Two Distributed Algorithms for the Frequency Assignment Problem in the Field of Radio Broadcasting

Lhassane Idoumghar and René Schott

***Abstract*—The frequency assignment problem involves the assignment of discrete frequencies to the transmitters of a radio network, such as a radio broadcasting network. Frequency separation is necessary to avoid interference by other transmitters to the signal received from the wanted transmitter at the reception region. Here, it is of major importance to minimize the interference while at the same time using the spectrum efficiently. In this paper we present two original distributed algorithms implemented on clusters of PCs used to solve the frequency assignment problem in the field of radio broadcasting. The first one is based on the island distributed [1] implementation of our hybrid genetic algorithm [2]. The second one uses a distributed cooperative Tabu Search. Experimental results show that our algorithms, applied to several instances given by TDF-C2R, lead to important time performance improvements.**

***Index Terms*—Cooperative tabu search algorithm, frequency assignment problem, hybrid algorithms, parallel genetic algorithms, radio broadcasting.**

I. Introduction

In order to perform suitable geographic and/or demographic coverages for radio broadcasting or radio communications, it is necessary to distribute large numbers of transmitters over the geographic areas. To each transmitter is assigned one frequency. Unfortunately, to each diffusion system[1] is assigned a finite frequency spectrum, which makes it necessary to reuse frequencies. This problem generates interference phenomena due to same or adjacent frequencies allocated to neighboring transmitters. Here, it is of major importance to minimize the interference while at the same time using the spectrum efficiently. This problem, known to be NP-Hard, can be stated as finding an optimal assignment of frequencies to a set of transmitters, under various constraints.

Due to the NP-Hardness of this problem [3] heuristic methods must be used for large graphs. Many heuristic and metaheuristic methods have been developed in the literature:

- [4]–[8] present some constructive algorithms and some basic/notable local searches.
- [9]–[12] present some approaches based on genetic algorithm paradigm.
- [8], [13], [14] present an Approximate Non deterministic Tree Search (ANTS) algorithm.
- [15], [16] present a guided local search (GLS) algorithm.
- [17]–[21] present some approaches based on Tabu Search algorithm.
- [2], [22]–[24] present some hybrid algorithm.
- [25], [26] presents a approach based on ant colony paradigms.

In this paper, we present two original distributed algorithms for solving the frequency assignment problem. The first one implements our Hybrid Genetic algorithm [2] on clusters of PCs. The second one uses a distributed cooperative Tabu Search. Applications of these new algorithms to real instances of frequency assignment problems provided by TDF-C2R Broadcasting and Wireless Research Center show drastic improvements of run time and performances.

This paper is organized as follows: Section II describes the radio network planning process. Section III recalls the frequency assignment problem. Section IV gives some hints on Genetic and Tabu Search algorithms. In Section V we present our distributed algorithms. Experimental results are given in Section VI. Concluding remarks are contained in the last section.

II. Radio Network Planning

As the demand for telecommunication services increases (interactive TV, mobile phones, FM radio, etc.), providers need fast and efficient design tools to help guarantee their commercial success. By using such tools the intent is to reduce the planning costs and shorten the duration of network deployment by employing various optimizations. Keeping the required hardware investments to a minimal level while achieving a high quality of service is the basic principle of network planning.

The planning process can be split into two main steps. The first one, called *dimensioning*, consists in spreading an optimal number of transmitters over an area in order for this area to be efficiently covered. This number should be as small as possible for obvious economic reasons. Transmitters are spread over the geographical area where providers wish to provide the users with their services, each transmitter covers a part of this geographical area called its *coverage area* (Fig. 1). The area around a transmitter where transmission conditions are favorable enough to have a good receipt of the signal is known as the *service area*

Manuscript received September 16, 2008; revised November 30, 2008. First published March 16, 2009; current version published May 22, 2009.

L. Idoumghar is with the LMIA, Université de Haute Alsace, 68093 Mulhouse, France (e-mail: lhassane.idoumghar@uha.fr).

R. Schott is with the IECN and LORIA, F-54506 Vandoeuvre-lès-Nancy, France (e-mail: Rene.Schott@loria.fr).

Digital Object Identifier 10.1109/TBC.2008.2012023

[1] Radio broadcasting, cellular phone networks, etc.

0018-9316/$25.00 © 2009 IEEE

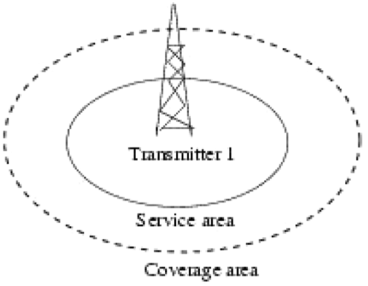

Fig. 1. Coverage and service area of a transmitter.

