A Tabu Search DSA Algorithm for Reward Maximization in Cellular Networks

Hany Kamal, Marceau Coupechoux, Philippe Godlewski
TELECOM ParisTech & CNRS LTCI
46, rue Barrault, Paris, France
{firstname.lastname}@Telecom-ParisTech.fr

***Abstract*—In this paper, we present and analyze a Tabu Search (TS) algorithm for DSA (Dynamic Spectrum Access) in cellular networks. We study a mono-operator case where the operator is providing packet services to the end-users. The objective of the cellular operator is to maximize its reward while taking into account the trade-off between the spectrum cost and the revenues obtained from end-users. These revenue are modeled here as an increasing function of the achieved throughput. Results show that the algorithm allows the operator to increase its reward by taking advantage of the spatial heterogeneity of the traffic in the network, rather than assuming homogeneous traffic for radio resource allocation. Our TS-based DSA algorithm is efficient in terms of the required memory space and convergence speed. Results show that the algorithm is fast enough to suit a dynamic context.**

I. Introduction

Due to the spectrum crowd situation and the high demands on spectral resources, spectrum sharing and DSA techniques have been active research topics. The existing spectrum allocation process, denoted as FSA (Fixed Spectrum Access), headed for static long term exclusive rights of spectrum usage [1] and shown to be inflexible [2].

Spectrum sharing has been proposed as a promising method for better usage of spectrum. Researchers have worked on spectrum sharing algorithms motivated by the incentives taken by FCC to promote a better usage of spectrum [3] [4]. For example, the authors of [5] propose a coordinated DSA system where a common pool of resources (CAB or Coordinated Access Band) is shared and controlled by a regional spectrum broker.

In this paper, we consider a framework of several operators sharing a common pool of resources (or a CAB) inspired by [5], and we focus on the strategy of one operator leasing spectrum from the broker. The operator does not own the spectrum, but rather has to lease it according to the demands. The operator is providing packet services for the end-users. We are interested in developing a DSA algorithm based on Tabu Search (TS) that provides the number of spectrum blocks to be acquired from the broker, as well as the frequency assignment corresponding to the maximum reward.

Several algorithms have been proposed to solve the CAP (Channel Assignment Problem) in cellular mobile networks. The classical CAP consists of assigning the channels to the cells within the mobile network while satisfying: (1) interference constraints (co-channel, adjacent channel or both together) and (2) the traffic load demands. The proposed algorithms in the literature could be categorized as follows: algorithms based on heuristic methods [6], others based on genetic algorithm [7], on graph coloring method [8], and on neural network methods [9].

Researchers have also studied frequency assignment using TS algorithm. For example, papers [10], [11], [12], and [13] make a partial list of the references proposed TS algorithm to solve the fixed-spectrum CAP.

It is worth mentioning that most of the work done using TS to solve the fixed-spectrum CAP in cellular networks, has focused on circuit switched traffic (i.e. voice traffic) with application to the GSM networks (see [11] for example). Treating voice traffic using TS has always been associated with a *hard interference requirement*: below a certain CIR (Carrier to Interference Ratio) threshold, the service is not accessible, while above this level, there is no significant increase of the service quality.

For this reason, the previous works have focused on the minimization of the interference (as an objective function), while satisfying the traffic demands. Note that in order to be able to perform the spectrum assignment, in fixed-spectrum CAP problems, it is necessary to know the number of channels required by each cell.

With the increasing demand of packet data services along with the development of new standards supporting packet applications, i.e. LTE and WiMAX, it would be interesting for DSA techniques to take into consideration the specificities of packet traffic. This is the challenge we are tackling in this paper. In contrast with the case of voice traffic, in packet traffic services, we see the interference constraint as a *soft interference requirement*, where interference can be tolerated without a *hard* threshold. A higher level of interference however induces a soft degradation of end-users throughput and consequently affects their satisfaction.

Different from references [8], [10], [11] and [12] that used TS algorithms, we set an objective function of maximizing the operators' reward. The reward is computed here as the sum of revenues obtained from end-users deduced from the spectrum cost. The revenue obtained from a user is in turn an increasing function of its throughput.

