An Application of Reinforcement Learning for Efficient Spectrum Usage in Next-Generation Mobile Cellular Networks

Francisco Bernardo, *Student Member, IEEE*, Ramon Agustí, *Member, IEEE*, Jordi Pérez-Romero, *Member, IEEE*, and Oriol Sallent

***Abstract*—This paper proposes reinforcement learning as a foundational stone of a framework for efficient spectrum usage in the context of next-generation mobile cellular networks. The objective of the framework is to efficiently use the spectrum in a cellular orthogonal frequency-division multiple access network while unnecessary spectrum is released for secondary spectrum usage within a private commons spectrum access model. Numerical results show that the proposed framework obtains the best performance compared with other approaches for spectrum assignment. Moreover, the framework is relatively simple to implement in terms of computational requirements and signaling overhead.**

***Index Terms*—Cellular mobile networks, frequency assignment, reinforcement learning (RL), spectral efficiency.**

I. Introduction

Next generation of mobile cellular networks devise a radio access network (RAN) based on orthogonal frequency-division multiple access (OFDMA) techniques. Such an interface divides a broad frequency band into many frequency subcarriers. In practice, the minimum radio resource in frequency is a small group of contiguous subcarriers, named hereafter as a *chunk*. This provides high robustness against typical variations of the frequency response of the mobile channel, enabling high-speed data communications [1]. However, an OFDMA RAN in a cellular scenario is highly affected by *intercell interference* (i.e., the interference that two or more neighbor cells using the same chunk cause each other), reducing the data rate that users can obtain in a given cell. Hence, such radio access interface requires strategies for selecting the cell-by-cell spectrum assignment in order to avoid intercell interference.

On the simplest extent, the cell-by-cell spectrum assignment strategy can be fixed by means of the frequency reuse factor (FRF) concept [2], where the available spectrum is divided into several equal subbands that are assigned to cells. For instance, in the case FRF = 1 the whole band is available in all cells, whereas in FRF = 3 the entire bandwidth is equally distributed among clusters of three cells. Also hybrid reuse schemes like partial reuse (PR) [3] or soft reuse (SR) [4], divide the entire frequency band of the system between a central and an edge subband, with FRF = 1 and FRF = 3 as frequency planning deployments, respectively. However, these fixed spectrum assignment strategies are proved to be clearly inefficient with variable spatial traffic demands, where different traffic loads per cell are given at different times of the day. Hence, higher flexibility in the spectrum management can be provided by dynamic spectrum assignment (DSA) strategies [5]. They are intended to automatically and dynamically decide which particular chunks are assigned to each cell in order to maintain the QoS of the ongoing users' sessions while smartly coping with the intercell interference. Therefore, DSA strategies would allow network operators to exploit the spectrum band allocated by the spectrum regulator more efficiently than when considering the FRF concept.

The DSA strategies could also be efficiently exploited in the framework of new envisaged regulatory rules. For instance, they can serve to achieve an efficient spectrum usage within the *private commons* initiative [6], which is a spectrum access model where *primary* (licensee) mobile network operators agree to open their spectrum for unlicensed *secondary* usage at the same time that they may charge a fee for each commercial secondary spectrum access [7]. In particular, DSA can provide the primary operators' networks with the proper cognitive mechanisms to automatically adapt the spectrum assignment to variable traffic demands, so that nonused chunks can be available for secondary usage.

In this context, reinforcement learning (RL) techniques show appealing cognitive capabilities since they try to learn the suitable set of actions to choose in order to maximize a numerical reward by following a cyclic interaction with an environment [8]. RL has been proposed for several applications in the field of mobile communications such as radio resource management [9], QoS provisioning and routing [10], [11], and joint management of multiradio and multioperator scenarios [12].

Regarding spectrum assignment tasks, RL was proposed in papers applied to dynamic channel assignment in former voice service oriented mobile networks [13], [14], but these approaches are not suitable to the future services based on data packets transmission. More recently, RL has been applied to spectrum sensing [15] or spectrum sharing [16] procedures in OFDMA-based networks from a secondary spectrum market but not from a primary operator perspective. On the other hand, in [17]– [19], we introduced a DSA framework with RL capabilities (in the following RL-DSA) for a multicell OFDMA packet RAN. The algorithm learns the most suitable spectrum assignment for a given set of cells in order to maximize a given reward, in accordance with certain cellular system's performance objective.

This paper extends our previous work by proposing first a new RL-DSA learning algorithm, providing *global* optimization of the reward signal, allowing RL-DSA to escape from local maxima. Second, a novel model for practical implementation of the reward signal is given, extending and exploiting what was briefly described in [17]– [19]. Third, an exhaustive performance comparison with, fixed, hybrid, and other DSA strategies within the private commons scenario is given, showing remarkable improvements over the rest of strategies. Finally, new results attending to the convergence behavior of RL-DSA and practical facts regarding the complexity and signaling overhead of the framework are given, illustrating that the implementation of our proposal is feasible and then could constitute a candidate solution for spectrum management in next-generation cellular systems.

In the following, Section II presents some useful definitions. Section III is devoted to present the RL-DSA algorithm, whereas Section IV describes how RL-DSA can be implemented in a typical next-generation cellular network. Next, Sections V and VI present the simulation model and obtained results, respectively. Finally Section VII concludes this paper.

II. Definitions and Performance Metrics

We consider a spectrum band of W Hz allocated to a primary operator that is shared between a set of K cells in the downlink OFDMA mobile cellular network. The band is divided into N chunks so that the

Manuscript received September 14, 2009; revised January 4, 2010; accepted January 8, 2010. Date of publication February 17, 2010; date of current version June 16, 2010. This work was supported by the Community's Seventh Framework Programme under Project E^3, by the Spanish Research Council under Formation of University Professors programme and COGNOS Grant TEC2007-60985, and by the European Regional Development Funds. This paper was recommended by Associate Editor B. Chaib-draa.

The authors are with the Signal Theory and Communications Department, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain (e-mail: fbernardo@tsc.upc.edu; ramon@tsc.upc.edu; jorperez@tsc.upc.edu; sallent@tsc.upc.edu).

Digital Object Identifier 10.1109/TSMCC.2010.2041230

bandwidth of a chunk is $B = W/N$ Hz. We define a *spectrum assignment* as a distribution of the N available chunks over the K cells, being possible the same chunk to be assigned to more than one cell, causing potential intercell interference. Different spectrum assignment strategies are possible, leading to different performances over the cellular network. In order to compare different schemes and for the remainder of the paper, we define the following performance metrics.