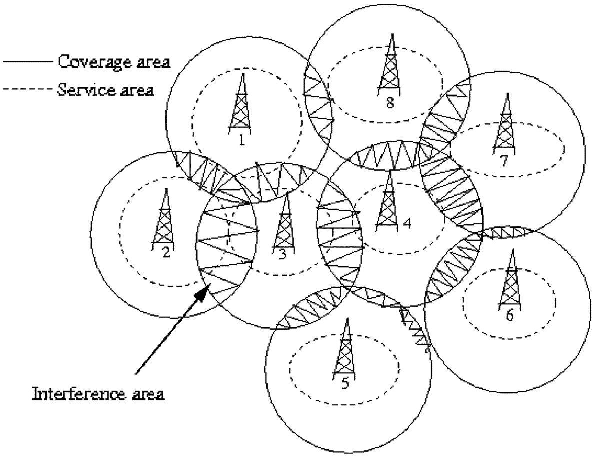

Fig. 2. Influence of a transmitter beyond its service area.

of the transmitter. The service area is the portion of the coverage area that is not jammed by other transmitters. The transmission quality and thus the shapes of these regions will depend heavily on the propagation conditions and current interferences produced by the other transmitters. Therefore, coverage areas are frequently of highly irregular shape.

Let W be a set of transmitters distributed across the geographical area. We build a constraints graph as follows:

1) apply a propagation routine to each element of W,
2) for each pair of elements in W, compute the intersection of the two corresponding coverage areas,
3) each element in W defines a vertex of the graph.

There exists an edge between two vertices if the overlapping of the two corresponding coverage areas is too significant to be ignored. In Fig. 2 at interference area the signal sent by transmitter 2 interferes with the one sent by transmitter 1. An example of a constraints graph of the radio network given in Fig. 2 is given in Fig. 3.

The weight of each edge depends on the kind of radio network. For example, in radio broadcasting the weight assigned to an edge stands for the gap between the frequencies assigned to the vertices at the two endpoints. For example, it is represented by an integer value ranging from 1 to 5. Consequently, at the end of the dimensioning step we are provided with a weighted graph.

In the second step, which is the *frequency assignment* itself, we look for a best frequency to be assigned to each transmitter. This problem will be described in the next section.

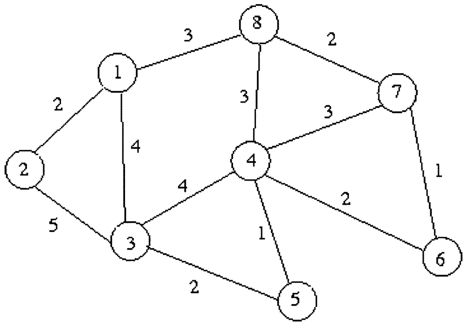

Fig. 3. Constraints graph.

III. The Frequency Assignment Problem

The frequency assignment problem can be represented as a generalized graph coloring problem [27]. To formally define the frequency assignment problem we need some definitions:

- $G = (W, E)$: an undirected graph, where $W = \{w_1, \ldots, w_n\}$ is the set of vertices. Each vertex corresponds to a transmitter of the original frequency assignment problem. $E = \{(w_i, w_j)/w_i, w_j \in W\}$ is the set of edges. Each edge of this graph represents a constraint on the frequencies assigned to the two transmitters connected by this edge. An example of constraints graph is given in Fig. 3.
- C: set of constraints. $c_{i,j}$ with $i \neq j$ represents the minimum frequency separation required to satisfy the constraint between vertices w_i and w_j. If we indicate with f_i the frequency assigned to transmitter w_i, then if $|f_i - f_j| > c_{i,j}$, the interference involving the two transmitters is acceptable.
- F: set of consecutive frequencies available for every transmitter in G.
- I: is a cost to be paid if the separation between the frequencies of transmitters w_i and w_j is less than or equal to $c_{i,j}$.

Formally, the frequency assignment problem is a mapping $f : W \rightarrow F$ (where F is a set of consecutive integers $0, \ldots,$ k representing frequencies) such that the constraints $|f_i - f_j| > c_{i,j}$ are satisfied for all $((w_i, w_j) \in E$. A feasible solution of this problem is represented by a vector $\overrightarrow{F}(f_0, f_2, \ldots, f_k)$ where $f_i \in F$ for all $i \in [0, k]$. The components of $\overrightarrow{F}$ are essentially the frequencies assigned to the vertices of G. This solution must satisfy all constraints if possible, otherwise the number of violated constraints must be minimized. In this case, our priority is to satisfy all highest level constraints.

IV. Background

A. Genetic Algorithms

Genetic algorithms were developed by Holland [28] to study the adaptative process of natural systems and to develop artificial systems that mimic the adaptative mechanism of natural systems. A review of their theoretical and practical aspects can be found in [29].

Recently, genetic algorithms have been successfully applied to various optimization problems, such as the frequency assignment problem [2]. Genetic algorithms differ from traditional optimization methods in the following ways.

