An Efficient Tabu Search DSA Algorithm for Heterogeneous Traffic 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 propose and analyze a TS (Tabu Search) algorithm for DSA (Dynamic Spectrum Access) in cellular networks. We consider a scenario where cellular operators share a common access band, and we focus on the strategy of one operator providing packet services to the end-users. We consider a soft interference requirement for the algorithm's design that suits the packet traffic context. The operator's objective is to maximize its reward while taking into account the trade-off between the spectrum cost and the revenues obtained from end-users. We focus on the temporal heterogeneity of the traffic and we show that our algorithm allows the operator to increase its reward by taking advantage of this heterogeneity, rather than assuming homogeneous traffic for spectrum allocation. We study the dynamicity of the algorithm through event-based simulations. Results show that our algorithm uses less spectrum and achieves less blocking probability than the FSA (Fixed Spectrum Access) case at the price of reduced user throughput.**

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 is 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 in [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 in order to provide packet services for the end-users. We are interested in developing a DSA algorithm based on TS (Tabu Search) 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, the references [10], [11], [12], and [8] make a partial list of the references proposing 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, 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 account 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.

Our formulations presented in this paper leads to a simple algorithm that does not require excessive memory space, and

suits the implementation in a dynamic context. Hereafter we summarize our main contributions with respect to the related work: proposing and analyzing a DSA algorithm based on adapted TS method, where (1) we consider packet traffic services, (2) we address the spectrum pricing issue, (3) we set an objective function for maximizing the operator's reward, and (4) we introduce adaptations to TS for a dynamic deployment of the algorithm.

In this paper we extend our work in [13] and [14], where we supposed a *priori* that a classical frequency reuse scheme is deployed. We also extend the work in [15] by focusing on the temporal heterogeneity of the network's traffic, rather than the spatial heterogeneity as done in [15].

In this paper we also introduce amendements to the algorithm, in [15], to better suit the implementation in a dynamic context: (1) enhancing the initialization process, (2) increasing the algorithm's convergence speed through: a) limiting the number of generated neighbors at each new event depending on the event type, b) reducing the complexity of calculating the reward corresponding to each neighbor by selecting only the concerned cells. Finally (3) we evaluate the performance of the algorithm's dynamicity using event-based simulations.

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 finally 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 deploy a 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.

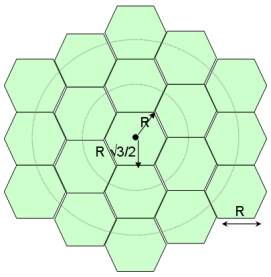


Fig. 1. Hexagonal network of study.

We assume a fair scheduling 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 [14].

We suppose a DSA decision is taken by the operator at each new 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.

At the very beginning, the operator is supposed to launch the TS for a sufficient number of iterations (which depends on the users' distribution in the RAN, as shown in [15], and that ensures the algorithm is able to track the traffic changes), then at each new event the operator launches TS-based DSA for a reasonably limited number of iterations.

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 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:

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

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 C_c is the sum of capacities provided by the frequency blocks used by the cell:

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

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

The basic idea is to forbid a move that would return to recently visited solutions by classifying them as tabu. The algorithm uses a memory structure called *TL (Tabu List)* to avoid cycles. 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]. The TL has a limited size called *Tabu Tenure TT*.

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 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.

As the minimum *number of iterations* required to reach an "efficient" solution using TS is very dependent on the initial start point, the basic idea of our TS-based DSA algorithm makes use of this dependency and applies the TS at each new event. In a dynamic context, the algorithm is launched at each new event where it starts from the last reached allocation solution.

B. Definitions

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

A solution s, in our context, 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 equals to 1 if frequency f is assigned to cell c, and 0 otherwise.

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, 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 that 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 solution are added to the TL. We have chosen to consider the reward corresponding to each selected solution (among all neighbors at each iteration) as its attribute (see section III-C).

C. Implementation

We present in Algorithm 1 our TS algorithm that suits the CAP for packet services.

We illustrate now the details of the steps given in Algorithm 1, and we illustrate the improvements with respect to the work in [15] that suit a dynamic deployment of the algorithm.