1) *User dissatisfaction probability* $P^{\mathrm{th}_{\mathrm{target}}}$: It is defined as the percentage of seconds in which user throughput is below a target throughput $\mathrm{th}_{\mathrm{target}}$ called *satisfaction throughput*.
2) *Spectrum usage:* It is defined as $W_k = N_k B$, where N_k is the number of chunks assigned to cell k.
3) *Spectral efficiency*: It denotes the throughput per unit of spectrum. It is defined as $\eta = (1/K)\sum_{k=1}^{K} \eta_k$, in bits/s/Hz, where $\eta_k = \mathrm{TH}_k / W_k$ is the spectral efficiency per cell and TH_k is the aggregate throughput of all users in cell k.

III. RL-DSA Algorithm

Reinforcement Learning-DSA functional architecture is composed of several RL agents, whose operation procedure is described in the following, together with the new learning algorithm and the RL-DSA architecture.

A. Single RL Agent

Let us consider that a single agent i interacts with an environment in a succession of time steps (denoted with t), where for each action a reward signal $r_i(t)$ is returned to the agent. The proposed RL-DSA algorithm is based on the modified REINFORCE methods presented in [20], which have been proven to converge to *global maximum* of reward signal in the long term because of the climbing of an appropriate gradient of the average reward, and to the inclusion of a perturbation term to allow RL to get out of local maxima. In this paper, we focus on the simplest REINFORCE agent, which is based on Bernoulli distributions and logistic functions. Hence, each RL agent's output $y_i(t) \in Y = \{0, 1\}$ is a Bernoulli random variable with action selection probability $p_i(t)$. That is, it is a two-action agent, where $\Pr(y_i(t) = 1) = p_i(t)$ and $\Pr(y_i(t) = 0) = 1 - p_i(t)$. Moreover, agent's input $x_i(t)$ and internal status $w_i(t)$ are related with $p_i(t)$ by means of the logistic function as

$$p_i(t) = \left(1 + e^{-x_i(t) w_i(t)}\right)^{-1}. \tag{1}$$

The learning capability of the agent is condensed step-by-step in the internal status $w_i(t)$, which is updated in accordance with the following learning rule:

$$w_i(t) = w_i(t-1) + \Delta w_i(t) \tag{2}$$

$$\Delta w_i(t) = \alpha(t)[r_i(t) - \bar{r}_i(t-1)][y_i(t-1) - p_i(t-1)]x_i(t-1) + \alpha(t)\xi\left(w_i(t-1)\right) + \sqrt{\alpha(t)}\zeta_i(t). \tag{3}$$

Note that, by varying the internal status $w_i(t)$ the learning algorithm is varying the selection probabilities in (1), so knowledge acquired from current reward is certainly enforced in the agent. The first term in (3) performs the gradient ascent of the reward signal as in the learning algorithm considered in [17]–[19]. Parameter $\alpha(t) > 0$ is called the learning rate (for details regarding its update see [17]). $\bar{r}_i(t)$ is the average reward obtained as $\bar{r}_i(t) = \beta r_i(t) + (1-\beta)\bar{r}_i(t-1)$, with $0 < \beta \leqslant 1$. The second term in (3) bounds the operation of the algorithm and introduces an exploratory behavior in the agent, with a small exploratory probability $0 < p_{\mathrm{explore}} \ll 1$ through the next function

$$\xi\left(w_i(t)\right) = \begin{cases} L - w_i(t), & w_i(t) \geqslant L \\ 0, & |w_i(t)| < L \\ -L - w_i(t), & w_i(t) \leqslant -L \end{cases} \tag{4}$$

where $L = \ln((1 - p_{\mathrm{explore}})/p_{\mathrm{explore}})$. This probability is necessary to allow the algorithm to explore new actions not taken in the past in order to seek for better reward responses. Finally, the third term introduces a perturbation parameter $\zeta_i(t)$, which is a random variable of zero mean and variance σ^2 (e.g., in this paper, $\zeta_i(t)$ takes the value either $+\sigma$ or $-\sigma$ with equal probability, being σ a positive constant). This term was proposed to give the algorithm the capability of escaping from local maxima and reaching global maximum of the average reward with a sufficient small value of σ and a sufficient number of iterations of the learning loop [20].

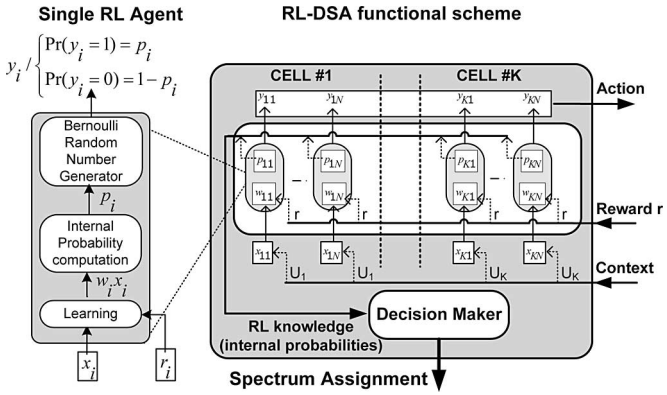

Fig. 1. RL-DSA algorithm functional scheme.

B. RL-DSA Functional Scheme and Procedure

Fig. 1 depicts the functional architecture of RL-DSA, which is composed of KN RL agents. The knth agent is devoted to learn whether the nth chunk is assigned to the kth cell. Obviously, in order to face a real-world problem with RL, it is necessary to appropriately select the physical meaning of the context, output, and reward sets. Particularly, we use RL-DSA to associate different cellular network traffic distributions (context inputs) to different spectrum assignments (output actions). Then, the context x_{kn} reflects the load status of the kth cell. That is, $x_{kn} = U_k \quad \forall n$, being U_k the average number of users in the kth cell. This context remains constant during an RL-DSA execution so that RL-DSA is able to associate solutions to both homogeneous and heterogeneous spatial distributions of the traffic load (i.e., users per cell). On the other hand, the action taken by RL-DSA is a binary vector $\Upsilon = (y_{11}, y_{12}, \ldots, y_{KN})$ that represents a *candidate* chunk-to-cell assignment in each RL step. To this end, it is considered that the nth chunk is assigned to the kth cell if the output y_{kn} is 1 (and not assigned in case y_{kn} is 0). Finally, the physical meaning of the reward signal r, common to all agents, is given in next section.

