Available online at www.sciencedirect.com
SCIENCE DIRECT®

Discrete Applied Mathematics 129 (2003) 155–169

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

An enumerative algorithm for the frequency assignment problem

Carlo Mannino*, Antonio Sassano

Dipartimento di Informatica e Sistemistica, Università di Roma La Sapienza, Via Buonarroti 12, 00185 Roma, Italy

Abstract

We present an algorithm to solve the frequency assignment problem for mobile cellular systems and radio and television broadcasting. Frequencies must be assigned to transmitters in order to meet interference requirements so that the overall signal/noise ratio is satisfactory. The basic scheme is an exact enumerative method provided with fixing criteria to reduce the size of the instances. Larger instances are solved by applying the algorithm to suitable subinstances, eventually extending the solutions found. We were able to solve large real-life instances arising in radio broadcasting and mobile cellular systems. Computational results outperform previous results reported in the literature.

Keywords: Telecommunication systems; Frequency assignment

1. Introduction

The radio spectrum is a limited resource; moreover, the demand for frequencies has grown very fast in the last decades. Thus, it is crucial to develop effective ways of managing this scarce resource. The frequency assignment problem (FAP) is the problem of efficiently assigning a limited number of radio frequencies to the transmitters of a network in such a way that interference requirements are satisfied. This sort of problems arise in telecommunication systems, such as mobile telephone and radio and television broadcasting. Let T be the set of transmitters and let $A_i \subseteq \{1,\dots,k\}$ be the set of available frequencies of transmitter i, for all $i \in T$. A_i is called the *frequency domain* of transmitter i. A *frequency assignment* is a vector $x \in Z_+^{|T|}$, such that $x_i \in A_i$, for $i = 1,\dots,|T|$. Distance requirements due to pairwise interference are represented by a

* Corresponding author.
E-mail addresses: mannino@dis.uniroma1.it (C. Mannino), sassano@dis.uniroma1.it (A. Sassano).

0166-218X/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0166-218X(02)00239-1

symmetric integer square matrix D of size $|T| \times |T|$. D is called the *distance matrix*. In what follows, $[D]_{ij}$ will be denoted by d_{ij}. In order to reduce to zero pairwise interference we must have

$$|x_i - x_j| \geqslant d_{ij}, \quad \forall i, j \in T. \tag{1}$$

An assignment satisfying (1) is called a *feasible assignment*.

In many applications we must take into account the so-called *cumulative interference*, i.e. the effect of the simultaneous interference of all transmitters versus a single one. The pairwise interference between transmitters is described by a matrix Q, where the entry $[Q]_{ij} = q_{ij}$ represents the noise/signal ratio in $i \in T$ when $i \in T$ and $j \in T$ use the same frequency. Due to asymmetries in the propagation of the signal, Q is not necessarily symmetric. In general, the value of the interference produced by j in i depends on the quantity $|x_i - x_j|$, by the inverse of the factor $NFD_{|x_i - x_j|}$, with $NFD_0 = 1$. NFD_1 is called the *net filter discriminator*. Finally, the cumulative interference in transmitter i (denoted by δ_i) is

$$\delta_i = \sum_{j \in T} \frac{1}{NFD_{|x_i - x_j|}} q_{ij}. \tag{2}$$

In practice, NFD_r is very large for all $r > 1$, and so (2) becomes

$$\delta_i = \sum_{j\,:\,x_j = x_i} q_{ij} + \sum_{j\,:\,|x_j - x_i| = 1} \frac{1}{NFD_1} q_{ij}. \tag{3}$$

Several objective functions can be considered: one can wish to minimize the number of different frequencies or the maximum frequency assigned, or the maximum cumulative interference in each transmitter. Typically, the last two objectives are transformed into feasibility problems of the form: does there exist a feasible assignment such that (i) $\delta_i \leqslant \bar{\delta}$ for all $i \in T$, and (ii) the maximum frequency assigned is not greater than f_{MAX}?

Requirement (i) can be expressed by the following constraint:

$$\delta_i = \sum_{j\,:\,x_j = x_i} q_{ij} + \sum_{j\,:\,|x_j - x_i| = 1} \frac{1}{NFD_1} q_{ij} \leqslant \bar{\delta}, \quad \forall i \in T \tag{4}$$

while requirement (ii) is obtained by letting $A_i \subseteq \{1, 2, \ldots, f_{\mathrm{MAX}}\}$.

In the following, we denote by FAP1 the problem of minimizing the largest frequency assigned, s.t. constraints of type (1) are satisfied, while we denote by FAP2 the problem of finding a feasible solution s.t. constraints of types (1) and (4) are satisfied. Let P be an instance of FAP1. The difference between the largest and the smallest frequency in an optimum assignment of P will be denoted by $Span(P)$. If P is an instance of graph coloring, and G is the corresponding graph, then $Span(P) = \chi(G) - 1$.