Initialization: From our experience, we have noticed that the TS algorithm assigns a number of blocks to each cell that is proportional to the number of users in the cell. Accordingly, in this paper we add an amendment to the initialization method used in [15] making use of this property, to get a good starting solution closer to the zone where the optimal solution exists.

We have chosen an initialization method based on randomly formed solutions. Note that the operator does not have any

Algorithm 1 TS algorithm for reward maximization in packet services context

1: ***Initialization***: an initial solution s_{init} is found.
2: $s \leftarrow s_{init}$
3: $g_{max} \leftarrow g(s_{init})$
4: **while** Nb. of iterations $\leq MAXITER$ **do**
5: ***Neighborhood formation***: all *possible* neighbors of the initial solution s are created, except those who are listed as tabu.
6: ***Neighbor selection***: the solution s' that achieves the maximum reward is chosen, among the set of neighbors, $s \leftarrow s'$
7: ***Tabu list update***: the reward $g(s')$ corresponding to the selected solution s' is added to the TL.
8: ***Maximum reward update***: g_{max} is updated: **if** $g(s') > g_{max}$, **then** $g_{max} \leftarrow g(s')$ **end if**
9: **end while**

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 F the operator can lease, $F \in (1, ..., F_{max})$. We generate randomly 300 *possible* solutions for each search zone with the conditions: (1) only one block is assigned to the cell(s) having one user, (2) a number of blocks equals to F is assigned to the cell(s) having the maximum number of users. 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*.

For an efficient algorithm that can be executed at each new event, a limitation of the number of neighbors for the first iteration is introduced, in this paper, depending on the event type (i.e. arrival or departure of a user). In case of arrival or departure of a user, replacing one of the used blocks is considered, however: (a) adding a non-used block to a random cell is only considered in case of arrival, and (b) removing an assigned block from a random cell is only considered in case of departure.

Neighbor selection: In order to choose one TS neighbor, the operator needs to calculate the reward for all neighbors. This process might affect the efficiency of the algorithm. In order to reduce the algorithm's complexity, we introduce in this paper the following method for the selection of cells while calculating the reward for each TS neighbor.

For each move that creates a TS neighbor, only the affected cells by the move are chosen for the reward update. For example: In case of replacing f_5 by f_4 in a cell, then only the cells using f_5 or f_4 need to be updated.

Tabu list update: we have chosen to consider the reward corresponding to each selected solution (among all neighbors at each iteration) as its attribute. There is two main advantages behind this approach: 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 who 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.

IV. Numerical results

A. Simulation scenario 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 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: $K_u = 20$ euros and $K_B = 100$ euros. The maximum number of users that each cell accepts (according to the Connection Admission Control configuration) equals to 10 users. We suppose a heterogenous traffic (arrival rate per cell) in the RAN. There is a high concentration of traffic in the central cell, and this concentration decreases as the distance from the cluster's center increases. We assume the arrival rate for the central cell $\lambda_1 = 4\lambda_2$, where λ_2 is the arrival rate per cell for the cells in the middle-circle, and $\lambda_1 = 6\lambda_3$, where λ_3 is the arrival rate per cell for the cells in the outer-circle.

TS algorithm parameters are set as follows: TT $= 200$, the maximum number of iterations $MAXITER = 300$ iterations at the very begining of launching the TS, and 10 iterations at each new event. The presented results are the average of $20K$ events.

According to the defined parameter set as well as to the neighbor definition, the total number of possible neighbors created from a solution s, in the worst case, equals to 285 neighbors. It is clear that, for our case study, generating all possible neighbors at each iteration is very feasible.

We compare the reward obtained by two operators; (1) one assumes a homogeneous traffic over the RAN to perform its channel assignment, and (2) a second operator uses DSA, and considers the exact (heterogenous) distribution of the traffic to dynamically assign spectrum blocks.

In the coming section we evaluate the performance of the algorithm's dynamicity using event-based simulations.

B. Results

We compare between FSA and TS-based DSA in terms of the obtained reward, CAB utilization, blocking probability, and the user throughput. Note that in FSA case the operator uses the whole spectrum of the CAB and applies reuse 3 scheme.