1) Genetic algorithms use a coding of the parameter set rather than the parameters themselves.
2) Genetic algorithms search from a population of search solutions instead from a single one.
3) Genetic algorithms use probabilistic transition rules.

A genetic algorithm consists of a string representation ("genes") of the solutions (called individuals) in the search space, a set of genetic operators for generating new search individuals, and a stochastic assignment to control the genetic operators.

Typically, a genetic algorithm consists of the following steps.

1) Initialization—an initial population of the search solutions is randomly generated.
2) Evaluation of the fitness function—the fitness value of each individual is calculated according to the fitness function (objective function).
3) Genetic operators—new individuals are generated randomly by examining the fitness value of the search individuals and applying the genetic operators to the individuals.
4) Repeat steps 2 and 3 until the algorithm converges.

From the above description, we can see that genetic algorithms use the notion of survival of the fittest by passing "good" genes to the next generation of strings and combining different strings to explore new search solutions. The construction of genetic algorithm for any problem can be separated into four distinct and yet related tasks.

1) The choice of the representation of the strings,
2) The determination of the fitness function,
3) The design of the genetic operators, and
4) The determination of the probabilities controlling the genetic operators.

Each of the above four components greatly affects the solution obtained as well as the performance of the genetic algorithm.

In Section V, we examine each of them for the frequency assignment problem.

B. *Tabu Search Algorithm*

The basic idea of the *tabu* search metaheuristic [18] is to explore the *search space* of all feasible solutions by a sequence of moves. A move from one solution to another is generally the best available. However, in order to prevent oscillation and to provide a mechanism for escaping from locally optimal but not globally optimal solutions, some moves, at one particular iteration, are classified as forbidden or *tabu*. Moves are regarded as tabu by consideration of short-term and long-term history of the sequence of moves. A very simple use of this idea might be to classify a move as tabu if the reverse move has been made recently or frequently. There is also an *aspiration criterion*, which overrides the tabu moves in particular circumstances. These circumstances might include the cases when, by forgetting that a move is tabu, a solution which is the best so far is obtained.

Suppose it is required to minimize some cost function F on the search space S. For combinatorially hard problems it may only be possible to obtain sub-optimal solutions, in which F is close to its minimum value. Sub-optimal problems may be obtained when a certain threshold for an acceptable solution has been achieved or when a certain number of iterations have been completed.

Distributed HGA algorithm running on each processor

```
• Intitialize $p_m$, $p_c \in ]0,1]$, K $> 0$, maxGen $> 0$, $cpt \leftarrow 0$ and $i \leftarrow 1$
• Construct constraints graph
• Generate subpopulation $P_0$
• Evaluate $P_0$ and find the best solution $\pi^*$
• $\pi_{Elite} \leftarrow \pi^*$
• Repeat
    – $P_i \leftarrow \emptyset$
    – For j:= 1 to $PopSize/2$ do
        * Select two parents $p_1$ and $p_2$ from $P_{i-1}$ offspring $\leftarrow (p_1, p_2)$
        * With probability $p_c$, perform offspring := crossover($p_1$,$p_2$)
        * With probability $p_m$, mutate offspring by using the Tabu Search Algorithm
        * Evaluate offspring and add it to $P_i$
    – Add $P_{i-1}$ to $P_i$
    – if Migrate_condtion then Receive $n$ Individuals and add its to $P_i$
    – Sort $P_i$
    – Keep the $PopSize$ best solution in $P_i$
    – Find the best solution $\pi^*$ in $P_i$
    – Stop criteria
        * if $\pi_{Elite} = \pi^*$ then $cpt \leftarrow cpt + 1$
        * $\pi^*$ is better than $\pi_{Elite}$ then cpt := 0 and $\pi_{Elite} \leftarrow \pi^*$
        * if $fitness(\pi_{Elite}) = 0$ OR $cpt = K$ OR $i = maxGen$ then terminate HGA
    – $i \leftarrow i + 1$
```

Fig. 4. Genetic algorithm that uses TS approach as mutation operator.

A characterization of the search space S for which Tabu Search can be applied is when there exists a set of k moves $D = \{d_1, \ldots, d_k\}$ such that the application of the moves to a feasible solution $s \in S$ leads to k, usually distinct, solutions $D(s) = \{d_1(s), \ldots, d_k(s)\}$. The subset $N_{set}(s) \subseteq D(s)$ of feasible solutions is known as the neighborhood of s.

The method starts with (possibly random) solution $s_0 \in S$ and determines a sequence of solutions $s_0, s_1, \ldots, s_n \in S$. At each iteration, s_{j+1} $(0 \leq j < n)$ is selected from the neighborhood $N_{set}(s_j)$. The selection process is first to determine the tabu set $T_{set}(s_j) \subseteq N_{set}(s_j)$ and the aspiration set $A_{set}(s_j) \subseteq N_{set}(s_j)$. Then s_{j+1} is the neighbor of s_j which is either an aspirant or not tabu and for which $F(s_{j+1})$ is minimal; that is, $F(s_{j+1}) \leq F(s_i)$ for all $s_i \in (N_{set}(s_j) - T_{set}(s_j)) \cup A_{set}(s_j)$.