In our formulation of the dynamic-CAP for packet service, the operator does not know the number of frequency blocks required by the cells. Assigning only *one*, but poorly interfered, block to a given cell might provide higher throughput to the end-users than assigning *two (or more)* highly interfered blocks. The operator needs to find a certain level of compromise in order to maximize its reward (Section II-B).

In [8], the authors have used TS method to solve the *minimum interference DSA* problem. Our approach differs from [8] mainly due to the consideration of packet context. Consequently our formulation of the objective function, the neighborhood structure, and the tabu list is different and adapted to the packet traffic assumption. Our formulations presented in this paper leads to obtain a simple algorithm that does not require excessive memory space, and can be envisaged for the implementation in a dynamic context. Hereafter we summarize our main contributions: Presenting and analyzing a DSA algorithm based on adapted TS method, where (1) we consider packet traffic services, (2) we address the spectrum pricing issue, and (3) we set an objective function for maximizing the operator's reward.

In this paper we extend our work in [14] and [15] by presenting an algorithm for heterogeneous traffic, where we do not suppose *a priori* that a classical frequency reuse (e.g. reuse 1 or reuse 3) is deployed as it is done in the above references.

The paper is organised as follows: Section II presents the network model in terms of system model, DSA principle, cell capacity calculations, and reward model. In section III, we illustrate the TS framework and we give our algorithm's details. Section IV gives the numerical results. Conclusion is finaly given in section V.

II. Network model

A. System model

We study DSA on the cell level and we focus on a mono-operator case. The operator is supposed to depoly one RAN (Radio Access Network) providing packet services to the end-users. The operator does not own the spectrum but rather has to lease it according to the traffic load. We are considering a hexagonal topology for the RAN, consisting of one central cell and two rings of cells surrounding the central cell. Fig. 1 gives the hexagonal model of our study, where parameter R is the cell radius.

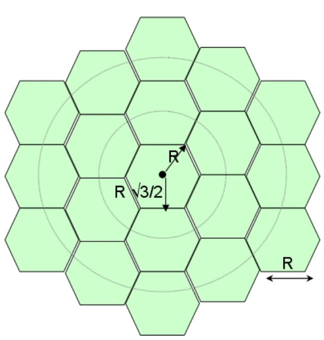

Fig. 1. Hexagonal network of study.

It is worth mentioning that the usage of a hexagonal model is only for the sake of simple simulations. Our algorithm behaves the same way no matter the type of network topology.

We assume a scheduling fair in throughput for the users of a given cell. The average data rate accessible by users in a cell is proportional to the bandwidth allocated to the cell and is equally divided among all users of the cell.

B. Dynamic spectrum access

In the considered system model, the core issue for the operator lies in the trade-off to be found between spectrum cost and revenues obtained from users: more spectrum means a higher cost for the operator but also higher throughputs for users that are encouraged to pay more for the service [15].

We suppose a DSA decision is taken by the operator at each event, i.e. arrival of a new user, or a user departure in any cell. A DSA decision assigns spectrum blocks to each cell in the RAN. We assume that at least one spectrum block is always available to each cell, so that starvation is not possible.

C. CIR and cell capacity

Clearly, the bit-rate obtained by the end-users depends on the perceived CINR (Carrier to Interference plus Noise Ratio) level. The CINR level depends on the frequency assignment.

The exact CINR distribution in a cell is hard to be determined in practice. For the sake of simplicity, we rely on an approximate calculation for the CINR by focusing on the cell edge, which is a worst case in terms of interference.

We consider an urban environment, and hence we neglect the noise and we focus on the CIR (Carrier to Interference Ratio). The users are assumed to be located on the cell border and facing the highest level of interference from the interefering cells.

According to the previous assumptions, the CIR perceived by the users in cell c on frequency block f, can be denoted by:

$$CIR_c^f = \frac{R^{-\alpha}}{\sum_{i=1}^{B_f} (d_{c,i} - R)^{-\alpha}},$$

where R is the cell radius, α is the path-loss exponent, $d_{c,i}$ is the distance between the victim cell c and the interfering cell i, and B_f is set of all cells using the frequency block f.

