**RAIRO Operations Research**
Will be set by the publisher

# THREE TABU SEARCH METHODS FOR THE MI-FAP APPLIED TO 802.11 NETWORKS*

Sacha Varone[1] and Nicolas Zufferey[2]

**Abstract**. Wireless LAN using IEEE 802.11 networks are now widely deployed at home by residential users or in hot spots by telecommunication operators. A hot spot is a place where a set of access points (APs) are located nearby each other and can serve many users. Since perturbations can degrade the quality of the signal, a careful channel assignment to each AP has to be done. Channel assignment of APs at hot spots, and more generally setup configuration and management, is still often done manually. In this paper, we consider a modeling that enables optimization of channel assignment with respect to the dynamic behavior of end-users. We prove our problem's formulation to correspond to the *Minimum Interference Frequency Assignment Problem*, and hence the problem to be NP-hard. We propose and compare three different *tabu search methods* to solve the problem of channel assignment in 802.11 WLAN networks. The first one, called *TabuObj*, tackles the problem using directly the objective function associated with the model. The second one, called *TabuApproxObj*, uses a simplified and approximate objective function in order to visit more solutions during the same amount of time, i.e. to be quicker than *TabuObj*. The third one, called *TabuLevel*, is even more quicker and is based on the following philosophy: under time constraints, it could be judicious to explore very quickly lots of solutions, rather than spending much computation time for the evaluation of each solution, and hence only considering a few solutions. Those three methods are then compared based on time constraints and on the quality of their solutions.

**Keywords:** WLAN, Channel Assignment,Coloring, Tabu Search.

**Mathematics Subject Classification.** 65K10, 68T20, 90B80

28 October 2007.

* *This work was supported by grant PA002-104974 received by the first author from the Swiss National Science Foundation*

[1] Haute École de Gestion (HEG), Bâtiment F, Route de Drize 7, 1227 Carouge, Switzerland, sacha.varone@hesge.ch

[2] Université Laval, Département Opérations et Systèmes de Décisions, Faculté des Sciences de l'Administration, Québec (QC), G1K 7P4, Canada, nicolas.zufferey@fsa.ulaval.ca

## Introduction

The ISM band allows everybody a free usage of a spectrum for wireless networks. As 802.11 b or g networks are now widely deployed, the consequence is that overlapping frequencies generate interferences and hence degrade the performance of such networks. The next generation of retro-compatible WLAN networks are on the point to be defined such as 802.11 n or WiMax. The common point between those networks is that they all have to face interferences, since each device has to transmit and to receive signals on only one channel. IEEE 802.11 b or g allows only three non-overlapping channels, and hence should not generate interferences. Therefore, most of the time, administrators only use channels 1, 6 and 11 for their network. A draft interaccess-point protocol (IAPP) has been specified to standardize the communication between APs over the wired interface IEEE 802.11F, but it does not solve completely the interference problem (IEEE trial-use recommended practice for multi-vendor access point inter-operability via an inter-access point protocol across distribution systems supporting IEEE 802.11 operation, IEEE Std 802.11F-2003). Leung and Kim [14] proposed a formulation based only on effective channel utilisation for solving a dynamic radio channel allocation. Ming *et al.* [15] used a heuristic based on the estimation of the interference for each access point. Luo and Shankaranarayanan applied a probabilistic algorithm for the same problem [13]. All those methods only uses (three) non-overlapping channels. Since the proliferation of private and commercial access points, interference problems arise. The three non-overlapping channels are no longer sufficient. However, in an environment of mobile users such as a campus, a fixed channel allocation may be inadequate. For example, an adequate throughput should be allocated in a meeting room when people are present, but throughput may fall when people leave that room, for the benefit of another crowded meeting room. That is why a solution of dynamic allocation of channels has to be found. The goal of this paper is therefore to propose a dynamic channel allocation in IEEE 802.11 networks by means of centralized iterative methods, allowing the use of all available channels in real time.

The paper is organized as follows. In Section 1, we present the modeling of channel assignment problem in 802.11 networks and state its complexity. We prove our problem's formulation to correspond to the *Minimum Interference Frequency Assignment Problem*, and hence the problem to be NP-hard. In Section 2, we propose two tabu search methods to solve the problem: $TabuObj$ which tries to minimize the true objective function $Obj$ associated with the model, and $TabuApproxObj$, which tries to minimize an approximate function of $Obj$, which is quicker to evaluate. Consequently, $TabuApproxObj$ will be quicker than $TabuObj$. In Section 3, we propose a third tabu search approach, called $TabuLevel$, which uses some graph coloring ingredients, that is much more quicker than $TabuApproxObj$. Numerical results and a comparison of the three methods are given in Section 4. We end the paper with a conclusion in Section 5.

## 1. Problem formulation and its complexity

We consider a set of nearby located access points (APs) as an undirected graph $G = (V, E)$, where the set of vertices $V$ is the set of APs, and there is an edge $[i, j]$ in $E$ if a transmission or reception on AP $i$ may generate some perturbation on AP $j$. The set of adjacent vertices of $i$ is denoted $N(i)$. A weight function $w : V \times V \to [0, 1]$ quantifies the amount of influence that two APs may have on each other. $w$ depends on the distance and on the medium between two APs. If two APs $i$ and $j$ with channels $c_i$ and $c_j$ respectively are only a few meters from each other, without any other medium than air, then the perturbation is maximal and is referred to as the theoretical perturbation $tp$. The measured perturbation $tp_{c_i,c_j}$ induced by two APs $i$ and $j$ depends on the difference between their assigned channel numbers $c_i$ and $c_j$ respectively, i.e. $|c_i - c_j|$. Table 1 represents these measures transformed into relative error rates.

