Models and methods for frequency allocation with cumulative interference constraints

Mireille Palpant[1]**, Cristian Oliva**[2]**, Christian Artigues**[1]**, Philippe Michelon**[1]**, Mohamed Didi Biha**[3]

[1]Laboratoire d'Informatique d'Avignon, FRE CNRS 2487, 339 chemin des Meinajariés,

84911 Avignon Cedex 9, FRANCE

[2] Departamento de Ingenierá Industrial, Universidad de Concepción, Casilla 160-C, Correo 3, Concepción, CHILE

[3] Laboratoire d'Analyse non linéaire et géométrie, 33, rue Louis Pasteur

84000 Avignon, FRANCE

Abstract: *We consider a realistic modelling of interferences for frequency allocation in hertzian telecommunication networks. In contrast with traditional interference models based only on binary interference constraints, this new approach considers the case of a receiver disrupted simultaneously by several senders yielding cumulative disruptions that are modelled through a unique non-binary constraint. To deal with these complex constraints, we propose extensions of classical integer linear programming formulations. On a set of realistic instances provided by the CELAR, we propose several exact and heuristic solution methods including branch and cut, constraint programming, and large neighbourhood search. We also compare the performances of our best methods with those of existing heuristics and we show how the end-user benefits from using the cumulative model instead of the traditional one.*

Keywords: *frequency allocation, cumulative interference constraints, linear programming, constraint programming, large neighbourhood search*

1 Frequency allocation with cumulative interference constraints

In this paper, we consider a frequency allocation problem in an hertzian telecommunication network. The network is made of geographic sites on which antennas are located, each antenna being connected with senders and/or receivers. A given site may include several antennas. Two distinct geographic sites can be connected by one or several unidirectional links between two antennas, each link being defined from the sender of the first antenna to the receiver of the second antenna, as shown in figure 1.

Let T denote the set of links for a given problem. Frequency allocation aims at giving to each link a frequency value which guarantees a satisfying communication quality.

Each link i of T is associated with a frequency domain F_i which defines the set of discrete frequencies that can be allocated to i. The domain results from legal issues, hardware limitations and geographic localization of the equipments.

Communication quality is based on electromagnetic compatibility computations. These computations consist, for a given receiver, to take into account the different emissions of neighbor senders that may disrupt it. For instance, the "C/I"

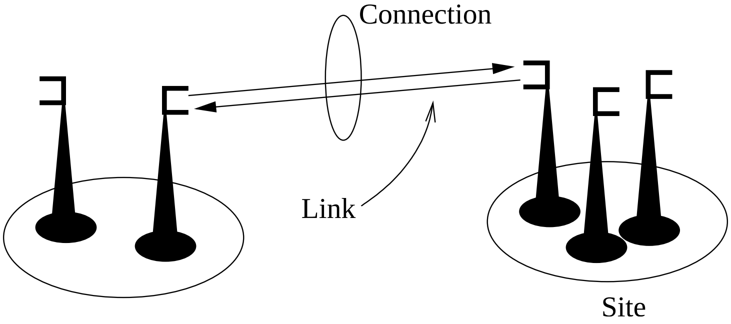

Figure 1: Sites, connections and links

criterion expresses an acceptable threshold between the useful power received by the disrupted receiver and the received power coming from all neighbor senders.

Usually (Aardal *et al.*, 2003) the ”right to disturb”, defined through the ”C/I” criterion, is equally distributed among all the disrupters. Such a distribution allows to transform a situation with N senders disrupting a receiver into N elementary situations with a single disrupter sender and a single disrupted receiver.

This binary structure gives the following usual interference constraints, where δ_{ij} is a minimal gap between frequency f_i allocated to link i and frequency f_j allocated to link j. Let CEM_1 denote the set of pairs of links concerned by the binary interference constraints.

$$|f_i - f_j| \geq \delta_{ij} \qquad \forall (i,j) \in \text{CEM}_1 \tag{1}$$

Nevertheless, this simplifying equal distribution is made considering practical solution issues without any realistic justification. In this paper, we propose to drop this simplification.