Then, for a succession of RL-steps $t = 1, 2, \ldots$ RL-DSA works as follows:

1. REPEAT
2. Receive reward $r(t)$ from the environment.
3. Update average reward $\bar{r}(t)$.
4. **FOR** all $k \in \{1, 2, \ldots, K\}$ and all $n \in \{1, 2, \ldots, N\}$
5. Update internal status $w_{kn}(t)$ following (2) and (3)
6. Compute internal probabilities $p_{kn}(t)$ according to (1)

7. Generate an action $y_{kn}(t)$ as a Bernoulli random variable with action selection probability $p_{kn}(t)$
8. **END FOR**
9. **UNTIL** $(t > MAX\_STEPS)$
10. **FOR** all $k \in \{1, 2, \ldots, K\}$ and all $n \in \{1, 2, \ldots, N\}$
11. **IF** $p_{kn} > 0.5$
12. Assign the nth chunk to the kth cell.
13. **ELSE**
14. Do not assign the nth chunk to the kth cell.
15. **END IF**
16. **END FOR**

Condition in step 9 finishes the learning loop when a maximum number of steps (MAX_STEPS) is reached. Steps from 10 to 16 are executed by *decision maker* module in Fig. 1 to decide the final spectrum assignment for the real network. Notice that RL-DSA bases on the knowledge stored in internal probabilities $p_{kn}(t)$ and not on the very last random action to decide the spectrum assignment to the network area. Moreover, note that a given chunk can be assigned by the algorithm to more than one cell.

Regarding the initial step, the first time that RL-DSA is triggered, full assignment is set, i.e., $y_{kn}(0) = 1$, and accordingly $p_{kn}(0) = 1 - p_{\text{explore}}$ and $w_{kn}(0) = L\ \forall n, k$. Moreover, $\bar{r}_{kn}(0) = 0$ $\forall n, k$. Notice that because of the exploratory probability, the spectrum assignment chosen by RL-DSA in the following steps can be different from full assignment (because the outputs are Bernoulli random variables). This situation triggers the learning of RL-DSA, causing that internal status and consequently action selection probabilities will evolve according to the learning rule until the end of the learning loop. Moreover, the perturbation term included in this paper in the learning rule makes that the update of the internal status is different for each chunk in one cell even if the reward and the outputs do not vary in two consecutive steps, allowing RL-DSA to escape from local maxima of the reward signal.

Finally, in subsequent triggers, RL-DSA begins from the assignment learned in the previous run, so that the knowledge acquired until that moment in internal status and probabilities is exploited.

C. Reward Signal Formulation

The target of the DSA strategy in this paper is twofold. First, it should assure a given QoS in terms of a minimum average user throughput in the primary cellular network. Second, it should improve spectrum usage in a private commons scenario, where opportunities for secondary spectrum usage in primary nonused spectrum are generated. Based on these targets, the reward signal per step $r(t)$, common for all RL agents in RL-DSA, is defined as follows:

$$r(t) = \sum_{k=1}^{K} r_k(t) + \sum_{j=1}^{s(t)-1} jR \tag{5}$$

$$r_k(t) = \begin{cases} 0, & \text{if } \widehat{\text{th}}_k(t) < \text{th}_{\text{target}} \\ \lambda\hat{\eta}_k(t) + \mu(N - N_k(t)), & \text{otherwise.} \end{cases} \tag{6}$$

$r_k(t)$ constitutes the reward signal per cell, R is an upper bound for all $r_k(t)$, and $s(t)$ stands for the number of cells that fulfill a QoS constraint $r_k(t) > 0$, as explained in the following. $\widehat{\text{th}}_k(t)$ is the estimated average user throughput for cell k in bits/s, $\hat{\eta}_k(t)$ is the estimated average spectral efficiency in bits/s/Hz, and $N - N_k(t)$ is the number of nonused chunks in that cell. $\lambda > 0$ and $\mu > 0$ are appropriate scaling constants. Then, the reward for a given cell is zero if the average user throughput is below the user satisfaction throughput target $\text{th}_{\text{target}}$. On the other hand, if the QoS is fulfilled in the kth cell, $r_k(t)$ is a positive real value, which, in practice, is upper bounded due to the existence of a maximum achievable spectral efficiency $\eta_{\max}$ and a finite number of available chunks. Then, let $R = \lambda\eta_{\max} + \mu N$ be this upper bound that fulfills $0 \leqslant r_k(t) < R \quad \forall k$. The inclusion of the second term in (5) assures that $r(t)$ increases monotonically with $s(t)$, as proved in the Appendix. Thus, RL-DSA will tend to select a spectrum assignment that maximizes the reward while at the same time maximizing the number of cells fulfilling the QoS constraint.

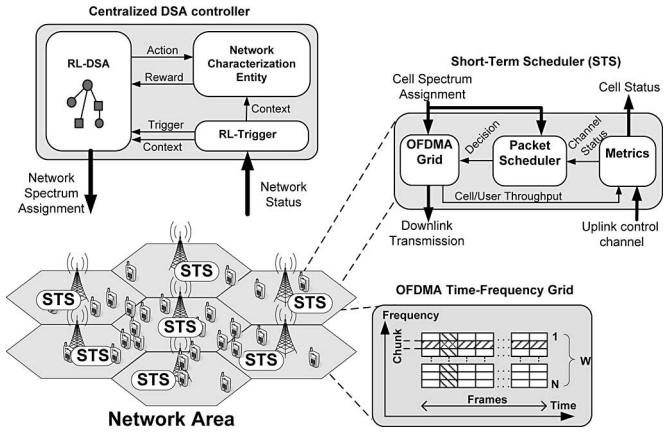

Fig. 2. DSA framework for RL-DSA execution in a next-generation OFDMA mobile cellular network.

IV. DSA Framework With Learning Capabilities

Fig. 2 depicts a hierarchical architecture for an operator who deploys a multicell system with an OFDMA-based radio interface. Typically, OFDMA builds a time-frequency grid (see Fig. 2 lower right corner). The whole available band is divided into *chunks*, whereas time it is divided into *frames* of a very short duration (e.g., ms), which allows combating the rapid fluctuations of the radio channel, also known as fast fading [1]. From a resource allocation perspective, we have to decide which chunks are allocated to each cell, and which of those chunks in a cell are assigned to a user in a given frame. Trying to perform this chunk-to-cell-to-user assignment simultaneously in the short-term can be very costly in terms of signaling exchange and computational requirements. Our hierarchical approach can reduce these costs by decoupling the resource assignment problem into two temporal scales:

