Urban planning for Radio Communications

SP 2

Measurements of propagation and cohabitation in Île-de-France

Task 2.1.3

***Dynamic Spectrum Allocation algorithms* (V2)**

Edition 2 04/12/2009

Editions

Edition	Date	Authors	Comments
Ed. 0	20/04/09	P. Tortelier (FT)	Initial version of TOC
Ed. 1	01/09/09	P. Tortelier (FT)	Description of the DSA Algorithm 2
Ed. 2	04/12/09	R.King(Py-automation)	Description of the DSA Algorithm 3

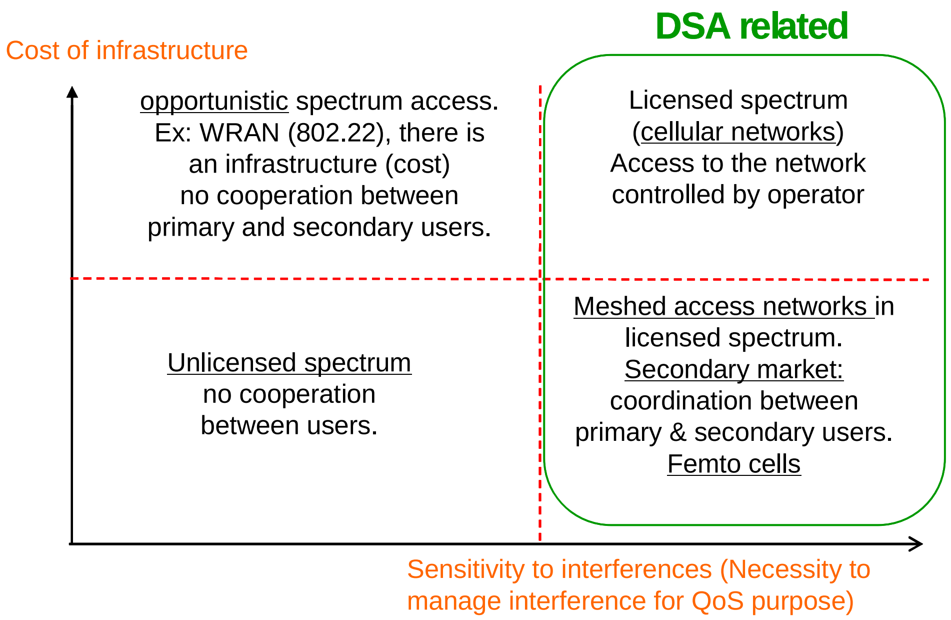
Contents

1 Introduction

This document is a sequel of a first version of the same deliverable D2.1.3 [1] where a DSA algorithm (hereafter referred as **algorithm 1**) was described and implemented by FT. As DSA is a major component of a better use of spectrum, it is worth to compare several algorithms based on different methods. It is therefore advisable to study, implement and compare performances of other DSA solutions. This new version of the deliverable will propose two such solutions.

The approach we describe in this document is a centralized one: a central entity has some information on the expected demand in spectrum, as a function of space and time; this entity regularly updates the way spectrum is allocated with respect with time and space, in order to use spectrum with the greatest possible efficiency. As a consequence, there is an underlying assumption that we are in what is called a dynamic exclusive use in the classification of [2].

We give an alternative two dimensional representation of the various spectrum access schemes which are located with respect to two criterion: the cost of deployment (cost of infrastructure) and the necessity to control interferences in order to achieve a required QoS. DSA relevant schemes are related to licensed spectrum required to manage interferences. This encompasses exclusive uses of [2] as well "private commons" which is also described in this paper. Both schemes are related to licensed spectrum.



2 DSA algorithm 2: Min Interference DSA by Tabu Search

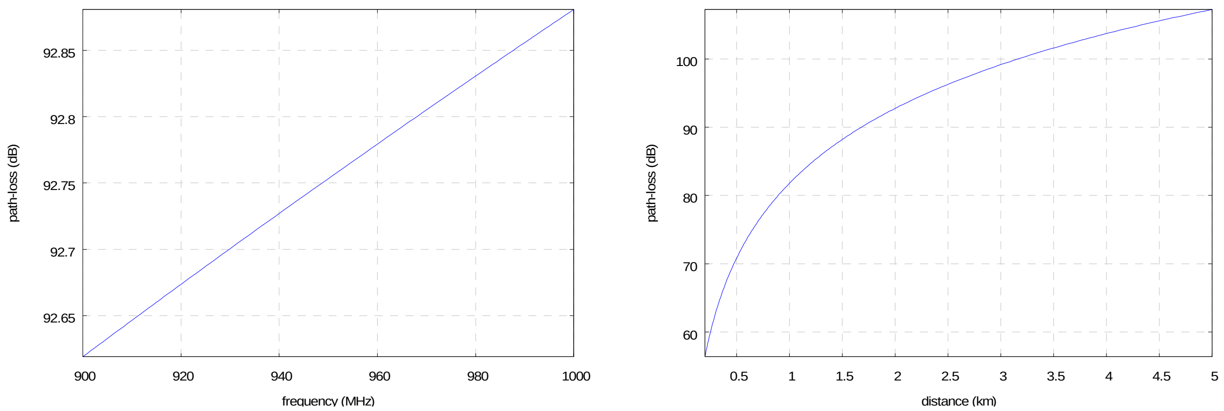
The first new algorithm addressed in this document was in fact partially described in [1] section 7 (Theoretical limits of DSA performance) not as a solution but as a mean to assess what can be achieved by a DSA algorithm. The algorithm was initially designed in the context of a third party (spectrum broker) which holds a license for a given frequency band and grants temporary licenses to wireless operators [2]-[5]. This frequency bands is used like a pool of frequency, and is called Coordinated Access Band (CAB) in the aforementioned publications. This broker try to find an allocation that fulfills the spectrum demands of operators while minimizing overall interference (**Min-Interference DSA**).

In this document we have a slightly different approach: a wireless operator has a license for an amount of spectrum, and we suppose a flexible regulation which is neutral with respect to the radio technology to be used in frequency bands. This operator tries to optimize its spectrum use while minimizing overall interference; this is a Min-Interference DSA. We describe this algorithm and an implementation using Tabu search.

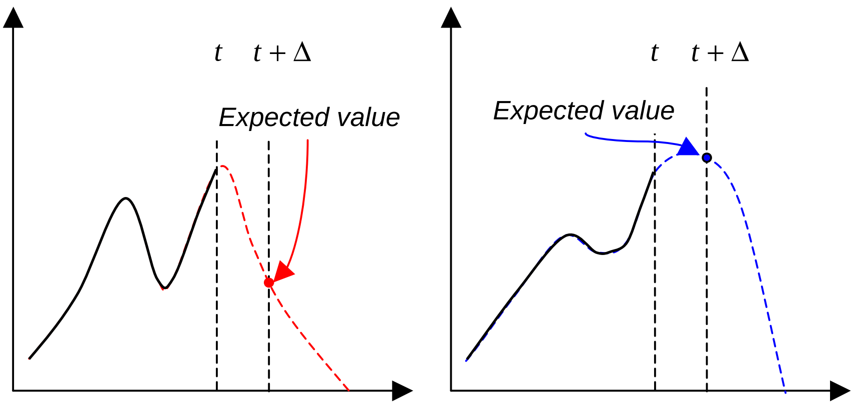
2.1 System model and assumptions

We remind in this section some basic assumptions we make.

- First of all, we assume we know the position (X_i, Y_i) of every base station, $i = 0 \cdots N_{BTS} - 1$ where N_{BTS} is the number of base stations. As a consequence we also know the propagation losses between any pair of nodes of the network because it depends on the distance $d_{i,j}$ of nodes i and j, be it the result of a path loss model (e.g. Hata path loss model) or the values given by terrain propagation models as in the URC simulator.
- Moreover, as this path loss depends much more heavily on the distance than on the frequency, we make the assumption of a constant path loss over the frequency band *B* of interest. The following figures are examples; we assume heights of transmitter and receiver are equal to 20m; left picture is the variation with respect to frequency on a 100 MHz width frequency band and at a distance equal to 1km (a total variation less than 0.3 dB on the band), while right picture is for a central frequency of 950 MHz and a distance varying from 0.2 to 5 km (a variation greater than 50 dB).



- We assume we know the demand in spectrum as a function of space and time, that is to say we have a spatio-temporal traffic model and a mean to convert traffic value to a demand in spectrum. Due to this spatial and temporal variation of traffic we should consider DSA algorithms taking into account each new demand as it appears. This is a too complex problem, and as the traffic model can be used to predict the evolution of demands, we envision rather a batch processing where all spectrum allocations are regularly computed (for instance hourly) for a given region.



Two traffic models corresponding to two different zones; at time t *a batch processing compute a new spectrum allocations corresponding to expected demands at time* $t+\Delta$*, taking advantage of spatial variations (red curve is different from blue curve)*

- We suppose that the total available bandwidth *B* is divided in *N* sub-bands $B_i, i=0\cdots N-1$ of equal width. To each node i of the network is associated a binary vector $\mathbf{f}_i=\left(f_{i,0}, f_{i,1},\cdots, f_{i,N-1}\right), i=0\cdots N_{BTS}-1$ where $f_{i,k}=1$ if sub-band B_k is used at node i, otherwise $f_{i,k}=0$. This binary vector $\mathbf{f}_i=\left(f_{i,0}, f_{i,1},\cdots, f_{i,N-1}\right)$ is the set of **active frequencies** at node i. The demand constraints can therefore be written as $\sum_k f_{i,k}=DEM(i), i=0\cdots N_{BTS}-1$ where $DEM(i)$ is the number of sub-bands required at node i to satisfy the spectrum demand.

2.2 The interference graph

The main problem we face when trying to spatially reuse frequency bands is interference. We look for spectrum allocation solutions which minimize the sum of interference at each node of a network while compliant with demands (frequency band needed at each node). This is a Min-Interference problem, as described in [3][4][5].

We first give a short description of the interference graph which is in fact a constraint graph whose nodes are the base stations of the network. Two nodes will be connected by an edge if they interfere when using a same frequency band: with this definition all nodes are connected to all other nodes (the graph would be a clique). If we remind that attenuation increases greatly with distance (see **Erreur ! Source du renvoi introuvable.**) "neighbor" nodes will contribute to the interference more heavily than distant nodes. To capture this phenomenon we introduce a threshold (in dB) and say that two nodes will be connected by an edge if the path loss (in dB) between them is less than this threshold.

Obviously the greater the threshold the greater the number of edges in the graph, this is depicted in Figure 1 and Figure 2 for a network of 150 nodes and thresholds respectively equal to 125 dB and 130 dB.

Each edge of this graph is weighted by the value (in a linear scale) of the attenuations between nodes, we will denote $g_{i,j}$ the weight of an edge between nodes i and j; we also make use of the neighborhood $V(i)$ of node i, the set of nodes j such that there is an edge between i and j.