Indeed, the above distance constraints can be replaced by weaker but more complex constraints which simultaneously take the N disrupter links into account. Let T_{ij} denote a function of $\mathbb{N} \to \mathbb{R}$ such that $T_{ij}(x)$ represents the disruption of link j on link i when $|f_i - f_j| = x$. Let Λ_i denote an acceptable threshold for the receiver of link i, computed according to the ”C/I” criterion. The influence of a disrupter link $j \neq i$ can be weighted by a multiplier λ_{ij}, which take the geographical distance and the respective orientations of both disrupter and disrupted considered links into account. Let CEM_2 denote the set of links involved in a cumulative constraint as a disrupted link and P_i the set of links able to disrupt link i. The new interference constraints are now expressed as follows:

$$\sum_{j \in P_i} \lambda_{ij} T_{ij}(|f_i - f_j|) \leq \Lambda_i \qquad \forall i \in \text{CEM}_2 \tag{2}$$

Function T_{ij} is positive, decreasing and tends to 0 when x tends to infinity.

This more realistic model has been studied only recently in the literature (Dunkin *et al.*, 1998), (Mannino and Sassano, 2003).

More precisely two distinct cases are considered:

- when the disrupter senders are located on the same site as the disrupted receiver, (”near field” disruptions) the constraints are kept in the binary form (1);
- when the disrupter senders are not on the same site as the disrupted receiver (”far field disruptions”), the new formulation (2) is involved.

For problems of realistic size, trying to satisfy all interference constraints (1) and (2) can reveal itself impractical. In consequence, a classical solution approach consists in relaxing these constraints and minimising the weighted sum of their respective violations. In addition, other imperative binary constraints are involved in the studied problems: these are equality (3), difference (4), fixed distance (5) and forbidden distance (6) constraints. CI_1, CI_2, CI_3 and CI_4 denote the set of links pairs involved in each type of imperative constraints, respectively.

$$f_i = f_j \qquad \forall (i,j) \in \mathrm{CI}_1 \tag{3}$$

$$f_i \neq f_j \qquad \forall (i,j) \in \mathrm{CI}_2 \tag{4}$$

$$|f_i - f_j| = \epsilon_{ij} \qquad \forall (i,j) \in \mathrm{CI}_3 \tag{5}$$

$$|f_i - f_j| \neq \epsilon_{ij} \qquad \forall (i,j) \in \mathrm{CI}_4 \tag{6}$$

Such constraints appear when two links connect the same sites (duplex constraints).

Last, in the case where a solution is found with no violation of any interference constraints, a secondary objective is to minimise the span:

$$\min(\max_{i \in T} f_i - \min_{i \in T} f_i) \tag{7}$$

2 Related work and position of the problem

Since the problem studied in the present work introduces a new formulation, most of the approaches that can be encountered in the literature usually do not deal with it but mostly with classical MI-FAP (Minimum Interference) or MS-FAP (Minimum Span). For a complete description of these approaches, we refer to Aardal *et al.*, 2003.

Because of the strong connection between graph colouring and frequency assignment, many methods involves techniques that have been shown very effective on the first class of problems. These methods rank from the simplest constructive algorithms to standard metaheuristics. Among the first class of methods, the generalisation of the DSATUR procedure

(Brélaz, 1979) constitutes the basis of Costa's work (Costa, 1993). Slight modifications can be performed to tackle the specificities of MI-FAP: (Borndörfer, 1998), Generalised Saturation Degree (Valenzuela, 1998). Other constructive methods are based on this analogy like the Generalised Sequential Packing procedure (Sung and Wong, 1997). More sophisticated approaches, like genetic algorithms (Valenzuela, 1998) or local search ones (Borgne, 1994) also invoke graph colouring methodologies.

Integer linear programs have been proposed for the binary interference case Aardal *et al.*, 2003. Constraint programming has been tested by (Walser, 1996) for span minimization only.

Apart from that kind of methods, standard metaheuristics can be encountered, like evolutionary approaches: genetic algorithms (Crompton, 1994), ANTS (Manezzio and Carbonaro, 2000), (Montemanni, 2002). An original genetic algorithm is described in (Kolen, 1999). The latter works on a redefinition of the crossing and mutation operators allowing to perform optimisation operations on the individuals of the population.

Local search also constitutes a common approach for solving frequency assignment: guided local search (Tsang and Voudouris, 1998), simulated annealing (Knälmann, 1994) and tabu search (Capone and Trubian, 1999), (Hao, 1999). Among these methods, two have been applied to the specific problem presented here. The simulated annealing procedure described in (Sarzeaud, 2003) involves Gibbs sampling for the choice of neighbour. In (Vlasak, 2003), the consistent neighbourhood tabu search method attempts to solve series of MAX-CSP while decreasing the span value each try.

