Optimal, Heuristic and Q-learning Based DSA Policies for Cellular Networks with Coordinated Access Band

Hany Kamal, Marceau Coupechoux, Philippe Godlewski, Jean-Marc Kélif

EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Euro. Trans. Telecomms. **00**: 1–10 (2007)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/ett.0000

Optimal, Heuristic and Q-learning based DSA Policies for Cellular Networks with Coordinated Access Band†

Hany Kamal[1]*, Marceau Coupechoux[1], Philippe Godlewski[1], Jean-Marc Kelif[2]

[1] *Telecom ParisTech & CNRS LTCI, France*
[2] *Orange Labs, France*

SUMMARY

Due to the increasing demands for higher data rate applications, also due to the actual spectrum crowd situation, DSA (Dynamic Spectrum Access) turned into an active research topic. In this paper, we analyze DSA in cellular networks context, where a CAB (Coordinated Access Band) is shared between RANs (Radio Access Networks). We propose an SMDP (Semi Markov Decision Process) approach to derive the optimal DSA policies in terms of operator reward. In order to overcome the limitations induced by optimal policy implementation, we also propose two simple, though sub-optimal, DSA algorithms: a Q-learning (QL) based algorithm and a heuristic algorithm. The achieved reward using the latter is shown to be very close to the optimal case and thus to significantly exceed the reward obtained with FSA (Fixed Spectrum Access). The rewards achieved using the QL-based algorithm are shown to exceed those obtained using FSA. Higher rewards and better spectrum utilization with DSA optimal and heuristic methods are however obtained at the price of a reduced average user throughput.
Keywords: *Cellular systems, Dynamic Spectrum Access, Markov Desicion Process, Q learning.* —

1. INTRODUCTION

Wireless networks are facing increasing demand for high data rate applications, and hence their demand for spectral resource increases. Researchers have started working on DSA algorithms as a solution to the spectrum scarcity problem encouraged by the rapid progress in SDR (Software Defined Radio) systems that enable the required reconfigurability for DSA and cognitive radio equipments.

In [1], the spectrum management models are divided into four main axis: command and control, exclusive-use, primary/secondary usage, and commons. The exclusive-use model includes a dynamic mode, where spectrum is owned by a single operator at any given point in space or time; owner and usage of the spectrum can however dynamically change. This model is thus particularly adapted to cellular networks. In this context, the IEEE P1900.4 working group has detailed three use cases [2] [3] with increasing levels of reconfigurability and joint management of resources. In this paper, we focus on the first one and consider a single operator with several RANs, able to dynamically distribute its frequency bands between its RANs.

Several papers are dealing with DSA for cellular networks. For example, in [4], authors propose a coordinated DSA system where a pool of resources (CAB or Coordinated Access Band) is shared and controlled by a regional spectrum broker. In [5], authors made use of the genetic algorithm to analyze the DSA in WCDMA networks. In [6], authors propose a MAC protocol enabling ad-hoc secondary users to utilize the unused resources of a GSM system.

It is however difficult to separate technical from pricing aspects when DSA is considered, especially for cellular operators who pay very high prices for the license. The wide interest in DSA is indeed mainly driven by the

*Correspondence to: Hany Kamal, 46 rue Barrault, Paris, France. E-mail: hany.kamal@telecom-paristech.fr
†A short version of this paper was presented in the 15th European Wireless Conference (EW 2009), Aalborg, Denmark

Received 9 November 2009
Revised
Accepted

expected benefits resulting from sharing the spectrum [7]. Reference [8] analyzes a network model where the service providers base stations are sharing a common amount of spectrum. A distributed DSA algorithm is proposed where each user maximizes its utility (bit rate) minus the payment for the spectrum. In [9], authors have considered a spectrum market, where they propose a Rubinstein-Stahl method for the spectrum trading.

In this paper, we present an approach based on SMDP to analyze DSA in a cellular context. We analyze a network model, where different RANs are sharing a CAB, inspired by the idea of resource sharing proposed in [4] and by the single operator use case presented in [3]. We take into account the spectrum price, and maximizing the operator revenue is our main concern.

MDP approach has been used to solve several optimization problems in telecommunication networks. In the context of cognitive radio, reference [10] proposes a cognitive medium access protocol that maximizes the throughput while limiting the interference affecting the primary user. The authors formulated the problem within the framework of constrained MDP. In [11], a POMDP (Partially Observable MDP) framework is proposed to optimize the performance of the secondary users while limiting the interference perceived by the primary users. These references however focus on the primary/secondary usage model.

In [12], the SMDP framework is used in a JRRM (Joint Radio Resource Managment) context in order to take an optimal CAC (Call Admission Control) decision, whether to accept a new coming call or to reject it. The reward function in [12] presents the end-user throughput. Different from [12], SMDP is used in this paper to find the optimal spectrum bands allocations. The reward function in this paper takes into account both user satisfaction and spectrum price.

The paper is organised as follows: section 2 presents the network model in terms of system model, traffic model and the principle of DSA operation. The SMDP approach is presented in section 3. In section 4, we propose a sub-optimal DSA heuristic easier to implement for an operator than optimal policies. Section 5 proposes an alternative solution based on Q-learning. The performances of the three approaches (optimal DSA, heuristic DSA and QL based DSA) are compared to the FSA case in section 6. Conclusion is given in section 7.