We approximate the cell capacity (in bps) using Shannon's classical formula. The cell capacity is the sum of capacities provided by the frequency blocks used by the cell. Formaly, the cell capacity C_c of cell c is denoted by:

$$C_c = \sum_{f=1}^{F_c} W_f \ log_2(1 + CIR_c^f),$$

where W_f is the block size of frequency f in Hz, CIR_c^f is the CIR perceived by cell c on frequency block f and F_c is the number of frequency blocks used by cell c.

As we consider fair throughput scheduling between users of a given cell. The data-rate D_c obtained by each of the users in cell c is given by: $D_c = C_c/N_c$, where N_c is the number of users in cell c.

D. Reward model

The challenging issue in DSA techniques for the operator lies in the trade-off between the cost paid for the spectrum and the revenues obtained from the satisfied users: more spectrum per cell means a higher cost for the operator but also means higher throughputs for the end-users. Based on this principle we define a reward model that takes into account both the user date-rate as well as the spectrum price.

The reward function depends on the revenue expected by the operators. The higher the satisfaction of users, the higher the operator revenue. The revenue obtained from a given customer in cell c increases with its satisfaction:

$$\phi_c(D_c) = K_u(1 - \exp(-D_c/D_{com})),$$

where K_u is a constant in euros per unit of satisfaction, D_{com} is a constant called comfort data-rate, and the satisfaction is an increasing function of the user data rate (without unit) [16].

We consider the spectrum price to be fixed by MHz. The cost paid by the operator for the spectrum can be given as:

$$K_B \ W_f \ F,$$

where F is the number of frequency blocks used by the RAN, W_f is the block size in Hz, and K_B is a constant in euros per block. Note that considering a different spectrum price function (for example a function that depends on the demands in the market as considered in [14] and [15]) will not affect the behavior nor the performance of our algorithm.

The reward obtained by the operator can thus be written:

$$g = \sum_{c=1}^{B} N_c \phi_c(D_c) - K_B \ W_f \ F,$$

where B is the total number of cells in the cluster area where DSA is performed.

III. TABU SEARCH

A. Principle

Tabu search is a *metaheuristic* that guides a local heuristic search procedure to explore the solution space beyond local optimality (by allowing a degenerated solution) [12]. TS was originally presented by Glover in [17] and [18].

The basic idea is to forbid a move that would return to recently visited solutions by classifying them as tabu. Hereafter we give the fundamentals of TS. Let S be the set containing the possible solutions to a problem. For each solution $s \in S$ there exists a subset of S called *neighborhood* of s. The neighborhood contains feasible solutions, each of them is obtained by making a simple *move* from the solution s. The algorithm uses a memory structure called *TL (Tabu List)* to avoid cycles. The algorithm forbids the selection of a solution among the neighborhood, if this solution have been visited in a previous iteration. At each iteration the TS updates the TL by adding attributes of the selected solution. Note that such attributes do not contain the complete solution otherwise handling the TL will become costly (in terms of required memory) when the number of iterations increases [12]. Note that the TL has a limited size (called *TT* for Tabu Tenure) and the choice of the TT has an impact on the obtained result. The smaller the TT, the higher the chance to have cycles (visiting previously visited solutions) and hence TS cannot go beyond the local optimal solution. However if the TT is very large, very few options will be left for the neighborhood formation.

The initial point of the TS algorithm has its importance in determining the time (i.e. number of iterations) required to reach the optimal solution. Starting from a solution very far away from the zone where the optimal solution exists, will require more iterations to explore the different zones. An *initialization* process aims at facilitating the search procedure for the algorithm, through the reduction of the time required to reach the optimal solution. Usually an *initialization* process is based on some heuristic method.

B. Definitions

Before illustrating our implementation of the TS algorithm, we give the following key definitions.