In the basic model, a single frequency must be assigned to each transmitter; however, in many applications we must assign $w_i \in Z_+$ different frequencies to each transmitter i. Due to self-interference, distinct frequencies assigned to the same transmitter i must satisfy a distance requirement d_{ii}.

Examples of multiple frequency assignment are mobile radio systems, where each transmitter is a single antenna called *cell*, and the demand of frequencies of each cell is equal to the maximum number of simultaneous calls to be served; the distance requirement between frequencies assigned to the same cell is called *co-cell constraint*. In addition, more antennas can be mounted on a same physical support and the corresponding cells are grouped into a cluster called *site*. Frequencies assigned to different cells belonging to the same site must also satisfy a distance requirement, the so-called *co-site constraint*.

The basic model can still be applied to handle multiple frequency demand. This is obtained by splitting a transmitter i into w_i "twin" transmitters, each one having unit demand, and generating co-cell constraints of type (1) for each pair of twins. All other constraints involving cell i will be replicated for each of the twin transmitters.

Due to its theoretical and practical relevance, FAP has been widely approached in the literature. A systematic, graph theoretical approach has been developed in [11].

FAP is a generalization of the well-known graph-coloring problem (see, for example, [11]), and thus it is an NP-hard problem. For this reason, many authors concentrated their efforts in developing heuristics to find feasible solutions. In particular, Tabu Search, Simulated Annealing, Neural Networks and Genetic Algorithms have been applied in [2–6,15,17]. Ad hoc heuristics are presented in [14,16]. A comparison among several heuristic approaches can be found in [12]. Lower bounds based on different relaxations are presented in [8,13,19]. All of these works deal with FAP without cumulative interference (FAP1). Branch-and-cut methods are presented in [1] (FAP1) and [7] (FAP2).

In principle, implicit enumeration schemes such as branch-and-cut and branch-and-bound are able to answer the question as to whether a given instance is feasible or not. However, it is common experience that the bounds computed are of little help in answering the feasibility question, despite the huge computational effort necessary to solve a linear relaxation in each node of the enumeration tree. In this paper, we develop an implicit enumeration method—alternative to branch-and-cut or branch-and-bound—which quickly explores a large number of alternatives and is able to find, at an early stage, a feasible solution. In addition, the proposed method is also able to prove the infeasibility of real life instances quite efficiently. Speed in the enumeration process is obtained by renouncing solving linear relaxations at each subproblem. Effective branching criteria, which do not make use of the fractional components of the optimal solution of the relaxed problem, are developed. In addition, pre-processing and fixing are used to reduce the size of the instances, while restricted backtracking is used to reduce the size of the enumeration tree. For the largest instances, feasible solutions are found by solving suitable subinstances and by extending the solutions so obtained.

Our algorithm has been tested on real-life instances of both FAP1 and FAP2. For FAP1, we were able to find a feasible solution and prove its optimality for instances up to 857 transmitters. For FAP2, we were able to solve all instances presented in [7] except for one, outperforming the results reported in [7], both in terms of running time and quality of the solutions found.

2. Enumeration scheme

Our goal is to solve the following feasibility problem: find an assignment of frequencies to transmitters so that (i) only available frequencies are assigned, (ii) all distance constraints are satisfied and (iii) the cumulative interference in each transmitter is at most $\bar{\delta}$.

In the sequel, an instance of FAP will be denoted by $P=(T,\mathscr{A},D,Q,\bar{\delta},W,x)$, where T is the transmitter set, $\mathscr{A}=\{A_i\colon i\in T\}$ is the family of feasible frequency domains, D the distance matrix ($[D]_{ij}=d_{ij}$), Q the interference matrix ($[Q]_{ij}=q_{ij}$), $\bar{\delta}$ the maximum cumulative interference allowed in each transmitter, W the set of assigned transmitters and $x\in Z_+^{|T|}$ the current solution (for $i\in T-W$, x_i is undefined). When P represents a problem of type FAP1, matrix Q is the null matrix. When W and x are irrelevant, we adopt the simplified notation $P=(T,\mathscr{A},D,Q,\bar{\delta})$.

When a frequency $\bar{f}$ is assigned to a transmitter i, the feasible domains A_j for all $j\in T-i$ must be modified by removing from A_j all frequencies $f\in A_j$ such that $f<\bar{f}+d_{ij}$ and $f>\bar{f}-d_{ij}$. In fact, such frequencies cannot be assigned without violating a constraint of type (1). We denote by $\mathscr{A}(i,\bar{f})$ the family of feasible domains so obtained.

Our enumeration scheme is summarized by the following recursive procedure, denoted by **BBfreq**. A few subroutines are included in the scheme. In particular, Procedures **Choose_t** and **Choose_f** select an active transmitter and a frequency in its current domain, respectively, by applying the branching rules described in the sequel. Procedure **Remove_f** removes from the domain of the selected transmitter all frequencies whose assignment violates (4).

