reoptimization using the parameters given in the row below;
- Tabu search with cell optimization (two columns): In each case, a single result using tabu search with cell reoptimization, using the parameters given in the row below;
- Tabu search with cell optimization (best): The best result from one or more runs using tabu search with cell reoptimization, using the parameters given in the row below. Sometimes a final improvement phase (typically affecting the second decimal place) has been made using one or two move random hillclimbing or low-threshold accepting. When β is very close to 1, this phase can be used to terminate the run quickly, without waiting for the tabu list to reduce fully;
- Previous best: The best result for these problems given in [16]. In all cases except Swisscom, this is obtained using the algorithm developed by Hellebrandt and Heller (see [13]). The results of column 6 either significantly improve this previous best or are close to it. Even the results of columns 3 and 4 (obtained with slightly smaller values of β and, therefore, are faster) are better than the results quoted at [16] for the algorithms of other authors.

TABLE IV
RESULTS FOR SOME COST 259 PROBLEMS

Problem Name	Threshold Accepting	Tabu Search	Tabu Search with Cell Optimisation	Tabu Search with Cell Optimisation	Tabu Search with Cell Optimisation (best)	Previous best
Parameters	$I_{\text{ta}} = 10^4$ $\beta_{\text{ta}} = 0.99995$	$I_{\text{ts}} = 10^3$ $\beta = 0.99999$	$I_{\text{ts}} = 10^3$ $\beta = 0.99999$ $N_{\text{opt}} = 10^4$	$I_{\text{ts}} = 10^3$ $\beta = 0.99995$ $N_{\text{opt}} = 10^2$	$I_{\text{ts}} = 10^3$ $1 > \beta \geq 0.99999$ $N_{\text{opt}} = 10^4$ (typically)	
Siemens 2	15.56	14.84	14.49	14.81	14.275	14.75
Siemens 3	6.37	5.68	5.60	6.28	5.186	5.26
Siemens 4	90.6	85.25	84.43	95.09	81.876	80.97
K	0.52	0.461	-	-	0.447	0.458
Swisscom	28.21	27.57	27.21	27.40	27.211	27.36

The actual assignments obtained have been added to the collection of available assignments in [16].

1) Parameter Sensitivity and Robustness: The choice of the parameter N_{opt} has already been discussed and is not critical. Similarly, I_{ts} is not critical, although β must be chosen with respect to the chosen value for I_{ts}. The initial length of the tabu list T_{init} is important. For the COST 259 problems, it can be observed that too small a value may lead to a failure to solve the hard constraints and too high a value simply wastes time early in the algorithm. The best choice is a value that is just large enough for the hard constraints to be solved quickly. The choice of 0.3 $\times$(the total number of moves possible in the problem) has been found to be successful.

In order to illustrate the robustness of the algorithm, the variation with different random starting assignments and sensitivity to the choice of the parameter β, some further results (with smaller β) are presented for Siemens 3 in Table V.

Comparing Table V with the Siemens 3 results from Table IV, the relationship between solution quality and the value of β chosen is clear. There is also some limited variation in cost, depending on the random starting assignment used. For small β, the best of several (relatively short) runs should be taken, but a single long run with a value of β closer to 1 is generally a better strategy.

TABLE V
FURTHER RESULTS FOR *SIEMENS 3* USING TABU SEARCH WITH CELL OPTIMISATION WITH SMALLER β

Random start	$\beta = 0.999$	$\beta = 0.9999$
random_start0	7.00	6.00
random_start1	7.06	5.84
random_start2	6.83	6.02
random_start3	7.00	6.12
random_start4	6.92	5.94
random_start5	6.94	6.05
random_start6	6.85	6.22
random_start7	6.87	6.42
random_start8	6.99	6.08
random_start9	7.05	5.82

IX. RELATIONSHIP BETWEEN COST FUNCTION AND THE SIR THRESHOLD

It is useful to consider the relationship between the value of the cost function minimized and the pattern of the SIR over the service area of the network. Such an analysis is usually only possible for simulated problem data. Real benchmark data rarely contains necessary information such as transmitter powers and directions and knowledge of the precise location of the network, which would allow the correct area of a terrain database to be selected.