Table 1. Theoretical perturbation

| Channel dist. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $tp$ | 0.37 | 1.0 | 0.56 | 0.3 | 0.16 | 0.11 | 0.08 | 0.06 | 0.04 | 0.03 | 0.02 | 0.01 | 0.005 |

We remark that two APs with the same channel cause each other less perturbation than if they are one channel away from each other. The inherent processus of 802.11 networks may explain this surprising fact: a computer can send a packet only if the medium is "free". Consider the situation in which two APs share the same channel. Therefore, the activity of one AP can be detected by the other. This is no more the case if the two APs are one channel away from each other: sent packets may collapse and therefore a perturbation occurs.

Let $A(i)$ be the activity of an AP $i$, then

$$A(i) = \frac{5 \cdot UR(i) + AR(i)}{6} \tag{1}$$

where the usage rate $UR(i)$ of AP $i$ is the ratio between the number of data frames sent or received by AP $i$ and the maximal number of such data frames, and the association rate $AR(i)$ of AP $i$ is the ratio between the number of associations (i.e. the number of devices associated with AP $i$) and the maximal number of devices that can be associated with AP $i$.

The objective function $Obj$ to minimize is described below and is accepted by some practitioners [18]. We show that $Obj$ can be rewritten as the standard *Minimum Interference Frequency Assignment Problem*, MI-FAP for short (see [1] for details on the MI-FAP).

The objective function $Obj$ to be minimized is the sum of three local objectives (see Equation (2)). The first part concerns the interferences on the AP $i$ itself; the second concerns the influence of AP $i$ on partners APs, which are in $N^+(i) \subseteq$

$N(i)$, and the last is the influence of AP $i$ on competitors APs, which are in $N^-(i) \subseteq N(i)$. Partner APs are supposed to be manageable, which is not the case for competitors APs. $\alpha, \beta, \gamma$ are parameters that define a strategy. For example, the choice $\alpha = \beta = 0.5$ and $\gamma = -0.5$ leads to an aggressive strategy where the goal is to improve its own network meanwhile trying to degrade the network of competitors. Let $x_{ic_i} = 1$ if we give channel $c_i$ to AP $i$, and $x_{ic_i} = 0$ otherwise. With such a notation, we can easily represent any channel assignments to the set of APs with a vector $x$.

$$
\begin{aligned}
&Obj(x)\\
&:= \min_x \sum_i \left( \alpha\ A(i) \sum_{j \in N(i)} A(j) \sum_{c_i, c_j} \frac{w_{ij} tp_{c_i c_j} x_{ic_i} x_{jc_j}}{\sum\limits_{l \in N(i)} w_{il}} \right.\\
&\quad \left. + \beta\ A(i) \sum_{j \in N^+(i)} \sum_{c_i, c_j} \frac{w_{ij} tp_{c_i c_j} x_{ic_i} x_{jc_j}}{\sum\limits_{l \in N^+(i)} w_{il}} + \gamma\ A(i) \sum_{j \in N^-(i)} \sum_{c_i, c_j} \frac{w_{ij} tp_{c_i c_j} x_{ic_i} x_{jc_j}}{\sum\limits_{l} \in N^-(i)} w_{il} \right) \qquad (2)\\
&= \min_x \sum_i \left( \sum_{j \in N(i)} \sum_{c_i, c_j} \alpha\ A(i) A(j) \frac{w_{ij} tp_{c_i c_j}}{\sum\limits_{l \in N(i)} w_{il}} x_{ic_i} x_{jc_j} \right.\\
&\quad \left. + \sum_{j \in N^+(i)} \sum_{c_i, c_j} \beta\ A(i) \frac{w_{ij} tp_{c_i c_j}}{\sum\limits_{l \in N(i)} w_{il}} x_{ic_i} x_{jc_j} + \sum_{j \in N^-(i)} \sum_{c_i, c_j} \gamma\ A(i) \frac{w_{ij} tp_{c_i c_j}}{\sum\limits_{l \in N(i)} w_{il}} x_{ic_i} x_{jc_j} \right)
\end{aligned}
$$

Let us define

$$
p'_{ijc_ic_j} = \begin{cases} \alpha A(i) A(j) \frac{w_{ij} tp_{c_i c_j}}{\sum\limits_{l \in N(i)} w_{il}} & \text{if } j \in N(i) \\ 0 & \text{otherwise} \end{cases}
$$

$$
p''_{ijc_ic_j} = \begin{cases} \beta A(i) \frac{w_{ij} tp_{c_i c_j}}{\sum\limits_{l \in N(i)} w_{il}} & \text{if } j \in N^+(i) \\ 0 & \text{otherwise} \end{cases} \qquad p'''_{ijc_ic_j} = \begin{cases} \gamma A(i) \frac{w_{ij} tp_{c_i c_j}}{\sum\limits_{l \in N(i)} w_{il}} & \text{if } j \in N^-(i) \\ 0 & \text{otherwise} \end{cases}
$$