Finally, an original methodology, called Solve and Extend, has been applied in (Smith, 1998) and (Mannino and Sassano, 2003). The procedure executes in two distinct phases: during the first one, a significative sub-problem is chosen and solved (Solve phase); it is then extended in order to obtain a solution to the global problem (Extend phase). The two-phase process is iterated until a stopping criterion is met.

3 Integer linear programming formulations

We propose a mixed integer linear programming formulation of the problem inspired from the classical formulation (Aardal *et al.*, 2003), based on variables indexed by the frequency value. For each link $i \in T$ and for each possible value $v \in F_i$ for f_i, we introduce a binary variable x_{iv} equal to 1 if and only if $f_i = v$. The solution of the following integer linear program solves the feasibility problem.

$$\min \quad \alpha \sum_{(i,j)\in \mathrm{CEM}_1} c_{ij} + \beta \sum_{i \in \mathrm{CEM}_2} d_i \tag{8}$$

$$\text{s-t} \quad \sum_{v \in F_i} x_{iv} = 1 \qquad \forall i \in T \tag{9}$$

$$x_{iv} + x_{jv} \leq 1 \qquad \forall (i,j) \in \mathrm{CI}_2, \forall v \in F_i \cap F_j \tag{10}$$

$$x_{iv} \leq x_{j(v+\epsilon_{ij})} + x_{j(v-\epsilon_{ij})} \qquad \forall (i,j) \in \mathrm{CI}_3, \forall v \in F_i \tag{11}$$

$$x_{j(v+\epsilon_{ij})} + x_{j(v-\epsilon_{ij})} + x_{iv} \leq 1 \qquad \forall (i,j) \in \mathrm{CI}_4, \forall v \in F_i \tag{12}$$

$$x_{iv} + \sum_{u \in V_{ijv}} x_{ju} \leq 1 + c_{ij} \qquad \forall (i,j) \in \text{CEM}_1, \forall v \in F_i, V_{ijv} \neq \emptyset \tag{13}$$

$$\sum_{j \in P_i} \lambda_{ij} \sum_{w \in F_j} T_{ijvw} x_{jw} \leq \Lambda_i + M(1 - x_{iv} + d_i) \qquad \forall i \in \text{CEM}_2, \forall v \in F_i \tag{14}$$

$$d_i \in \{0,1\} \qquad \forall i \in \text{CEM}_2 \tag{15}$$

$$c_{ij} \in \{0,1\} \qquad \forall (i,j) \in \text{CEM}_1 \tag{16}$$

$$x_{iv} \in \{0,1\} \qquad \forall i \in T, \forall v \in F_i \tag{17}$$

where

- constraints (9) state that one and only one frequency has to be assigned to each link.
- constraints (10), (11) and (12) represent imperative difference constraints (4), fixed distance constraints (5) and forbidden distance constraints (6), respectively. Imperative equality constraints (4) are not translated into linear constraints since a straightforward preprocessing keeps only one variable.
- constraints (13) correspond to classic binary interference constraints (1). V_{ijv} denote the set of frequencies $u \in F_j$ such that $|f_v - f_u| < \delta_{ij}$, i.e. assignments of f_j violating the constraint when $f_i = v$. Hence for a distance constraint (1) there are as many linear constraints as possible values v for i such that V_{ijv} is non empty. Binary variable c_{ij} indicates if the constraint is violated. Indeed, the distance δ_{ij} between f_i and f_j is respected if and only if $c_{ij} = 0$.
- constraints (14) correspond to cumulative interference constraints (2). Let $T_{ijvw} = T_{ij}(|v - w|)$ denote the interference value of link j induced on link i if $f_i = v$ and $f_j = w$. Then, the left member of the constraint represent the sum of interferences on link i when $f_i = v$. If $x_{iv} = 0$ the constraint is always satisfied whenever constant M is large enough. Binary variable d_i allows to verify if the constraint is violated. If $x_{iv} = 1$, the constraint is satisfied and the interference sum is not grater than threshold Λ_i if and only if $d_i = 0$.
- (8) is the objective function minimising the weighted sum of violated interference constraints, where α is the weight of binary constraints CEM_1 and β is the weight of cumulative constraints CEM_2.