The input of **BBfreq** is an instance P of FAP and the constant MAXTIME, which is the available processing time. The output is a frequency assignment x, and two boolean variables, namely FEASIBLE and TIMEOFF. If the algorithm terminates with TIMEOFF = true, then the enumeration was not completed within the time limits. Otherwise, the enumeration has been completed and FEASIBLE is true iff P is feasible: in this case, x is a feasible assignment.

Procedure: BBfreq.

Input: An instance $P=(T,\mathscr{A},D,Q,\bar{\delta},W,x)$ of FAP, MAXTIME.
Output: x, FEASIBLE and TIMEOFF.

1. *If* $W=T$
  1.1 FEASIBLE = true
  1.2 *return*
 EndIf
2. FEASIBLE = false
3. *If* $A_j=\emptyset$ for some $j\in T-W$ *return*
4. $i=$ **Choose**_$t(T-W,\mathscr{A},D)$.
5. **Remove**_$f(W,Q,\bar{\delta},x,i,A_i)$
6. *While* ($A_i\neq\emptyset$) and (FEASIBLE = false) and (TIMEOFF = false)

7. $\bar{f} = \mathbf{Choose\_}f(A_i, W, I, \bar{\delta}, x)$.
8. $A_i = A_i - \bar{f}$. $W = W \cup \{i\}$. $x_i = \bar{f}$.
9. $\mathbf{BBfreq}(T, \mathscr{A}(i, \bar{f}), D, Q, \bar{\delta}, W, x)$
EndWhile

10. *return*

Procedure **Choose_t** selects a transmitter to be assigned. We denote by $d(i,Z)$ the quantity $\sum_{j\in Z} d_{ij}$. We compared the following three different criteria:

(b1) $i = \operatorname*{argmax}_{z\in T-W} \quad d(z, T - W)/|A_z|$,

(b2) $i = \operatorname*{argmax}_{z\in T-W} \quad d(z, T - W)$,

(b3) $i = \operatorname*{argmin}_{z\in T-W} \quad |A_z|$.

Criteria (b2) and (b3) can be considered as generalizations of the maximum degree branching criterion (b2) and the minimum saturation degree criterion (b3) for the graph coloring problem (see [18]).

Extensive testing led us to prefer (b1) for FAP1 and (b3) for FAP2; in the latter case, ties are broken by (b2).

Procedure **Choose_f** selects a frequency in the current frequency domain of the selected transmitter i. When assigning a frequency to an unassigned transmitter, the cumulative interference in all assigned transmitters will grow according to (3) are satisfied. Let $\delta_j(f)$ be the cumulative interference at the current iteration for all $j \in W \cup \{i\}$ when $x_i = f$. Then, we assign to i the lowest frequency $\bar{f}$ such that

$$\text{(b4)} \quad \delta_j(\bar{f}) \leqslant \alpha\bar{\delta}, \quad \forall j \in W \cup \{i\}, \tag{5}$$

where α is a given parameter with $0 \leqslant \alpha \leqslant 1$. Observe that $\alpha = 1$ corresponds to choosing the lowest available frequency. If none of the frequencies in A_i satisfies (5), then we assign to i a frequency minimizing the largest cumulative interference. When Q is the null matrix (FAP1), our branching criterion corresponds to selecting the lowest frequency in the domain.

2.1. Restricted backtracking

Procedure **BBfreq** can be viewed as performing a depth-first-search (dfs) on a branching tree $G=(N,A)$: each node $u \in N$ corresponds to a subproblem P and the children $u_1, \ldots, u_r$ of u correspond to the r subproblems obtained from P by selecting an active transmitter and assigning to it the frequencies $f_1, \ldots, f_r$ of its domain (ordered by the branching criterion). When the size of the instances increases, the number of nodes in the branching tree can grow very large, and the time needed to explore it can easily exceed prescribed limits. In this case, we need to limit the search by renouncing to perform a complete visit of the branching tree. This is obtained in the following way. Let $v_1, v_2, \ldots, v_{|N|}$ be the unique ordering of the nodes of G corresponding to the dfs, that is $t > s$ iff v_t is visited after v_s. Clearly, v_1 is the root of G and corresponds to the initial problem. We denote by $depth(z)$, $z \in N$, the length of the unique path from v_1 to z in G.

Table 1
Instances solved by restricted backtracking

Name	Size	Nodes	Time	dB
a14	470	13524	49.64	13
b13	536	165080	682.12	12
c16	486	39580	147.92	11
d14	525	22767	125.53	15
R7	857	21080	95.73	*
T6	350	187051	1681.90	*

Restricted backtracking consists of visiting only a subtree $G' = (N', A')$ of G: again, if $v_t, v_s \in N'$ and $t > s$, then v_t is visited after v_s. Let $v_p \in N$, let k be a non-negative integer and let l be the minimum index such that $l \geqslant p$ and $depth(v_l) \leqslant k$. G' is said to be obtained from G by a *k-jump* in v_p iff $N' = N - \{v_{p+1}, \ldots, v_{l-1}\}$. The node $v_l \in N'$ is the *final* node of the k-jump. Informally, we can say that the dfs is interrupted at node v_p and restarted at node v_l.