Consequently,

$$\begin{aligned}
&Obj(x)\\
&= \min_x \sum_i \left( \sum_{j\in N(i)} \sum_{c_i,c_j} p^{'}_{ijc_ic_j} x_{ic_i} x_{jc_j} + \sum_{j\in N^+(i)} \sum_{c_i,c_j} p^{''}_{ijc_ic_j} x_{ic_i} x_{jc_j} + \sum_{j\in N^-(i)} \sum_{c_i,c_j} p^{'''}_{ijc_ic_j} x_{ic_i} x_{jc_j} \right)\\
&= \min_x \sum_{i,j} \sum_{c_i,c_j} \left( \underbrace{p^{'}_{ijc_ic_j} + p^{''}_{ijc_ic_j} + p^{'''}_{ijc_ic_j}}_{\tilde{p}_{ijc_ic_j}} \right) x_{ic_i} x_{jc_j}\\
&= \min_x \sum_{i,j} \sum_{c_i,c_j} \tilde{p}_{ijc_ic_j} x_{ic_i} x_{jc_j} \qquad (3)\\
&=: \text{MI-FAP}
\end{aligned}$$

As stated in [4], the MI-FAP is strongly NP-hard. Other formulations for 802.11 [14] networks have also been proposed in the literature.

## 2. Two tabu search procedures

Tabu search (TS), originally proposed by Glover [6, 7] and Hansen [10], is an iterative procedure that starts from an initial solution and iteratively search in its neighborhood a new solution. The modification of the current solution to get a neighbor solution is called a *move*. TS stops when a criterion is satisfied, and then the best solution found during the search is returned. To avoid cycling, each reverse move is forbidden during $r$ iterations, which is referred to as the duration of the tabu status (TL). As it may occur that some forbidden moves could lead to a best ever met solution, an *aspiration function* is usually used to cancel the forbidden attribute of a move in such cases. A basic version of the tabu algorithm is presented in Algorithm 1, where $NS(x)$ is the set of neighbor solutions of $x$.

Many variants and extensions of Algorithm 1 can be found in Glover and Laguna [8]. An example of a possible improvement of a tabu search is to use *reactive* tabu tenures, as proposed in [2] or in [3].

Tabu search adaptation to a specific problem needs the definition of the neighborhood structure (i.e. the nature of a move), the nature of the tabu status and the way to select a neighbor solution. In the considered problem, we propose solution $x'$ to be a neighbor of solution $x$ if we can obtain $x'$ from $x$ by changing the channel of a single AP $i$. Let $c_i$ and $c'_i$ be the channels of AP $i$ in solution $x$ and $x'$ respectively. The associated move is denoted by $(i, c_i, c'_i)$. Two types of tabu status are defined: (1) any other channel's allocation of AP $i$; (2) a reassignment of channel $c_i$ to AP $i$. In the former case the tabu list $TL$ is a set of AP's, whereas in the latter case $TL$ is composed by a set of forbidden assignments $(i, c_i)$. Preliminary experiments with different durations of tabu status showed that the use of the second type of tabu status leads to better results. Only this type of

**Algorithm 1** Tabu Search

```
generate an initial solution $x$
set $x^* = x$, $f^* = f(x)$, and $TL = \emptyset$
**repeat**
  determine $NS'(x) = \{x' \in NS(x)$ such that $x'$ is obtained from $s$ by performing a move that is not tabu, or such that $f(x') < f^*\}$
  set $x' = \arg \min_{x'' \in NS'(x)} f(x'')$
  update the tabu status and set $x = x'$
  **if** $f(x') < f^*$ **then**
    set $f^* = f(x')$ and $x^* = x'$
  **end if**
**until** a maximum number of iterations without improving $x^*$ have been performed
```

tabu status will be considered from now. In addition, when a move $(i, c_i, c_i')$ is performed in order to generate solution $x'$ from $x$, then we set the tabu duration of the move $(i, c_i', c_i)$ as follows, where $\text{UNIF}(a, b)$ generates, based on the uniform distribution, an integer number in the set $\{a, \ldots, b\}$:

- if $Obj(x') < Obj(x)$, then it is forbidden to give $c_i$ back to $i$ during $\text{UNIF}(5, 30)$ iterations;
- if $Obj(x') = Obj(x)$, then the duration of the tabu status is $\text{UNIF}(5, 20)$ iterations;
- if $Obj(x') > Obj(x)$, then the duration of the tabu status is $\text{UNIF}(5, 10)$ iterations.

Such a strategy is quite straightforward: if we perform a good move (i.e. we decrease $Obj$), then the reverse of such a move should be forbidden for a long time.

In order to choose a neighbor solution of $x$ we can evaluate:

(1) every possible move (such a strategy is called *exhaustive*);
(2) a random sample of moves;
(3) a selective sample of moves: some AP's are more interesting (i.e. have a stronger impact on the objective function) than others. More precisely an AP $i$ is *attractive* for a move if it meets one of the following criteria.
   - $A(i) \geq A_0$;
   - $\exists\, j \in N(i)$ such that $A(j) \geq A_0$, $w_{ij} \geq W_0$;

   where $W_0$ and $A_0$ are parameters. Then, among the set of moves $(i, c_i, c_i')$ such that AP $i$ is attractive, we select the best possible one minimizing $Obj$. Such a strategy is called *attractive sample*.

Preliminary experiments, which will, again, not be detailed here, showed that the attractive sample strategy performed better than the exhaustive and random ones within time constraints. The so obtained tabu search method will be called *TabuObj* from now and will be considered for comparisons in Section 4.