The interference graph allow to account for the overall interference at a given node; if we denote $P_{j,k}$ the transmitted power of node j in sub-band B_k when (i.e., when $f_{j,k}=1$) then the overall interference at node i is given by:

$$I_i=\sum_{j\in V(i)}\sum_k\left\{g_{i,j}(k)\, f_{i,k}\, f_{j,k}\, P_{j,k}\right\}$$

Where the term $f_{i,k}\ f_{j,k}$ is non zero when both nodes i and j are using the same frequency band and potentially interfere. Using the approximation $g_{i,j}(k) \approx g_{i,j}$ (path loss near constant on the frequency band *B*) then I_i is roughly equal to:

$$I_i \approx \sum_{j \in V(i)} g_{i,j} \sum_k \left\{ f_{i,k}\ f_{j,k}\ P_{j,k} \right\}$$

Which can be further simplified if we suppose, to begin with, that all nodes use the same radio technology with the same transmit power *P*:

$$I_i \approx P \times \sum_{j \in V(i)} g_{i,j} \sum_k \left\{ f_{i,k}\ f_{j,k} \right\}$$

This last identity makes apparent the quantity $\sum_k \left\{ f_{i,k}\ f_{j,k} \right\}$ which is simply the number of conflicts between nodes i and j, equal to the number of frequency band used jointly by these nodes.

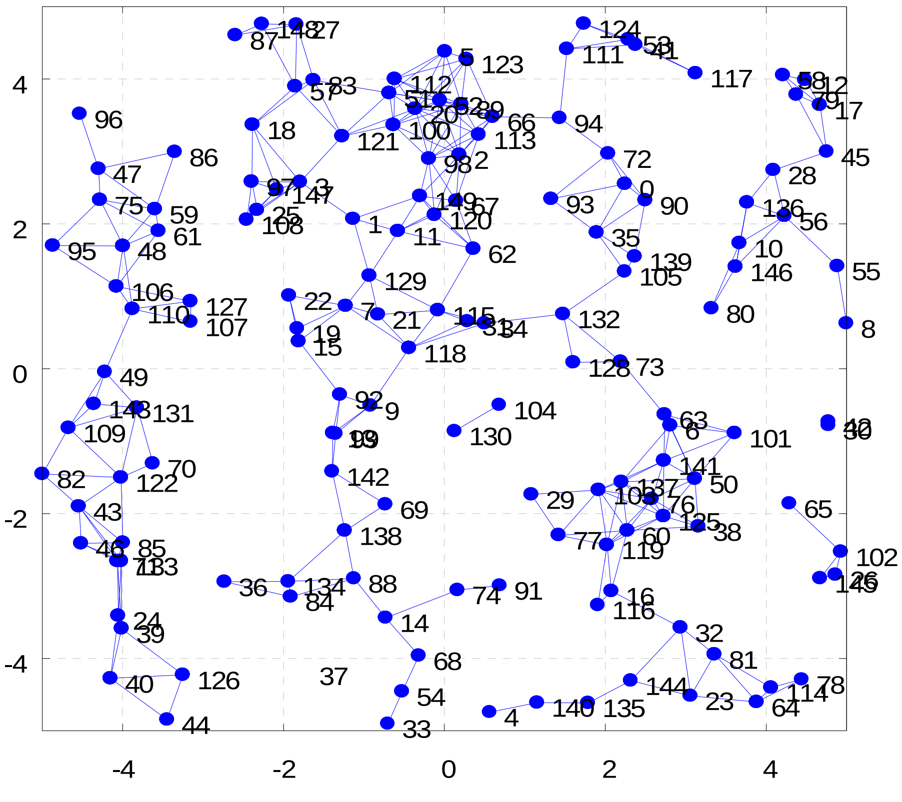


Figure 1: interference graph, threshold = 125 dB

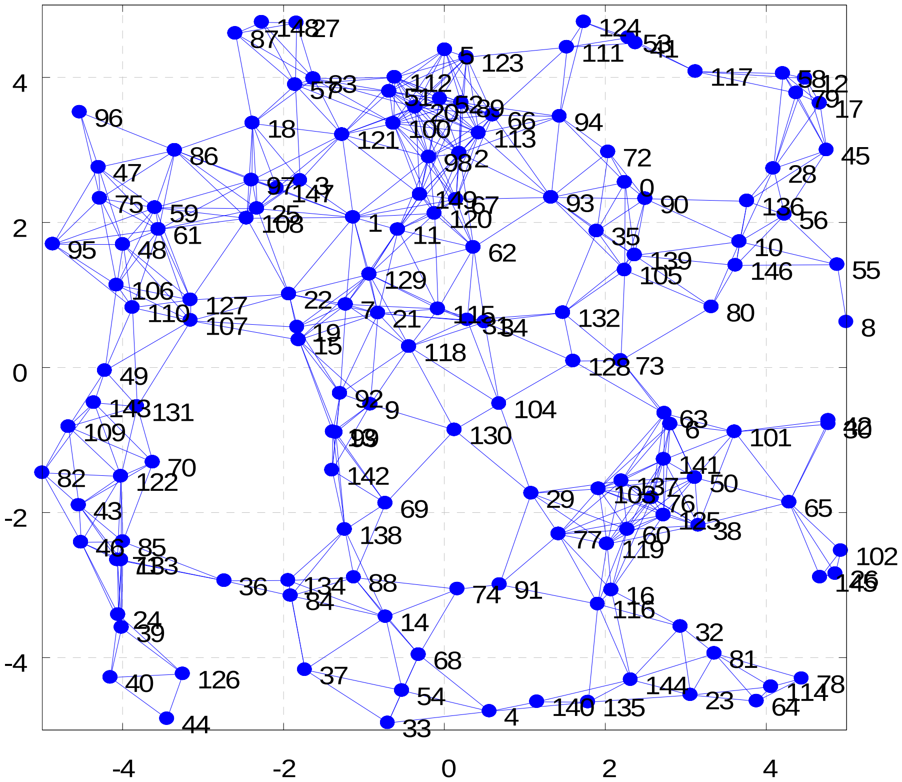


Figure 2: interference graph, threshold = 130 dB: more edges.

2.3 The optimization problem

With our assumptions and notations the problem we have to solve is to minimize the overall interference (the cost function)

$$\min_{\{f_{i,k}\}} \sum_i I_i \approx P \times \sum_i \left(\sum_{j \in V(i)} g_{i,j} \sum_k \{f_{i,k}\ f_{j,k}\} \right)$$

The minimization is performed over the set of binary vectors $\{f_{i,k}\}$ (N_{BTS} vectors of N bits) subject to the demand constraints:

$$\sum_k f_{i,k} = DEM(i), i = 0 \cdots N_{BTS} - 1$$

The search space is thus a subset of $\{0,1\}^{N_{BTS} \times N}$; its dimension is generally huge (consider for instance 150 nodes and 50 frequencies), an exhaustive search is impossible. The problem as we have stated it is a frequency assignment problem (FAP) with binary variables $f_{i,k}$. Although some software packages for this kind of problem are available they generally lead to very time consuming runs, see for example [5] section 8.1 where GLPK and DSDP where used to obtain performance metrics and where the authors said that they where not able to evaluate networks with more the 50 nodes due to *unreasonably long computation time*.

We therefore have to consider a different class of optimization algorithms generalizing local search. A number of possibilities exist (simulated annealing, genetic algorithms, memetic algorithms,…) and we decided to try Tabu Search (TS) algorithms introduced by Glover in [6][7], one of the reason that led us to this choice is that TS is already in use for many FAP variants [10][11]. Many tutorials on Tabu Search are available, we have used [9] and references therein.

2.3.1 A remark

As described above the available bandwidth is an input of the algorithm which then tries to minimize a criterion (the sum of all interference terms) within an already allocated frequency band.

Generally we are also interested in minimizing the required bandwidth; a simple way to proceed without implementing an iterative process where the allowable bandwidth shrinks at each iteration (we then need a termination criterion: a threshold of allowable total interference) is to introduce the used bandwidth in the criterion, which becomes

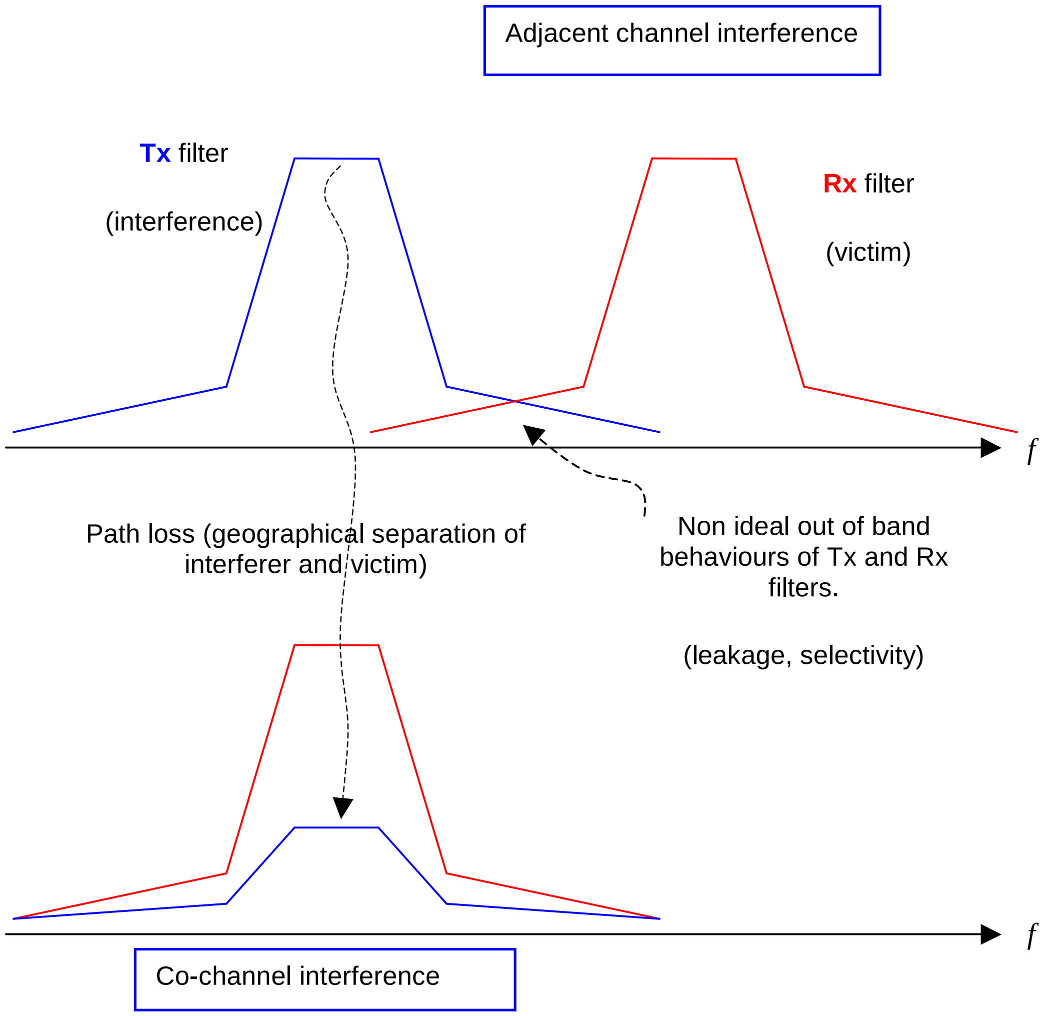
$$objective = \min_{\{f_{i,k}\}} \sum_i I_i + \lambda \left(\max_{i,k}\left\{f_{i,k} \neq 0\right\} - \min_{i,k}\left\{f_{i,k} \neq 0\right\} \right)$$

where λ is the weight of the total frequency span in the objective function. This parameter should be tuned according to the importance given to the need in shrinking used spectrum with respect to the interference criterion.