Our branching tree G' is obtained from G by performing a sequence of m consecutive k-jumps. If we denote by $v_{f(i)}$ the final node of the ith k-jump, we have $f(i) < f(i+1)$, for $i = 1, \ldots, m-1$. In the following, $v_{f(0)} = v_1$. The dimension of G' is controlled by the assumption that exactly a nodes must be visited between two consecutive k-jumps, namely that $|\{v_j \in N': f(i) \leqslant j \leqslant f(i+1)\}| = a$ for $i = 0, \ldots, m-1$, where a is a given parameter.

Restricted backtracking is embodied in our solution method in the following way: Procedure **BBfreq** is first called on the initial problem with no restricted backtracking; if no solution is found within the time limits a second call to Procedure **BBfreq** is made with restricted backtracking. A number of instances solved by restricted backtracking are shown in Table 1 (all instances will be described in detail in Section 5). For all problems, $k = 2$ and $a = 1000$. The dB column reports the quality of the signal/noise ratio; a star denotes that we are dealing with an instance of FAP1 (no cumulative interference). The instances of FAP1 reported in the table are solved more efficiently by applying the technique described in Section 4.

Different approaches to restricted backtracking have been used for example in [9,10].

3. Fixing

Fixing is a technique to reduce the size of instances of optimization problems. This is of great importance in order to reduce the computational effort of the algorithm, and it is crucial for solving the largest instances. In this paper, we present two types of fixing for FAP: frequency fixing and transmitter fixing.

3.1. Frequency fixing

This type of fixing allows us to reduce the number of available frequencies in the frequency domains. Let $P=(T,\mathscr{A},D,Q,\bar{\delta})$ be an instance of FAP. Let $i,j\in T$ and let A_i and A_j be the corresponding frequency domains. Let $f_1=\min_{f\in A_j} f$ and let $f_2=\max_{f\in A_j}$. Let $\bar{f}\in A_i$. We say that $\bar{f}$ *covers* A_j iff $\bar{f}-d_{ij}<f_1$ and $\bar{f}+d_{ij}>f_2$. Clearly, if $\bar{f}$ covers A_j, then A_j is empty in $\mathscr{A}(i,\bar{f})$, i.e. the assignment of frequency $\bar{f}$ to transmitter i is infeasible. Thus, we can generate from P a new problem P' by removing frequency $\bar{f}$ from A_i: P' is such that (i) if P has a feasible solution, then P' has a feasible solution; (ii) if P is infeasible then P' is infeasible. Obviously, a feasible assignment of P' is also a feasible assignment of P, so we can solve P' to find a solution for P.

In the following we report the scheme of our fixing algorithm, denoted by **Fix_f**. The input is a set of transmitters T, a family of frequency domains $\mathscr{A}$ and the distance matrix D. It returns as output a reduced family of frequency domains $\mathscr{A}'$.

Procedure: Fix_f.

Input: $T,\mathscr{A},D$.
Output: A (reduced) family $\mathscr{A}'$.

0. $\mathscr{A}'=\mathscr{A}$. REDUCED = true.
1. *While* (REDUCED = true)
  2. REDUCED = false
  3. *for* $i\in T$ and $f\in A_i'$
    4. *for* $j\in T-i$
      5. $f_1=\min\{f\colon f\in A_j'\}$
      6. $f_2=\max\{f\colon f\in A_j'\}$
      7. *If* $(f-d_{ij}<f_1)$ and $(f+d_{ij}>f_2)$
      8. $A_i'=A_i'-f$.
      9. REDUCED = true
      EndIf
    EndFor
  EndFor
EndWhile
10. *return*

A naive implementation of the above procedure leads to a worst-case complexity $O(f_{MAX}^2|T|^3)$. Procedure **BBfreq** can be easily amended to include frequency fixing. This is done by the insertion of the following Step 0:

0. Fix_f$(T-W,\mathscr{A},D)$

Table 2 shows the effectiveness of frequency fixing both in terms of time and number of subproblems in the search tree, when solving three different instances of FAP1 from our test set. For problem T5, fixing is essential to avoid exceeding time limits.

Table 2
Effects of frequency fixing

		Fixing		No fixing	
Name	Size	Time	Prob.	Time	Prob.
T3	45	27.76	23,501	57.27	54,958
T4	100	63.46	28,625	236.50	92,752
T5	200	686	141,115	*	*

What described above allows us to fix "out" frequencies from the frequency domains. Another type of fixing, which can be used only when solving instances of FAP1, can be applied to the transmitter set.