In order to evaluate a neighbor solution $x'$ of $x$, one may propose an incremental computation of $Obj$. Unfortunately, preliminary investigations showed us that it is not possible to implement such a computation in a more efficient way than performing the whole computation of $Obj(x')$. In order to evaluate a neighbor solution more quickly, we propose to use an approximate function *ApproxObj* of $Obj$, which is $\sum\limits_{[i,j]\in E} w_{ij} tp_{c_i c_j}$. The associated tabu search procedure will be called *TabuApproxObj*. In this case, an incremental computation of *ApproxObj* is possible.

In order to generate an initial solution, we start from an empty solution $x$, i.e. without any assignment. At each step, we choose an unassigned AP $i$ such that $i$ has the largest number of already assigned adjacent vertices. We break ties with the *degrees* (i.e. the degree of a vertex $i$ is the number of adjacent vertices to $i$) if there are several possibilities. As soon as $i$ is selected, we give to $i$ the best possible channel, i.e. the one minimizing the augmentation of $Obj(x)$. If there are many possible channels, we randomly choose one. We stop as soon as every AP has a channel. We call this procedure *GreedyBySaturation*.

## 3. A graph coloring approach

In *TabuObj* and *TabuApproxObj*, the evaluation of the neighbor solutions could be time consuming for large and dense networks. In order to accelerate the algorithm, we propose to work level by level and to use a quick objective function for each level. The idea is to adapt algorithms used in GSM networks [4] to WLAN networks.

Consider the graph $G = (V, E)$, where $V$ is the set of APs and there is an edge $[i, j]$ in $E$ if a perturbation between AP $i$ and AP $j$ may occur. Suppose first that all edges have the same weight. Let $c : V \mapsto \mathbb{N}^+$ be a function that assigns a positive integer (called *color*) to each vertex $v \in V$. Let $\delta : \mathbb{N}\times\mathbb{N} \mapsto \mathbb{N}$ be a function such that $\delta(p, q) = |p-q|$. Let $F$ be the set of forbidden values for $\delta$. The problem denoted by $Color(F)$, also called a *$T$-coloring problem* in [1]) consists in a color assignment to each vertex of $V$ such that $\delta(c(i), c(j)) \notin F$, $\forall [i, j] \in E$. If $F = \{0\}$ then the problem corresponds to the usual and well-known *graph coloring problem.* A solution which satisfies $Color(F)$ is said to be *admissible.* This generalization of vertex coloring to practical frequency assignment was first introduced by Hale [9] in the context of GSM networks.

As it is possible that no admissible solution exists according to some $F$, the goal can be changed into the minimization of the number of *conflicts* or *violations*, which is the number of times $\delta(c(i), c(j)) \in F$, with $[i, j] \in E$. Finding an admissible solution might be too ambitious. As interferences depend on the channel distances, successive problems can be solved, from the easiest one to the most difficult one: $Color(F_0), \ldots, Color(F_6)$, where $F_0 = \emptyset, F_1 = \{1\}, F_2 = \{1, 2\}, F_3 = \{0, 1, 2\}, F_4 = \{0, 1, 2, 3\}, F_5 = \{0, 1, 2, 3, 4\}, F_6 = \{0, 1, 2, 3, 4, 5\}$. That list of sets is built according to decreasing *tp* values of Table 1. Since there is theoretically no spectral density overlapping for channels distant of more than four units [17], it is not worth trying to solve $Color(F)$ for sets $F \supset F_6$ larger than $F_6$, i.e. including inter-channel distances larger than five. A solution which satisfies $F_k$ is said to be *$k$-admissible*. It is important to mention, for example, that a solution $x_2$ which is not 2-admissible could be better than a 1-admissible solution $x_1$, in the sense of the objective function $Obj$, i.e. $Obj(x_2) < Obj(x_1)$.

The strategy proposed in Algorithm 2 can be applied to search for 6-admissible solutions, i.e. to indirectly try to minimize $Obj$. Note that randomly assigning a channel to every AP yields a 0-admissible solution.

**Algorithm 2** Solving $Color(F)$

```
set $k = 0$
generate an initial solution using *GreedyBySaturation*
**while** a $k$-admissible solution of $Color(F_k)$ is found and $k < 6$, **do**
  set k = k+1
  try to find a $k$-admissible solution with $TabuLevel(F_k)$ (as specified below)
**end while**
Output: solution $x^*$ of highest admissibility level
```

For a fixed $k$, the main issue is now to propose a tabu search method for searching $k$-admissible solutions. Such a method is denoted by $TabuLevel(F_k)$ and is very close to *Tabucol*, a graph coloring tabu search algorithm proposed by Hertz and de Werra in [12], and improved by Galinier and Hao in [5]. The objective function $Conf^{(k)}(x)$ to minimize is simply the number of conflicts at level $k$ (i.e. associated with $F_k$) that are in solution $x$. Thus, $Conf^{(k)}(x)$ is much more quicker to evaluate than $ApproxObj(x)$. Even more, in order to evaluate a move $(i, c_i, c_i')$, we only have to compute $\Delta Conf^{(k)}(i, c_i, c_i')$, which is the difference between the number of additional conflicts and the number of removed conflicts that occur if we assign channel, i.e. color, $c_i'$ instead of $c_i$ to AP $i$. In order to select a neighbor solution, an *attractive sample* strategy is used (as it is the case for *Tabucol* in [5,12]): every AP $i$ such that $i$ is involved in a conflict at level $k$ is attractive. Then, among the set of moves $(i, c_i, c_i')$ such that AP $i$ is attractive, a non tabu move which minimizes $\Delta Conf^{(k)}(i, c_i, c_i')$ is selected. Ties are broken by using the accurate function $Obj$. At the end of Algorithm 2, the output solution, denoted by $x^*$, is the one associated with the highest admissibility level. The computation of $Obj(x^*)$ is then required to know the true value of $x^*$. Note that the duration of the tabu status is set as in [5] as follows: when channel $c_i'$ is given to AP $i$ instead of $c_i$ in order to generate a neighbor solution $x'$ from $x$, then it is forbidden to give channel $c_i$ to $i$ for $\text{UNIF}(0,9) + 0.6 \cdot n_c$ iterations, where $n_c$ is the number of conflicts associated with solution $x$.