In the next section we will present different elements used by our algorithm based on a refined modeling of the input data and on a *tabu search*.

V. Our Distributed Algorithms

A. *Distributed Hybrid Genetic Algorithm*

The principles of our hybrid algorithm are described in Fig. 4.

We use independent subpopulations of individuals with their own fitness functions which evolve in isolation, except for an exchange of some individuals (migration). A set of m individuals is assigned to each of the P processors, for a total population size of $m \times P$. The set assigned to each processor is its subpopulation.

The processors are connected by an interconnection network with a ring topology. Initial subpopulations consist of a randomly constructed assignment created at each processor. Each processor, disjointly and in parallel, executes the sequential hybrid genetic algorithm on its subpopulation for a certain number of generations $(maxGen)$. Afterwards, each subpopulation exchanges a specific number of individuals (migrants) with its neighbors. We exchange the individuals themselves, i.e. the migrants are removed from one subpopulation and added to another. Hence the size of the population remains the same after migration. The process continues with the separate evolution of each subpopulation for a certain number of generations. At the end of the process the best individual that exists constitutes the final assignment.

1) Description of an Individual: Each individual represents a possible solution to the problem and is composed of a string of genes. In our implementation a gene is coded by an integer array. The ith value corresponds to the frequency assigned to the vertex w_i in a constraint graph. In such graph, some vertices (transmitters) would have fixed frequencies. These vertices are placed at the end of the string. The size of each individual is $|W|$.

2) Initial Subpopulation: Initial subpopulation P_0 at each processor is created randomly. In Fig. 4, $PopSize$ is the size of every subpopulation P_i. During each of the $maxGen$ generations, $PopSize$ offsprings are generated through the crossover of parents selected from the subpopulation.

3) Fitness Evaluation: The fitness function in hybrid genetic algorithms is typically the objective function that we want to minimize in the problem. It serves for each individual to be tested for suitability to the environment under consideration. Our objective function is defined as follows: $I = \sum_{i,j} \rho_{i,j}\delta_{i,j}$, where $\rho_{i,j}$ is a weight associated with constraint $m_{i,j}$ ($\rho_{i,j}$ and $m_{i,j}$ are positive integers). $\delta_{i,j}$ is defined by:

$$\delta_{i,j} = \begin{cases} 0 & \text{if } |f(w_i) - f(w_j)| \geq m_{i,j} \\ m_{i,j} - |f(w_i) - f(w_j)| & \text{otherwise} \end{cases}$$

4) Selection Operator: In our approach, we use a Roulette-wheel selection that is a simple method for implementing fitness-proportionate selection. It is conceptually equal to giving each individual a slice of a circular roulette wheel equal in area to the individuals fitness [30]. The wheel is spun N times, where N is the number of the individuals in the population (in our case $N = PopSize$). On each spin, the individual under wheels marker is selected to be in the pool of parents for the next generation [30].

5) Crossover Operator: The crossover is a random process defined by a probability p_c and applied sequentially to pairs of parents chosen randomly in the population. It consists in exchanging parts of the genetic material of the parents in order to create two children (offspring).

In our approach we have used the uniform crossover [31] obtained from a binary mask which possesses a number of genes equal to the number of genes of the individuals of the population. This mask is usually uniformly randomly generated: each bit has value 0 or 1 with equal probability. The used uniform crossover is parameterized by a function of the probability $p_0 = 0.5$ [32], corresponding to appearance of the values (0 and 1) in each bit of the mask. The first child is created by taking the genes of the first parent corresponding to value 1 in the mask. The second child is obtained similarly but the complementary part of the mask is used.

Sequential Tabu Search Algorithm

```
• Let $X$ be an initial solution and let $X^* \leftarrow X$ be a best solution
• stop ← false, tabu_list ← $\emptyset$,
• While not stop Do
    – Let $s^*$ be the best neighbouring configuration which is not
      forbidden by the tabu_list. This configuration is chosen by
      using a uniform probability in Skiplist.
    – Choose $s_t$ in the best neighbouring configuration forbidden
      by tabu_list (Aspiration criterion).
    – Choose the best configuration $X$ between $s_t$ and $s^*$ which
      minimizes a cost function.
    – Adjust tabu_list
    – Update the interferences matrix $M$.
    – If $X$ is better than $X^*$ Then $X^* \leftarrow X$
    – stop ← termination criterion
```

Fig. 5. Tabu search approach used as a mutation operator.