3.2. Transmitter fixing

Consider again two transmitters, $i, j \in T$, and let A_i be the frequency domain of transmitter i. Suppose now we assign to j a frequency $\bar{f}$. Every frequency in the set $A_i \cap \{\bar{f} - d_{ij} + 1, \ldots, \bar{f} + d_{ij} - 1\}$ will be forbidden for i (due to (1)), and must be removed from A_i. So, if we assign a frequency to transmitter j, we have that the maximum number of distinct frequencies which will be removed from A_i is $2d_{ij} - 1$. That is, the maximum number of frequencies which will be removed from A_i when all $j \in T - i$ are assigned is $\sigma(i, T) = \sum_{j \in T - i}(2d_{ij} - 1)$. If $\sigma(i, T) < |A_i|$ then any feasible assignment to all other transmitters will leave an assignable frequency in the domain A_i. This implies (when cumulative interference is not involved) that transmitter i can be assigned (one of its residual frequencies) after all other transmitters have been independently assigned. This allows us to generate a new instance P' obtained from P by removing i from the set T. Again, if P has a feasible solution, so does P'; if P is infeasible, P' is infeasible. A solution of P can be obtained from a solution of P' by assigning to i a frequency which does not violate (1).

The following procedure **Fix_t** requires in input a set of transmitters T, a family of frequency domains $\mathscr{A}$ and the distance matrix D. It returns as output a (reduced) set of transmitters $T' \subseteq T$.

Procedure: Fix_t.

Input: $T, \mathscr{A}, D$.
Output: A (reduced) transmitter set T'

1. *While* there exists $i \in T'$ such that $\sigma(i, T') < |A_i|$
  2. $T' = T' - i$.

EndWhile

The worst-case complexity of the above procedure is $O(|T|^3)$. This type of fixing can be applied at any node of the branching tree. However, there is a trade-off between the time saving due to the reduction of the size of the subproblems and the time increase due to the additional call to **Fix_t**. Computational experience showed that it is convenient to apply transmitter fixing only in the pre-processing stage, i.e. at depth 0 of the branching tree. Transmitter fixing has been applied successfully to reduce large instances of FAP1. In particular, we were able to reduce the size of S7 from 857 transmitters to 548 and the size of T7 from 857 transmitters to 626.

4. Core search

Let $P=(T,A,D,Q,\bar{\delta})$ be an instance of FAP and let $T' \subseteq T$. We denote by $P[T']=(T',\mathscr{A}',D',Q',\bar{\delta}')$ an instance of FAP obtained from P in the following way:

- $\mathscr{A}'=\{A_i \in \mathscr{A}:\ i \in T'\}$.
- D' is a $|T'| \times |T'|$ matrix with $[D']_{ij}=d_{ij}$, for $i \in T'$, $j \in T'$.
- Q' is a $|T'| \times |T'|$ matrix with $[Q']_{ij}=q_{ij}$, for $i \in T'$, $j \in T'$.
- $\bar{\delta}'=\bar{\delta}$.

$P'=P[T']$ is said to be an *induced subinstance* of P.

Extensive testing on real-life problems showed that most of large-size instances contain an induced subinstance of much smaller size which is feasible iff the original problem is. In other words, the minimum bandwidth (i.e. the span) necessary for a feasible assignment is the same for the two problems. Typically, these subinstances correspond to densely populated geographical areas.

Let P be an instance of *FAP* and let T be its transmitter set. In the following, we denote by *core* any subinstance $P'=P[T']$ of P such that $Span(P')=Span(P)$ while all induced subinstances of P' have smaller span.

If such a subinstance can be identified and it is small enough to be solved without exceeding time limits by Procedure **BBfreq** with no-restricted backtracking, then its solution provides us with (i) a proof of the infeasibility of the whole problem (if the core is infeasible) or (ii) a partial solution which can sometimes be extended to a solution of the whole problem.

The problem of finding a core P' of P is NP-hard. This can be easily shown by a reduction from graph coloring. In fact, consider the special case when P is an instance of graph coloring and let G be the corresponding graph. If we are able to identify in polynomial time a minimal-induced subgraph G' such that $\chi(G')=\chi(G)$, then it is possible to compute in polynomial time the chromatic number of G. In fact, let v be any vertex of G': since G' is minimal, $\chi(G'-v)=\chi(G')-1=\chi(G)-1$. The thesis follows by induction.

Due to this, we content ourselves with heuristically searching for a difficult subproblem, small enough to be solved exactly by Procedure **BBfreq**, but large enough to represent the *hardness* of the original problem. Similar approaches have been followed, both in coloring and FAP, for example, in [9,18,20]. The relevance of this approach is stressed in the paper by Sewell [18], where an exact (potentially) exponential–time algorithm, is used to find an initial subinstance.

First of all, we need a procedure to identify hard subinstances. Next procedure selects a subinstance of P of size *subsize*, where the size is the number of transmitters in the subinstance. Subinstances are completely identified by their set of active transmitters T'. Remind that we denote by $d(i,Z)$ the quantity $\sum_{j\in Z} d_{ij}$.