We note that solving $Color(F_0)$ with only three non-overlapping channels (1, 6 and 11) corresponds to the problem as it is today usually handle by administrators. Such an approach was outperformed by our three tabu search methods.

Preliminary experiments showed that such a method does not perform well at all. We point out that the reason is the following: Algorithm 2 stops too early and it never tries to search for 6-admissible solutions. Thus, we compared Algorithm 2 with $TabuLevel(F_6)$ only, and we remarked that $TabuLevel(F_6)$ performed much better according to *Obj*! Consequently, only $TabuLevel(F_6)$ ($TabuLevel$ for short) will be used for comparisons in Section 4.

## 4. Numerical results and conclusion

In this section, we first describe the considered instances and then compare the three proposed methods: $TabuObj$, $TabuApproxObj$ and $TabuLevel$. All the experiments were done on a Pentium 3 (670 Mhz, 512 MB RAM). We will compare those methods according to the *Obj* objective function after 10 seconds of CPU time, 60 seconds, and 300 seconds. These values correspond to acceptable time-waiting period. For example, a meeting is held in a conference room, followed by a discussion in the coffee room. It is therefore necessary to have a sufficient throughput available first in the conference room, and then in the coffee room, within a sufficiently short period of time (e.g., 10 seconds, 1 minute, or even 5 minutes). The room that is not, or least used, can suffer from a lower throughput.

The density $d$ of a network (or a graph) is the average number of adjacent vertices of any single AP (or vertex) $i$. The number $n$ of vertices and the edge density $d$ define the size of an instance. Given $n$ and $d$, an instance is basically defined by the following information: for each AP $i$, we should know its activity and the group it belongs to (partners or competitors). For each link between a pair $\{i, j\}$ of APs, we should indicate its weight $w_{ij}$. The value $A(i)$ and $w_{ij}$ follow a uniform distribution in $[0, 1]$. Note that instead of such a way of simulating the activities, one may simulate or compute each of its component, namely the usage rate $UR(i)$ and the association rate $AR(i)$ (see Section 1 for more details). Note that the presented experiments were performed with $(\alpha, \beta, \gamma) = (3; 1; 0)$ (i.e. there is no competitor group), but our analysis still holds for different values of $\alpha, \beta$ and $\gamma$.

The obtained results are summarized in Table 2. For each instance, we indicate the *Obj* value obtained by the algorithm *GreedyBySaturation* (*GBS* for short). Such solutions are always obtained after a fraction of second. For each tabu search method (namely $TabuObj$, $TabuApproxObj$ and $TabuLevel$), we indicate the obtained value of *Obj* after 10 seconds (denoted by ”10 s”), 60 seconds and 300 seconds. Each value of *Obj* is actually an average over three executions of the considered method (one different seed for each execution). Different seeds have been used, which lead to similar values of *Obj* (within 1%). Therefore we will not give the results for each single seed. Note in addition that for each instance and each stopping criterion, the best results are written in bold style. The method which obtains the best results after 300 seconds is indicated in bold too.

In Figure (A), we compare the three tabu search methods on graphs with a density $d = 0.3$, $n \in \{10, 25, 50, 75, 100\}$ and a time limit of 10 seconds. The figure illustrates the improvement percentage of $Obj$ obtained by the considered method $GreedyBySaturation$. For example, for $n = 25$, $TabuObj$ was able to improve the $Obj$ value obtained by $GreedyBySaturation$ by about 30 %. The same elements are showed in Figures (B), (C) and (D), but with different densities, which are respectively 0.5, 0.8 and 1. In a real situation as a university campus, the density of a network of access points is not very high (about 0.1 to 0.3). The greatest the network is, the lower its density. However, in urban areas, it is not uncommon that network density raised locally: a dozen of access points may cause interferences.

We can generally remark that the larger $n$ is, the closer the methods are according to the improvements they obtain. Another observation is the peak for improvement around 25 access points. Both observations are due to the complexity of the problem, and to the time limitation constraint for its resolution. With few access points and a low density of possible interactions, the problem could still be treated satisfactorily in a very short time (reminder: 10 seconds time limitation). But this is no longer the case when the number of access points increases.