1) In the short-term (i.e., frame-by-frame), the so-called *short-term scheduler* (STS) in each cell decides how to schedule users' transmissions into available chunks, depending on the channel status reported by users. There are several possibilities for the scheduling strategy. We consider in this paper two of them: round robin (RR) and proportional fair (PF). RR is a nonchannel aware strategy that cyclically allocates chunks to users regardless whether the radio channel status is appropriate or not for the selected user in the scheduled frame. On the other hand, PF is a channel-aware strategy that is able to exploit the channel in time and frequency domain [21]. It takes into account the instantaneous signal-to-interference-plus-noise ratio (SINR) $\gamma_{m,n}$ perceived by the mth user in the cell in the nth chunk to schedule users' transmissions. We consider constant transmitted chunk power and the typical radio channel propagation features (i.e., the distance-dependant pathloss, slow fading, and frequency selective fading) for the computation of $\gamma_{m,n}$ [1]. Frequency selective fading makes that the channel gain varies from one chunk to another for all m and n. In addition, STS takes care of the so-called adaptive coding and modulation (ACM) procedure, which decides the modulation and coding rate that mth user should employ in transmission

TABLE I
MODULATION AND CODING SCHEMES [22]

SINR threshold (dB)	Modulation b (bits/s/Hz)	Coding Rate r	Spectral efficiency q=b*r (bits/s/Hz)
$<$ 0.9	0	0	0
$\geq$ 0.9	2 (QPSK)	1/3	0.66
$\geq$ 2.1	2 (QPSK)	1/2	1
$\geq$ 3.8	2 (QPSK)	2/3	1.33
$\geq$ 7.7	4 (16QAM)	1/2	2
$\geq$ 9.8	4 (16QAM)	2/3	2.66
$\geq$ 12.6	4 (16QAM)	5/6	3.33
$\geq$ 15.0	6 (64QAM)	2/3	4
$\geq$ 18.2	6 (64QAM)	5/6	5

over the chunk n for a given $\gamma_{m,n}$. In this paper, ACM bases on Table I [22]. Finally, users can be granted with more than one chunk in a frame, but a chunk can only be assigned to a single user.

2) In the longterm (e.g., thousands of frames), the controller of a cluster of cells (named hereafter *DSA controller*) decides which chunks should be used by each cell under control (i.e., it performs cell-by-cell spectrum assignment). Long-term execution can be considered because a chunk-to-cell assignment can be valid for a given spatial traffic distribution, which usually changes in a slow way. The functional architecture of the DSA controller is depicted in Fig. 2. RL-DSA implements the decision and learning functionalities as explained before in Section III. RL-DSA execution is supported by two functional entities that constitute its environment: the *RL-trigger entity* and the *network characterization entity* (NCE), as explained next.

A. RL-Trigger Entity

The *RL-trigger* entity *observes* and *analyzes* the network variable status. Then, it detects the instants when the current spectrum assignment is no longer valid to achieve a given users' QoS performance. Let $P_k^{\mathrm{th}_{\mathrm{target}}}$ be the average dissatisfaction probability per cell k over a period of l seconds. Based on this period, each cell reports to the DSA controller $P_k^{\mathrm{th}_{\mathrm{target}}}$ and its average number of users U_k. Then, RL-trigger computes the average network dissatisfaction probability as

$$P^{\mathrm{th}_{\mathrm{target}}} = \frac{\left(\sum_{k=1}^{K} U_k P_k^{\mathrm{th}_{\mathrm{target}}}\right)}{\left(\sum_{k=1}^{K} U_k\right)} \tag{7}$$

where K is the number of cells of the network area. Then, the RL-trigger entity triggers the execution of the RL-DSA algorithm if $P^{\mathrm{th}_{\mathrm{target}}}$ is either above a given threshold δ^{up} or below a threshold δ^{down}, since in these cases current assigned resources would be either insufficient or overprovisioned, respectively, in accordance with the desired QoS. In addition, RL-trigger entity provides the *execution context* at the beginning of a new execution, which in this paper is the average number of users per cell U_k. This context orients the RL-DSA learning and hence allows RL-DSA to adapt to potential uneven distributions of the traffic load.

B. Network Characterization Entity

Once RL-DSA is executed, its intermediate actions are applied *offline* to an NCE. Each action of RL-DSA represents a *candidate* spectrum assignment for the network, and in turn, the NCE returns the reward signal given in (5). Hence NCE constitutes a *model* of the network's response in terms of reward for a given spectrum assignment and network context. We propose in this paper a practical model to implement NCE.[1]

In order to build the reward signal, the NCE needs to estimate the average spectral efficiency $\hat{\eta}_k$, the average user throughput for cell $\widehat{\mathrm{th}}_k$, and the number of nonused chunks in that cell $N - N_k$. This last term can be easily obtained from the input spectrum assignment provided by the RL-DSA. Alternatively, $\hat{\eta}_k$ and $\widehat{\mathrm{th}}_k$ can be obtained as follows.

Assuming a cellular system with uniformly distributed users per cell, the average spectral efficiency in bits/s/Hz for a given cell k is given by

$$\hat{\eta}_k = N_k^{-1} \sum\nolimits_{n \in C_k} \left[G(\Phi_{k,n}, U_k) \times \int\int_A A^{-1} q\left(SINR(\Phi_{k,n}, \rho, \theta)\right) \rho d\rho d\theta \right] \tag{8}$$

where N_k and C_k are the number and the set of chunks assigned to a given cell k, respectively. $\Phi_{k,n}$ is the set of cells that cause interference for a specific cell k in a specific chunk n, and U_k is the average number of users in the cell. $G(\Phi_{k,n}, U_k)$ is a gain factor that captures the characteristics of the short-term scheduling strategy (RR or PF) used in the cell. $q\left(\mathrm{SINR}(\Phi_{k,n}, \rho, \theta)\right)$ is the spectral efficiency in bits/s/Hz for a given value of the SINR ($\mathrm{SINR}(\Phi_{k,n}, \rho, \theta)$) at a given point (ρ, θ) of the cell in polar coordinates. For instance, function q can be the mapping table given in Table I. Hence, (8) is the average spectral efficiency for all points in the area A covered by the cell and for all assigned chunks to that cell.