2. NETWORK MODEL

2.1. System model

We intend to study cell-by-cell DSA between two access networks. RANs are supposed to be homogeneous in propagation and in traffic, and the operator is assumed to deploy classical frequency reuse schemes (i.e., reuse 1 or reuse 3). Based on these assumptions, all cells of a RAN statistically behave the same way, we can thus focus on a single cell per RAN.

The system is thus made of two cells of two different RANs (in this paper, terms cell and RAN will be used indifferently). RANs do not have their own spectrum bands but rather have to dynamically access to a CAB. The CAB is sub-divided into m_{max} elementary spectrum bands (blocks) that can be used indifferently by any RAN. As traffic grows, a RAN can lease a new elementary band (one block) and as it decreases, the RAN can leave it free for the common pool (section 3.3). We assume that the average data rate accessible by users in a RAN is proportional to the bandwidth allocated to the RAN and is equally divided among all users of the RAN (section 2.2). The model is shown in Fig. 1. Parameters n_i, $i = 1, 2$ are the number of active users in RAN1 and RAN2. Parameter m_i is the current number of elementary bands leased by RAN i from the CAB.

Both RANs are operated by a single operator responsible for attributing or freeing elementary bands to each RAN. On the one hand, revenue is assumed to be proportional to the satisfaction of the users. On the other hand, it is supposed that spectrum cost follows the law of supply and demand: as free spectrum diminishes, spectrum cost increases (section 3.2). We are interested in the optimal policy that assigns bandwidth to the RANs.

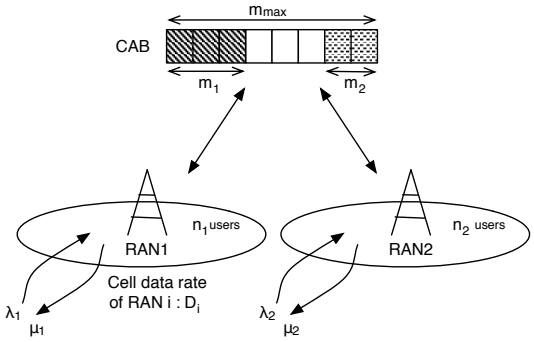

Figure 1. System model: two RANs access to a CAB according to their needs.

Our model could be coherent with multi-carrier HSPA (High Speed Packet Access) systems. According to 3GPP release 8, an evolution of the HSPA systems will indeed allow the aggregation of 5 MHz carriers [13].

Our model could also be coherent with SOFDMA (Scalable Orthogonal Frequency Division Multiple Access) cellular networks (i.e. WiMAX, 3GPP-LTE), where the bandwidth of the system is scalable [14]. In these systems the operator has indeed an additional flexibility in resource allocation through the possibility of scaling the bandwidth.

2.2. *Traffic*

We consider a bursty packet traffic, such as web browsing or file downloading on the downlink: a user alternates between packet calls (several packets are transferred in a very short time) and reading times (there is no transfer). In this paper, we focus on the packet call level and so we neglect the details of the packet level.

We assume Poisson arrivals of user downlink packet calls with rate λ_1 in RAN1 and λ_2 in RAN2 (see Fig. 1). Traffic is supposed to be elastic: the packet call size is exponentially distributed with mean X_{ON} bits in both RANs and so the service rate depends on the available RAN throughput. We assume a throughput fair scheduling between users of a given RAN. For RAN i let D_i be the cell data rate (in bits/s) accessible with an elementary spectrum band. Then, the service rates can be written as:

$$\mu_i = \frac{m_i D_i}{X_{ON}}. \quad (1)$$

An illustration of the traffic model is shown in Fig. 2. Arrows on the time axis represent Poisson arrivals of new packet calls and grey rectangles their duration. Packet calls are made of several packets that together represent X_{ON} bits.

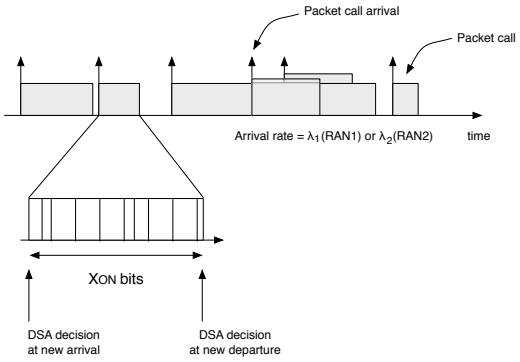

Figure 2. Assumed traffic model.

2.3. *Dynamic spectrum access*

In the considered system model, the core issue for the operator of the RANs lies in the trade-off to be found between spectrum cost and revenues obtained from users: more spectrum per RAN means a higher lease cost for the operator but also higher throughputs for users that are encouraged to pay more for the service. As the CAB size is limited and as spectrum cost increases with increasing demand, there is a strong interaction between RANs.

In this paper, a DSA policy is a strategy that dynamically attributes spectrum bands to each RAN from the CAB. We assume that a DSA decision is taken at each new event, i.e., a new packet call arrival or a packet call departure in any RAN (see Fig. 2). A DSA decision is supposed to increase the number of spectrum bands for a RAN by a single block, to decrease by a single block this number, or to keep constant the spectrum of a RAN. We thus do not allow too abrupt changes in resource allocation.

We further assume that at least one spectrum block is always available to each RAN, so that starvation is not possible. We are now interested in the optimal DSA policies in terms of operator revenue.