Figure (A): density 0.3, CPU time 10 seconds

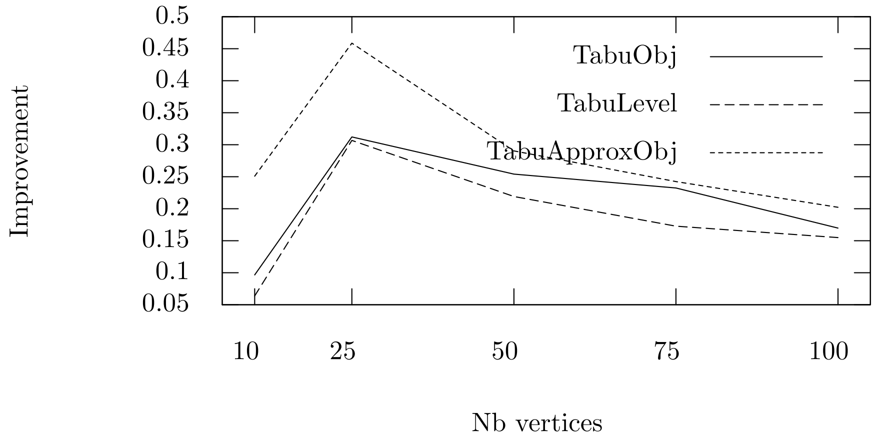


Figure (B): density 0.5, CPU time 10 seconds

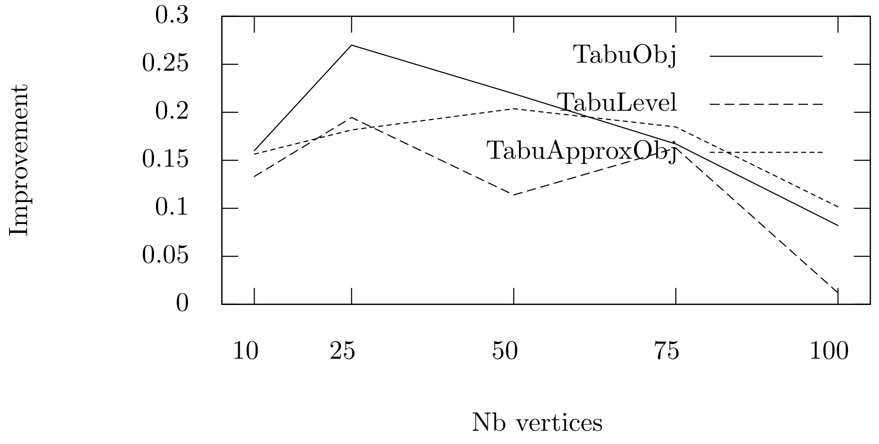


Figure (C): density 0.8, CPU time 10 seconds

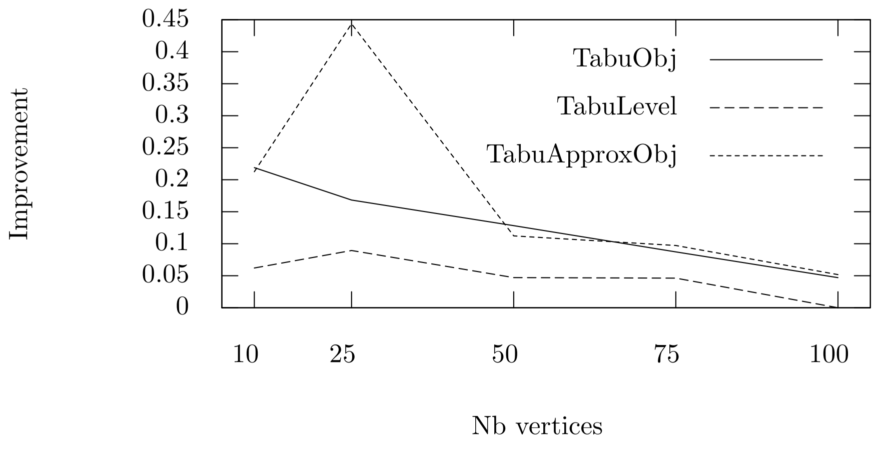
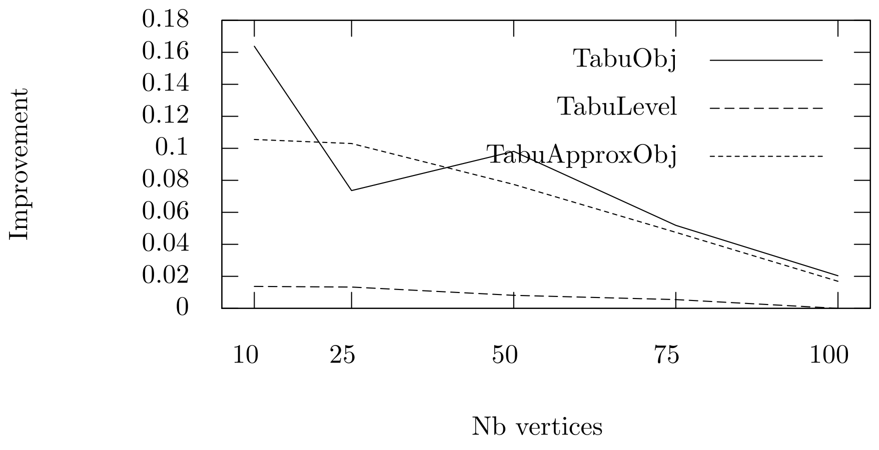