6) Mutation Operator: As indicated previously, we use *Tabu Search Algorithm* described in Fig. 5. This algorithm implements some particular features, which are not present in the general schema and which we list below. These features improve the performances of our Tabu Search Algorithm. Namely:

- **Neighboring configuration:** an solution X' is in the neighborhood of the current solution X if X' differs from X in the frequency assigned to exactly one transmitter. *Skiplist*[2] data structure [33] allows us to choose efficiently and quickly the transmitter who violates more constraints. To do this, all transmitters are ranked in descending order of their *local cost function*[3] *Skiplist* contains pairs $\langle I_{local}(w_i), w_i \rangle$. The transmitter for which we must change the frequency is the first transmitter w in *skiplist*. For this transmitter we choose the best assignment f_{best} by using the *interferences matrix M* (see next point). The move $\langle w, f_{old} \rangle$ which produces X' from X is added to the tabu list and *Skiplist* is updated (remove $\langle I_{local}(\langle w_i, f_{old} \rangle), w_i \rangle$ and add $\langle I_{local}(\langle w_i, f_{new} \rangle), w_i \rangle$)
- **Local cost function:** to improve performances of our algorithm, especially when we compute the objective function, we have implemented a two dimensional array $|W||F|$, called the *interferences matrix M* inspired by [23], [34] where each element $M(w_i, f_j)$ corresponds to the *local cost (Interferences)* $I_{local}(w_i)$ obtained by assigning the frequency f_j to transmitter w_i. The best frequency f_{new} to be assigned to a transmitter w_i, corresponds to the frequency which has the smallest $M(w_i, f_{new})$ value. Each time a move $\langle w_i, f_{new} \rangle$ is chosen, only rows corresponding to all w_j adjacents to w_i, are updated accordingly. More precisely, for each w_j only the positions of $M(w_j, f)$ corresponding to the frequencies f which interfere with the

[2] A skip list of size n is a probabilistic data structure, based on multiple parallel, sorted linked lists, with efficiency comparable to a binary search tree (insertion, deletion, query have an average time complexity in $O(log(n))$).

[3] $I_{local}(w_i) = \sum_{w_j \in adjacent(w_i)} \rho_{w_i,w_j} \delta_{w_i,w_j}$.

```
Cooperative Distributed Tabu Search Algorithm
• Let $X$ be an initial solution and let $X^* \longleftarrow X$ be a best solution
• stop ⟵ false, tabu_list ⟵ $\emptyset$,
• While not stop Do
    – cooperation_send_receive(best_solution)
    – If best_solution is better than X then
        * tabu_list ⟵ $\emptyset$, $X \longleftarrow$ best_solution
    – else
        * Let $s^*$ be the best neighbouring configuration which
          is not forbidden by the tabu_list. This configuration is
          chosen by using a uniform probability in Skiplist.
        * Choose $s_t$ in the best neighbouring configuration for-
          bidden by tabu_list (Aspiration criterion).
        * Choose the best configuration $X$ between $s_t$ and $s^*$
          which minimizes a cost function.
        * Adjust tabu_list
        * Update the interferences matrix $M$.
    – If $X$ is better than $X^*$ Then $X^* \longleftarrow X$
    – stop ⟵ termination criterion
```

Fig. 6. Distributed tabu search algorithm.

f_{old} or the f_{new} frequency of the transmitter w_i are modified. Thus, the objective function of a current solution X is computed as follows:

$$I_{new}(X) = I_{old}(X) - I_{local}\left(\langle w_i, f_{old}\rangle\right) + I_{local}\left(\langle w_i, f_{new}\rangle\right)$$

- **Aspiration criterion:** the tabu status of a move is cancelled if the move leads to a solution better than the best solution X^* found so far.
- **Tabu list:** the tabu list in our approach contains pairs $\langle w_i, f\rangle$, where w is a transmitter and f a frequency. Each time a move $\langle w_i, f\rangle$ is chosen, we insert it into the tabu list and it is classified tabu for l iterations, where l is a tabu list size.
- **Termination criterion:** The algorithm stops when the maximum number of iterations has been exceeded or if there was no improvement of the best solution in the last k iterations.

B. Cooperative Distributed Tabu Search Algorithm

The principles of this algorithm, described in Fig. 6, are as follows:

The processors are connected by an interconnection network with a ring topology. Each processor, disjointly and in parallel, executes the sequential Tabu Search Algorithm on its current solution. After each k iterations, it exchanges its best solution (obtained so far) with its neighbors by calling *cooperation* method.

The *cooperation_send_receive* method implements:

- *Synchronous* mode (*PTSSy*): at the kth iteration, each processor sends its best solution and waits to receive one solution from its neighbor.
- *Asynchronous* mode (*PTSAs*): at the kth iteration, each processor sends its best solution and continues the improvement of its current solution and verifies if it does not receive a solution from its neighbor.

The process continues with the separate improvement of each current solution for a maximum number of iterations. At the end of the process the best solution that exists constitutes the final assignment.