In the case where the problem is feasible, another linear program represents the problem of minimising the span, by setting $c_{ij} = 0, \forall (i,j) \in \text{CEM}_1$ and $d_i = 0, \forall i \in \text{CEM}_2$.

$$\min \qquad f_{\max} - f_{\min} \tag{18}$$

s-t $\quad (9), (10), (11), (12), (13), (14), (17)$

$$f_{\max} \geq \sum_{v \in F_i} v x_{iv} \qquad \forall i \in T \tag{19}$$

$$f_{\min} \leq \sum_{v \in F_i} v x_{iv} \qquad \forall i \in T \tag{20}$$

$$f_{\max}, f_{\min} \geq 0 \tag{21}$$

Constraints (19) and (20) are used to compute the maximal and minimal frequencies, respectively. Objective function (18) is the linear expression of constraint(7).

Recall that the classical representation of the interference constraints is obtained by replacing all CEM_2 constraints (14) by CEM_1 constraints (13), through a uniform distribution of the ”right to disrupt”, which of course gives a more constrained problem.

4 Constraint programming formulation

The constraint programming formulation of the problem is very close to the natural one already described. It is based on the following decision variables:

- f_i for all $i \in T$ with domain F_i. These variables represent directly the frequencies assigned to the links.
- d_{ij} for any couple of links (i, j) involved in a constraint. d_{ij} is equal to $|f_j - f_i|$.
- t_{ij} for any couple of links (i, j) involved in a cumulative constraint CEM_2. t_{ij} represents the value of the discrete perturbation function.

The constraints are:

$$d_{ij} = |fi - fj| \tag{22}$$

$$d_{ij} = \epsilon_{ij} \qquad \forall (i, j) \in \text{CI}_3 \tag{23}$$

$$d_{ij} \neq \epsilon_{ij} \qquad \forall (i, j) \in \text{CI}_4 \tag{24}$$

$$d_{ij} \geq \delta_{ij} \qquad \forall (i, j) \in \text{CEM}_1 \tag{25}$$

$$t_{ij} = T_{ij}[d_{ij}] \qquad \forall (i, j) \in \text{CEM}_2 \tag{26}$$

$$\sum_{j \in P_i} \lambda_{ij} \sum_{w \in F_j} t_{ij} \leq \Lambda_i \quad \forall i \in \text{CEM}_2 \tag{27}$$

Constraints (26), named ELEMENT constraints force variable t_{ij} to be the element of an array (the perturbation function) indexed by another decision variable: distance d_{ij}.

5 Resolution

5.1 The CELAR FAPPG instances

From data issued from military applications, the CELAR generated 30 instances of the frequency allocation problem with cumulative interference constraints, named FAPPG instances.

The instances have from 16 to 2166 links, from 10 to 1229 imperative constraints of type (3)...(6), from 16 to 2015 cumulative interference constraints (2) and from 16 to 4155 binary interference constraints (1).

We will propose in what follows several exact and heuristic methods to solve these instances. Each method aims in a first phase at minimizing the weighted number of violated constraints, then minimizing in a second phase the span, only when the number of violated constraints is equal to 0. All methods use a common preprocessing phase. All the tests have been run on a biprocessor PC with clocked at 1,3 GHz with 1 GB RAM.

5.2 Preprocessing

During the preprocessing phase, we try to decompose the set of links of the studied instance into connected components, such that there is no constraint linking two links belonging to distinct components. This decomposition, that is really useful if we consider the interference minimization criterion, is not directly usable during the second phase where the objective function involves links belonging to distinct connected components. Table 1 shows the number of connected components for each of the 30 instances.

The frequency domains can be reduced by propagating the imperative fixed distance constraints (5) as follows:

$$v \in F_i, \quad \text{if } \nexists w \in F_j : |f_i - f_j| = \epsilon_{ij} \Rightarrow F_i = F_i - v$$
$$w \in F_j, \quad \text{if } \nexists v \in F_i : |f_i - f_j| = \epsilon_{ij} \Rightarrow F_j = F_j - w$$

This simple reduction rule has a positive impact on 14 out of 30 instances as shown in table 2.

We propose a special preprocessing for the cumulative constraints, taking into account jointly these constraints and the fixed distance constraints (5). On one hand, the cumulative constraints depend on the distances between f_i and $f_j \in P_i$ for any $i \in$CEM$_2$. On the other hand, the fixed distance constraints provide us with the exact value ϵ_{ij} of this distance for any $(i, j) \in$CI$_3$. Hence we can use these values directly in the cumulative constraints where couples of frequencies linked with duplex constraints are involved. This preprocessing is sufficient to prove that instances 25 and 27 are unfeasible with a lower bound of 2 violated cumulative constraints.