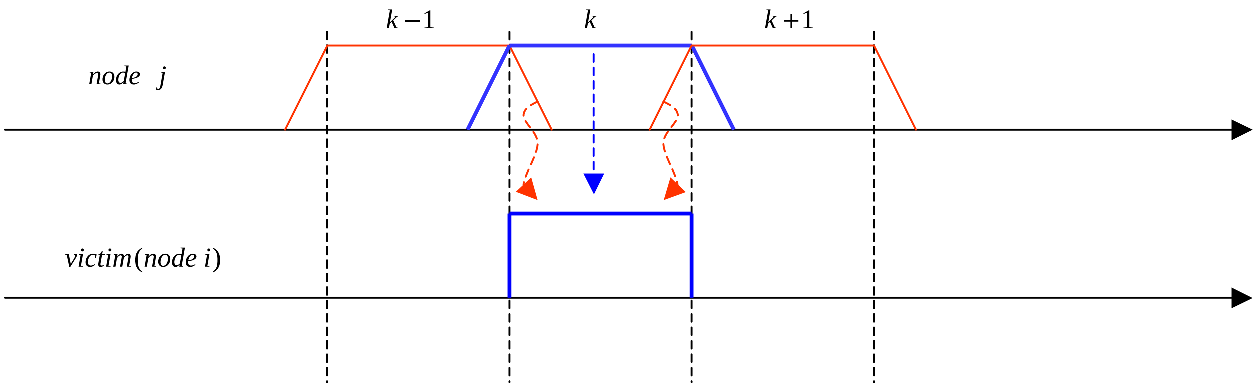
This is a possible source of discrepancy when comparing different algorithms: the need of taking the frequency span into account must be clearly stated in the scenario.

2.4 Transmitter impairments

The problem as we described it accounts only for co-channel interference; it is possible to broaden its scope so as to account for some real-world impairments of transmitters and receivers, the simplest to take into account for our problem being the non ideal behaviors of transmit and receive filters which do not have a brick wall transition band from pass band to cut band (leakage, selectivity).



$$I_i \approx P \times \sum_{j \in V(i)} g_{i,j} \sum_k \left\{ \alpha\, f_{i,k}\, f_{j,k-1} + f_{i,k}\, f_{j,k} + \alpha\, f_{i,k}\, f_{j,k+1} \right\}$$



$$I_i = \sum_{j \in V(i)} \left\{ \sum_k g_{i,j}(k)\, f_{i,k} \left(\alpha_{-1} f_{j,k-1} + f_{j,k} + \alpha_{+1} f_{j,k+1} \right) \right\}$$

$$I_i \approx \sum_j g_{i,j} \underbrace{\left\{ \sum_k f_{i,k} \left(\alpha_{-1} f_{j,k-1} + f_{j,k} + \alpha_{+1} f_{j,k+1} \right) \right\}}_{co-channel\ +adjacent\ channels}$$

$$\alpha_{\pm 1} << 1$$

Generally, coefficients α_{-1}, α_{+1} are equal (due to symmetry of transmit filters with respect to central frequency) and are very small, so that we can neglect them in a first approximation and do not implement this functionality in the current version of the DSA-v2 algorithm, due to a tight schedule.

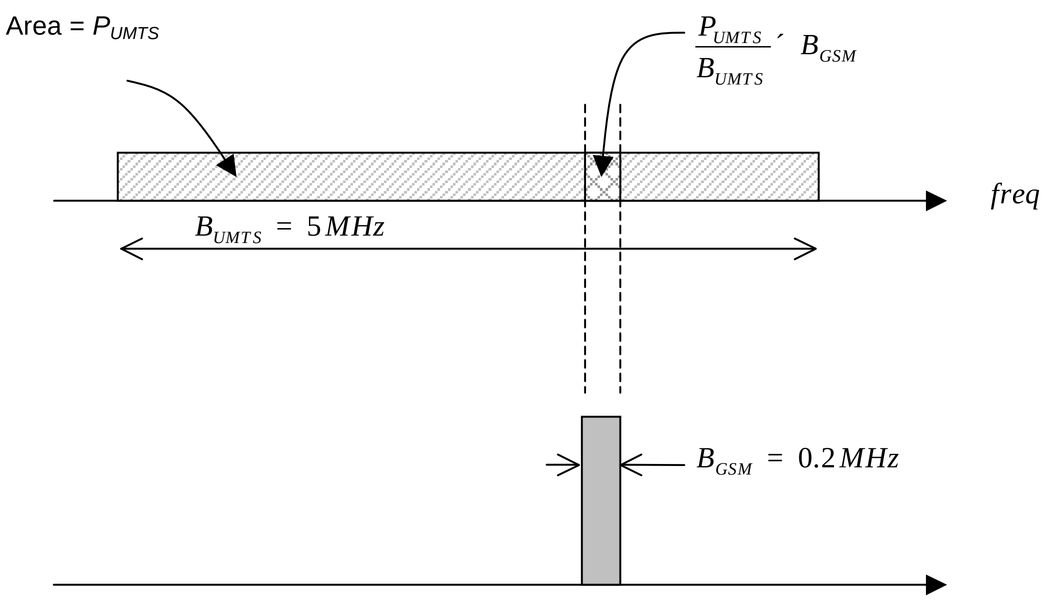
We keep in mind it should be included if we are to consider two systems with very different transmit powers, for instance if one of them is a TV transmitter with far greater transmit power than cellular base stations.

2.5 Managing GSM and UMTS signals

Scenario under consideration in URC project involves GSM and UMTS transmitters, we then have to account for it when computing interferences between both systems.

2.5.1 Interference from a UMTS BTS on a GSM one

This is the simplest case ; the power of a UMTS signal is spread over a 5 MHz frequency band ; we just have to account for the signal power falling into the narrow band of a GSM signal, $P_{UMTS} / B_{UMTS} \times B_{GSM}$, as is depicted below. With the propagation attenuation between UMTS transmitter and GSM receiver, the interference UMTS α GSM est is given by $PL_{linéaire} \times P_{UMTS} / B_{UMTS} \times B_{GSM}$



2.5.2 A short reminder on CDMA

Narrowband signals[1] have the general following representation

$$s(t) = \mathrm{Re}\left[s_b(t)\exp(2\pi i f_0 t) \right]$$

where f_0 is the central frequency (carrier frequency) of the transmitted signal, for example $f_0 = 2.45GHz$. The shape of the power spectral density is given by its baseband signal $s_b(t)$ (the complex envelop of the signal) which, for digital modulations, has the general form

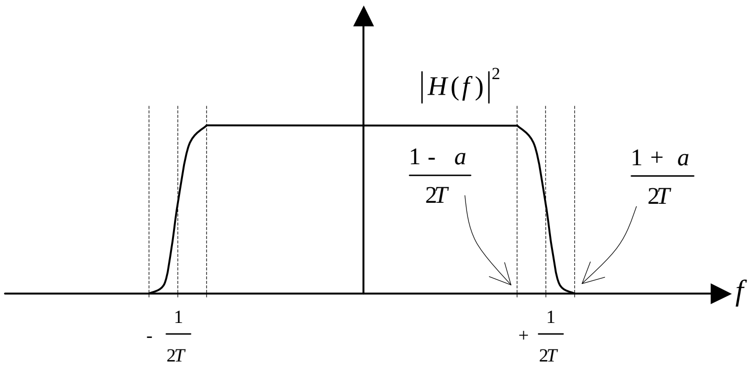
$$s_b(t) = A \times \sum_n d_n\, h(t - nT),\ d_n \in \mathbb{C}$$

$$E\left[d_n \overline{d_m} \right] = 0 \text{ si } m \neq n$$

This is the sum of shifted versions of the pulse $h(t)$ modulated by the complex symbols d_n of the modulation (BPSK : $d_n \in \{-1,+1\}$, QPSK : $d_n = (\pm 1 \pm i)/\sqrt{2}$, 16-QAM,...). The power spectral density (psd) of the baseband signal is given by the formula (Bennett) :

$$\Gamma(f) = \frac{A^2}{T} \times E\left[|d_n|^2 \right] \times |H(f)|^2$$

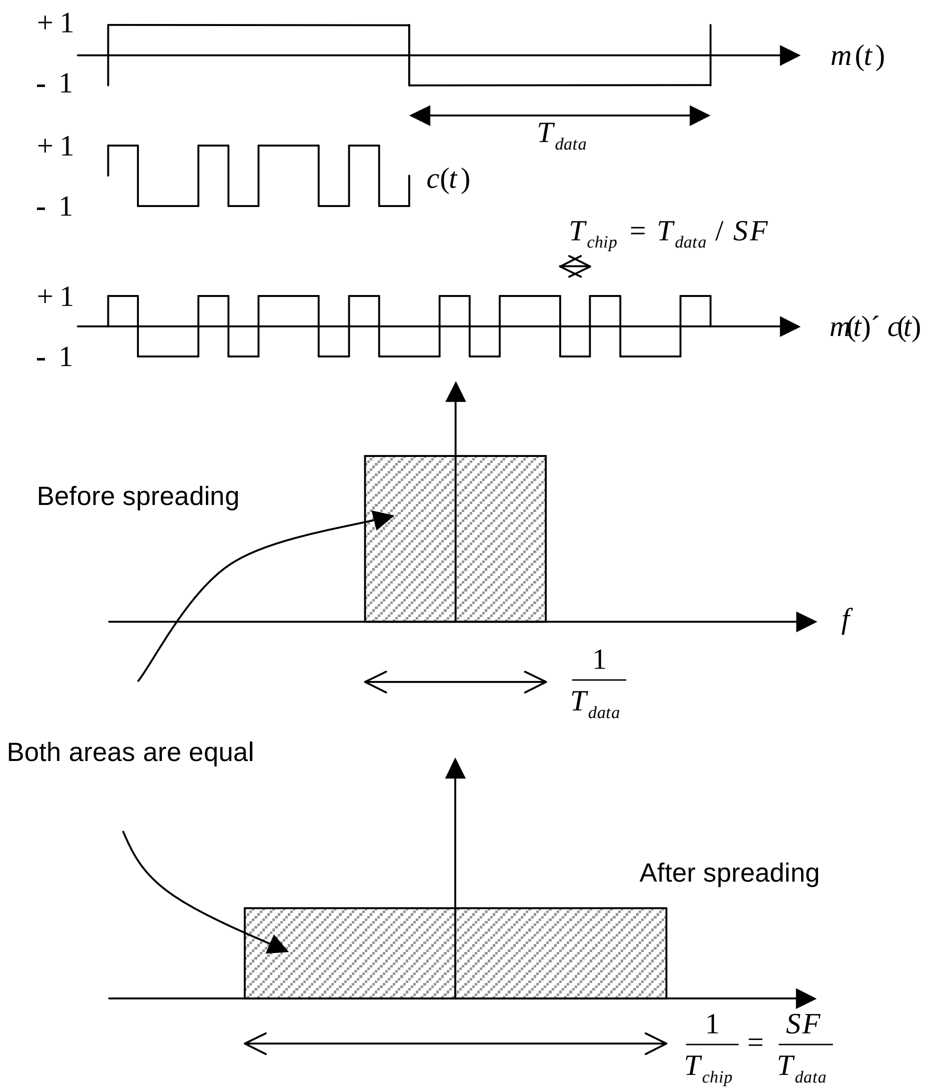
Within a multiplicative factor, the shape of the psd is determined by the square modulus of the transfer function $H(f)$, the Fourier transform of the pulse $h(t)$. The Root Raised Cosine filter is generally used because it allows a transmission of symbols without inter-symbol interference with a spectrum occupancy slightly greater than the minimum value $1/T$ where *T* is the symbol duration. The square modulus of $H(f)$ is depicted by the following picture :