3. OPTIMAL DSA POLICIES

In order to achieve this goal, we rely on the SMDP framework. We first define the SMDP and the reward function, then use uniformization to obtain an MDP and rely on the policy iteration algorithm to find the optimal DSA policy.

3.1. *State space*

The system state is given by all four-tuple (n_1, m_1, n_2, m_2) with constraints $n_1 \leq n_1^{max}$, $n_2 \leq n_2^{max}$ and $m_1 + m_2 \leq m_{max}$. The limitation imposed to the number of active users is equivalent to setting a minimum throughput per RAN. Let S be the state space.

3.2. *Reward function*

The reward function is based on the revenue expected by the operator. The higher the satisfaction of users, the higher the operator revenue; the higher the amount of bandwidth leased by RAN, the higher the cost to lease this spectrum band. We define a comfort service rate μ_{com}. The revenue obtained from a given customer in RAN i increases with its satisfaction:

$$\phi_i(n_i, m_i) = K_u(1 - \exp(-\mu_i/n_i\mu_{com})),$$

where K_u is a constant in euros per unit of satisfaction. Satisfaction, defined in [15], is an increasing function of the user data rate and is without unit. As the scheduling is fair in throughput, each user gets a data rate proportional to μ_i/n_i in RAN i. Thus the total revenue obtained by the operator in state $s = (n_1, m_1, n_2, m_2)$ is

$$g_1(s) = n_1\phi_1(n_1, m_1) + n_2\phi_2(n_2, m_2).$$

We assume that the spectrum price is increasing when the amount of free spectrum decreases and we define it as:

$$g_2(s) = K_B(m_1 + m_2)\exp\left(-\frac{m_{max} - m_1 - m_2}{m_{com}}\right),$$

where m_{com} is a constant that controls the variation of the price and K_B is a constant in euros per MHz (it is the equivalent spectrum price per cell). If m_{com} is high, the exponential function is close to 1 whatever the state. If m_{com} is small, there is a high discount when the CAB is free. Note that the price paid by the operator for a given elementary band varies with the occupation of the CAB. The global reward function per time unit can thus be written in state s:

$$g(s) = g_1(s) - g_2(s). \tag{2}$$

Note that $g(s)$ is defined per time unit because the longer the spectrum is used, the more the operator pays. In the same way, for a given throughput, the longer a user is using the bandwidth, the more he has to pay.

3.3. Action space

In each state, the operator is allowed to increase, decrease or leave unchanged the spectrum of each RAN. As shown in Fig. 2, a decision epoch occurs at each packet call arrival, or departure. As state transitions occur only at the arrival or the departure of a single user, we assume that the band assigned to a single RAN can be increased or decreased by a single elementary band. This leads to nine possible actions of the form $a = (a_1, a_2),\ a_i \in \{0, -1, +1\}$ given in Tab. 1.

The effective action space depends on the state. If $m_i = 1$ the spectrum band of RAN i cannot decrease. If the CAB is blocked, i.e., if $m_1 + m_2 = m_{max}$, no band can increase.

Table 1. List of possible actions

Action	a vector	action index
Band1 constant and Band2 constant	$(0, 0)$	1
Band1 constant and Band2 increases	$(0, +1)$	2
Band1 constant and Band2 decreases	$(0, -1)$	3
Band1 increases and Band2 constant	$(+1, 0)$	4
Band1 increases and Band2 increases	$(+1, +1)$	5
Band1 increases and Band2 decreases	$(+1, -1)$	6
Band1 decreases and Band2 constant	$(-1, 0)$	7
Band1 decreases and Band2 increases	$(-1, +1)$	8
Band1 decreases and Band2 decreases	$(-1, -1)$	9

3.4. Transition probabilities

Let $p_{s,s'}(a)$ be the probability that at the next decision epoch (i.e., at the next transition), the system will be in state $s' = (n_1', m_1', n_2', m_2')$ if a is chosen in state $s = (n_1, m_1, n_2, m_2)$. Let $1/\nu_s(a)$ be the expected time until next decision epoch if action a is chosen in state s:

$$\begin{aligned}\nu_s(a) \quad = \quad & \mathbb{1}_{\{n_1 < n_1^{max}\}}\lambda_1 + \mathbb{1}_{\{n_2 < n_2^{max}\}}\lambda_2 \\ & + \mathbb{1}_{\{n_1 > 0\}}\mu_1 + \mathbb{1}_{\{n_2 > 0\}}\mu_2.\end{aligned}$$

Transition probabilities are given by:

$$p_{s,s'}(a) = \begin{cases} \lambda_i/\nu_s(a) & \text{if } (n_i' = n_i + 1) \\ & \text{and } (\forall j\ m_j' = m_j + a_j), \\ \mu_i/\nu_s(a) & \text{if } (n_i' = n_i - 1) \\ & \text{and } (\forall j\ m_j' = m_j + a_j), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

3.5. Uniformization

A step of uniformization is now needed in order to transform the continuous time Markov chain into an equivalent discrete time Markov chain [16]. This is done by choosing a sufficiently small transition step $1/\nu$ ($\forall s, a,\ \nu_s(a) \leq \nu$) and allowing self transitions from a state to itself.

Transition probabilities are modified in the following way:

$$\tilde{p}_{s,s'}(a) = \begin{cases} p_{s,s'}(a)\nu_s(a)/\nu & \text{if } s \neq s', \\ 1 - \sum_{s' \neq s}\tilde{p}_{s,s'}(a) & \text{otherwise.} \end{cases} \tag{4}$$

A DSA policy R associates to each system state s, an action $R(s)$ from the action space of s.