- A solution s is defined as a Boolean matrix of size $F_{max} \times B$, where F_{max} is the CAB size (or the maximum number of blocks the operator can lease) and B is the number of cells in the RAN. An element s_{fc} of the matrix is defined as:

  $$s_{fc} = \begin{cases} 1, \text{ if frequency } f \text{ is assigned to cell } c, \\ 0, \text{ otherwise.} \end{cases}$$

  Taking an example of $F_{max} = 3$ blocks, and $B = 5$ cells, then a "possible" solution s can be given as:

  $$s = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

  In this simple example, only 2 blocks are used by the RAN ($F = 2$) and the operator pays for the corresponding spectrum size.
- According to our model, a "possible" solution means there is at least one block assigned to any cell. Practically, this assumption helps reducing the search space for the algorithm, and hence increasing the chance of reaching a better solution in less number of iterations. The assumption is also realistic that avoids a starvation situation.
- For each solution $s \in S$, we define the set of moves $M(s)$ which can be applied to s in order to obtain a new solution s'.
- A *neighbor* s' of the solution s is created by applying one move m, where $m \in M(s)$.
- The move m is a Boolean matrix of the same size as s, all its elements equal to *zero* except one, or two elements equal to *one*.
- The reward $g(s)$ achieved using a solution s is calculated as illustrated previously in sections II-C and II-D. The maximum reward ever-reached during the search process is denoted g_{max}.

- At each iteration, *attributes* of the selected solutions are added to the TL.

We have chosen to consider the reward corresponding to each selected (among all neighbors at each iteration) solution as its attribute. There is two main advantages behind choosing the reward as the attribute of the visited solution. First, adding the reward $g(s)$ (of the selected solution s) in the TL, will not only forbid the TS from selecting s as a valid solution for the following iterations, but will also forbid visiting all solutions that achieve the same reward as $g(s)$.

The second advantage is related to the required memory space for the TL. Our TL is composed of a single vector of size TT, each of its elements being equal to the reward $g(s)$ corresponding to the selected solution s. Note that, in our case, $g(s)$ holds the complete needed information (from the operator's perspective) of the solution matrix s.

C. Implementation

We present hereafter our TS algorithm that suits the CAP for packet services:

Algorithm 1 TS algorithm for reward maximization in packet services context

---

Initialization: an initial solution s_{init} is found.
$s \leftarrow s_{init}$
$g_{max} \leftarrow g(s_{init})$
while Nb. of iterations $\leq MAXITER$ **do**
    Neighborhood formation: all *possible* neighbors of the initial solution s are created, except those who are listed as tabu.
    Neighbor selection: one solution s' is chosen among the set of neighbors to be considered as the "initial" solution for the next iteration.
    Tabu list update: the reward $g(s')$ corresponding to the selected solution s' is added to the TL.
    Max. reward update: the maximum ever-obtained reward g_{max} is updated:
    if $g(s') > g_{max}$, **then** $g_{max} \leftarrow g(s')$ **end if**
end while

We illustrate now the details of the steps given in Algorithm 1.

Initialization: We have chosen an initialization method based on randomly formed solutions. Note that the operator does not have any requirements on the number of blocks to assign to the cells (unlike [8], [10] and [12]), hence the total number of blocks to be used is unknown to the operator. We divide the search zones according to the total number of blocks the operator can lease $(1, ..., F_{max})$. Note that the search zone with a single block is a trivial one because in this case, there is a single possible solution corresponding to frequency reuse 1. We generate randomly 300 *possible* solutions for each search zone. The TS algorithm starts using the solution corresponding to the maximum obtained reward among all randomly created solutions.

Neighborhood formation: All possible neighbors are created by whether: (1) removing an assigned block from a random cell, (2) adding a non-used block to a random cell, or (3) replacing one of the used blocks in a random cell by a non-used block. Note that adding, removing, or replacing a frequency can be performed by a simple XOR operation. The neighbor $s' = s \oplus m$, where m is a Boolean matrix that contains *zeros* except one element equal to *one* in case of adding or removing a block. In case of replacing a block, two elements of the matrix m equal to *one*.

Neighbor selection: According to our defined objective function, the neighbor that achieves the maximum reward among all neighbors is selected.

IV. Numerical results

A. Simulation scenarios and parameters

We consider a RAN consisting of 19 cells. The RAN's topology is formed of one central cell and two rings of cells surrounding the central cell as shown in Fig. 1.