Last, depending on the size of the problem a heuristic decomposition method is applied. Each subproblem is then solved independently and the constraint linking distinct subproblems are ignored. The subproblem are generated such that:

instance number	number of links	number of components	instance number	number of links	number of components
01	16	1	16	0038	1
02	18	1	17	0040	1
03	66	2	18	0052	1
04	64	2	19	0770	1
05	64	1	20	1930	136
06	182	4	21	1088	1
07	182	4	22	0768	2
08	608	20	23	0034	1
09	1460	65	24	0048	1
10	1698	73	25	0106	1
11	0164	3	26	0140	2
12	0902	23	27	0154	1
13	0306	2	28	0398	9
14	0194	1	29	0526	17
15	2454	4	30	2166	46

Table 1: Number of connected components for the 30 CELAR instances

instance number	total domain size	number of removed values	percent of removed values
01	10200	240	2.35%
02	11600	240	2.07%
03	39200	1280	3.26%
07	108700	3520	3.24%
11	97800	2240	2.29%
12	540000	18320	3.39%
16	3800	760	20.00%
17	4000	800	20.00%
21	533120	4352	0.81%
22	299520	4608	1.54%
27	9240	3080	33.33%
28	99500	19900	20.00%
29	210400	39976	19.00%
306	866400	164616	19.00%

Table 2: Domain reduction for 14 out of 30 CELAR instances

- the number of constraints linking distinct subproblems are minimized;
- the imperative constraints can not be defined for two links of distinct subproblems.

5.3 Branch and cut

The resolution of the ILP is achieved by the CPLEX 7.5 solver. The linear programs representing the FAPPG instances are of considerable size, even after a decomposition in connected components. The number of binary variables x_{iv} grows with the product of the number of links times the size of domains. Hence we have from 1700 to 866400 binary variables x_{iv}. The number of linear cumulative interference constraints (14) has the same growth. Furthermore, preliminary tests have shown that constraints (14) yield poor relaxations. Hence, we propose to replace these constraints by a set of stronger ones.

For each cumulative constraint $i \in$CEM$_2$ (2), there is a set of equivalent constraints of the type "cover inequalities":

$$x_{iv} + \sum_{(j,w) \in P} x_{jw} \leq |P| + d_i, \forall i \in T, \forall v \in F_i, \forall P \in \mathcal{P}_{iv} \qquad (28)$$

where $P \in \mathcal{P}_{iv}$ is a set of couples (link,value) violating the cumulative constraint for $f_i = v$. The drawback of these constraints is their exponential number. To overcome this difficulty, we can solve the two phases (minimisation of the weighted sum of violated constraints and span minimisation) with the branch and cut method described bellow:

- a branching method named Generalised Upper Bound (or Special Ordered Set type 1) is applied. Instead of branching on $x_{iv} = 0$ and $x_{iv} = 1$, the domain F_i is partitioned into two sub-domains F_{i1}, F_{i2} and the branching is done on $\sum_{v \in F_{i1}} x_{iv} = 0$ and $\sum_{v \in F_{i2}} x_{iv} = 0$.
- at each node of the tree search, constraints (28) violated by the current assignment are generated as cuts.

5.4 Hybrid Constraint Programming and Integer Linear Programming method

The CP model presented in Section 4 is solved by a constraint programming solver (ILOG solver 5.0) by specifying the branching rule and using the standard constraint propagation algorithms of the solver. The simple branching rule we use consists in selecting the variable f_i with the smallest domain and exploring the values of domain F_i in an increasing order. For optimization, series of feasibility problems are solved by generating constraints on the objective function.

To enhance the CP-based method, we have coupled the solving of the CP model with the solving of a relaxation of the problem based on the concept of cliques in a constraint graph. We consider the cumulative interference constraints (2):

$$\sum_{j \in P_i} \lambda_{ij} T_{ij}(|f_i - f_j|) \leq \Lambda_i \qquad \forall i \in \text{CEM}_2$$

For each $j \in P_i$, we can deduce classical binary interference constraints (1) as follows:

$$|f_i - f_j| \geq \min_{e \in IN}\{e : \lambda_{ij} T_{ij}(e) \leq \Lambda_i\} \quad \forall j \in P_i, i \in \text{CEM}_2 \tag{29}$$

Using these deduced constraints, we build a constraint graph where the nodes are the links and the edges are the original binary interference constraints (1) and fixed distance constraints (1) plus the deduced binary interference constraints. Each edge is weighted by the distance, i.e. the right term of the corresponding constraint. Let us consider a k-clique int the constraint graph. It is well known that solving the perfect matching, a relaxation of the travelling salesman problem in this clique gives a lower bound for the span criterion. Hence we can deduce that there is no solution to the problem when the lower bound is greater than the difference between the largest and the smallest frequency values of all frequency domains. This relaxation is used as a global constraint to prune the search during the CP resolution phase.