Procedure: FindSub.

Input: An instance P of FAP, *subsize*.
Output: A subset T' of the transmitter set.

1. *Set* $T' = \emptyset$.
2. *While* $|T'| < subsize$
  3. *If* $|T'| = 0$
    3.1 $i = argmax_{z\in T} \quad d(z, T)$.
  4. *Else*
    4.1 $i = argmax_{z\in T-T'} \quad d(z, T')$.
  EndIf
  5. $T' = T' \cup \{i\}$.
EndWhile

The selection rule in Step 4.1 intends to identify a subinstance which is highly connected and such that the sum of the entries in the distance matrix is as large as possible. The selection rule at Step 3.1 selects the first transmitter which is the one of maximum weighting degree (ties are broken randomly).

The proposed algorithm is summarized by Procedure **SolveFap**. The input is an instance of FAP, and three constants, MAXTIME1, MAXTIME2, and INC. MAXTIME1 is the maximum available time for each execution of Procedure **BBfreq** at Step 2, while MAXTIME2 is the maximum available time for each execution of Procedure **BBfreq** at Step 7. We typically choose MAXTIME1 $\gg$ MAXTIME2. INC is used to increment the size of the current subinstance at Step 9. The output is as for Procedure **BBfreq**. The body of the algorithm is embodied in a loop which terminates when the problem is solved (feasible or infeasible), or the time limits are exceeded. At each iteration, a subinstance SUBP of size *subsize* is re-computed (Step 1). If the time limits are not exceeded, we try to extend the solution found. This is done by generating a new problem $\bar{P}$ obtained from the original one P by removing the assigned transmitters and by updating the frequency domains of the remaining transmitters to take into account the pre-assignments.

Procedure: SolveFap.

Input: An instance P of FAP, MAXTIME1, MAXTIME2, INC.
Output: An assignment x for P, FEASIBLE, TIMEOFF.

0. $subsize = INC$
1. **FindSub**$(P, subsize)$
2. **BBfreq**$(SUBP, \text{MAXTIME1})$.
3. *If* TIMEOFF = true *return*

Table 3
Effects of application of core search

Name	Size	Time	Prob.	Freq	Feas	Subinst.
T4	100	21.51	16,121	16	Yes	0
T4	100	20.65	32,738	16	Yes	6

4. *If* FEASIBLE = false *return*
5. *If subsize* = $|T|$ *return.*
6. *Generate* $\bar{P}$.
7. **BBfreq**($\bar{P}$, MAXTIME2)
8. *If* (TIMEOFF = true) or (FEASIBLE = false)
  9. *subsize* = *subsize* + *INC*
  10. *Goto* Step 1
EndIf

Searching and solving "hard" subinstances has been crucial both in proving feasibility as well as infeasibility of large instances of FAP1 as it will be shown in the next section. However, also when the problem is solved within time limits by direct application of Procedure **BBfreq**, we can still have savings in time and number of problems of the branching tree by applying Procedure **SolveFap**.

In Table 3 we show the effect of applying core search to a feasible instance. Column *subinst.* reports the number of subinstances computed and solved by **SolveFap**. Observe that, even though the number of problems in the branching tree increases, the overall time decreases. This is a consequence of the fact that the average size of the problems solved by Procedure **SolveFap** is smaller.

5. Computational experience

The algorithm has been implemented in C, and run on a IBM-RISC System 6000 Power-Station 475. We tested our algorithm on two different sets of instances, one of FAP2 and the other of FAP1.

5.1. Instances of FAP2 (with cumulative interference)

These instances arise from real-life problems of mobile cellular systems, and were provided by CSELT, a research laboratory operating with the main Italian mobile radio system operator. All transmitters (cells) have multiple demands ranging from 2 to 4. The original threshold value $\bar{\bar{\delta}}$ is set to 0.125687: this corresponds to a 9 dB signal/noise ratio, which is considered a satisfactory quality level for the signal. For this application, the net filter discriminator $NFD_1 = 63.1$, while $NFD_r = \infty$ for all $r > 1$.

The test set is subdivided into six clusters, denoted by names aa, bb, cc, dd, ee, ff; each of the clusters corresponds to a different geographical area. Within each cluster, all the instances are induced subinstances of the largest one.