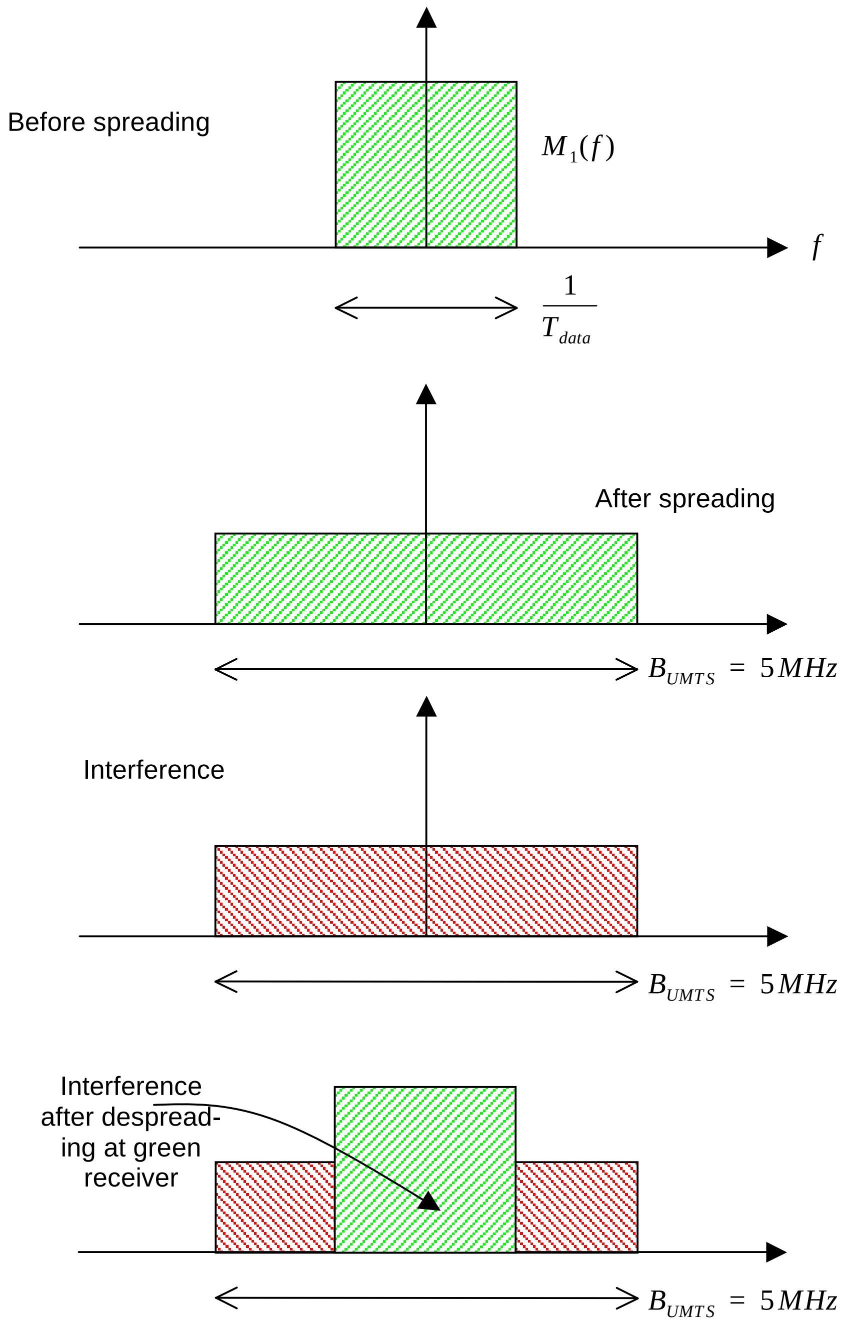
the rolloff coefficient α pertains to the excess bandwith with respect to the minimum required $1/T$. For UMTS the chip rate is $1/T = 3.84MHz$ and $\alpha = 0.22$ so that occupied banwidth is equal to $3.84 \times 1.22 = 4.6848\,\mathrm{MHz}$; the 5 MHz space between two UMTS carriers is slightly greater than this value.

The very basic principle of CDMA is that when a symbole sequence with symbol duration *T* is multiplied with a pseudo-random sequence $c(t)$ with a shorter chip duration, the transmitted sequence looks like a random sequence varying at the same rate than $c(t)$, its bandwidth is then *SF* times larger than that of the original signal (whence the name SF, for Spreading Factor).

[1] Their bandwidth is much smaller than their center frequency. Despite the name « wideband CDMA » UMTS signals fall into this category, for their 5 MHz bandwith is very small compared to a center frequency equal to 2 GHz.

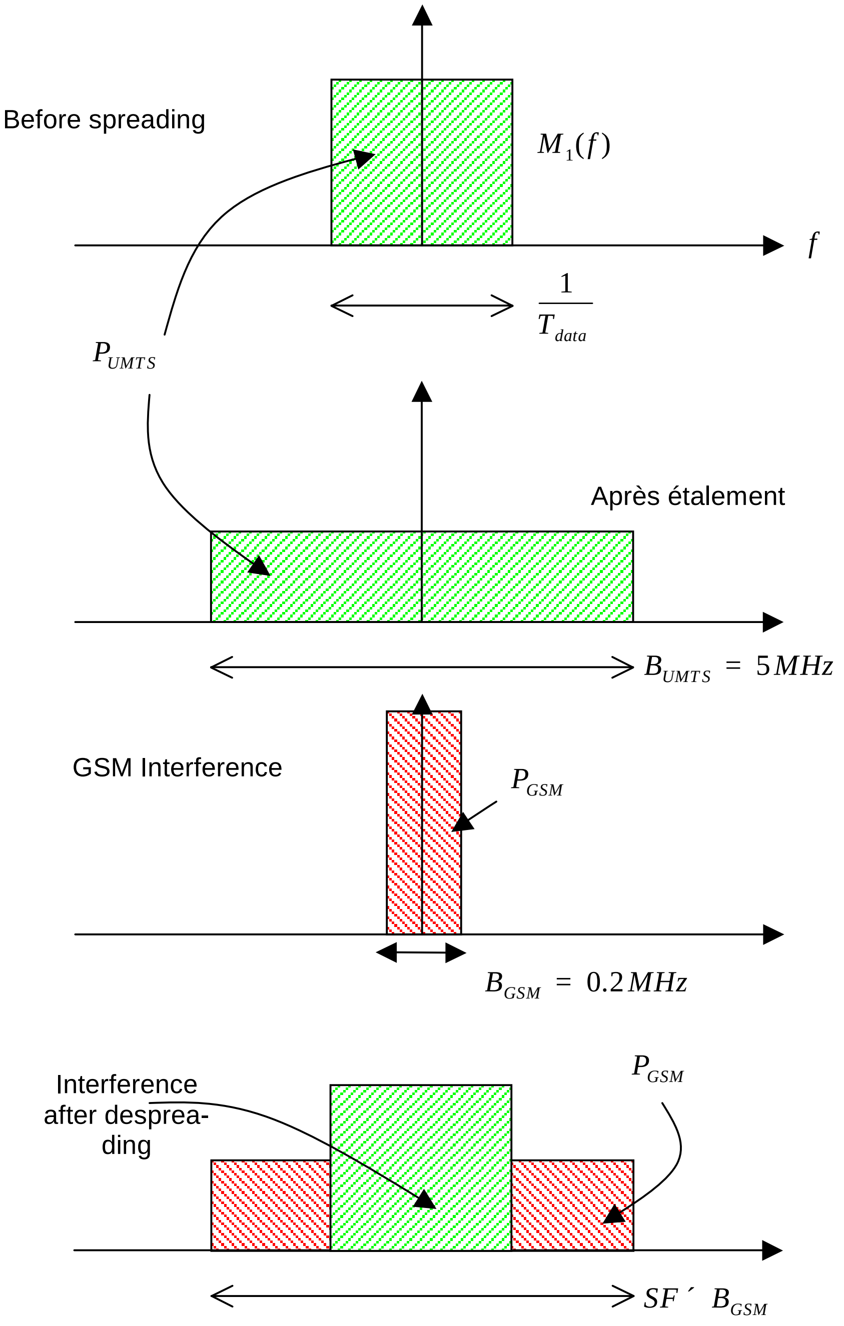


When a receiver performs the multiplication af the received signal with $c(t)$ (it means it knows the sequence and has some means to synchronize it with the received signal) it performs the multiplication $m(t)\times c(t)^2 = m(t)$ because $c(t)$ is in $\{-1,+1\}$ for all t. If using another sequence $c'(t)$ it would obtain $m(t)\times c(t)\times c'(t) \neq m(t)$, indeed $c(t)\times c'(t)$ is another random sequence and $m(t)\times c(t)\times c'(t)$ is also spread in frequency. This is depicted in the following picture where two different users (with respective signals in red and green) share a same frequency band thanks to two different spreading sequences.



After despreading with the good sequence $c(t)$ the receiver has a useful signal with power P_1 (the power of user N°1); the second user's remain spread and only the part which falls into first user's frequency band ,equal to P_2 / SF , will contribute to interference.

2.5.3 Interference from GSM on UMTS



The incoming interference from a GSM transmitter in a UMTS band after despreading is equal to:

$$\frac{P_{GSM}}{SF \times B_{GSM}} \times \frac{B_{UMTS}}{SF}$$

(to be multiplied by the path-loss, in linear scale).