5.5 Solving the problem with large neighbourhood search

For most instances, the high number of variables and the important domains size don't allow the use of an exact ILP or CP solution method. We have then developed a heuristic scheme based on the Large Neighborhood Search methodology described in detail in (Palpant *et al.*, 2002). In the same way as before, the method runs in two phases: if a feasible solution (i.e. that satisfies all interference constraints) is found during a first phase, a second phase is initiated that consists of minimizing span.

For each connected component, we compute an initial solution with the help of a greedy algorithm. This heuristic performs in $|T|$ steps, each one selecting the most constrained link (i.e. the one that appears in the biggest number of constraints) and assigning it to the lowest possible frequency that minimizes the increase of the objective function value.

Starting from this initial solution, a given number of links are fixed to the frequency they're assigned to in the current solution while the other ones are unassigned. We then consider the subsequent sub-problem and solve it with the help of an appropriate method. The result of this optimization is obviously feasible regarding to imperative constraints and strictly better than the previous one for additional constraints impose an improve of the objective function value. This principle is iterated until a maximum execution time is reached. The crucial and difficult point of this method, that can be stated very simply, lies in the sub-problem generation and the solution method employed to solve it.

At a given iteration, a sub-problem is generated for each connected component c. Let us denote p_c the number of free variables (the ones that are unassigned). The selection of these variables is performed starting from an initial link i chosen in a random way. In the case of interference minimization, this link is assumed to belong to a constraint violated by the current assignment. We then extend the sub-problem by selecting every link related to i by a constraint. At the end of this step, if p_c is not reached, the process is iterated starting from a link already included in the sub-problem or a new randomly chosen link.

For solving the sub-problem, we apply a greedy algorithm or a tree search procedure. The choice of the solution method employed at the current iteration lies on global strategies: the idea is to apply the heuristic scheme when the current solution is likely to be easily improved (i.e. during the first iterations) and then apply the exact solution method in order to intensify the search. The greedy heuristic involved is of the same type of the one used to compute the initial solution. The CP tree search procedure is the one presented in 3.3. The search is here truncated as soon as an improving solution is discovered or when a maximum execution time is reached. Finally, an adjustment of sub-problem size p_c is made when the current solution is not improved by the greedy heuristic during a given number of iterations. This parameter can vary between an empirically chosen value $\min(|T_c| - 1, 15)$ and $|T_c|$, that is the number of links in connected component c. When the exact solution method is employed, we limit for computational times considerations this parameter to a fixed empirically chosen value $p_c = \min(|T_c| - 1, 10)$.

This Large Neighborhood Search method has shown its effectiveness and applicability to big problems. In particular, it has been able to find all the solutions proved optimal by the CP method (see section 5.4).

6 Comparison of the methods

Table 3 gives the results of the proposed methods on the CELAR instances. The results are given in terms of number of violated constraints and minimum span value. Column BC, CP and LNS give the results of the Branch and Cut, Constraint Programming method and Large Neighborhood Search method respectively. In addition we give the results obtained by other heuristics on the same problem: the ones provided by the Simulated Annealing method of Sarzeaud (2003) are given in column SA and those of the Tabu Search method of Vlasak and Vasquez (2003) are given in column TS. For each approach, the number of violated constraints and the minimum span value are indicated. The latter value is indicated only when the problem is feasible for the Branch and Cut and Constraint Programming. All methods are run during 1 hour of CPU times on the same machine, which makes the comparison significant. The best results are indicated in bold.

The results show that the LNS, SA and TS methods are far better than the branch and cut and constraint programming methods. Indeed, the latter method cannot tackle each instance globally and the problems have to be decomposed heuristically (see Section 5.2) with a significant loss of quality. However the relaxation based on the perfect matching was able to find lower bounds for the number of violated constraints on some instances (column #viol(LB)). For instances 25 and 27, the lower bound proves the optimality of the number of violated constraints found by the heuristics (2 violated constraints). For the span criterion, the problem is solved to optimality by the CP-based methods for instances 1, 23 and 24. The branch and cut method is unable to solve any instance to optimality.