Fig. 2 gives the obtained reward as well as the CAB utilization versus the mean arrival rate λ using both TS-based DSA and FSA.

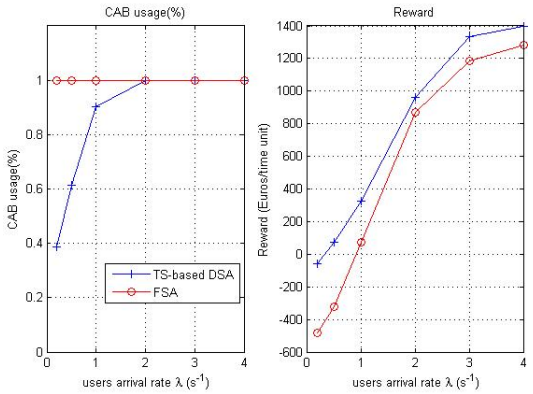


Fig. 2. CAB utilization and obtained reward obtained using FSA and TS-based DSA.

We can notice that the obtained rewards using TS-based DSA exceed significantly the rewards obtained using FSA for all values of simulated λ, even for $2 \leq \lambda \leq 4s^{-1}$ where both techniques use 100% of the CAB. At $\lambda = 1s^{-1}$, the reward obtained using TS-based DSA is 334.5 compared to 75.9 using FSA ($+345\%$). We can also see from Fig. 2 that a considerable spectrum conservation using the proposed DSA algorithm is achieved with respect to FSA for $\lambda < 2s^{-1}$.

Fig. 3 gives the end-user throughput as well as the blocking probability obtained using both FSA and TS-based DSA.

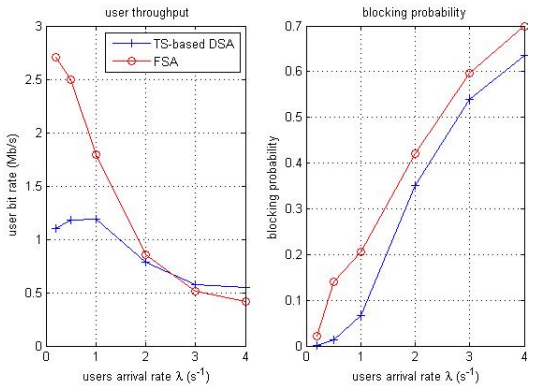


Fig. 3. Blocking probability and end-user throughput obtained using FSA and TS-based DSA.

As the FSA uses more spectrum than TS-based DSA, consequently the achieved user throughput is reduced for the DSA case, especially for low values of λ, as shown in Fig. 3. Note that in FSA case a single user in a cell takes advantage of using 2 blocks assigned to the cell, while in DSA case only 1 block is assigned to the user. As the spectrum allocation depends on the cells' traffic loads, the blocking probability decreases using DSA with respect to FSA.

V. Conclusion

We have proposed and analyzed a TS-based DSA for cellular systems adapted to packet services. We have considered a spectrum sharing context and we focused on the strategy adapted by one operator to maximize its reward. The revenues are modeled as an increasing function of the achieved throughput of end-users. We adapted the TS objective function to suit packet traffic. Our TS-based DSA algorithm is simple, does not require an excessive memory space. The proposed algorithm allows the operator to increase its rewards, to use less spectrum, and to achieve less blocking probability, however at the price of reduced user throughput.

References

[1] M. Buddhikot, "Understanding Dynamic Spectrum Allocation: Models, Taxonomy and Challenges," *Proc. IEEE DySPAN'07*, 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, "DIMSUMNet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access," *Proc. IEEE WoWMoM'05*, 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, 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] H. Kamal, M. Coupechoux, and Ph. Godlewski, "Inter-Operator Spectrum Sharing for Celullar Networks using Game Theory," *Proc. of IEEE PIMRC*, 2009.

[14] 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.

[15] H. Kamal, M. Coupechoux, and Ph. Godlewski, "A Tabu Search DSA Algorithm for Reward Maximization in Cellular Networks," *Proc. of IEEE WiMoB*, 2010.

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

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