Hereafter we define the parameters we used for our simulations. The maximum number of blocks F_{max} the operator can lease is assumed to be 6 blocks, with block size of 1 MHz. The comfort bit-rate for the user D_{com} = 500 Kbps, the cell radius $R = 1$ Km, and the path-loss exponent $\alpha = 3$. The pricing constants are fixed as follows: $K_u = 10$ euros and $K_B = 50$ euros.

TS algorithm parameters are set as follows: Tabu Tenure $= 200$ and the maximum number of iterations $MAXITER =$ 800 iterations. According to the defined parameter set as well as to the neighbor definition, the total number of possible neighbors created from a solution s equals to:

$$F_{max}\ B - B_{s0} + \sum_{c}^{B} F_c\ \bar{F}_c,$$

where B_{s0} is the number of cells having one block in s, B is the total number of cells in the RAN, F_c is the number of frequency blocks used by cell c, and $\bar{F}_c$ is the number of blocks not used by cell c. Note that $F_c + \bar{F}_c = F_{max}$. The first part of the equation $(F_{max}\ B - B_{s0})$ represents the total number of possible neighbors created due to adding or removing a block from a cell, knowing that at least one block should be assigned to any cell. The summation part represents the total number of possible neighbors created due to the replacement of a block.

It is clear that, for our case study, generating all possible neighbors at each iteration is very feasible.

B. Homogeneous versus heterogenous traffic

Now we are going to use TS to compare the obtained reward through serving a specific amount of traffic load (determined by the total number of active users) in two different cases: (1) the case of an operator using FSA, who assumed a homogeneous traffic over the RAN to perform its channel assignment, and (2) the case of an operator using DSA, who considers the exact (heterogenous) distribution of the traffic to dynamically assign spectrum blocks.

We have considered the parameter set given in section IV-A, with a total number of users equal to 57 users. We suppose the distribution of the users is following a decreasing function of the distance from the central cell. There is a high concentration of users in the central cell, and this concentration decreases with the distance from the center of the cluster. Tab. I gives all studied users' distributions following this criterion with a total number of users equals to 57 users. It gives the number of users/cell for the central cell, the middle-circle cells, and the outer-circle cells as well as the distributions' standard-deviation σ. The homogeneous traffic scenario is also included with its zero standard deviation (all cells have the same number of users).

TABLE I
STUDIED USERS' DISTRIBUTIONS AND CORRESPONDING STANDARD DEVIATIONS σ

central cell	middle-circle cells	outer-circle cells	σ
33	2	1	7.28
27	3	1	5.88
21	4	1	4.58
15	5	1	3.46
9	6	1	2.76
9	4	2	1.73
3	3	3	0

The operator using FSA has assigned frequency blocks to the cells while assuming a homogeneous traffic (last line of Tab. I). For a fair comparison, the assignment is obtained using the TS algorithm but remains fixed whatever the traffic conditions.

The operator using DSA adapts its frequency assignment according to the dynamic of the traffic and try to maximize its reward whatever its heterogeneity using the proposed TS based DSA algorithm.

Fig. 2 gives the obtained reward versus the standard deviation σ for both operators. Each point corresponds to a line in Tab. I. For $\sigma = 0$, as both operators launch the TS algorithm, obtained reward are equal. As heterogeneity grows up (σ increases), the FSA allocation remains the same for the first operator, while for the second, DSA strategy adapts the assignment to the traffic.

We can notice that the obtained reward decreases as σ increases for both cases, even for the operator who considers the real traffic in the RAN. Fig. 2 shows however that the reward obtained by the DSA operator exceeds the reward obtained by the FSA operator for all values of σ.

We give in Fig 3 the spectrum assignment obtained using TS-based DSA algorithm for $\sigma = 7.28$.

The numbers indicated on the cells represent the block numbers. The TS-based DSA algorithm has assigned one block to all the cells, except the central cell. The algorithm has assigned 3 blocks to the central cell; two of them (block 4 and block 5) are not assigned to any of the other cells, while the third block (block 2) is assigned to some of the cells on the outer-circle. We see this assignment is coherent with the distribution of users in the cells. Note that the central cell has 33 users (see Table I).