The LNS method is in general superior to the SA and TS methods, with the notable exception of instance 19. This latter particular instance is the only one that involves different weights in the expression of the sum of violated constraints.

instance	link	BC		CP		LNS		SA Sarzeaud *et al*		TS Vlasak *et al*	
	number	#viol	span	#viol(LB)	span	#viol	span	#viol	span	#viol	span
1	16	0	640	**0**	**548**	**0**	**548**	0	549	**0**	**548**
2	18	4		0	650	**0**	**629**	**0**	**629**	**0**	**629**
3	66	18		43		2	599	**2**	**580**	2	623
4	64	3		0	547	0	520	0	520	**0**	**519**
5	64	31		7		**0**	**599**	0	623	0	676
6	182	185		20		**0**	**718**	**0**	**718**	0	758
7	182	185		176		6	666	**4**	**698**	8	687
8	608	-		15		**0**	**620**	0	646	0	642
9	1460	-		21		**0**	**544**	0	656	0	860
10	1698	-		23		**0**	**412**	0	692	0	849
11	164	104		5		0	604	0	656	**0**	**601**
12	902	-		33		**0**	**572**	1	639	2	634
13	306	312		120		9	399	**6**	**399**	8	380
14	194	212		265		0	398	0	360	**0**	**354**
15	2454	-		2823		**31**	**399**	73	399	44	399
16	38	69		69 (1)		**46**	**146**	**46**	**146**	**46**	**146**
17	40	51		51 (2)		46	99	**45**	**98**	**45**	**98**
18	52	158		20		**0**	**404**	1	476	0	408
19	770	-		882(4)		4385	492	4375	496	**3998**	**496**
20	1930	-		285(1)		**150**	**492**	193	496	152	496
21	1088	-		383		**0**	**982**	2	964	4	994
22	768	918		157		**0**	**788**	0	818	1	894
23	34	0	540	**0**	**380**	**0**	**380**	**0**	**380**	**0**	**380**
24	48	0	540	**0**	**410**	**0**	**410**	0	430	**0**	**410**
25	106	10		29(2)		2	540	**2**	**490**	2	540
26	140	13		88		**0**	**480**	1	492	**0**	**480**
27	154	48		82(2)		**2**	**490**	4	490	**2**	**490**
28	398	41		94		**0**	**610**	0	646	0	638
29	526	236		131		**0**	**542**	0	852	0	866
30	2166	-		871		**23**	**912**	27	912	32	912

Table 3: Computational results on the FAPPG CELAR instances

7 Comparison of the models

Table 4 gives the comparison between the model including the cumulative interference constraints (columns $\#viol_1$, $viol_1$, cpu_1) with the more constrained model which replaces the cumulative interference constraints by more constrained binary interference constraints (columns $\#viol_2$, $span_2$, cpu_2) for the first 20 instances. The results are obtained with the best heuristic (Large Neighbourhood Search method). The cpu times are here indicated for comparison purposes.

There exists in theory, because of the construction of the constraints, a solution for the problem with the cumulative constraints as least as good as the best solution of the classical model.

However, the objective of the current study is to determine whether the methods are able to find these solutions in a reasonable amount of time. In other word, is the increase of complexity of the constraints balanced by the quality of the obtained solutions?

The results show that the model with cumulative interferences obtains a larger number of feasible solutions compared to the model with only binary interference constraints with a reasonable cpu time increase. This shows clearly the benefit of introducing the cumulative interference constraints to solve practical frequency allocation problems.

8 Conclusion

We have performed an experimental comparison of two models and several methods to solve frequency assignment problems with cumulative interferences.

The large neighborhood search method we propose is superior to the existing heuristics, although it does not dominate them on all instances.

Last, our study establishes that good heuristic methods can take advantages of a direct representation of the cumulative constraints, despite their complexity. Such a result is of practical importance to solve real assignment problems since practitioners would benefit from switching, at least partially, to the new model.

As suggested by our preliminary encouraging results in this way, a future direction of research may consist in designing efficient constraint propagation techniques for the cumulative interference constraints.

Acknowledgements

The authors wish to thank Thierry Defaix from the CELAR for the generation of the problem instances and the organization of the computational experiments.