Table 4
Instances with cumulative interference

			Nodes		Time		Opt.	
Name	n	$\sum w_i$	BC	MS	BC	MS	BC	MS
a11	139	328	275	328	817	4.19	Yes	Yes
a12	160	375	268	375	736	5.52	No	Yes
a13	181	425	318	425	970	7.22	No	Yes
a14	200	470	347	471	1854	9.42	No	Yes
b10	121	316	227	317	382	4.02	Yes	Yes
b11	162	430	356	635	916	9.42	No	Yes
b12	179	473	393	958	5100	11.66	No	Yes
b13	203	536	449	557	3493	12.84	No	Yes
c13	140	352	314	381	1221	4.98	Yes	Yes
c14	161	396	400	624	3212	6.25	No	Yes
c15	181	444	416	621	5995	9.35	No	Yes
c16	200	486	424	812	3104	9.24	No	Yes
d11	141	370	280	370	289	5.38	Yes	Yes
d12	159	419	336	419	489	7.63	Yes	Yes
d13	181	476	350	476	690	9.69	Yes	Yes
d14	201	525	398	542	881	11.57	Yes	Yes
e12	139	349	284	419	636	5.24	Yes	Yes
e13	161	408	374	579	1368	6.83	Yes	Yes
e14	179	457	395	542	1777	11.57	No	Yes
e15	201	514	436	542	5919	11.57	No	No
f10	141	314	314	337	650	3.27	Yes	Yes
f11	161	357	266	561	504	4.81	No	Yes
f12	182	404	366	701	1015	6.16	No	Yes
f13	201	446	392	555	1528	8.46	No	Yes

The value of the parameter α in Procedure **Choose_f** is set to 0.5 in all experiments and the maximum available time is equal to 3600 s.

We compare our results with those presented in [7]. The results are shown in Table 4, where BC denotes the branch-and-cut columns, while MS are the **BBfreq** columns, and:

- *name*: is the name of the instance.
- n: the number of cells.
- $\sum w_i$: the overall demand, corresponding in our model to the size of the transmitter set.
- *prob.*: the number of nodes in the branching tree.
- *time*: the overall running time
- *solved*: $=$*yes* if a feasible solution has been found.

Table 5
Improved signal/noise ratio

Name	Size	Nodes	Time	dB
a14	470	13 524	49.64	13
b13	536	165 080	682.12	12
c16	486	39 580	147.92	11
d14	525	21 080	95.73	15

The results reported in the columns (BC) are taken from [7] and all experiments were run on a SUN ULTRA1 workstation with 160 MHz. All instances are available on request (see [7]).

Observe that we have been able to solve the feasibility problem with 9 dB for all instances solved by BC. In addition, we have been able to solve 13 more instances, corresponding to the largest instances of test set aa, bb, cc, dd and instance e14. Due to the effectiveness of branching rules (b3) and (b4), the number of subproblems needed is, for each instance, comparable with BC. On the other hand, our combinatorial approach implies a tremendous speed up. In fact, solutions are often found in $< 1/100$ of the BC times. Running times are directly comparable since the computers used have similar performances. We remark that we were not able to solve one instance in the test set, specifically the largest instance of the set ee.

All instances but e14 and e15 were solved with no restricted backtracking. Instance e15 is unsolved, while e14 was solved by restricted backtracking (with $k = 2$ and $a = 1000$). By applying restricted backtracking, we were able to improve the initial requirement of 9 dB of the signal/noise ratio for several instances. Computational results for the largest instances of test sets aa, bb, cc, dd are shown in Table 5. In particular, for instance d14, we were able to improve the quality up to 15 dB.

5.2. Instances of FAP1 (no cumulative interference)

We present computational results on two test sets, the set R and the set T, both arising in radio broadcasting. The two largest instances, $R7$ and $T7$, are real-life problems corresponding to a major Italian network. The largest entries of matrix D are equal to 6 for $R7$ and 3 for $T7$. All other instances in the set R and in the set T are obtained as induced subinstances from the largest one. We applied Procedure **SolveFap** to all instances, with MAXTIME1 = 3600 s, MAXTIME2 = 36 s and INC = 10. The results are shown in Tables 6 and 7, where the column:

- *name*: is the name of the instance.
- *size*: the number of transmitters.
- *time*: the overall running time
- *prob.*: the number of subproblems in the branching tree.
- *feas*: feasible or not.
- *core*: number of core searches

Table 6
Results with No cumulative interference—R instances

Name	Size	Time	Prob.	Freq	Feas	Cores
$R5$	200	2.04	4140	17	No	2
$R6$	350	2.15	4140	17	No	2
$R7$	857	2.46	4140	17	No	2
$R5$	200	4.76	7120	18	Yes	4
$R6$	350	5.55	7272	18	Yes	4
$R7$	857	11.35	7806	18	Yes	4
$R5$	200	0.36	438	19	Yes	3
$R6$	350	40.85	15012	19	Yes	7
$R7$	857	5.23	1002	19	Yes	3

Table 7
Results with no cumulative interference—T instances

Name	Size	Time	Prob.	Freq	Feas	Cores
$T2$	30	0.05	34	15	Yes	0
$T6$	350	18.04	27 192	15	No	4
$T7$	857	23.82	27 085	15	No	4
$T2$	30	0.01	40	16	Yes	1
$T6$	350	26.19	33 262	16	Yes	6
$T7$	857	40.92	38 109	16	Yes	6
$T2$	40	0.1	30	17	Yes	1
$T6$	350	16.28	19 887	17	Yes	6
$T7$	857	8.09	3689	17	Yes	6