VI. Experimental Results

The island distributed hybrid genetic and Tabu Search algorithms have been implemented on a cluster of PCs running under Windows XP Professional version 2002. The cluster is composed of 5 Pentium (D) machines running at 3 Ghz. Each processor has 1 Gb of memory.

The results presented in this paper have been achieved with the machines running their normal daily loads in addition to our algorithm.

Each approach described in this paper is written in C++ using MPICH2 (version 1.0.6) library for communication across the processes. The C++ compiler used is gcc version 2.95.2 (Dev-cpp).

This section presents the main results obtained after several runs performed over three real instances, which can be obtained from the corresponding author, of frequency assignment problems provided by TDF-C2R Broadcasting and Wireless Research Center:

- *instance1*: 972 transmitters and 12933 constraints,
- *instance2*: 972 transmitters and 25550 constraints,
- *instance3*: 972 transmitters and 43807 constraints.

Each vertex of the constraints graph may take $m = 200$ possible values (bandwidth 87.5–107.5 MHz, $|107.5 - 87.5\ \mathrm{MHz}| = 20\ \mathrm{MHz} = \mathrm{m} \times 100\ \mathrm{kHz}$).

We recall that the main objective is to satisfy all constraints (interference constraints) if possible, otherwise high level constraints have to be satisfied first.

The results produced by the approach presented in this paper and those obtained by other approaches [2], [25] compared to the best operating solution in the field of radio broadcasting given by CSA [35] in France are given in Tables I–III.

In these tables:

- *SeqTS*: means that we use *Sequential Tabu Search Algorithm* described in Fig. 5.
- *SeqHGA*: means that we use a sequential hybrid genetic algorithm described in Fig. 4. This algorithm uses a Tabu Search algorithm described in Fig. 5 as a mutation operator.
- *ANT*: means that we use an ant algorithm [25]. This algorithm is a multi-agent system based on the idea of parallel search. In this algorithm, a given number of ants move around the vertices of the graph and change the color of each visited vertex according to a local criterion. At each iteration each ant k moves from vertex i to vertex j and changes its color, it remains there until the next iteration, when it will move again towards one of the vertices of j's neighborhood. Each ant can "remember" at each step the former changes performed by the algorithm, and takes into account that these changes may have modified the cost function of the neighborhood of j. Therefore, at the next step, the ant k will normally try to arrange the coloring of the worst adjacent vertex to j. Any single action depends strongly on the last move of each ant; this dependence reinforces the results of recent modifications. At a given iteration each ant moves from the current vertex to the adjacent vertex with the highest level of violations, and replaces the old color of the vertex with a new color that minimizes these violations. For a given vertex, the highest

TABLE I
COMPARISON OF THE RESULTS OBTAINED BY DIFFERENT METHODS USED TO SOLVE *INSTANCE1*

	SeqHGA	ParHGA	ANT	PTSAs	PTSSy	SeqTS	CSA
Total violated constraints	64	64	70	66	66	68	142
constraints 1	20	23	17	23	21	27	37
constraints 2	39	36	41	37	34	35	68
constraints 3	4	4	9	4	4	4	15
constraints 4	1	1	3	2	2	2	6
constraints 5	0	0	0	0	0	0	16
run time	17 sec	6 sec	-	2 sec	2 sec	2 sec	-

TABLE II
COMPARISON OF THE RESULTS OBTAINED BY DIFFERENT METHODS USED TO SOLVE *INSTANCE2*

	SeqHGA	ParHGA	ANT	PTSAs	PTSSy	SeqTS	CSA
Total violated constraints	81	77	99	82	81	84	271
constraints 1	36	38	47	42	40	43	92
constraints 2	42	35	42	34	38	35	117
constraints 3	2	3	8	3	1	3	27
constraints 4	1	1	2	3	2	3	10
constraints 5	0	0	0	0	0	0	25
run time	27 sec	15 sec	-	3 sec	3 sec	3 sec	-

TABLE III
COMPARISON OF THE RESULTS OBTAINED BY DIFFERENT METHODS USED TO SOLVE *INSTANCE3*

	SeqHGA	ParHGA	ANT	PTSAs	PTSSy	SeqTS	CSA
Total violated constraints	104	104	151	114	105	119	466
constraints 1	59	59	74	67	57	68	157
constraints 2	42	41	68	43	44	47	202
constraints 3	2	2	6	2	2	2	44
constraints 4	1	2	3	2	2	2	25
constraints 5	0	0	0	0	0	0	38
run time	1 min 47 sec	17 sec	-	5 sec	5 sec	5 sec	-

TABLE IV
HYBRID GENETIC ALGORITHM: DEFAULT VALUES OF THE PARAMETERS

Population size (PopSize)	200 for the sequential version. 40 for each sub-population in distributed version
Mutation probability m_m	0.5
Crossover probability p_c	0.8
Migration rate	5
Stopping criterion	The stopping criterion used is the non evolution of the best solution during $K = 15$ generations, once the minimal number of generations has been overpassed ($maxGen = 200$).

level of violations is computed by using the *Cost* function described in Section IV. This action is randomly repeated for each ant: the ant moves to the worst adjacent vertex with a certain probability p_n (otherwise it moves to any other adjacent vertex randomly chosen), and assigns the best possible color with a probability p_c (otherwise any color is assigned at random). The probabilistic nature of the algorithm allows the ants to escape from local minima and to obtain bounds close to the absolute minimum. This process, which is carried out simultaneously by the set of ants, is repeated until the optimal solution is found or the algorithm converges. The number of ants that move along the graph is an adjustable parameter and increases with the order of the graph.