Mobile cellular networks are interference limited systems, that is, noise at the receiver can be usually neglected when compared with the received interference. Then, we approximate the SINR as a signal to interference ratio as follows:

$$\mathrm{SINR}(\Phi_{k,n}, \rho, \theta) = \left(\sum_{j \in \Phi_{k,n}} (1 + (d_j/\rho)^2 - 2(d_j/\rho)\cos(\theta - \phi_j))^{-0.5\chi} \right)^{-1} \tag{9}$$

where it has been considered that any interfering cell j is located, in polar coordinates, at a point (d_j, ϕ_j) with respect to the reference cell k. Also, the same transmission power and antenna gains are assumed for all cells, and χ denotes the pathloss exponent [1]. On the other hand, $G(\Phi_{k,n}, U_k)$ will depend on the considered packet scheduling strategy. In particular, it is well known that the achieved spectral efficiency for an RR short-term scheduling strategy does not depend on the number of users in the cell, because RR leads to equal users' transmission probability [23]. Then, for an RR strategy a proper setting would be $G(\Phi_{k,n}, U_k) = 1$. However, a channel-aware scheduler with unequal users' transmission probabilities, such as PF, leads to a dependence of the achieved spectral efficiency with the number of users in the cell under certain SINR patterns. For this type of channel-aware schedulers, $G(\Phi_{k,n}, U_k)$ would correspond to a gain factor over the RR spectral efficiency. For PF, gain factor concept was developed by recent studies [23]. This gain factor depends on the number of users and the SINR distribution over the cell, as shown in Fig. 3. This figure plots a set of

[1] It is worth noting that the offline learning considered here brings two major advantages over on-line learning (i.e., actions applied directly to the live real network). First, the physical time taken by the real network to return a proper averaged reward can be unacceptably slow in terms of elapsed time compared with the quick response that NCE could provide. Second, actions taken by RL-DSA during the learning process are random, bringing in some cases prohibitive costs in performance for the live network if on-line learning is implemented. On the contrary, applying those actions to an offline environment only supposes a cost in *simulation* time, allowing RL-DSA to take actions from the whole action space without any restriction.

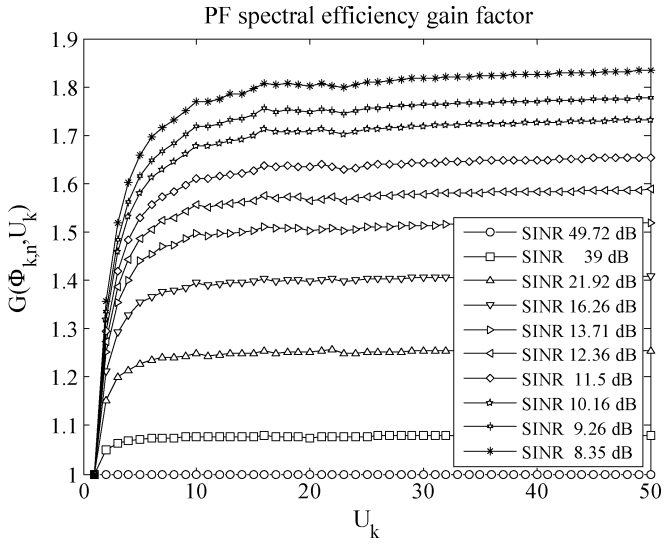

Fig. 3. Sample of considered curves for spectral efficiency gain factor of a PF scheduler over an RR scheduler.

PF gain factor curves obtained for exhaustive simulations that focus on the central cell in a two-ring macrocell scenario with different intercell interference patterns $\Phi_{k,n}$, leading to different average SINRs (other simulation values are detailed in Table II). Finally, NCE estimates the average user throughput per cell as $\widehat{\text{th}}_k = W_k \hat{\eta}_k / U_k$, where W_k is the bandwidth assigned to cell k.

V. Simulation Model

Results presented in this paper focus on the assessment of the proposed framework in a dynamic downlink OFDMA-based multicell scenario, where both temporal and spatial variations of the load per cell are considered. Configuration parameters are summarized in Table II. The scenario is composed of $K = 19$ cells and a maximum of $N = 12$ available chunks, which is one of the possible spectrum deployments of 3GPP LTE [24]. Users are distributed homogeneously within a cell, and they move at the speed of 3 km/h following a random walk model [24]. Users always remain within their cell (i.e., handovers are not considered). Users are assumed to have always data ready to be sent (i.e., full-buffer traffic model), so that each user tries to obtain as much capacity as possible. The target throughput $\text{th}_{\text{target}}$ is 256 kbits/s. The STS implements a PF strategy, although similar results have been obtained with an RR short-term scheduling strategy.

The performance of the system is evaluated during 1 h to capture changes in the spatial distribution of the load (users). In this respect, three types of cells can be distinguished in the scenario, as shown in Fig. 4. At the beginning, all cells are equally loaded with 15 users. After 25 min, type 1 cell increases the number of users in two users per minute. Type 2 cells increase the number of users in one user per minute, whereas type 3 cells decrease the number of users in one user per minute. These variations take place only during a 10 min period between 25 to 35 min. After 35 min, users are heterogeneously distributed among cells. Note that this pattern tries to reflect a temporal evolution of the load in the scenario that progressively would tend to concentrate the traffic load within a single cell (i.e., cell 1 in Fig. 4). Finally, RL-DSA is executed each time that the system dissatisfaction probability falls outside the interval $[\delta^{\text{down}}, \delta^{\text{up}}] = [0.001, 0.1]$. Default RL configuration parameters are included in Table II based on experimental results.

TABLE II
Simulation Parameters

Number of cells [K]	19
Cell Radius [R]	500 meters
Antenna Patterns	Omnidirectional
Frame duration	2 ms
Averaging window [l]	15000 frames
Carrier Frequency	2 GHz
Number of chunks [N]	12
Chunk bandwidth [B]	375 kHz
Total Bandwidth [P]	33 dBm
Path Loss at d Km in dB	128.1+37.6log10(d) [24]
Path Loss Exponent [χ]	3.76
Slow Fading standard deviation	8 dB [24]
Slow Fading decorrelation distance	5 m [24]
Fast Fading Model	ITU Ped. A [24]
UE thermal noise	-174 dBm/Hz
UE noise factor	9 dB
UE speed	3 km/h
Trigger Threshold [$\delta_{down}, \delta_{up}$]	[0.001, 0.1]
User's satisfaction throughput [th_{target}]	256 kbits/s
Maximum spectral efficiency [$\eta_{\max}$]	5 bits/s/Hz
Short Term Scheduling method	Proportional Fair [21]
PF Averaging window	50 frames
RL parameters [$\alpha, \beta, \sigma, \Delta, p_{\text{explore}}$]	[100, 0.01, 0.05, 10-6, 0.001]
Reward Parameters [R, λ, μ]	[6.2, 1, 0.1]
MAX_STEPS	1000000
Partial Reuse (PR) specific parameters	
Number of chunks of central cell	6
Number of chunks of edge cell	2
Centre-Edge pathloss threshold	-100 dBm
Soft Reuse (SR) specific parameters	
Number of chunks of central cell	8
Number of chunks of edge cell	4
Centre-Edge pathloss threshold	-100 dBm
Heur-DSA specific parameters	
Initial Margin Factor	1.5
Margin factor step	0.25

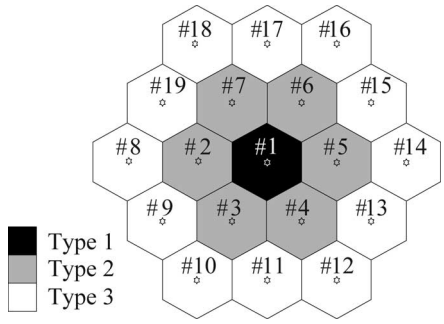

Fig. 4. Scenario layout for simulation.