Pb	$\#viol_1$	$span_1$	cpu_1	$\#viol_2$	$span_2$	cpu_2
01	0	548	2	0	569	3
02	0	629	66	1	691	2
03	2	599	57	2	619	3
04	0	520	172	0	717	2
05	0	599	2589	1	796	196
06	0	718	3592	0	773	48
07	6	666	253	13	699	1551
08	0	620	3595	0	700	24
09	0	526	3599	0	756	18
10	0	412	3591	0	730	31
11	0	623	3592	0	600	12
12	0	560	3599	0	618	550
13	6	399	1798	15	398	3601
14	0	348	728	2	397	3091
15	25	399	3358	61	399	2703
16	46	146	145	126	149	4
17	45	99	1487	84	99	62
18	0	404	3497	0	488	20
19	550	492	3506	1455	492	2310
20	149	492	1288	105	488	644

Table 4: Benefits of using the cumulative interference constraints

9 References

K. I. Aardal, C. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino, A. Sassano, Models and Solution Techniques for the Frequency Assignment Problem, 4OR, 1(4):261-317, 2003.

L. Borgne, Automatic frequency assignment for cellular networks using local search heuristics, PhD thesis, Uppsala University, 1994.

R. Borndörfer, A. Eisenblätter, M. Grötschel, A. Martin, frequency assignment in cellular phone networks, Annals of Operations Research, 76 :73-93, 1998.

D. Brélaz, New methods to color the vertices of a graph, Communications of the ACM, 22 :251-256, 1979.

A. Capone, M. Trubian, Channel assignment problem in cellular systems: a new model and a tabu search algorithm, IEEE Transactions on Vehicular Technology, 48(4) :1252-1260, 1999.

D. Costa, On the use of some known methods for t-colouring of graphs, Annals of Operations Research, 41 :343-358, 1993.

W. Crompton, S. Hurley, N. M. Stephens, A parallel genetic algorithm for frequency assignment problems, in proceedings of IMACS/IEEE International Symposium on Signal Processing, Robotics and Neural Networks, 81-84, Lille, France, 1994.

N. W. Dunkin, J. E. Bater, P. G. Jeavons and D. A. Cohen (1998), Towards High Order Constraint Representations for the Frequency Assignment Problem, CSD-TR-98-05, Royal Holloway, University of London, Egham, Surrey, UK.

J.-K. Hao, L. Perrier, Tabu search for the frequency assignment problem in cellular radio network, European Journal of Operational Research, 1999.

A. Knälmann, A. Quellmalz, Solving the frequency assignment problem with simulated annealing, IEEE conference publication, 396 :233-240, 1994.

A. W. J. Kolen, A genetic algorithm for frequency assignment, Technical report, Universiteit Maastricht, 1999.

V. Manezzio, A. Carbonaro, An ANTS heuristic for the frequency assignment problem, Future Generation Computer Systems, 16 :927-935, 2000.

C. Mannino, A. Sassano, An Enumerative Algorithm for the Frequency Assignment Problem, Discrete Applied Mathematics, 129(1) :155-169, 2003.

R. Montemanni, D. H. Smith, S. M. Allen, An ANTS algorithm for the minimum span frequency assignment with multiple interference, IEEE transactions on Vehicular Technology, 51(5) :949-953, 2002.

M. Palpant, C. Artigues, P. Michelon (2002), A heuristic for solving the frequency assignment problem, proceedings of the XI Latin-Iberian American Congress of Operations Research (CLAIO), CD-Rom, Concepción, Chile http://www.lia.univ-avignon.fr/fich_art/321-palpant-claio.zip

O. Sarzeaud, Allocation de fréquences par échantillonnage de Gibbs, recuit simulé et apprentissage par renforcement, in proceedings of the fifth congress of the French OR society, ROADEF 2003, Avignon, France, 2003.

D. H. Smith, S. Hurley, S. U. Thiel, Improving heuristics for the frequency assignment problem, European Journal of Operational Research, 107 :220-229, 1994.

C. W. Sung, W. S. Wong, Sequential packing algorithm for channel assignment under cochannel and adjacent-channel interference constraints, IEEE transactions on Vehicular Technology, 46(3) :676-686, 1997.

E. Tsang, C. Voudouris, Solving the radio link assignment problem using guided local search, in proceedings of NATO Symposium on Radio Length Frequency Assignment, Aalborg, Denmark, 1998.

C. Valenzuela, S. Hurley, D. H. Smith, A permutation based genetic algorithm for minimum span frequency assignment, Lecture Notes in Computer Science, 1498 :907-916, 1998.

J. Vlasak, M. Vasquez, Résolution du problème d'attribution de fréquences avec sommation de perturbateurs, in proceedings of the fifth congress of the French OR society, ROADEF 2003, Avignon, France, 2003.

J.P. Walser, Feasible cellular frequency assignment using constraint programming abstractions, in proceedings of the Workshop on Constraint Programming Applications (CP96), Cambridge, USA, 1996.