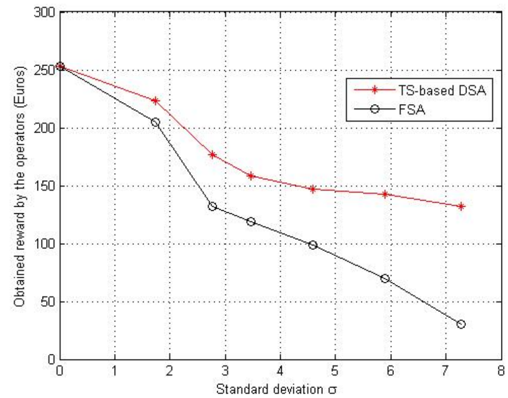

Fig. 2. Obtained reward by the two operators as a function of σ.

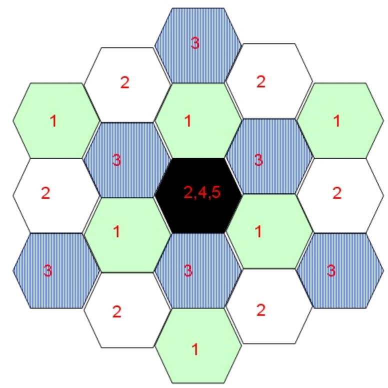

Fig. 3. Obtained spectrum assignment using TS-based DSA for $\sigma = 7.28$.

C. Performance of TS

In this section, we evaluate the performance of our TS algorithm in terms of the minimum number of iterations required to reach an "efficient" solution. This metric is important from the dynamicity point of view of the algorithm. We assume the same parameter set as given in section IV-A. The RAN has a total number of 57 users. We compare different cases of users' distributions: homogeneous and heterogeneous distributions.

Fig. 4 gives the mean obtained reward using TS as a function of the number of iterations for $\sigma = 0,\ 2.76$ and 3.46 (see Table I for the exact number of users in each cell). Each of the presented curves in Fig. 4 is the output of averaging 250 trials.

It is clear that the higher the number of iterations, the higher the chance to get a better solution. A too high number of iterations would however prevent an operator from using the proposed algorithm in a dynamic context.

We can notice from Fig. 4 that the mean value of the reward increases with the increase of the iterations' number until it stabilizes (this is true for all values of σ).

The minimum number of iterations required for the mean reward to stabilize is found to be approximately 200, 240, and 320 iterations for $\sigma = 0,\ 2.76$, and 3.46 respectively.

In the homogenenous distribution case ($\sigma = 0$) the mean re-

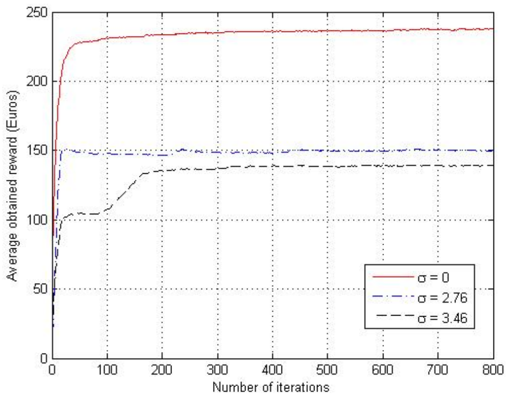

Fig. 4. Average reward versus the number of iterations.

ward curve stabilizes very fast. However in the heterogeneous case, the mean reward curve needs higher number of iterations to stablize. Obviously, finding the allocation which maximizes the reward in a heterogenous traffic case is more challenging.

We can notice that the required minimum number of iterations are reasonable enough to allow the operator to launch the algorithm at each new event.

It is very important to note that the obtained minimum number of iterations is very dependant on the initial start point. In a dynamic context, and at each new event, the operator is supposed to start TS algorithm from the last reached allocation solution. Hence the minimum number of iterations are expected to be reduced.

V. Conclusion and future work

We have presented and analyzed a TS algorithm for DSA in cellular systems adapted to packet services. We have studied a mono-operator case assuming the operator is sharing the spectrum. We focused on packet traffic context where we considered the revenues obtained from users as well as the spectrum price. We modeled an objective function for the maximization of the operator's reward. Our TS based DSA algorithm is simple, does not require an excessive memory space, and hence can be envisaged for the implementation in a dynamic context. Results have also shown that the algorithm is efficient in terms of convergence speed.