DOI: 10.1002/ett

3.6. Policy iteration

We are interested in finding the optimal policy R^* of the continuous-time average cost problem described above. For that, we apply the policy iteration algorithm to the auxiliary discrete-time average cost problem obtained after uniformization (see [16], vol.2, p.315). The iterative algorithm is now succinctly described, it iteratively solves Bellman equations in a synchronous manner.

Algorithm 1 Policy Iteration

1: **Initialization**: Let R be an arbitrary stationary policy.
2: **Value-determination**: For the current policy R, we solve the system of linear equations whose unknowns are the variables $\{J_R, h_R(s)\}$: $h_R(1) = 0$ and

$$h_R(s) = g(s) - J_R + \sum_{s' \in S} \tilde{p}_{s,s'}(R(s)) h_R(s').$$

3: **Policy improvement**: For each $s \in S$, we find:

$$R'(s) = arg \max_{a \in A(s)} \{g(s) - J_R \quad + \sum_{s' \in S} \tilde{p}_{s,s'}(a) h_R(s') \Big\}.$$

4: **Convergence test**: If $R' = R$, the algorithm is stopped, otherwise, we go to step 2 with $R := R'$.

The SMDP approach has the advantage of providing optimal policies and an upper bound on the achievable reward. The policy iteration algorithm takes into account not only RANs loads, the number of active users and RANs interactions but also the whole dynamics of the system. Optimal policies are thus strongly dependent on the system parameters and simple examples cannot be easily generalized when the number of system states increases. In the next section, we propose a sub-optimal DSA heuristic that overcomes these limitations for an operator, while still providing a high reward.

4. HEURISTIC DSA

4.1. DSA policies implementation

In order to implement optimal policies, an operator would have to run the policy iteration algorithm for all possible system parameter sets and store results to be dynamically used according to the context. Running policy iteration on a real-time basis seems indeed difficult, especially when the number of system states increases (for example if many cells or users are considered). The proposed DSA heuristic intends to ease DSA implementation for an operator. With this heuristic, massive storage of data is not needed and computations can be done on the fly.

4.2. Proposed DSA heuristic

Optimal DSA policies decisions are taken at each new event (a packet call arrival or departure) and thus depend not only on the arrival rates (λ_1, λ_2) but also on the variations of the number of users (n_1, n_2). In order to obtain a simple heuristic, we focus only on the arrival rates and neglect the variations of (n_1, n_2).

Let us now consider that (m_1, m_2) is fixed for a given couple (λ_1, λ_2). In this case, each of the RANs can be considered as a M/M/1/n_i^{max} system. The service rate μ_i is indeed constant (see Eq. 1) and in every state n_i, the departure rate is $\mu_i = n_i \times \mu_i / n_i$ because of the throughput fairness scheduling assumption.

With these assumptions, the average heuristic reward for the operator, g_H, can be easily computed for all possible combinations of alloacted bands (m_1, m_2), along with the corresponding λ_i values. The average reward is the sum of the rewards obtained from the two RANs. For a given $(\lambda_1, \lambda_2, m_1, m_2)$:

$$g_H(\lambda_1, \lambda_2, m_1, m_2) = \sum_{i=1}^{2} \sum_{n_i=0}^{n_i^{max}} \pi_{n_i}(\lambda_i) n_i \phi_i(n_i, m_i) - g_2(n_1, m_1, n_2, m_2), \quad (5)$$

where the $\pi_{n_i}(\lambda_i),\ i \in \{1, 2\},\ n_i \in \{0, ..., n_i^{max}\}$ are the steady state probabilities of a M/M/1/n_i^{max} with arrival rate λ_i and service rate μ_i. We use this result for the proposed DSA heuristic:

Algorithm 2 Heuristic DSA

1: Estimate arrival rates λ_1 and λ_2.
2: **for all** (m_1, m_2) **do**
3: Compute the average reward g_H according to Eq. 5.
4: **end for**
5: Allocate bandwidth according to the tuple (m_1, m_2) that maximizes the average reward g_H.

Eq. 5 can be instantaneously computed for realistic values of the n_i^{max} and can be easily extended to several cells. Note that this proposed heuristic algorithm still need the knowledge of several system parameters, like λ_1, λ_2, μ_1, μ_2 and X_{ON}, which can not be always easily obtained

Euro. Trans. Telecomms. **00**: 1–10 (2007)
DOI: 10.1002/ett

by the operator. We now propose an alternative solution based on Q-learning in order to overcome this limitation.

5. Q-LEARNING BASED DSA

5.1. Reinforcement learning

Reinforcement learning (RL) is a simulation-based dynammic programming technique used to solve complex MDPs (Markov Decision Problems) without the need of knowing the transition probabilities. RL is concerned about learning how to take an action that maximizes a specific metric, typically long-term rewards. The algorithm trains an agent to take the appropriate action in response to the environment reactions. The agent learns by analyzing its actions through the evaluation of the received rewards (for each action it takes). In RL, the agent / environment interactions are usually modeled by an MDP (Markov Decision Process) [17], [18]. In this paper, we are going to use the Q-learning technique that solves a MDP using the value-iteration method.