2.6 Tabu search (TS) algorithm

The first step, after the definition of the search space, is to define a neighborhood structure. Indeed, TS is a metaheuristics based on a local search. Given a frequency allocation *S* which is a set of N_{BTS} binary words of *N* bits indicating what channels are in use at a given node (*S* is the current solution) a *neighbor* of *S* is the result of a local *move* from *S*, that is to say, another solution *S'* obtained from *S* by a "small" modification. Of course several definitions of a neighbor are possible (several neighborhood structures) and this is a major difficulty in designing a good TS implementation. Moreover this neighborhood structure should keep some information on the physical underlying problem under consideration.

For our frequency allocation under demand constraint a possible definition of a move could be:

1. choose a node *i* at random
2. choose at random two channels k_1, k_2 such that $f_{i,k1} = 1,\ f_{i,k2} = 0$ and swap them; after the move they become $f'_{i,k1} = 0,\ f'_{i,k2} = 1$.

For instance, we show below a binary vector $\mathbf{f}_i = \left(f_{i,0}, f_{i,1}, \cdots, f_{i,N-1}\right)$ before and after a swap of two frequency uses (swapped frequencies are shown in red):

Before: 10110011101010101

After: 10110001101110101

We shall denote $swap(i; k_1, k_2)$ the swap of frequencies k_1, k_2 at node i.

This choice has the advantage that it keeps constant the number of allocated frequencies after the move. Provided we have an initial allocation satisfying the demands (a feasible solution), all neighbors that will be under consideration at any step of the TS algorithm will fulfill the demands at each node.

The set of all neighbors of a solution *S* will be denoted $N(S)$.

Tabu search can be seen as a modification of the neighborhood $N(S)$ according to past moves: once a swap of frequencies k_1, k_2 at node i is performed it is declared forbidden (or Tabu, whence the name Tabu Search) for a number of steps in order to avoid cycling between a set of moves, and the next iterations will consider only neighbours of current solution *S* which are not Tabu, *i.e.* elements of $N(S) - T$ (difference between two sets).

A first simplified version of TS is thus given by the following pseudo code:

$$
\begin{array}{l}
\text{Initialization:} \\
S \leftarrow S_0 \text{ // initial solution obeying demand constraints} \\
S_* \leftarrow S_0 \text{//current best solution} \\
f_* \leftarrow f(S_0) \text{// current best criterion} \\
T = \varnothing \text{ //initial tabu list is empty} \\
\text{repeat}\{ \\
\quad S \leftarrow \underset{S' \in Neighborhood(S) - T}{\arg\min} f(S') \\
\quad \text{if } f(S) < f_* \text{ then}\{ \\
\qquad S_* \leftarrow S; \\
\qquad f_* \leftarrow f(S); \\
\qquad \text{put move}(S_* \rightarrow S) \text{ in } T \\
\qquad \} \\
\}\text{ until } \underbrace{\text{termination\_criterion}}_{??}
\end{array}
$$

We need a termination criterion; many are possible: a given number of iterations, or a maximum value of the cost, in that case the repeat loop looks like

$$\text{repeat}\,\{\,\cdots\,\}\text{until}\,(\,f_* \le f_{\max}\,)$$

2.7 Management of the Tabu list

At a given node i a swap of two frequencies k_1, k_2 can be decomposed in two steps if we make the distinction between the frequency which is no longer in use (say k_1) and the one which was not in use (say k_2): we say that frequency k_1 **quits** the set of active frequencies at node i and that frequency k_2 **joins** this set. We then introduce two time stamps to indicate that frequency k_1 cannot return in the active frequencies before a given number of iterations and that frequency k_2 cannot quit this active set before a (possibly different) given number of iterations: at each node we have two arrays, no_quit_before[] and no_return_before[] so that, at time *t* and after a swap of frequencies k_1, k_2, no_quit_before[k2] = t+DT and no_return_before[k1] = t+DT, where DT is a delay after which we can perform the steps (return of k_1 in the active set, or k_2 quits the active set).

To know if a swap is not Tabu it is then enough to test if (no_quit_before[k1] < t) && (no_return_before[k2] < t).

As pointed out in [8] this is a simplification aiming at reducing the cost of management of the Tabu list (the table look up to decide if a move either forbidden or not is very easy to implement), but the true move is the conjunction of the two attributes (quit a frequency and join another). In the simplified management above, after a swap of frequencies 5 and 10 at a given node, a further swap fo frequencies 5 and 22 is forbidden for a while, although swap(5,10) is not the equal to swap(5,22).

2.8 Some results

We present here some first results of our implementation of the Min-interference DSA with Tabu-Search. All runs were performed on a standard PC (AMD Athlon Dual Core 2.10 GHz, 1.93 Go RAM).

2.8.1 Choice of a move definition

A first simulation was intended to compare two neighborhood structures: one mentioned above: swap of two frequencies, and a new one named 'insert' and defined on an example:

Before: 10110011101010101

After: 10110011010110101

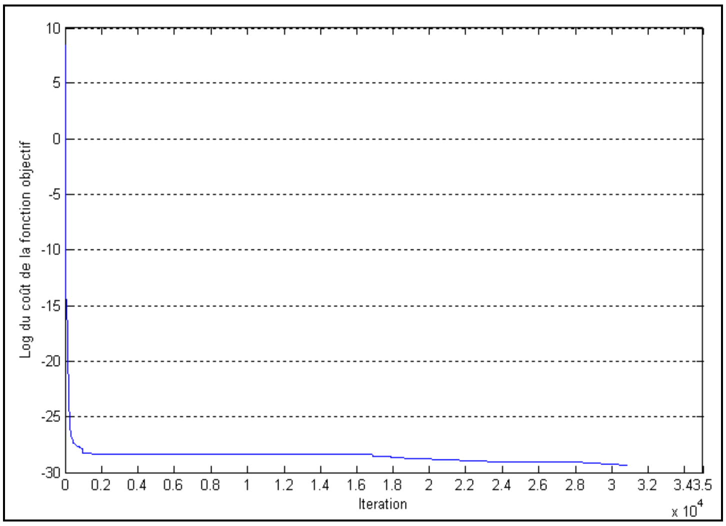
We choose a first bit at a given position (blue underlined), then a number of subsequent bits (red) shifted one position left, the first bit is placed after these shifted bits. The Hamming distance between the two binary words is 4.

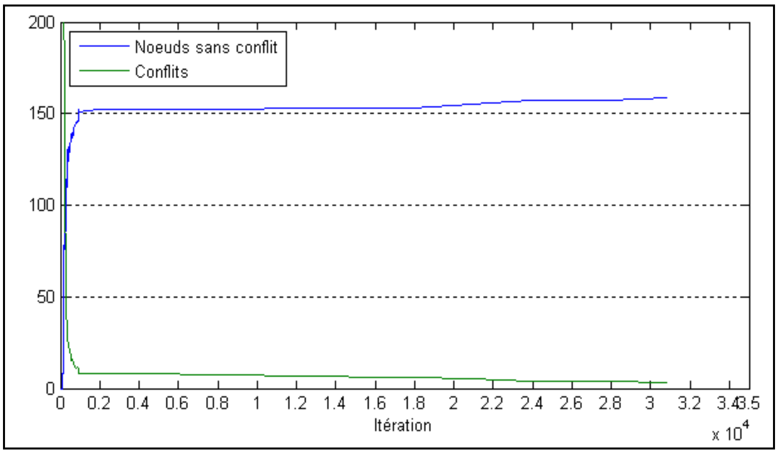
The following table summarizes the comparison between these two definitions of a move, we compare the objective function (cost), the time (or number of iterations), the number of nodes without frequency conflicts with their neighbors (i.e. without interference):

2 RATS n = 165 edges = 8165	Time (sec)	Cost	Conflicts	#nodes w/o conflicts
Initial	0	3.6 e 3	1873	0
Swap	~900	5.4 e -15	3	159
Insert	~1300	1.6 e -12	25	125

As could be expected the 'insert' move has worse performances than the swap, for the Hamming distance of the frequencies active sets (before and after the move) is generally greater for the 'insert' move than the swap move, as seen on the example above.

The two figures below depict the evolution of the objective function and the number of conflicts with respect to the iteration number.





As a result of this comparison all subsequent simulation were based on a swap of two frequencies as a move towards a neighbor. A logarithmic scale was also chosen for the horizontal axis.

The following examples depict the evolution of the objective function (cost function) with respect to the numer of iterations (left curve) and the number of conflicts during the algorithm (right), for several scenarios.

2.8.2 Tests with two RATs (UMTS, GSM) and 165 BTS

We have used the data from URC simulator, including the geographic positions of the BTS

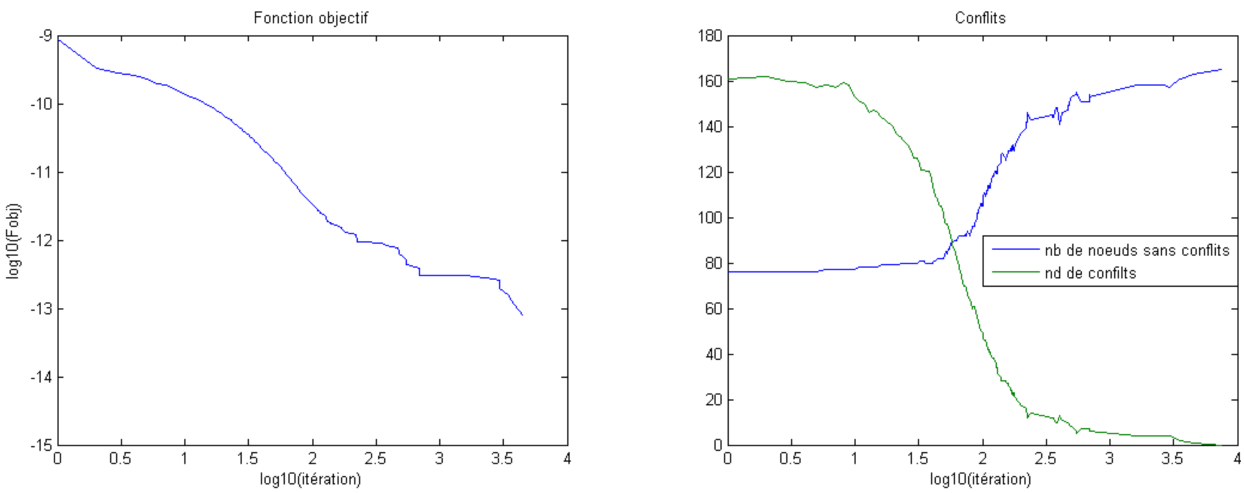
- UMTS : 72 BTS in frequency band 800 – 825 MHz
- GSM : 93 BTS in frequency band 800 – 826 MHz

The Min interference problem we have described (from [5]) and implemented use a random graph to represent the possible interferences, with a threshold to decide if two nodes can interfere or not, as described in section 2.2.