In our future work, we plan to study the temporal aspects of the traffic using event based simulations. We also plan to study a distributed approach that allows the operators to perform DSA on a larger RAN with high number of cells.

References

[1] M. Buddhikot, "Understanding Dynamic Spectrum Allocation: Models, Taxonomy and Challenges," *Proc. IEEE DySPAN'07*, pp. 649-663, 2007.

[2] T. Kamakaris, M. Buddhikot, and R. Iyer, "A Case of Coordinated Dynamic Spectrum Access in Cellular Networks," *Proc. of IEEE DySPAN*, 2005.

[3] FCC Spectrum Policy Task Force, "Report of the Spectrum Efficiency Working Group," 2002.

[4] FCC Spectrum Policy Task Force, "Report of the Spectrum Rights and Responsibilities Working Group," 2002.

[5] M. Buddhikot, P. Kolodzy, K. Ryan, J. Evans, and S. Miller, "DIMSUM-Net: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access," *Proc. IEEE WoWMoM'05*, pp. 78-85, 2005.

[6] Goutam Chakraborty, "An Efficient Heuristic Algorithm for Channel Assignment Problem in Cellular Radio Networks," *IEEE transactions on vehicular technology*, vol. 50, No. 6, 2001.

[7] D. Thilakawardana, K. Moessner, and R. Tafazolli, "Darwinian Approach for Dynamic Spectrum Allocation in Next Generation Systems," *IET Communications*, vol. 2, No. 6, pp. 827-836, 2008.

[8] A.P. Subramanian et al, "Fast Spectrum Allocation in Coordinated Dynamic Spectrum Access Based Cellular Networks," *Proc. IEEE DySPAN'07*, 2007.

[9] Y. T. Wang , J. P. Sheu, "A Dynamic Channel-Borrowing Approach with Fuzzy Logic Control in Distributed Cellular Networks," *Elsevier, Simulation Modelling Practice and Theory*, vol. 12, Issues 3-4, Modeling and Simulation of Distributed Systems and Networks, 2004.

[10] Yangjie Peng, Lipo Wang, and Boon Hee Soong, "Optimal Channel Assignment in Cellular Systems using Tabu Search," *Proc. IEEE PIMRC'03*, 2003.

[11] R. Montemanni, Jim N.J. Moon, and D.H. Smith "An Improved Tabu Search Algorithm for the Fixed-Spectrum Frequency-Assignment problem," *IEEE transactions on vehicular technology*, vol. 52, No. 3, May, 2003.

[12] A. Capone, and M. Trubian "Channel Assignment Problem in Cellular Systems: A New Model and a Tabu Search Algorithm," *IEEE transactions on vehicular technology*, vol. 48, No. 4, July, 1999.

[13] Ph. Tortelier "Measurements of Propagation and Cohabitation in Ile-de-France," *URC project document,SP2, Task2.1.3, Dynamic Spectrum Allocation algorithms (v2)*,2009

[14] H. Kamal, M. Coupechoux, and Ph. Godlewski, "Inter-Operator Spectrum Sharing for Cellular Networks using Game Theory," *Proc. of IEEE Personal, Indoor and Mobile Radio Communications Symposium (PIMRC)*, 2009.

[15] M. Coupechoux, H. Kamal, Ph. Godlewski, and J.M. Kelif, "Optimal and Heuristic DSA Policies for Cellular Networks with Coordinated Access Band," *Proc. of European Wireless*, 2009.

[16] N. Enderlé and X. Lagrange, "User Satisfaction Models and Scheduling Algorithms for Packet-Switched Services in UMTS," *Proc. VTC'03*, vol. 3, pp. 1704-1709, 2003.

[17] Glover F., "Tabu Search, Part I" *ORSA Journal on Computing*, vol 1, 1989.

[18] Glover F., "Tabu Search, Part II" *ORSA Journal on Computing*, vol 2, 1990.