5.2. Q-learning for continuous average-cost problem

In Q-learning, the agent learns the action-value function with the target of determining a policy that maximizes long-term rewards. The value function is a function (known as Q function) that gives the expected long-term reward obtained by applying a certain policy. The Q function represents an evaluation of each action, taken by the agent, and associates it with the environment-state at the moment of executing this action.

Most approaches to RL, including QL, are developed to optimize discrete discounted-reward problems. The original QL algorithm presented by Watkins [20], was based on discounted reward value iteration [21]. Discounted optimization is motivated by domains in which reward is money that earns interest in each time step [17].

In case of average cost problems, QL does not apply immediately. Setting the discount factor to 1 in the Q-learning algorithm would be equivalent to base the method on the average cost value iteration, which is known to be unstable. Using a high discount factor would cause the learning convergence to be too slow. Authors of [19] and [17] have proposed RL algorithms that solve the average cost problem. They are however designed for discrete-time models. An interesting solution for both average cost and continuous-time problems can be found in [21].

5.3. Gosavi algorithm

In this section we give the details of the *Gosavi* algorithm [21] that we used to implement a QL-based DSA for our average-reward continuous-time problem.

5.3.1. Q factors update: Like in the traditional Q-learning algorithm, the agent in state s_t takes an action a_t; this causes the system to move to state s_{t+1}; reward r_t is observed and the value of the action is denoted $Q(s_t, a_t)$. The Q function (or the Q-factor) is updated each time there is a state transition. The action is then taken at the instant of a new event. The Q function is updated formally, according to [21], as follows:

$$Q(s_t, a_t) \quad = (1-\alpha)Q(s_t, a_t) + \alpha\, r_t - \alpha\, \rho\, \delta_t \quad (6)$$
$$+ \alpha\, arg\ \max_{a \in A(s)} \{Q(s_{t+1}, a)\},$$

where α is a learning factor, ρ is the estimated average-reward, and r_t denotes the reward obtained upon spending a period δ_t in state s_t.

Fig. 3, gives an illustration of the Q-factor updating principle, along with the instants where the actions are executed, as implemented in our simulator.

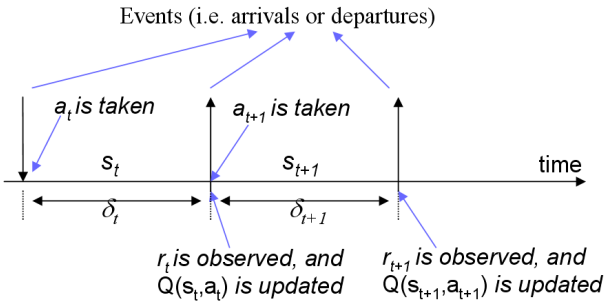

Figure 3. Illustration of the Q-value updating principle in our system-model context.

The algorithm's main idea is a relative value iteration update. At every state-transition instant, the agent updates the old Q-factor according to the new information. When the system visits a state, the agent selects the action with the highest rewards, this is represented in the term $arg\ \max_{a \in A(s)} \{Q(s_{t+1}, a)\}$ in Eq. 6

Although average-reward value iteration is numerically unstable, Gosavi's algorithm uses a relative value-iteration method. The relative value iteration method differs by substracting some value ($\delta_t \rho r_t$) from the Q-factor.

5.3.2. Exploration-exploitation policy: In this paper, we use a *p-greedy* method to explore and exploite: with a probability p, the agent chooses a random action among the given set of authorized actions, and with a probability $1-p$, the agent exploits the Q-factors.

5.3.3. Learning factors: In case the agent chooses to exploit the Q-factors, he updates the estimated average-reward ρ, using a decreasing (and a second) learning factor β. The parameter ρ is updated as follows:

- The estimated total reward C is updated as:

$$C \leftarrow (1-\beta)C + \beta\, r_t. \tag{7}$$

- The estimated total time T is updated as:

$$T \leftarrow (1-\beta)T + \beta\, \delta_t. \tag{8}$$

- Parameter ρ is updated as:

$$\rho = C/T. \tag{9}$$

The *Gosavi* algorithm is thus a two time-scales QL algorithm, the average cost is approximated on one time scale and the Q-factor on the other [21].

Two learning factors on two time scales are used. Both of them decrease as the algorithm runs. The learning factor α depends on the number of times the state-action pair was tried until that decision epoch. The learning factor β depends on the number of decision epochs in which the Q-factors have been exploited.

In order to compare this approach to other DSA policies, we first launch the algorithm during a learning phase during which we use the exploration-exploitation policy and two decreasing learning factors (α and β). By the end of the learning phase, the QL algorithm provides us with the output policy. We then calculate analytically the average reward knowing the policy provided by the QL algorithm. The details of the algorithm are given in Algorithm 3.