Figure (D): density 1.0, CPU time 10 seconds



We can first easily observe from Table 2 that: (1) every tabu search method is able to improve the results obtained by *GreedyBySaturation*; (2) the larger $n$ is, the more often the method will provide a better solution if we run it for a longer time; (3) *TabuLevel* is never as good as the two other tabu search methods, except if $n = 1000$, where it is better than *TabuObj*; (4) *TabuObj* and *TabuApproxObj* are comparable methods, except if $n = 1000$, i.e. on a very large network, where *TabuApproxObj* is better.
If we have a closer look on *TabuApproxObj* and *TabuObj*, we can remark that

if $n > 50$ or if we only allow 10 seconds, $TabuApproxObj$ is generally better than $TabuObj$. This indicates that if the practitioners evolve towards real time computations and decisions, $TabuApproxObj$ is the most appropriate algorithm among the ones presented here. Note that on these instances, $TabuLevel$ is not competitive with the two other tabu search algorithms, which means that straightforward adaptation of efficient algorithms in GSM networks (see [11,16]) to WLAN networks are not appropriate. However, we feel that it may be appropriate for networks with very large value of $n \cdot d$ and for real time computation (i.e. when only one or two seconds are allowed), as it is able to visit a very large amount of solutions (between 3 and 5 times more solutions are visited compared to TabuObj).

## 5. Conclusion

In this paper, we propose a modeling that enables optimization of channel assignment with respect to the dynamic behavior of end-users in the context of wireless LAN using IEEE 802.11 networks. As we prove the considered problem to be NP-hard, the use of heuristics is appropriate to tackle it. We developed three tabu search algorithms, namely $TabuObj$, $TabuApproxObj$ and $TabuLevel$, where each algorithm has its own objective function. Depending on the context, the user may choose a different algorithm. $TabuLevel$ is more appropriate for very large instances or in situations where on-line optimization is required, because the associated objective function is very easy to compute. In contrast, $TabuObj$ uses the true objective function to minimize, which is more accurate but very cumbersome. $TabuApproxObj$ is a compromise between $TabuObj$ and $TabuLevel$.

GSM networks uses a dedicated frequency band, only available to operators, who can therefore completely optimize their network. 802.11 networks no more use dedicated private bandwidth, as today 80.11 a/b/g networks of for the upcoming 802.11n protocol. With the proliferation of 802.11 devices, operators have to face new challenges, among other the management of dynamic frequency allocation with the increased difficulty that, on the contrary to the GSM network, it is no more possible to control all aspects of the network. Therefore solutions to dynamic allocation of networks such as the one presented in this paper may contribute to improve the network quality and customers' satisfaction. Further area of research can be, together with the frequency allocation problem, the network power management as well as predictive methods for dynamic environment.

## References

[1] K.I. Aardal, S.P.M. van Hoesel, A.M.C.A. Koster, C. Mannino, and A. Sassano. Models and solution techniques for frequency assignment problems. *4OR*, 1:261–317, 2003.

[2] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6:126–140, 1994.

[3] I. Bloechliger and N. Zufferey. A Graph Coloring Heuristic using Partial Solutions and a Reactive Tabu Scheme. *Computers & Operations Research*, 35:960 975, 2008.

[4] A. Eisenblätter, M. Grötschel, and A. M. C. A. Koster. Frequency assignment and ramifications of coloring. *Discussiones Mathematicae Graph Theory*, 22:51–88, 2002.

[5] P. Galinier and J.-K. Hao. Hybrid Evolutionary Algorithms for Graph Coloring. *Journal of Combinatorial Optimization*, 3:379–397, 1999.

[6] F. Glover. Tabu search - part I. *ORSA Journal on Computing*, 1:190–205, 1989.

[7] F. Glover. Tabu search - part II. *ORSA Journal on Computing*, 2:4–32, 1990.

[8] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Boston, 1997.

[9] W. K. Hale. Frequency assignment: Theory and applications. In *Proceedings of the IEEE*, volume 68, pages 1497–1514, 1980.

[10] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. In *Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy*, 1986.

[11] Jin-Kao Hao, Raphaël Dorne, and Philippe Galinier. Tabu search for frequency assignment in mobile radio networks. *Journal of Heuristics*, 4(1):47–62, 1998.

[12] A. Hertz and D. de Werra. Using Tabu Search Techniques for Graph Coloring. *Computing*, 39:345–351, 1987.

[13] L. Hui and N.K. Shankaranarayanan. A distributed channel allocation technique for throughput improvement in a dense wlan environment. In *Acoustics, Speech, and Signal Processing*, volume 5, pages V–345–8, 2005.

[14] K.K. Leung and B.J. Kim. Frequency assignment for ieee 802.11 wireless networks. In *Vehicular Technology Conference*, volume 3, pages 1422–1426, 2003.

[15] Y. Ming, N. Karmarkar, and A. Malvankar. A dynamic radio channel allocation scheme for wireless lans. In *IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, pages 17–20, 2005.

[16] R. Montemanni, J.N.J. Moon, and D.H Smith. An improved tabu search algorithm for the fixed-spectrum frequency-assignment problem. *IEEE Transactions on Vehicular Technology*, 52:891–901, 2003.

[17] P802.11. *IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specification*, 1999. http://ieee802.org/11/.

[18] D. Rossier, S. Varone, J.-F. Wagen, F. Gamba, V. Inguscio, and E. Marchon. System für die dynamische zuweisung von trägerfrequenzen zu zugriffspunkten eines lokalen funknetzes. Patent 03405356.1, May 2003.