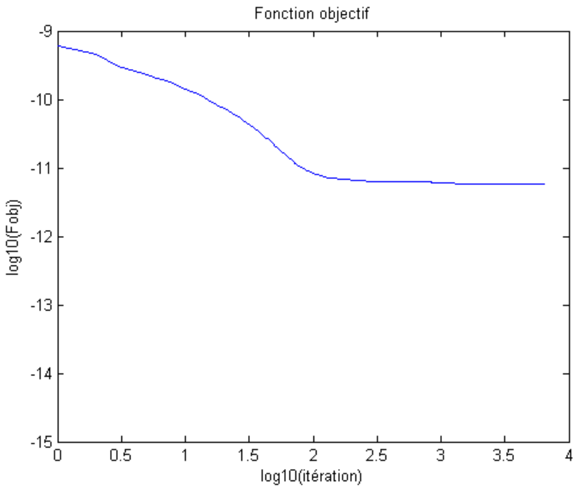
We run the TS for two values of this threshold, 130 dB and 150 dB.

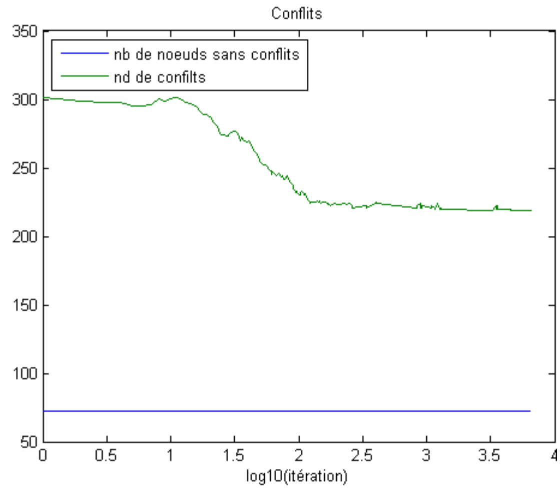
A) Threshold = 130 dB

(time : ~ 130 sec) : the optimum (= 0) is reached in less than 10000 iterations; there is no conflicts at the end of the optimization (the minimum is attained).



B) Threshold = 150 dB
The interference graph as a greater number of edges (see Figure 1 and Figure 2). The Tabu search does not find a solution without interference: the final value of the objective function is less than 1.0e-11, and the final number of conflicts is around 230. (temps : ~ 42 sec) :



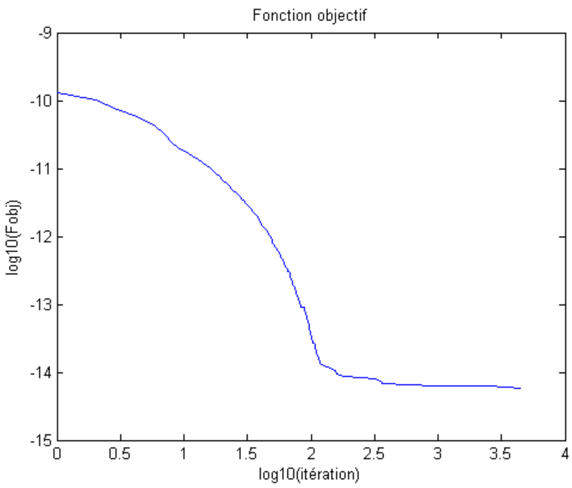


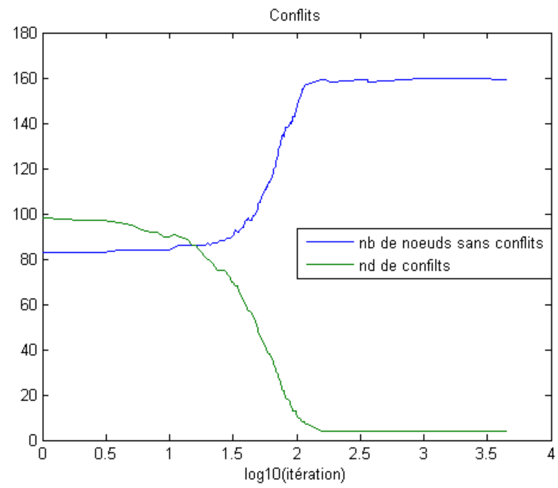
This test shows that the choice of the threshold value is very sensitive; a low threshold is easier to manage by the algorithm but it does not correspond to a real solution since it does not account for all existing interferences in the network. A greater value is more time consuming (more interferences terms are computed, for this example the simulation time is ~ 416 sec), but is more realistic.

2.8.3 Two RATs (UMTS, GSM) , 165 BTS, threshold = 150 dB

We allocate a greater available bandwith, 55 MHz, with the same threshold.

- UMTS : 72 BTS in frequency band 800 – 855 MHz
- GSM : 93 BTS in frequency band 800 – 855 MHz





A much better solution is found by the algorithm (1.0e-14), and the number of conflicts is very low.

2.9 Conclusion

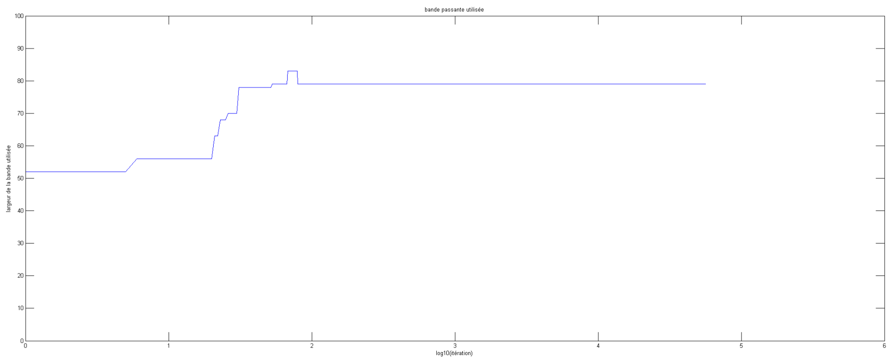
The first results given above show that Tabu Search is a promising solution to the Min-Interference optimization problem model of DSA. They indicate that the graph used to account possible interferences in the network has a great importance in the result obtained; a dense graph (which takes into account all possible interferences) lead to more computation time with a more realistic solution than a graph taking into account only nearest neighbors (those who will contribute more heavily to the total interference).

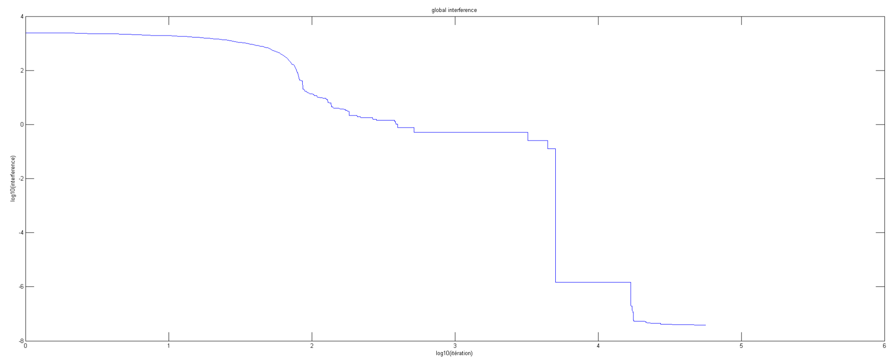
This remark could lead us to a modified version of Tabu Search based in successive problems based on embedded graphs: we could start with a sparse graph corresponding to the edges with the lowest path-loss, easily processed by the TS, leading to a first solution Sol(1); then we add edges with greater attenuations and run the TS from the previous solution Sol (1) as starting point and Sol(2) as final best solution; at iteration *k+1* we add some edges to the graph of iteration *k* and use Sol(k) as initial value of the solution, the TS leading to a new solution Sol(k+1). This scheme deserves a closer look.

$$objective = \min_{\{f_{i,k}\}} \sum_i I_i + \lambda \left(\max_{i,k}\{f_{i,k} \neq 0\} - \min_{i,k}\{f_{i,k} \neq 0\} \right)$$

The "true" quality of the obtained solutions will be assessed in the system performance evaluations (with system indicators, as blocking probability), and will be described in the D3.3.2 deliverable.

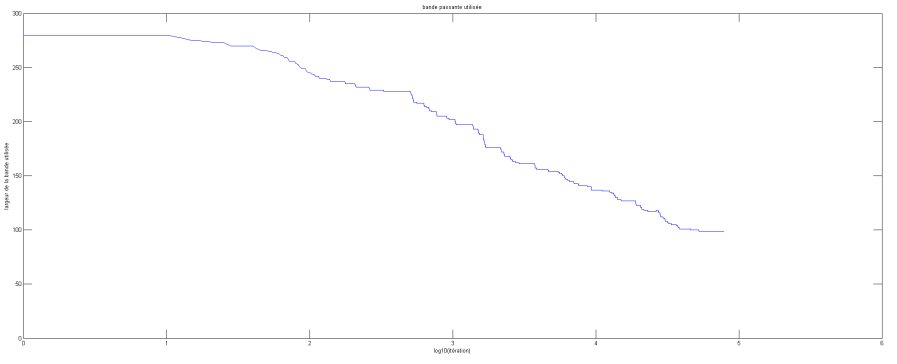
A very preliminary test shows that this is a potentially interesting solution; the two pictures below depict the evolution of the two parts of the compound criterion (occupied bandwith in number of 0.2 MHz bins, sum of interferences) versus iteration when the initial allocation was not enough (here: 10 MHz). The algorithm increases the allocated bandwith and simultaneously minimizes the interference.

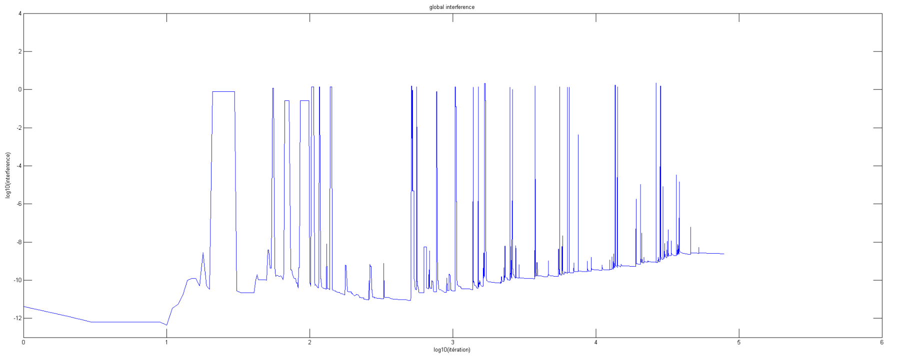




The second test allocate an overestimated bandwidth (50 MHz); we wanted to check if the algorithm would try to minimize total bandwidth; we see that the algorithm does decrease the allocated bandwidth while keeping the total interference under 1.0e-9.