VI. Results

A. Performance Evaluation

Results for the proposed RL-DSA algorithm are compared with fixed, hybrid, and dynamic strategies. As fixed and hybrid strategies, FRFs FRF1 (universal reuse), FRF3, PR [3], and SR [4] strategies are considered. As a dynamic strategy, the heuristic strategy DSA2 from [6] named here Heur-DSA is retained. Configuration parameters for these strategies can also be found in Table II. Heur-DSA is dynamically triggered following the same criterion as RL-DSA.

Fig. 5 depicts the average dissatisfaction probability evolution for all cells and each type of cell. Results for cells #1, #3, and #9 in Fig. 4 are presented, as representative ones for type of cells 1, 2, and 3, respectively (analogous results were obtained for other cells of the same type). RL-DSA improves the average dissatisfaction probability with respect

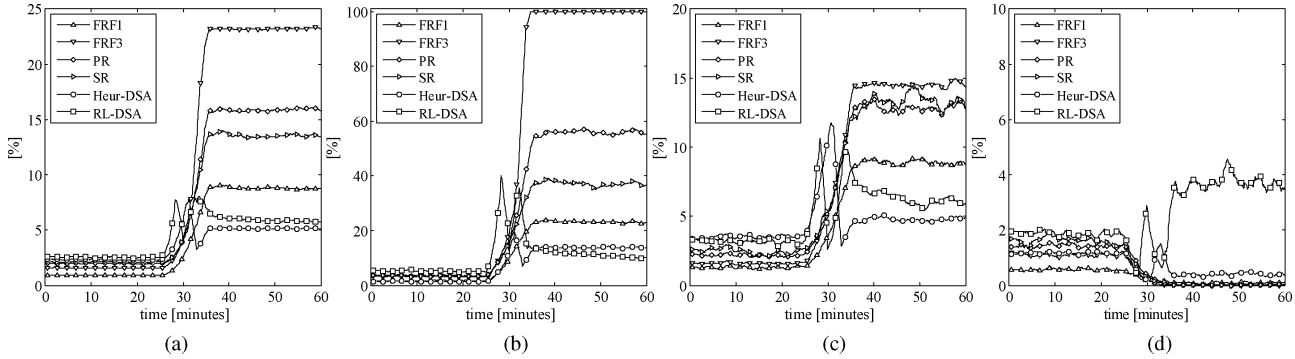

Fig. 5. Average dissatisfaction probability comparison between studied spectrum assignment strategies. (a) All cells, (b) cell type 1, (c) cell type 2, (d) cell type 3.

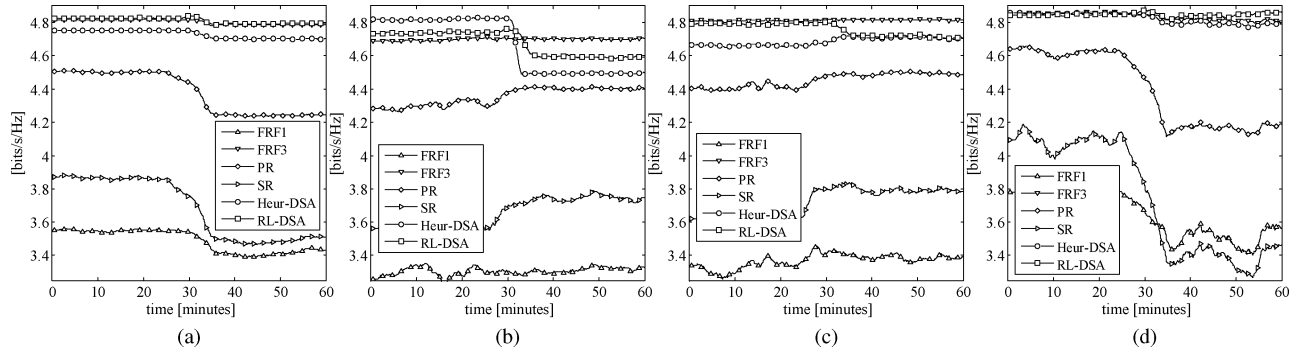

Fig. 6. Average spectral efficiency comparison between studied spectrum assignment strategies. (a) All cells, (b) cell type 1, (c) cell type 2, (d) cell type 3.

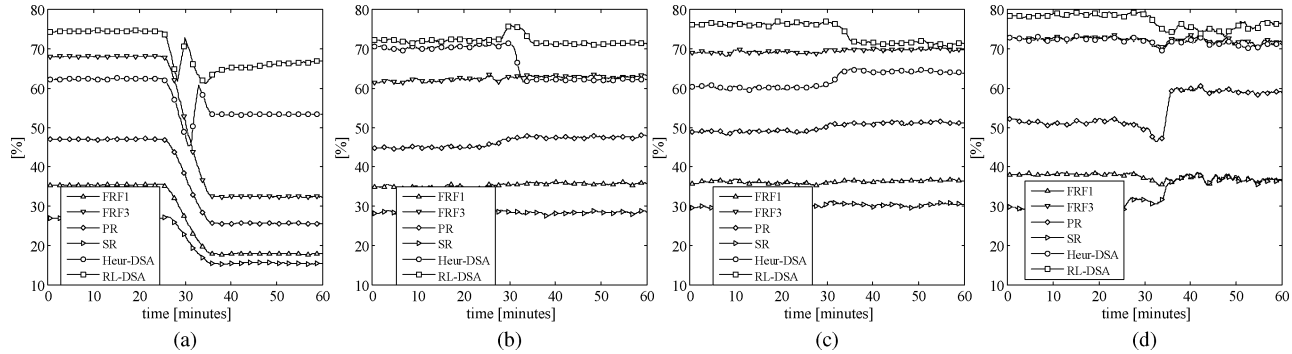

Fig. 7. Average fairness comparison between studied spectrum assignment strategies. (a) All cells, (b) cell type 1, (c) cell type 2, (d) cell type 3.

to the fixed spectrum assignment strategies and shows similar behavior with respect to Heur-DSA. Dynamic strategies provide dramatic improvements with respect to fixed strategies especially when the distribution of the load is heterogeneous (i.e., for time above 35 min). For instance, the dissatisfaction probability is more than four times lower than that for FRF3. Also, RL-DSA maintains a dissatisfaction probability below 10% target for all type of cells. Similarly, Fig. 6 depicts the spectral efficiency evolution. It can be observed that RL-DSA attains the best spectral efficiency for all cells. Thus, RL-DSA achieves the best tradeoff between users' satisfaction and spectral efficiency. Finally, it is shown in Fig. 7 that RL-DSA approach achieves the best results in fairness fulfillment. Fairness is defined as the fifth percentile of the average user throughput per cell normalized to mean throughput. It represents the balance between the throughput attained by the users in the center and in the edge of the cell.