We solved the R-set with 3 different bandwidths, corresponding to the frequency domains $\{1,\ldots,17\}$, $\{1,\ldots,18\}$, $\{1,\ldots,19\}$; analogously, we solved the T-set with the 3 different domains $\{1,\ldots,15\}$, $\{1,\ldots,16\}$, $\{1,\ldots,17\}$. Observe that infeasibility of instance $R7$ with domain $\{1,\ldots,17\}$ and feasibility of $R7$ with domain $\{1,\ldots,18\}$, imply that 18 is the minimum number of (contiguous) frequencies to solve $R7$. Similarly, we need at least 16 frequencies to solve $T7$.

References

[1] K.I. Aardal, A. Hipolito, C.P.M. van Hoesel, G. Jansen, A branch-and-cut algorithm for the frequency assignment problem, http://ftp.win.tue.nl/pub/techreports/CALMA/index.html, 1998.

[2] A. Bouju, J.F. Boyce, C.H.D. Dimitropoulos, G. vom Scheidt, J.G. Taylor, Tabu search for the radio links frequency assignment problem, in: Proceedings of the Conference on Applied Decision Technologies: Modern Heuristic Methods, Brunel University, 1995, pp. 233–250.

[3] D.J. Castelino, S. Hurley, N.M. Stephens, A tabu search algorithm for frequency assignment, Ann. Oper. Res. 63 (1996) 301–319.

[4] D. Costa, On the use of some known methods for T-colourings of graphs, Ann. Oper. Res. 41 (1993) 343–358.
[5] W. Crompton, S. Hurley, N.M. Stephens, A parallel genetic algorithm for frequency assignment problems, in: Proceedings of the IMACS/IEEE Conference on Signal Processing, Robotics and Neural Networks, Lille, France, 1994, pp. 81–84.
[6] M. Duque-Anton, D. Kunz, B. Ruber, Channel assignment for cellular radio using simulated annealing, IEEE Trans. Vehicular Technol. 42 (1993) 14–21.
[7] M. Fischetti, C. Lepschy, G. Minerva, G. Romanin-Jacur, E. Toto, Frequency assignment in mobile radio systems using branch-and-cut techniques, Eur. J. Op. Res. 123 (2000) 241–255.
[8] A. Gamst, Some lower bounds for a class of frequency assignment problems, IEEE Trans. Vehicular Technol. 35 (1986) 8–14.
[9] F. Glover, M. Parker, J. Ryan, Coloring by tabu branch and bound, in: D.S. Johnson, M.A. Trick (Eds.), Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Discrete Mathematics and Theoretical Computer Science, Vol. 26, American Mathematical Society, Providence, RI, 1996, pp. 285–307.
[10] M.K. Goldberg, R.D. Rivenburgh, Constructing cliques using restricted backtracking, in: D.S. Johnson, M.A. Trick (Eds.), Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Discrete Mathematics and Theoretical Computer Science, Vol. 26, American Mathematical Society, Providence, RI, 1996, pp. 285–307.
[11] W.K. Hale, Frequency assignment: theory and applications, Proc. IEEE 68 (1980) 1497–1514.
[12] S. Hurley, S.U. Thiel, D.H. Smith, A comparison of local search algorithms for radio link frequency assignment problems, in: ACM Symposium on Applied Computing, Philadelphia, 1996, pp. 251–257.
[13] J. Janssen, K. Kilakos, Polyhedral analysis of channel assignment problems: (I) Tours, Technical Report CDAM-96-17, London School of Economics, and on 1996.
[14] S. Kim, S.-L. Kim, A two phase algorithm for frequency assignment in cellular mobile systems, IEEE Trans. Vehicular Technol. 43 (1994) 542–548.
[15] D. Kunz, Channel assignment for cellular radio using neural networks, IEEE Trans. Vehicular Technol. 40 (1) (1991) 188–193.
[16] R. Leese, Tiling methods for channel assignment in radio communication networks, in: Proceedings of the Third ICIAM Congress, 1996.
[17] G.D. Lochtie, M.J. Mehler, Channel assignment using a subspace approach to neural networks, IEE Antennas Propagation (1995) 296–300.
[18] E.C. Sewell, An improved algorithm for exact graph coloring, in: D.S. Johnson, M.A. Trick (Eds.), Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Discrete Mathematics and Theoretical Computer Science, Vol. 26, American Mathematical Society, Providence, RI, 1996, pp. 359–372.
[19] D.H. Smith, S. Hurley, Bounds for the frequency assignment problem, Discrete Math. 167/168 (1997) 571–582.
[20] D.H. Smith, S. Hurley, S.U. Thiel, Improving heuristics for the frequency assignment problem, Eur. J. Oper. Res. 107 (1998) 76–86.