The idea sounds valid, a better tuning of the weighting of the two parts of the global criterion should be addressed.


bande passante utilisée
largeur de la bande utilisée
log10(itération)
0
50
100
150
200
250
300
0
1
2
3
4
5
6


global interference
log10(interference)
log10(itération)
4
2
0
-2
-4
-6
-8
-10
-12
0
1
2
3
4
5
6

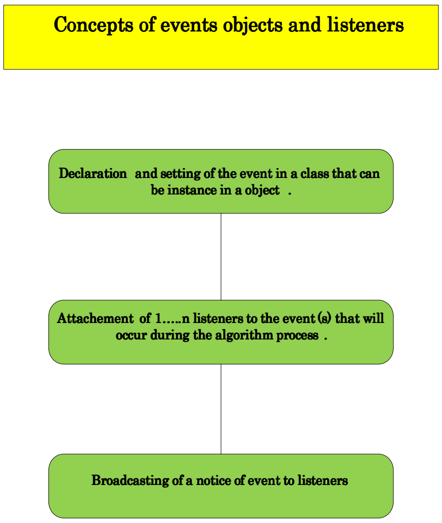
3 DSA algorithm 3: DSA PROCESSING BY BTS COMMUNICATION

3.1 Original approach of DSA

The conception of DSA algorithm was mainly based on an object oriented programming concept which is the management of events between instanced objects. Indeed, events represent a way to send and respond to messages.

3.1.1 Concepts of events objects and listeners

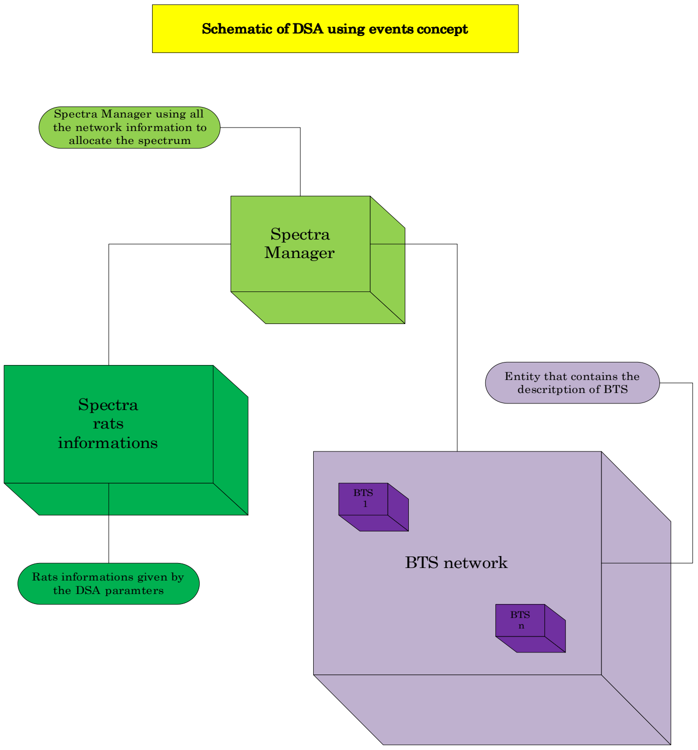
Events represent changes or actions that appear in objects. As a matter of fact, any activity that can be detected programmatically can generate an event and communicate information to other objects. A simple way to define the process that communicates the occurrence of events to others objects that need to respond to the events is:



It is pretty much like the M.V.C design pattern. Well known for Web application design.

3.1.2 Instancing of DSA using the concept of events objects

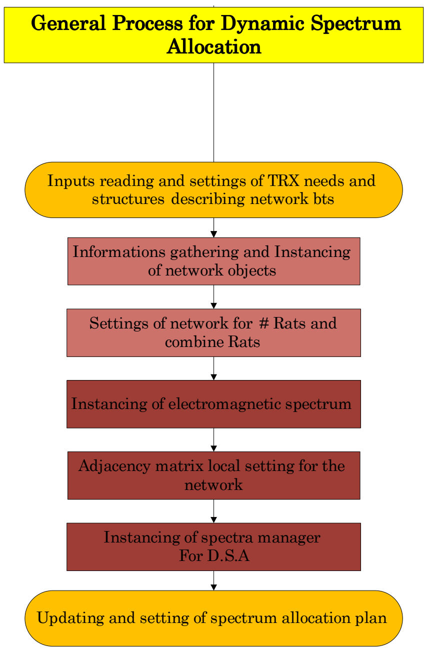
The main idea was to instance a BTS network that will allow the BTS to communicate with one another. Using concept of events and listeners [13]. The BTS network will be handled by a Manager that will guide the DSA procedure. In fact, the spectra allocation will be leaded by a spectra manager that will have a full knowledge of the electromagnetic spectrum and network configuration. The events will be launched by the BTS, the listener and coordinator who will be the spectra manager updating the spectrum.



3.2 General Process for Dynamic Spectrum Allocation

3.2.1 Overview

In the overall, the algorithm is linear, base on the assumption that each part of it could be coded apart. It should be written directly in the main function or in the main method of a static class of the source code. Here is a simple schematic of the process use for DSA. All different parts will be described further on [1].



The General process for DSA is mainly oriented around a goal which is managing spectrum considering constrains due to different technologies (RATS) taking in account the spectrum physical shape like UMTS, GSM [14] and even more like winmax or brand new RATS.

One of the first issues is to convert the structured information in usable and practical data. First of all, the whole different striking parameters of BTS have to be considered in order to materialize BTS network. Second of all, spectrum parameters must be taking in account in order to set the electromagnetic spectrum. Once data have been restructured, it is mandatory to gather all the information into a fast settable object.

The second step should focus on computation of the distance between BTS of same RATS and different RATS. The appealing criterion is the distance between entities because the space shape is the primary argument for wave motion. On one hand, interference between BTS depends on distance that electromagnetic waves need to cover. On the other hand, the interference is also measured by wave strength propagation. Nevertheless it should be taken as a secondary argument in the algorithm.

The fourth challenge has to be the electromagnetic spectrum instancing. After the whole constrains that have been gathered in the data object have been considered.

Towards DSA, the algorithm accuracy has to be based on the graph theory. The space setting of BTS network will lead directly to the creation of adjacency matrix (A). In order to raise accuracy of spectrum allocation the squared adjacency matrix should be compute. Indeed, $(A)^2$ represents the shortest path for the different entities considered in the graph.

The most challenging part should be the instancing of the Spectra Manager that process the dynamic allocation of the electromagnetic spectrum. It uses the implementation of the event concept in order to generate the communication and setting up of the spectrum. It is compelled to take in account the processing of the different path that are computed by the research of allocation between BTS of same and different RATS.

Finally, in order to stay on the same wave length that current software, it is mandatory to restructure the DSA plan at initial state.

3.2.2 Non exhaustive description of algorithm

3.2.2.1 Information gathering and instancing of network objects

The only work that has to be done is to clone the BTS structure fields and spectrum need. And then mute them in properties object in order to carry out a faster and well organize treatment on the data that describes the network in a 2D space.

3.2.2.2 Setting of network

The network has to be set in connection with the two different RATS that are considered in the study. At first, like it has been describe in the overview, network has to be set using the primary argument which is the distance between BTS that is directly connected to the phenomenon of wave propagation and interference.

We drastically implement the distance between to 2 BTS that belong to the network:

$$\overrightarrow{BTS_j} = \begin{pmatrix} X_j \\ Y_j \end{pmatrix} \qquad \overrightarrow{BTS_i} = \begin{pmatrix} X_i \\ Y_i \end{pmatrix}$$

Where we assume that,

$$\vec{r} = \overrightarrow{BTS_i BTS_j}$$

Thus the distance computation will be,

$$\|\vec{r}\| = \left(\left(X_j - X_i\right)^2 + \left(Y_j - Y_i\right)^2\right)^{1/2}$$

This is the main criterion used to set local networks in the main network:

The ensemble N (complete network):

- All RATS should be considered:

$$card\,(N) = card\left(\sum_{i=1}^{n=165} BTS_i\right)$$

The ensemble n (local network):

- For each RATS and combine RATS

$$card\,(l) = card\left(\sum_{i=1}^{n\prime} BTS_i\right)$$

Where,

$$n' \ll n$$

The Elements should be gathered in matrices using the distance criterion.

3.2.2.3 Instancing of electromagnetic spectrum

The electromagnetic spectrum should be instanced according to the DSA parameters. On one side, a spectrum class should be implemented for each RATS. On the other side, each RATS spectrum should be instance and create the cluster spectrum that contains spectra constrains for each RATS.

3.2.2.4 Adjacency matrix local setting for the network

The accuracy of the local treatment should be done using the graph theory. In order to determine the shortest path that exists between BTS of the area.

We consider that BTS are connected if the distance between BTS is lower than the interference criterion given by the DSA parameters.

To create the adjacency matrix one should consider the results that have been computed at 3.2.2.2 and gathered in matrices. And then combine it with graph theory.

Using the following algorithm:

```
if i == j
     obj.Matrix(i,j)=0;
elseif i~=j
     obj.Matrix(i,j)=1;
     obj.Matrix(j,i)=1;
end
```

And obtain the final result:

The Adjacency matrix [16] is written as below:

$$[A] = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

a_{11} Represents a path where the arrival and origin are similar, where

$a_{11} = 0$

a_{1n} Represents a path between the entity 1 and entity n, where

$a_{1n} = a_{n1} = 1$

Once the adjacency matrix has been created the shortest path could compute thanks to the squared adjacency matrix.

$$[A]^2 = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

The shortest path will be the result of matrix terms 1….then 2 and so on representing entities considered in the graph.

3.2.2.5 Instancing of spectra manager

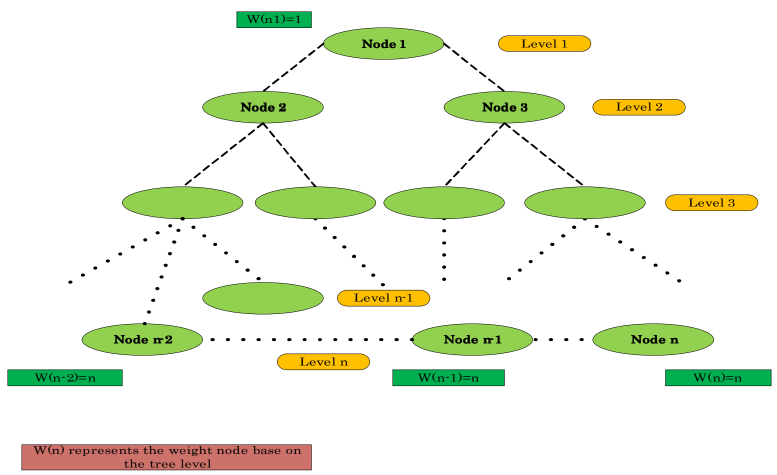
This particular manager is the observer that holds all the methods to handle and manipulate spectrum. Indeed, a smooth way to control the allocation of spectrum is to use a recursive method to create a tree [17] that contains a local BTS list needed for DSA. While the main procedure is executed, spectrum allocation could be optimized using the shortest path given by the adjacency matrix.

It could be implemented [13] as below:

```
classdef SpectraManager < handle
    properties
    NetWorkMatrix
    end

    methods
        %constructor
        function obj=SpectraManager(salloc,…)
            obj.NetWorkMatrix=…;
            ……………………….
            %calls NetWorkManaging on spectra manager
            addlistner(salloc,'Freebandspectra',@obj.NetWorkManaging)

        end
        %methods
        function NetWorkManaging(obj)
            ……………………….
            obj.TreeProcess %calls tree process on spectra manager
        end

    end
```

An interesting way to implement a tree would be to instance an object that contains node with node weight. The weight of nodes will be based on the tree level.