Fig. 8 depicts the chunk usage per cell. Specifically, Fig. 8(a) shows the average number of nonused chunks per cell in the scenario, demonstrating that RL-DSA is the strategy that leaves more free chunks. However, it is also interesting to see how these nonused chunks are distributed. Fig. 8(b) shows the average number of nonused chunks in clusters of adjacent cells in the scenario. That is, we count for chunks that are not used in a cell and all its adjacent cells. Such nonused chunks would be more appropriate for secondary usage since secondary transmissions would not cause interference in a wider region. Observe that RL-DSA is the unique strategy that generates these spectrum usage opportunities during the complete simulation.

Therefore, RL-DSA assigns the right amount of spectrum per cell so that users obtain the satisfaction throughput, but not more. In this way, there is spectrum free that can be used by a secondary spectrum market.

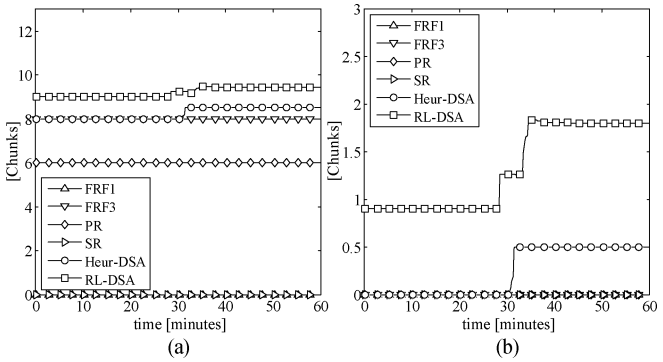

Fig. 8. (a) Average number of nonused chunks per cell. (b) Average number of nonused chunks in clusters of adjacent cells.

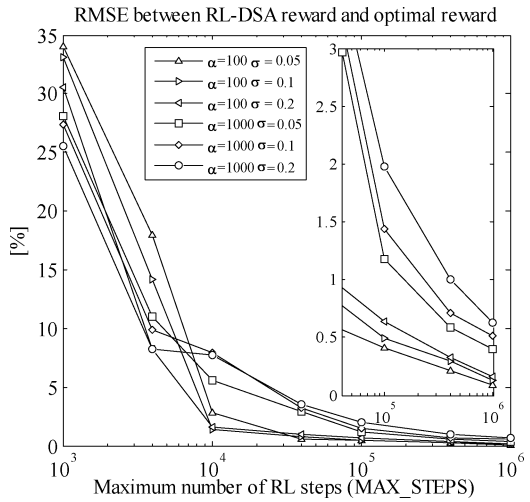

Fig. 9. RMSE between the final achieved RL-DSA reward and the optimal reward.

B. Convergence Behavior

The convergence behavior of RL-DSA for different values of its main parameters such as the learning rate (α), the maximum number of steps (MAX_STEPS), and the random perturbation term (σ) are studied hereafter. These results show a qualitative behavior that may be useful to setup RL-DSA.

Fig. 9 shows RMSE between the reward achieved by RL-DSA and the optimal reward in a given scenario. In order to make feasible the computation of the optimal reward, we have set a very particular scenario with 19 cells and 19 chunks, and 5 users per cell, where any spectrum assignment that gives one different chunk per cell (i.e., no intercell interference) was considered to be optimum, i.e., attained the best reward defined in (5). Note that an excellent RMSE of 1% can be reached in 10^4 steps for some of the configured parameters, which denotes a good convergence behavior of RL-DSA since the solution space for the scenario involves $2^{19\times 19}$ (4.69×10^{108}) different assignments. The study reveals that a high α reduces the number of steps needed to converge to the optimal solution. On the other hand, high values of σ perform better for a low number of steps but lower values obtain lower RMSE for a high number of steps. Finally, notice that for a number of steps above 10^5, the RMSE falls below 2% for all tested values of the parameters, revealing a robust behavior of RL-DSA with respect to the values selected for its parameters.

C. Implementation Issues

The required signaling exchange between the cells and the centralized DSA controller in the proposed framework in Section IV is low, since it is only produced on periods of l seconds. Moreover, only few bytes are needed to encode the dissatisfaction probability $P_k^{\mathrm{th}_{\mathrm{target}}}$ and the average number of users U_k per cell k, which act as inputs of the RL-trigger entity. That is, if $P_k^{\mathrm{th}_{\mathrm{target}}}$ and U_k are encoded with 8 and 1 bits is devoted to determine if a specific chunk is assigned to a cell (i.e., KN bits encode the resultant spectrum assignment), then $K(N+8+8)$ bits are needed to bear all the signaling for the execution of RL-DSA. For the numbers considered in Table II, only 532 bits are needed, which is clearly bearable for current communication trunks of a network operator.

Moreover, RL-DSA requires a small constant number of operations per step (i.e. few additions and products including simple forms of random number computation). Finally, memory requirements are low since only few records to store the weights, probabilities, rewards, and outputs are needed per RL-agent. These properties make the implementation of the RL-DSA scheme quite feasible.

VII. Conclusion

This paper has presented a framework for DSA in the context of next-generation downlink OFDMA-based networks. Decisions regarding spectrum assignment reside on an reinforcement learning-based algorithm (RL-DSA), which maximizes a reward signal defined targeting an efficient spectrum usage with QoS assurance. On the one hand, RL-DSA has shown the best tradeoff between spectral efficiency, QoS fulfillment and fairness among the different spectrum assignment strategies. On the other hand, RL-DSA was able to generate spectrum usage opportunities for secondary spectrum markets in a private commons spectrum access model. In addition, it is quite easy to change the optimization objective of the framework, by changing the reward signal formulation and the NCE functional block devoted to build such a reward signal. Finally, studies about convergence behavior show an excellent robustness of RL-DSA with respect to the values selected for its parameters.