Algorithm 3 Q-learning based DSA

1: **Initialize** the following parameters:
   - the initial state: $s_t = (0,0,1,1)$.
   - Q-factors: $Q(s,a) = 0, \forall s \in S$ and $a \in A(s)$.
   - the estimated total cost: $C = 0$.
   - the estimated average reward: ρ is set to average reward obtained with the heuristic DSA.
   - the number of times Q is exploited: $k = 0$.
   - the number of visits to the state-action pair (s,a): $N_v(s,a) = 0, \forall s \in S$ and $a \in A(s)$.
2: **repeat**
3: **Exploration-exploitation policy**: draw uniform random variable X on $[0,1]$.
4: **if** $X < p$ (exploration) **then**
5: Choose action a_t at random
6: **else**
7: Choose action a_t that maximizes $Q(s_t,a)$ on the set of actions in state s_t.
8: Update learning factors $\alpha = 1/(1+N_v(s,a))$ and $\beta = 1/(1+k)$.
9: Update $Q(s_t,a_t)$ according to Eq. 6.
10: Update the estimated average reward ρ according to Eq. 7, 8 and 9.
11: $k \leftarrow k+1$.
12: **end if**
13: $N_v(s_t,a_t) \leftarrow N_v(s_t,a_t)+1$.
14: $s_t \leftarrow s_{t+1}$.
15: $t \leftarrow t+1$.
16: **until** End of the learning period

6. PERFORMANCE EVALUATION

In this section, we compare the results obtained with optimal DSA policies, the proposed heuristic, the QL-based DSA and FSA in terms of operator reward, CAB utilization, and average user throughput.

6.1. Parameters

The CAB is assumed to have a size of 6 MHz, the elementary band (m_i = 1) has a size of 1 MHz, and m_{com} = 4 MHz. For the sake of simplicity, we assume that both RANs have the same characteristics: the average cell data rates D_i are considered to be 1250 Kbps, $X_{ON} = 3$ Mbits, $\lambda_1 = \lambda_2 = \lambda$, and $n_1^{max} = n_2^{max} = 8$. The pricing constants are fixed as follows: $K_u = 100$ euros, $K_B = 1$ euro, and $\mu_{com} = 0.167$ s^{-1}(which corresponds to a comfort throughput of 500 Kbps).

Concerning the QL-based DSA algorithm, the agent keeps learning (and updating the Q-factors) for 200 thousands events, and the results are averaged over 20 iterations.

6.2. *Arrival rate thresholds for heuristic DSA*

For the considered parameter set, Fig. 4 shows the average reward g_H (see Eq. 5) as a function of the arrival rate λ for different combinations of the allocated bands.

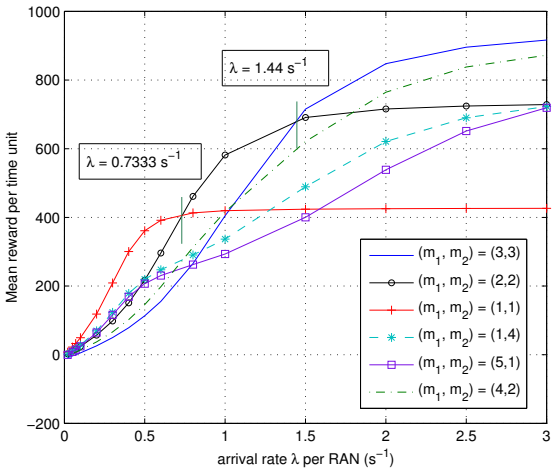

Figure 4. Operator reward obtained for different allocated bands combinations and load thresholds for heuristic DSA.

We can notice that the (m_1, m_2) values that give the maximum reward are: (1,1), (2,2), and (3,3) depending on the arrival rate λ. The maximum reward can then be obtained by dynamically allocating symmetric numbers of elementary bands to the RANs according to the cell load. This result was expected since in this simulation $\lambda_1 = \lambda_2$. Threshold values for λ are given on Fig. 4.

6.3. *Convergence of the QL-based DSA*

We illustrate in this section the convergence of the QL-based DSA algorithm through a study on the estimated average reward ρ. Fig. 5 represents the convergence of the estimated average reward ρ as a function of the number of events for two different arrival rates, $\lambda = 0.2$ and $\lambda = 1.5$ s^{-1}.

We can notice that the value of ρ fluctuates at the beginning of the learning phase and starts to stabilize after a certain duration (i.e., number of events). The duration required for ρ to stabilize is the period equivalent to about 200 thousands events in our examples. Note that the Q-learning algorithm theoretically converges to the optimal policy after an infinite duration. In our simulations, we stop learning after a realistic duration (200 thousands events) and exploit the obtained policy.

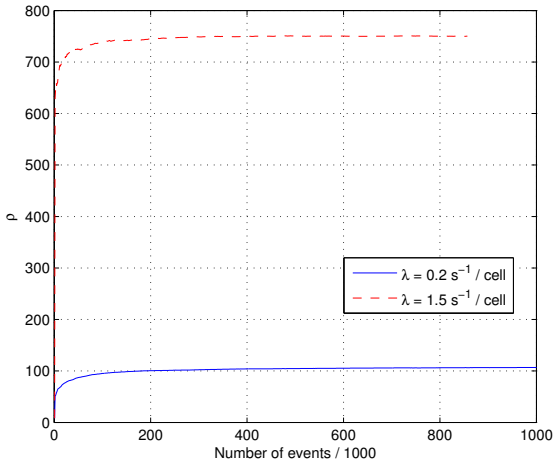

Figure 5. Convergence of the estimated average reward ρ for Q-learning based DSA.

6.4. *Operator reward, CAB utilization, user throughput*

Fig. 6 compares operator rewards obtained respectively with optimal DSA policy, the proposed heuristic, the QL-based DSA and FSA. By definition, FSA allocates $m_i = 3$ elementary bands to each RAN whatever the system state. It can be seen that optimal policies provide significant increases of the reward for low to intermediate values of λ (for example $+229\%$ at $\lambda = 0.5$ s^{-1}). At high load, FSA and optimal DSA policy converge as expected. Both the proposed heuristic and the QL-based DSA provide the optimal reward at low load and converges also to FSA at high load: only for intermediate values of λ, there is a small degradation of the reward (for example, -21% at $\lambda = 0.7$ s^{-1} for the heuristic method).