Tree schematic



This could represent a local network of BTS where each node is a BTS define by is Id, RAT, with coordinates (X, Y). The weight has not a physical reality; it is simply a trick to visit each node tree.

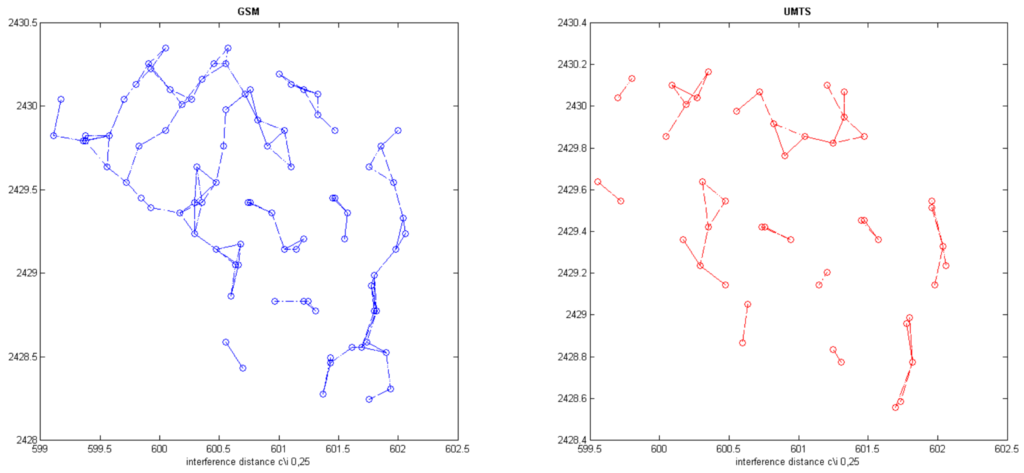
Once path allocation as been set, next step is to allocate spectrum from bottom to the top considering constrains due to each RAT while they are deployed in time and space.

3.3 Results given by the algorithm

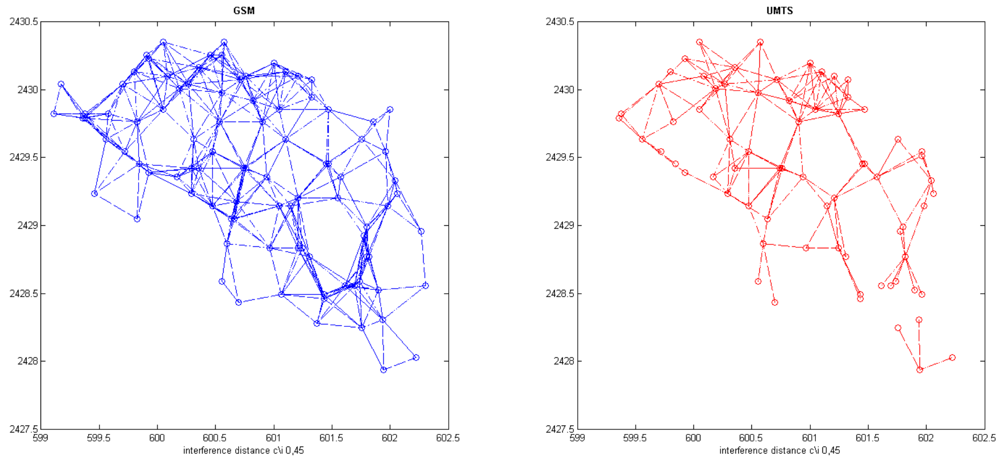
After the network is set, we can easily obtain usable results base on network spatial setting.

3.3.1 Results for same RATS

If we consider connection between BTS of each RAT, we note that allocation of spectrum should be a lot more accurate because local networks are distinguishable and independent from one another. In this particular case, the interference criterion could set the spectrum allocation, since UMTS BTS are less interfering with one another; allocation should start with this particular RAT, whereas GSM network should be set in the free spectrum band. The use case is minimum interference between network BTS. Nevertheless, spectrum sharing wouldn't be made on spectrum band sharing.



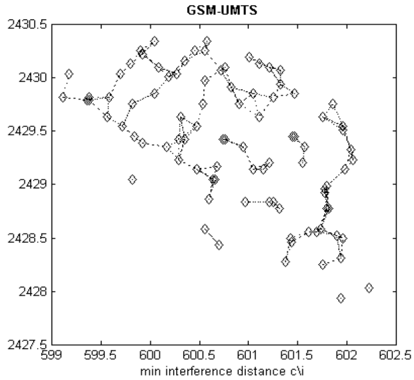
In the next case, we observe that most of BTS are connected but more in a local mean. The connection density implies a greater interference between BTS of network. This could end up in a greater sharing of the electromagnetic spectrum but a slower spectrum allocation. The free spectrum bands will be spread at greater frequencies because more interference implies a greater traffic jam of the carrier waves.



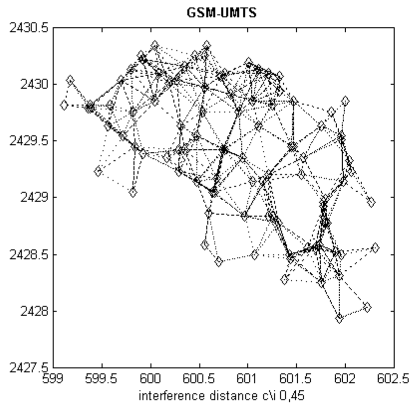
This type of setting up could be used on the different use cases. A general conditional procedure could be implemented in connection with the traffic jam.

3.3.2 Results for combine RATS (GSM-UMTS)

The combining of the different RATS leads to a result which quite similar to the last we have obtained splitting RATS.



In the following case, we observe that density has raise among close connected BTS which will lead to a much slower spectrum allocation due to the tree processing (recursive method) cost in CPU mention in the algorithm. It could also be useful to use it form a blend of FSA and DSA. Because local areas with high density would inherit of fix spectrum allocation while the other areas could be useful for quick allocation in case awkward situations or events.



3.4 Conclusion

This study of Dynamic spectrum allocation could be one of the stepping stone for the next generation software managing cellular networks. If this algorithm is rightly improved, especially on the aspect of local allocation, for instance in case of highly traffic we could quickly hook on another area (different local network) to share the lack of spectrum usage.

Indeed, since local networks seems to naturally appear when interference is low. We could imagine having a manager for each local network that regulates emitting power for BTS depending on local traffic in order to raise spectrum offers.

We should also notice that microscopic effects [15] were not taken in account like boundary effects between spectrum bands. Microscopic effects could also be a good study path to improve spectrum usage in case of dynamic allocation.

Finally, it could be interesting that each local manager could be a sort of Artificial Intelligence well designed and implemented in cellular network.

4 Bibliography

[1] UCR deliverable D2.1.3, Dynamic Spectrum Allocation algorithms,Ed.19, 23/04/2008.

[2] M. Buddhikot: Understanding Dynamic Spectrum Access: Models, Taxonomy and Challenges**,** *Proceedings of IEEE DySPAN 2007, Dublin, Ireland, April 17-21, 2007.*

[3] A.S. Prabhu, H. Gupta, M.M. Buddhikot, Fast spectrum allocation in coordinated dynamic spectrum access based cellular networks, *Proceedings of IEEE DySPAN 2007* .

[4] A.S. Prabhu, M. al Ayyoub, H. Gupta, S. Das, M.M. Buddhikot: Near-optimal dynamic spectrum allocation in cellular networks, *Proc. IEEE Dynamic Spectrum Access Networks (DySPAN), 2008.*

[5] A. Prabhu, H. Gupta, S. Das: Minimum interference channel assignment in multi-radio wireless mesh networks, *IEEE SECON, 2007.*

[6] F. Glover: Tabu Search - Part I, *ORSA Journal on Computing*, Vol. 1, No. 3, 190-206, 1989.

[7] F. Glover: Tabu Search - Part II, *ORSA Journal on Computing*, Vol 2, No. 1, 4-32, 1990.

[8] F. Glover: Tabu Search Fundamentals and Uses, Technical report, University of Colorado, Boulder, June 1995.

[9] M. Gendreau: an introduction to Tabu search, july 2002, available online at: http://www.ifi.uio.no/infheur/Bakgrunn/Intro_to_TS_Gendreau.htm

[10] A. Koster: Frequency Assignment – Models and algorithms, PhD, Ph.D. thesis University of Maastricht, Maastricht, The Netherlands, 1999, http://www.zib.de/koster/thesis.html.

[11] K.I. Ardal, S.P van Hoesel, A. Koster, C. Mannino, A. Sassano: Models and Solution Techniques for Frequency Assignment Problems, Konrad-Zuse-Zentrum für Informationstechnik Berlin, technical report ZIB-Report 01–40 (December 2001)

[12] R. Montemanni, J. Moon, D.H. Smith: An improved Tabu-search algorithm for the fixed-spectrum frequency-assignment problem, IEEE Trans. Vehicular technology, vol. 52, N° 3, may 2003, pp. 891-901.

[13] MATLAB 7-Oriented Object Programming- Events-Sending and Responding to Messages

[14] Réseaux cellulaires par Jean CELLMER- ingénieur des télécommunications (internet)

[15] Les Antennes – Tableau des distances de transmission (internet)

[16] Eléments de Théorie des Graphes - INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE- Didier Maquin

[17] Théorie des graphes - Université Bordeaux 1- Eric sopena

APPENDIX

This appendix describes the Hata model of path loss which we used to label the edges of the interference graph.

```
We used the following parameters : HT = 20m and HR = 3m
/*
 * modèle de path-loss de Hata, zone urbaine
 * freq = frequence en MHz,150 <= freq <= 2000
 * HT = hauteur de l'emetteur (en m), 30m <= HT <= 200m
 * HR = hauteur du récepteur (en m) ,  1m <= HR <=  10m
 *  d = distance entre emetteur et recepteur (en km),  1km <= d <= 20km
 * reference: COST action 231 :
 * EVOLUTION OF LAND MOBILE RADIO (INCLUDING PERSONAL) COMMUNICATIONS,
 * final report (chapitre 4), http://www.lx.it.pt/cost231/  *****/

double PL_COST231_Hata_dB(double freq, double d, double HT, double HR){
      double K0, K1, K2, A, logf, logh, res;
      // si dense urban:
      // K0 = 3.0
      // A  = 3.2 * (log10(11.75*HR))^2 - 4.97
      //
      K0 = 0;
      ASSERT((freq >= 150.0) && (freq <= 2000.0));
   if (freq <= 1500.0) {
      K1 = 69.55;
            K2 = 26.16;
            }
      else {
         K1 = 46.30;
            K2 = 33.9;
            }
      logf = log10(freq);
      logh = log10(HT);
      A  = (1.1*logf-0.7)*HR - (1.56*logf-0.8);

      res = K1 + K2 * logf - 13.82*logh - A + (44.9-6.55 *logh)*log10(d) + K0;
      return(res);
      }
```