The proposed framework could be extended for a distributed architecture in future work. Then, this architecture could enable the deployment of base stations that *autonomously* learn the best spectrum configuration to achieve eventually near-optimal spectral efficiency and QoS performance.

Appendix

In the following, it is proved that reward $r(t)$ increases monotonically with the number of cells $s(t)$ fulfilling the QoS constraint. The reward signal $r(t)$ from expression (5) can be bounded as

$$\sum_{j=1}^{s(t)-1} jR \leqslant r(t) < Rs(t) + \sum_{j=1}^{s(t)-1} jR. \tag{10}$$

By substituting the well-known result for the arithmetic sum and operating, we can get

$$0.5R[s^2(t) - s(t)] \leqslant r(t) < 0.5R[s^2(t) + s(t)]. \tag{11}$$

On the other hand, let us assume an increase in the number of cells fulfilling the QoS constraint $s(t)$ to $s'(t) = s(t) + 1$. Then, the corresponding reward $r'(t)$ obtained will be lower bounded as

$$0.5R[s'^2(t) - s'(t)] = 0.5R[s^2(t) + s(t)] \leqslant r'(t). \tag{12}$$

and $r'(t) > r(t)$ follows. This proves that $r(t)$ is a monotonically increasing function with the number of cells $s(t)$ that fulfill the QoS constraint (i.e., $r_k > 0$).

ACKNOWLEDGMENT

The authors would like to thank the contributions of colleagues from E^3 consortium. This paper reflects only the authors' views, and the Community is not liable for any use that may be made of the information contained therein.

REFERENCES

[1] T. S. Rappaport, *Wireless Communications Principles and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[2] Z. Wang and R. Stirling-Gallacher, "Frequency reuse scheme for cellular OFDM systems," *Electron. Lett.*, vol. 38, no. 8, pp. 387–388, Apr. 2002.

[3] M. Sternad, T. Ottosson, A. Ahlen, and A. Svensson, "Attaining both coverage and high spectral efficiency with adaptive OFDM downlinks," in *Proc. IEEE 58th Veh. Technol. Conf. VTC2003-Fall*, Oct., pp. 2486–2490.

[4] Huawei, "Soft frequency reuse scheme for UTRAN LTE," 3GPP TSG RAN WG1, Tech. Rep. TR1-050507, Orlando, FL, 2005.

[5] D. López-Pérez, A. Jüttner, and J. Zhang, "Dynamic frequency planning versus frequency re-use schemes in OFDMA networks," in *Proc. IEEE Veh. Technol. Conf.-Spring 2009*, Apr., pp. 1–5.

[6] F. Bernardo R. Agustí, J. Pérez-Romero, and O. Sallent, "Dynamic spectrum assignment in multicell OFDMA networks enabling a secondary spectrum usage," *Wireless Commun. Mobile Comput. (WCMC)*, vol. 9, pp. 1502–1519, 2009.

[7] M. M. Buddhikot, "Understanding dynamic spectrum access: Models, taxonomy and challenges," in *Proc. IEEE New Front. Dyn. Spectr. Access Netw. (DySPAN)*, 2007, pp. 649–663.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, Mar. 1998.

[9] Y.-S. Chen, C.-J. Chang, and F.-C. Ren, "Q-learning-based multirate transmission control scheme for RRM in multimedia WCDMA systems," *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 38–48, Jan. 2004.

[10] T.-K. Hui and C.-K. Tham, "Adaptive provisioning of differentiated services networks based on reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 33, no. 4, pp. 492–501, Nov. 2003.

[11] A. Vasilakos, M. Saltouros, A. Atlassis, and W. Pedrycz, "Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 33, no. 3, pp. 297–312, Aug. 2003.

[12] L. Giupponi and R. Agustí, J. Pérez-Romero, O. Sallent, "Fuzzy neural control for economic-driven radio resource management in beyond 3G networks," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 39, no. 2, pp. 170–189, Mar. 2009.

[13] J. Nie and S. Haykin, "A Q-learning-based dynamic channel assignment technique for mobile communication systems," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1676–1687, Sep. 1999.

[14] S. Singh and D. Bertsekas, "Reinforcement learning for dynamic channel allocation in cellular telephone systems," in *Advances in Neural Information Processing Systems*, vol. 9, Cambridge, MA: MIT Press, 1997, pp. 974–980.

[15] U. Berthold, F. Fu, M. van der Schaar, and F. Jondral, "Detection of spectral resources in cognitive radios using reinforcement learning," in *Proc. 3rd IEEE Symp. New Front. Dyn. Spectr. Access Netw. (DySPAN 2008)*, Oct., pp. 1–5.

[16] R. Farha, N. Abji, O. Sheikh, and A. Leon-Garcia, "Market-based resource management for cognitive radios using machine learning," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM 2007)*, Nov., pp. 4630–4635.

[17] F. Bernardo, R. Agustí, J. Perez-Romero, and O. Sallent, "A novel framework for dynamic spectrum assignment in multicell OFDMA networks based on reinforcement learning," in *IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2009, pp. 1–6.

[18] F. Bernardo and R. Agustí, J. Pérez-Romero, O. Sallent, "A self-organized spectrum assignment strategy in next generation OFDMA networks providing secondary spectrum access," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.

[19] F. Bernardo, R. Agustí, J. Perez-Romero, and O. Sallent, "Temporal and spatial spectrum assignment in next generation OFDMA networks through reinforcement learning," in *Proc. IEEE 69th Veh. Technol. Conf. VTC2009-Spring*, pp. 1–5.

[20] V. V. Phansalkar and T. M. A., "Local and global optimization algorithms for generalized learning automata," *Neural Comput.*, vol. 7, no. 5, pp. 950–973, Sep. 1995.

[21] C. Wengerter, J. Ohlhorst, and A. V. Elbwart, "Fairness and throughput analysis for generalized proportional fair frequency scheduling in OFDMA," in *Proc. IEEE 61st Veh. Technol. Conf. 2005-Spring*, vol. 3, pp. 1903–1907.

[22] R. Schoenen, R. Halfmann, and B. Walke, "MAC performance of a 3GPP-LTE multihop cellular network," in *Proc. IEEE Int. Conf. Commun. (ICC 2008)*, May, pp. 4819–4824.

[23] J.-G. Choi and S. Bahk, "Cell-throughput analysis of the proportional fair scheduler in the single-cell environment," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 766–778, Mar. 2007.

[24] 3GPP, "Physical layer aspects for evolved universal terrestrial radio access (UTRA)," 3GPP, Tech. Rep. TR 25.814 v7.1.0, Oct. 2006, release 7.