Table 2. Results obtained by *GreedyBySaturation* (*GBS*), *TabuObj*, *TabuApproxObj* and *TabuLevel*.

| **n** | **d** | **Method** | **GBS** | **10 s** | **60 s** | **300 s** |
|---|---|---|---|---|---|---|
| 1000 | 0.01 | TabuObj | 184.99 | 184 | 183.01 | 182.04 |
| | | TabuLevel | | 179.55 | 168.2 | 151.66 |
| | | **TabuApproxObj** | | **179.18** | **159.59** | **134.13** |
| 100 | 0.3 | **TabuObj** | 22.89 | 19 | **17.32** | **17.32** |
| | | TabuLevel | | 19.34 | 19.61 | 19.61 |
| | | TabuApproxObj | | **18.26** | 17.92 | 17.55 |
| 100 | 0.5 | **TabuObj** | 23.07 | 21.18 | **19.34** | **19.18** |
| | | TabuLevel | | 22.8 | 20.58 | 20.52 |
| | | TabuApproxObj | | **20.73** | 19.43 | 19.25 |
| 100 | 0.8 | **TabuObj** | 22.2 | 21.16 | 19.98 | **19.63** |
| | | TabuLevel | | 22.2 | 20.91 | 20.91 |
| | | TabuApproxObj | | **21.05** | **19.9** | 19.7 |
| 100 | 1 | **TabuObj** | 20.31 | **19.9** | **19.45** | **19.28** |
| | | TabuLevel | | 20.31 | 20.17 | 20.17 |
| | | TabuApproxObj | | 19.97 | 19.51 | 19.44 |
| 75 | 0.3 | TabuObj | 15.27 | 11.72 | 11.52 | 11.52 |
| | | TabuLevel | | 12.63 | 12.33 | 12.33 |
| | | **TabuApproxObj** | | **11.57** | **11.18** | **11.18** |
| 75 | 0.5 | TabuObj | 16.82 | 14.01 | 13.38 | 13.38 |
| | | TabuLevel | | 14.18 | 14.13 | 14.13 |
| | | **TabuApproxObj** | | **13.71** | **13.37** | **13.14** |
| 75 | 0.8 | **TabuObj** | 17.26 | 15.75 | **14.94** | **14.94** |
| | | TabuLevel | | 16.46 | 16.19 | 16.19 |
| | | TabuApproxObj | | **15.58** | 15.06 | 14.98 |
| 75 | 1 | TabuObj | 14.14 | **13.4** | **13.02** | 13.02 |
| | | TabuLevel | | 14.06 | 14.03 | 14.03 |
| | | **TabuApproxObj** | | 13.46 | 13.24 | **13** |
| 50 | 0.3 | TabuObj | 9.77 | 7.29 | 7.29 | 7.29 |
| | | TabuLevel | | 7.63 | 7.63 | 7.63 |
| | | **TabuApproxObj** | | **6.94** | **6.89** | **6.89** |
| 50 | 0.5 | **TabuObj** | 10.1 | **7.88** | **7.88** | **7.88** |
| | | TabuLevel | | 8.95 | 8.95 | 8.95 |
| | | TabuApproxObj | | 8.04 | 7.99 | 7.99 |
| 50 | 0.8 | TabuObj | 12.04 | 10.5 | 10.5 | 11.51 |
| | | TabuLevel | | 11.57 | 11.51 | 11.51 |
| | | **TabuApproxObj** | | **10.69** | **10.3** | **10.3** |
| 50 | 1 | TabuObj | 10.97 | **9.9** | **9.9** | **9.9** |
| | | TabuLevel | | 10.88 | 10.88 | 10.88 |
| | | **TabuApproxObj** | | 10.12 | 9.99 | 9.93 |
| 25 | 0.3 | TabuObj | 3.64 | 2.51 | 2.51 | 2.51 |
| | | TabuLevel | | 2.53 | 2.53 | 2.53 |
| | | **TabuApproxObj** | | **2.82** | **2.82** | **2.82** |
| 25 | 0.5 | **TabuObj** | 5.16 | **3.77** | **3.77** | **3.77** |
| | | TabuLevel | | 4.16 | 4.1 | 4.1 |
| | | TabuApproxObj | | 4.22 | 4.22 | 4.22 |
| 25 | 0.8 | TabuObj | 5.66 | 4.71 | 4.71 | 4.71 |
| | | TabuLevel | | 5.16 | 5.16 | 5.16 |
| | | **TabuApproxObj** | | **3.52** | **3.52** | **3.1** |
| 25 | 1 | TabuObj | 4.64 | 4.3 | 4.3 | 4.3 |
| | | TabuLevel | | 4.58 | 4.58 | 4.58 |
| | | **TabuApproxObj** | | **4.16** | **4.16** | **4.16** |
| 10 | 0.3 | TabuObj | 0.75 | 0.68 | 0.68 | 0.66 |
| | | TabuLevel | | 0.7 | 0.7 | 0.7 |
| | | **TabuApproxObj** | | **0.56** | **0.56** | **0.56** |
| 10 | 0.5 | **TabuObj** | 1.38 | **1.16** | **1.16** | **1.16** |
| | | TabuLevel | | 1.2 | 1.2 | 1.2 |
| | | TabuApproxObj | | 1.17 | 1.17 | 1.17 |
| 10 | 0.8 | **TabuObj** | 1.79 | **1.4** | **1.4** | **1.4** |
| | | TabuLevel | | 1.68 | 1.68 | 1.68 |
| | | TabuApproxObj | | 1.41 | 1.41 | 1.41 |
| 10 | 1 | **TabuObj** | 1.7 | **1.42** | **1.42** | **1.41** |
| | | TabuLevel | | 1.68 | 1.68 | 1.68 |
| | | TabuApproxObj | | 1.52 | 1.52 | 1.52 |