Fig 7 gives a comparision of the reward gains, in percentage, for the proposed DSA methods with respect to the rewards obtained using FSA.

We can notice that, the three proposed methods achieve gain in terms of rewards over FSA for arrival rate values $\lambda < 2\ s^{-1}$. All the proposed DSA methods give rewards that converge to the same reward values as FSA for arrival rates $\lambda > 2\ s^{-1}$.

These results can be explained by a better utilization of the sepctrum. CAB utilization is illustrated in Fig. 8 as a function of the arrival rate λ. Optimal DSA policy smoothly increases the CAB utilization as arrival rate increases. The proposed heuristic follows this trend with a step function. QL-based DSA, although a bit less efficient, has a similar behavior. It is worth mentioning the DSA gain in terms of spectral resource usage with respect to FSA.

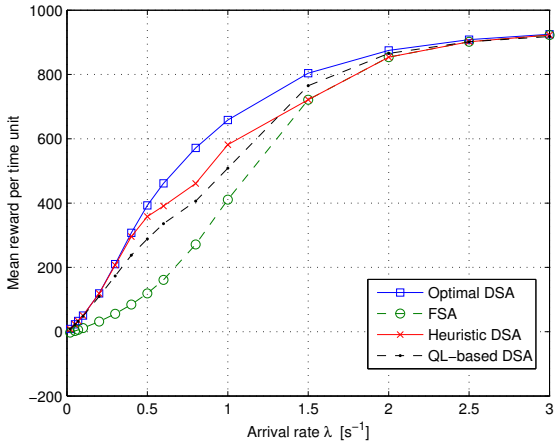

Figure 6. Operator reward obtained with optimal DSA, heuristic DSA, QL-based DSA, and FSA.

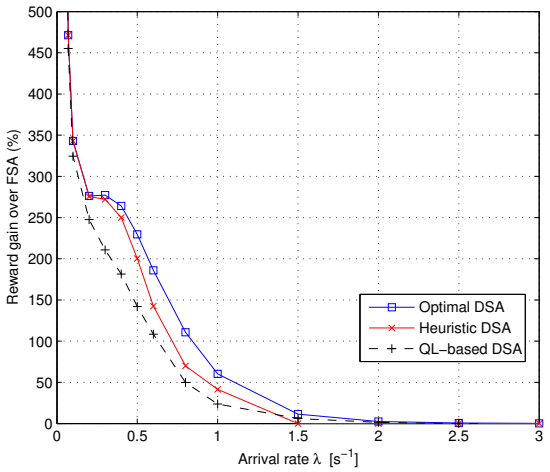

Figure 7. Reward gain with respect to FSA for optimal DSA, heuristic DSA, and QL-based DSA.

To explain the difference between the heuristic DSA and the QL-based method, it is worth mentioning that the heuristic method needs to know all the networks parameters such as, λ, X_{ON} and D_i, unlike the case of the QL-based algorithm. The fact that QL-based DSA is below the optimal DSA in terms of performance is explained by the fact that the learning phase is voluntary limited to a realistic duration.

Operator reward and better spectrum utilization with the three proposed approaches are however obtained at the price of a degradation of the average user throughput.

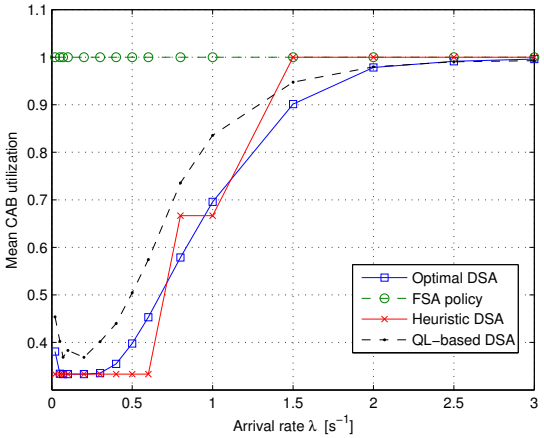

Figure 8. CAB utilization with optimal DSA, heuristic DSA, QL-based DSA, and FSA.

Fig. 9 illustrates the average user throughput as a function of the RANs load λ. Optimal DSA policy, the proposed heuristic and the QL-based DSA show again similar results.

The observed variations of the heuristic DSA between $\lambda = 0.5$ and 1.5 s^{-1} can be explained by the changes of resource allocation at threshold values 0.73 and 1.44 s^{-1} (see Fig. 4).

The achieved average user throughput with FSA is however much higher, especially at low loads. According to the traffic assumptions (see section 2.2), a single user is indeed allowed to take advantage of the whole bandwidth allocated to a RAN. At low loads, FSA allocates 3 MHz to each RAN, while DSA methods allocates only 1 MHz leading to lower user throughputs. Our assumptions represent thus a worst case scenario; a terminal or service limitation in maximum data rate would reduce the advantage of FSA at low loads.