- *PTSSy* and *PTSAs*: means that we use *Synchronous and Asynchronous* Distributed Tabu Search Algorithm described before.
- *ParHGA*: means that we use *Distributed hybrid genetic algorithm* described in Fig. 4.
- *CSA*: corresponds to the best operating solution in the field of radio broadcasting given by the CSA [35] in France,
- The second row gives the total number of constraints that are violated by each approach,
- The next five rows give the details of the number of total violated constraints for each constraint type,
- The last row gives the run time of each algorithm.

The analysis of the results given in these tables shows important time performance improvements.

Table IV gives the default values of the parameters used by the hybrid genetic algorithm.

VII. CONCLUDING REMARKS

We have presented two original distributed algorithms for solving the frequency assignment problem. Experimental results show important time performance improvements. The design of even better hybrid sequential and parallel algorithms for solving this problem is a real challenge.

Acknowledgment

The authors are very grateful to anonymous referees for their comments and suggestions which allowed to improve the contain and the presentation of this paper.

References

[1] M. Khabzaoui, C. Dhaenens, and E. Talbi, "Parallel genetic algorithms for multi-objective rule mining," in *6th Metaheuristics International Conference (MIC'2005)*, Vienne, Autriche, 2005, pp. 571–576.

[2] L. Idoumghar, M. Alabau, and R. Schott, "New hybrid genetic algorithms for the frequency assignment problem," *IEEE Trans. Broadcasting*, vol. 48, no. 1, pp. 27–34, Mar. 2002.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. San Francisco, CA: W.H. Freeman and Co., 1979.

[4] R. Borndorfer, A. Einsenblatter, M. Grotschel, and A. Martin, "Stable Set and Other Techniques for Frequency Assignment Problems," Preprint, Konrad-Zuse-Zentrum fur Informationstechnik Berlin.

[5] R. Borndorfer, A. Einsenblatter, M. Grotschel, and A. Martin, "Frequency assignment in cellular phone networks," *Annals of Operations Research*, vol. 76, pp. 73–93, 1998.

[6] A. Eisenblatter, "A Frequency Assignment Problem in Cellular Phone Networks (Extended Abstract)," Konrad-Zuse-Zentrum fur Informationstechnik Berlin, Technical Report SC 97-27, Jul. 1997.

[7] A. M. C. A. Koster, "Frequency Assignment. Models and Algorithms," PhD thesis, Maastricht University, Maastricht , The Netherlands, Nov. 1999.

[8] R. Montemanni, *Algoritmi Esatti ed Euristici per il Problema dell'Assegnamento di Frequenze* (in Italian). Bologna, Italy: Tesi di Laurea, Universita degli Studi di Bologna, Mar. 1999.

[9] A. W. J. Kolen, "A Genetic Algorithm for Frequency Assignment," Maastricht University, Technical report, 1999.

[10] T. L. Lau and E. P. K. Tsang, "Solving the radio link frequency assignment problem with the guided genetic algorithm," in *Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace)*, Aalborg, Denmark, Oct. 1998, pp. 14b.1–14b.27, NATO RTO-MP13, 1999.

[11] G. Colombo, "A genetic algorithm for frequency assignment with problem decomposition," *Int. J. Mobile Network Design and Innovation*, vol. 1, no. 2, pp. 102–112.

[12] S. Matsui, I. Watanabe, and K. Tokoro, "An Efficient Genetic Algorithm for a Fixed Frequency Assignment Problem with Limited Bandwidth Constraint," *Systems and Computers in Japan*, vol. 35, no. 10, pp. 32–39, 2004.

[13] V. Maniezzo and A. Carbonaro, "An ANTS heuristic for the frequency assignment problem," *Future Generation Computer Systems*, vol. 16, pp. 927–935, 2000.

[14] V. Maniezzo, A. Carbonaro, and R. Montemanni, "An approach to frequency assignment problem based on an ANTS heuristic," in *Proceedings of the Third Metaheuristic International Conference (MIC'99)*, 1999, pp. 311–316.

[15] C. Voudouris, "Guided Local Search for Combinatorial Optimisation Problems," PhD thesis, University of Essex, Colchester, U.K., 1997.