However, if we assume that the weights are generated using the rational method mentioned in the introduction, something can be said without this information. This method of weight generation is described in more detail in [14]. The most important constraints are hard (or are given a very large penalty) and are, therefore, satisfied in an optimized assignment. If transmitters i and j have a cochannel constraint, then the appropriate weight is the expected fraction of the area of the cell served by transmitter i, which has an unsatisfactory SIR as a result of interference from transmitter j plus the expected fraction of the area of the cell served by transmitter j, which has an unsatisfactory SIR as a result of interference from transmitter i if i and j are assigned the same frequency. Similarly, if transmitters i and j have an adjacent channel constraint, then the appropriate first-level weight is defined in the same way. A second-level weight is the expected fraction of the area of the cell served by transmitter i, which has an unsatisfactory SIR as a result of interference from transmitter j plus the expected fraction of the area of the cell served by transmitter j, which has an unsatisfactory SIR as a result of interference from transmitter i if i and $j$are assigned adjacent frequencies.

When the cost function is high, it might be the case that certain regions of cells have an unsatisfactory SIR as a result of more than one interferer (considered individually). When the cost function is low, this effect reduces or disappears. Similarly, multiple interference is ignored by this formulation. However, broadly, it can be said that when the cost function is small, a given further percentage reduction in cost results in almost the same percentage reduction in the area served that has inadequate SIR as a result of interference.

X. Conclusion

The effectiveness of the dynamic tabu search algorithm for the FS-FAP has been established. In particular, it is has been evaluated, with and without cell reoptimization, on some of the COST 259 benchmarks. For these cellular problems, it has, in combination with cell reoptimization, improved some of the best results known. For noncellular problems, when cell reoptimization is not appropriate, the improved performance of the tabu search algorithm is even more apparent.

References

[1] W. K. Hale, "New spectrum management tools," *Proc. IEEE Int. Symp. Electromag. Compatibility*, pp. 47–53, Aug. 1981.

[2] 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.

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

[4] D. H. Smith, S. Hurley, and S. M. Allen, "A new lower bound for the channel assignment problem," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 1265–1272, July 2000.

[5] S. M. Allen, D. H. Smith, and S. Hurley, "Generation of lower bounds for minimum span frequency assignment," *Discrete Appl. Math.*, vol. 119, no. 1–2, pp. 59–78, 2002.

[6] R. Montemanni, D. H. Smith, and S. M. Allen, "Lower bounds for fixed spectrum frequency assignment," *Ann. Oper. Res.*, vol. 107, no. 1–4, pp. 237–250, 2001.

[7] R. Montemanni, "Upper and Lower Bounds for the Fixed Spectrum Frequency Assignment Problem," Ph.D., University of Glamorgan, 2001.

[8] R. Montemanni, D. H. Smith, and S. M. Allen, "An improved lower bound for the fixed spectrum frequency assignment problem," Eur. J. Oper. Res., (available online at http://www.sciencedirect.com), to be published.

[9] D. J. Castellino, S. Hurley, and N. M. Stephens, "A tabu search algorithm for frequency assignment," *Ann. Oper. Res.*, vol. 63, pp. 301–319, 1996.

[10] A. Capone and M. Trubian, "Channel assignment problem in cellular systems: A new model and a tabu search algorithm," *IEEE Trans. Veh. Technol.*, vol. 48, no. 4, pp. 1252–1260, July 1999.

[11] S. Hurley, R. M. Whitaker, and D. H. Smith, "Channel assignment in cellular networks without channel separation constraints," in *Proc. IEEE Vehi. Technol. Conf. Fall'00*, Boston, MA, Sept. 2000, pp. 1714–1718.

[12] P. Hansson, C. Lewer-Allen, and J. Samuelsson, "Breakthroughs in frequency planning," *Mobile Commun. Int.*, pp. 89–90, Dec./Jan. 1997/1998.