7. CONCLUSION

In this paper, we have studied DSA in cellular networks context. We have used the SMDP framework to derive optimal DSA policies in terms of the operator reward. We have proposed two methods to defeat the generalization difficulty of the optimal policies over realistic systems: a simple heuristic DSA method and a QL-based DSA algorithm. The achieved reward using the heuristic DSA gives a very close reward to the optimal obtained by SMDP and thus significantly exceeds the reward obtained with FSA. The obtained reward using the QL-based has

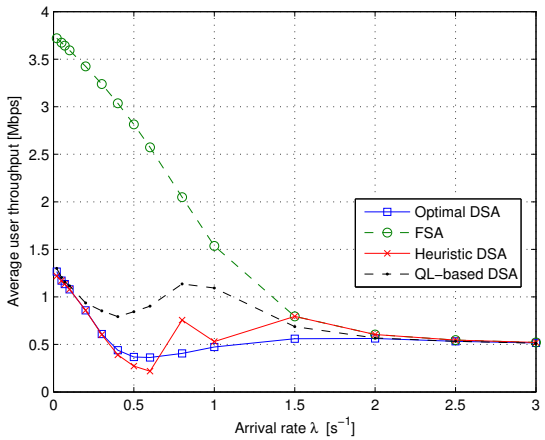

Figure 9. Average user throughput with optimal DSA, heuristic DSA, QL-based DSA, and FSA.

shown its gain over reward achieved using FSA. Although less gain is achieved using QL-based DSA, however the algorithm does not need to know the networks parameters. Operator revenue increases but better spectrum utilization is obtained at the price of a user throughput degradation.

ACKNOWLEDGEMENT

This work is part of the French program Systematic/URC funded by Paris Region and the national authorities.

REFERENCES

1. M. Buddhikot, "Understanding Dynamic Spectrum Allocation: Models, Taxonomy and Challenges," in *Proc. IEEE DySPAN'07*, pp. 649-663, 2007.
2. S. Buljore et al., "IEEE P1900.4 Standard: Reconfiguration of Multi-Radio Systems," in *Proc. IEEE SIBIRCON'08*, pp. 413-417, 2008.
3. S. Filin et al., "Dynamic Spectrum Assignment and Access Scenarios, System Architecture, Functional Architecture and Procedures for IEEE P1900.4 Management System," in *Proc. IEEE CrownCom'08*, pp. 1-7, 2008.
4. M. Buddhikot, P. Kolodzy, K. Ryan, J. Evans, and S. Miller, "DIMSUMNet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access," in *Proc. IEEE WoWMoM'05*, pp. 78-85, 2005.
5. 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.
6. S. Sankaranarayanan, P. Papadimitratos, A. Mishra, S. Hershey, "A Bandwidth Sharing Approach to Improve Licensed Spectrum Utilization," in *Proc. DySPAN'05*, pp. 279-288, 2005.
7. J.M. Chapin and W.H. Lehr, "Cognitive Radios for Dynamic Spectrum Access-The Path to Market Success for Dynamic Spectrum Access Technology," *IEEE Communications Magazine*, vol 45, no. 5, pp. 96-103, 2007.
8. J. Acharya, and R.D. Yates, "A Price Based Dynamic Spectrum Allocation Scheme," in *Proc. ACSSC'07*, pp. 797-801, 2007.
9. L. Vanbien, L. Yuewei, W. Xiaomeng, F. Zhiyong, and Z. Ping, "A Cell Based Dynamic Spectrum Management Scheme with Interference Mitigation for Cognitive Networks," in *Proc. VTC'08*, pp. 1594-1598, 2008.
10. S. Geirhofer, L. Tong, and B.M. Sadler, "Cognitive Medium Access: A Protocol for Enhancing Coexistence in WLAN Bands," in *Proc. IEEE GLOBECOM'07*, pp. 3558-3562, 2007.
11. Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework," *IEEE J. on Select. Areas in Commun.*, vol. 25, no. 3, pp. 589-600, 2007.
12. M. Coupechoux, J.M. Kelif, and Ph. Godlewski, "SMDP Approach for JRRM Analysis in Heterogeneous Networks," in *Proc. European Wireless'08*, pp. 1-7, 2008.
13. 3GPP, Overview of 3GPP Release 8 V0.0.8 (2009-09), Sept. 2009.
14. IEEE 802.16-2005 standard for local and metropolitain area networks.
15. N. Enderlé and X. Lagrange, "User Satisfaction Models and Scheduling Algorithms for Packet-Switched Services in UMTS," in *Proc. VTC'03*, vol. 3, pp. 1704-1709, 2003.
16. D. P. Bertsekas, "Dynamic Programming and Optimal Control," third edition, Athena Scientific, 2007.
17. P. Tadepalli, and D. Ok, "Model-based average reward reinforcement learning," Elsevier, Artificial Intelligence, vol 100, issue 1-2, pp. 177 - 224, 1998.
18. P.Y. Glorennec, "Reinforcement Learning: an Overview," European Symposium on Intelligent Techniques, pp. 17-35, 2000.
19. J. Abounadi and D. Bertsekas, "Learning algorithms for Markov decision processes with average cost," SIAM Journal on Control and Optimization, vol 40, Issue 3, pp. 681 698, 2001.
20. C.J. Watkins, "Learning from Delayed Rewards", Ph.D. Thesis, Kings College, Cambridge, England, 1989.
21. A. Gosavi, "Reinforcement learning for long-run average cost," Elsevier, European journal of operational research, Traffic and Transportation Systems Analysis, pp. 654-674, 2004.