[16] C. Voudouris and E. Tsang, "Solving the radio link frequency assignment problem using guided local search," in *Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace)*, Aalborg, Denmark, Oct. 1998, pp. 14a.1–14a.9, NATO RTO-MP13, 1999.

[17] J. M. Coupé, "Using a tabu search to solve the frequency assignment problem," Patent TDF, N. 9902768, 1999.

[18] F. Glover and E. Taillard, "A user's guide to Tabu search," *Annals of Operations Research*, vol. 41, pp. 3–28, 1993.

[19] L. Idoumghar and P. Debreux, "New modeling approach for the frequency assignment problem in broadcasting," *IEEE Trans. Broadcasting*, vol. 48, no. 4, pp. 293–298, Dec. 2002.

[20] R. Montemanni, D. H. Smith, and S. M. Allen, "An improved lower bound for the fixed spectrum frequency assignment problem," *IEEE Trans. Vehicular Technology*, vol. 52, no. 4, pp. 891–901, Jul. 2003.

[21] R. M. Whitaker, S. Hurley, and D. H. Smith, "Frequency assignment heuristics for area coverage problems," Cardi University, Technical report, Sep. 2000.

[22] L. Idoumghar and R. Schott, "A new hybrid GA-MDP algorithm for the frequency assignment problem," in *Proceedings of ICTAI'06* , Washington, DC, Nov. 13–15, 2006, pp. 18–25.

[23] C. Fleurant and J. A. Ferland, "Genetic and hybrid algorithms for graph coloring," *Annals of Operations Research*, vol. 63, pp. 437–461, 1996.

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

[25] F. Comellas and J. Ozón, "An ant algorithm for the graph colouring problem," in *Proc. ANTS'98- From Ant Colonies to Artificial Ants: First International Workshop on Ant Colony Optimiszation*, Brussels, Belgium, Oct. 1998.

[26] J. S. Graham, R. Montemanni, J. N. J. Moon, and D. H. Smith, "Frequency assignment, multiple interference and binary constraints," *Wireless Networks*, vol. 14, no. 4, pp. 449–464, Aug. 2008.

[27] S. Hurley, D. H. Smith, and S. U. Thiel, "Fasoft: A system for discrete frequency assignment," *Radio Science*, vol. 32, no. 5, pp. 1921–1939, 1997.

[28] J. H. Holland, "Adaptation in Natural and Artificial Systems," PhD thesis, University of Michigan Press., Ann Arbor, MI, 1974.

[29] D. E. Goldberg, "Genetic algorithms in search," in *Optimization and Machine Learning*. Reading, MA: Addison Wesley, 1989.

[30] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: The MIT Press, 1998.

[31] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proc. of Int. Conf. on Genetic Algorithms (ICGA'89)*, 1989, pp. 2–9.

[32] W. M. Spears and K. A. De Jong, "On the virtues of parameterized uniform crossover," in *Proc. of the 4th Int. Conf. on Genetic Algorithms and Their Applications*, 1991, pp. 230–236, University of California, San Diego, Morgan Kauffmann Publishers.

[33] W. Pugh, "Skip list: A probabilistic alternative to balanced trees," *CACM*, vol. 33, no. 6, pp. 668–676.

[34] J.-K. Hao, R. Dorne, and P. Galinier, "Tabu Search for frequency assignment in mobile radio networks," *Journal of Heuristics*, vol. 4, pp. 47–62, 1998.

[35] The "Conseil supérieur de l'audiovisuel" is an independent administrative authority which was created in France to guarantee broadcasting freedom in the conditions laid down by the modified Law of September 30th, 1986. [Online]. Available: http://www.csa.fr/multi/introduction/intro.php?l=uk

Lhassane Idoumghar is an associate Professor at the University of Haute Alsace in France since 2003. He obtained his Ph.D. at the University of Henri Poincaré—Nancy 1, France. 2000–2003 he worked in TDF-C2R Broadcasting and wireless Research Center. His research activities include the frequency assignment problem, planning processes, stochastic processes, meta-heuristics and applications in computer science.

René Schott is a full Professor at the University Henri Poincaré-Nancy 1, France since 1987. He has granted the Silver Medal from the National Center for Scientific Research (CNRS) with the team EURECA, CRIN (Centre de Recherches en Informatique de Nancy) in 1987 for his achievements in theoretical computer science. He is a member of the American Mathematical Society, the French Mathematical Society and the European Association for Theoretical Computer Science. His research interests include the study of stochastic processes and operator calculus on algebraic structures and their applications in computer science (probabilistic analysis of algorithms, random generators, randomized algorithms, optimization problems). More recently, he has been focusing on cellular networks and frequency assignment problems. He serves as referee for many journals and international conferences both in mathematics and in computer science. Dr. Schott is the co-author of seven books and has also published (as author or co-author) over 100 journal and conference proceedings papers. He received numerous collaborative research grants from such agencies as CNRS, INRIA, NATO, NSF and PROCOPE.