[13] L. M. Correia, Ed., *Wireless Flexible Personalised Communications; COST 259: Eur. Co-operation in Mobile Radio Research*. Chichester, U.K.: Wiley, 2001.

[14] D. H. Smith, S. M. Allen, and S. Hurley, "Characteristics of good metaheuristic algorithms for the frequency assignment problem," *Ann. Oper. Res.*, vol. 107, no. 1–4, pp. 285–301, 2001.

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

[16] FAP web, A. Eisenblatter and A. Koster. [Online]. Available: http://fap.zib.de/

[17] Roadef Challenge, V.-D. Chung, T. Defaix, M. Diamantini, and B. Rottembourg. (2001). [Online]. Available: http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2001/challenge2001_en.html#sujet

[18] B. Cabon, S. de Givry, L. Lobjois, T. Schiex, and J. P. Warners, "Radio link frequency assignment," *Constraints*, vol. 4, no. 1, pp. 79–89, Feb. 1999.

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

[20] D. H. Smith, S. Hurley, and S. U. Thiel, "Improving heuristics for the frequency assignment problem," *European J. Oper. Res.*, vol. 107, no. 1, pp. 76–86, 1998.

[21] K. I. Aardal, S. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino, and A. Sassano, "Models and Solution Techniques for Frequency Assignment Problems,", Berlin, Germany, ZIB-Rep. 01–40, 2001.

[22] C. Fleurant and J. A. Ferland, "Genetic and hybrid algorithms for graph coloring," *Ann. Oper. Res.*, vol. 63, no. 1, pp. 437–461, 1996.

[23] J.-K. Hao, R. Dorne, and P. Galinier, "Tabu search for frequency assignment in mobile radio networks," *J. Heuristics*, vol. 4, no. 1, pp. 47–62, 1998.

[24] F. Glover, "Heuristics for integer programming using surrogate constraints," *Decision Sci.*, vol. 8, pp. 156–166, 1977.

[25] F. Glover, E. Taillard, and D. de Werra, "A user's guide to tabu search," *Ann. Oper. Res.*, vol. 41, no. 1, pp. 3–28, 1993.

[26] A. Bouju, J. F. Boyce, C. H. D. Dimitropoulos, G. vom Scheidt, and J. G. Taylor, "Tabu search for the radio links frequency assignment problem," in *Proc. Conf. Appl. Decision Technol.: Modern Heuristic Methods*, Uxbridge, U.K., 1995, pp. 233–250.

[27] W. J. Watkins, S. Hurley, and D. H. Smith, "Evaluation of Models for Area Coverage," Cardiff University, no. 98 003, 1998.

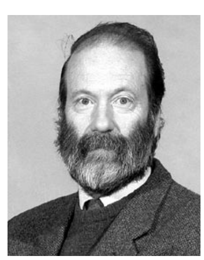
[28] R. M. Whitaker, S. Hurley, and D. H. Smith, "Frequency Assignment Heuristics for Area Coverage Problems," Cardiff University, Cardiff, U.K., 2000.

Roberto Montemanni was born in Ravenna, Italy, in 1975. He received the Laurea degree in computer science from the University of Bologna, Bologna, Italy, in 1999 and the Ph.D. degree in mathematics from the University of Glamorgan, Pontypridd, U.K., in 2002.

Since November 2001, he has been a researcher at Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland. His research interests include optimization problems arising in radio-frequency assignment, transportation, and ad hoc networks.

Jim N. J. Moon received the B.Sc. degree in computer studies and the Ph.D. degree in multiagent systems in marine simulation, both from the University of Glamorgan, Pontypridd, U.K. He also qualified as a Master Mariner in an earlier career at sea.

He is Senior Lecturer in software engineering at the University of Glamorgan. His research interests include radio-frequency assignment, software engineering, multiagent systems, and marine simulation.



Derek H. Smith received the B.Sc. and Ph.D. degrees in mathematics from the University of Southampton, Southampton, U.K.

He has been with the University of Glamorgan, Pontypridd, U.K., since 1971, where he is a Professor of mathematics. His research interests include radio-frequency assignment, graph theory, coding theory, data compression, and network reliability studies.