A Novel Framework for Dynamic Spectrum Management in MultiCell OFDMA Networks based on Reinforcement Learning

Francisco Bernardo, Ramón Agustí, Jordi Pérez-Romero and Oriol Sallent
Signal Theory and Communications Department
Universitat Politècnica de Catalunya (UPC)
08034 Barcelona, Spain
Email: [fbernardo, ramon, jorperez, sallent]@tsc.upc.edu

***Abstract*—In this work the feasibility of Reinforcement Learning (RL) for Dynamic Spectrum Management (DSM) in the context of next generation multicell Orthogonal Frequency Division Multiple Access (OFDMA) networks is studied. An RL-based algorithm is proposed and it is shown that the proposed scheme is able to dynamically find spectrum assignments per cell depending on the spatial distribution of the users over the scenario. In addition the proposed scheme is compared with other fixed and dynamic spectrum strategies showing the best tradeoff between spectral efficiency and Quality-of-Service (QoS).**

***Index Terms*—Dynamic Spectrum Assignment, Multicell OFDMA, Reinforcement Learning.**

I. Introduction

Future wireless networks will demand dynamic spectrum management policies that cope with the detected spectrum scarcity and its underutilization in current networks [1]. Thus, the limited available spectrum claims for a new paradigm of spectrum management that accounts for the different temporal and spatial spectrum demands, that is, a new concept in which the spectrum is used dynamically and opportunistically rather than in a fixed form [2].

In this context, the objective of Dynamic Spectrum Management (DSM) is to provide the procedures and algorithms to find a spectrum assignment over the radio interface that (a) adapts systems' spectrum to users' QoS requirements, taking into account the long-term spatial and temporal spectrum demands, and (b) improves the spectral efficiency. To this end, a flexible radio interface that eases the spectrum assignment becomes crucial. In this way, OFDMA (Orthogonal Frequency Division Multiple Access) is a multiple access technique that is in the main stream of current proposed systems (3G LTE, WiMax) while is suitable for cognitive radio usage [3].

Furthermore, DSM can be addressed from a Cognitive Radio Networks (CRN) perspective [4]. CRNs are composed of several cognitive elements whose aim is to improve the overall CRN performance. However, to the best of the authors' knowledge, so far cognitive approaches are being mainly developed to opportunistically take advantage of the unused spectrum, being DSM the natural consequence for such proposals at the link layer level [5].

In a general case, the CRN nodes should implement algorithms that adapt to the environment in which they are immersed. In this sense, Reinforcement Learning (RL) techniques show cognitive capabilities since they try to solve a given problem from the continuous interaction with an environment that returns a reward for each one of the executed RL actions [6]. RL has been studied in papers applied to dynamic channel assignment in multicell FDMA networks with different approaches like Q-learning [7], actor-critic TD(0) [8] and SARSA [9]. In those references comparison results with fixed channel allocation and dynamic schemes show that the RL approaches improve the network's performance in terms of blocking probability while reducing algorithms' complexity when frequency channels are assigned to incoming voice calls.

This paper goes beyond in several directions and proposes a novel framework for DSM operation based on RL techniques in the context of next generation multicell OFDMA networks. Specifically, the contributions of this paper are: (a) It embraces the DSM problem not just at cell level (i.e., the spectrum assignment in the particular wireless interface of a single cell) but at network level by sharing the whole spectrum among different cells. Thus, basic cognitive network functionalities to observe, analyze, learn and react to the different temporal and spatial spectrum demands are introduced in order to provide the network with the ability to automatically decide an adequate spectrum assignment to the different base station transmitters. (b) In accordance with new trends in wireless communications, it considers OFDMA data networks and variable data rate traffic instead of voice calls. (c) The ability of RL to learn from interaction with the network is exploited and a new RL-based algorithm that discovers proper dynamic spectrum assignments of groups of contiguous OFDMA subcarriers or *chunks* to cells is proposed. The algorithm is based on the RL REINFORCE methods [10] not used before in this

This work has been performed in the framework of the project E3, which has received research funding from the Community's Seventh Framework programme. Also, the Spanish Research Council and FEDER funds under COGNOS grant (ref. TEC2007-60985) have supported this work. 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. The contributions of colleagues from E3 consortium and the support of the Spanish Ministry of Science and Innovation via FPU grant AP20051165 are hereby acknowledged.

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

context to the best of the authors' knowledge. These methods are characterized by a low complexity and an optimal behavior in the sense that a maximum reward is guaranteed in the long-term.

Obtained results show that the proposed RL-based algorithm is able to dynamically learn spectrum assignments in the long-term (e.g. tenths of minutes, hours), following the rather slow network load variations. It is demonstrated that under these assignments the performance in terms of spectral efficiency and users' QoS satisfaction overcomes the performance obtained with other fixed and dynamic spectrum assignment strategies [11].

Following introduction, section II introduces RL concepts and the RL REINFORCE methodology adopted here. Section III describes the proposed RL-based model for DSM whereas section IV details the algorithm introduced in this paper. Finally, section V discusses results obtained with the proposed framework and section VI concludes this work.

II. Reinforcement Learning

Reinforcement Learning comes from the field of artificial intelligence and machine learning. It consists in learning the suitable set of *actions* to choose in order to maximize a *reward* in the long-term given that there is a continuous interaction with the outside world [6]. The reward is a numerical representation of the goal achievement and varies from one step to another.

The REINFORCE methods [10] adopted in this paper consider an RL agent i represented in Fig. 1 that interacts with an environment in a succession of time steps. Agent's interaction with the environment is composed of a reward signal r and M input signals x_{ij} biased by weighting values w_{ij}. Denoting $\mathbf{x}^{(i)}$ and $\mathbf{w}^{(i)}$ as real vectors containing the set of inputs to agent i and their corresponding weights respectively, it is useful to compact the input to the agent in a single scalar parameter z_i as

$$z_i = \mathbf{w}^{(i)^T}\mathbf{x}^{(i)} = \sum_{j=1}^{M} w_{ij}x_{ij}. \tag{1}$$

Then, the RL agent propagates this input to the output y_i that is a binary number representing two possible actions. In fact, the RL agent is also called Bernoulli-logistic unit (BLU) because the output is a Bernoulli random variable with parameter $p_i = f(z_i) = 1/\left(1+e^{-z_i}\right)$, and probability mass function

$$g_i(y_i, p_i) = \begin{cases} 1-p_i, & \text{if } y_i = 0 \\ p_i, & \text{if } y_i = 1 \end{cases} \tag{2}$$

That is, p_i represents the probability that the output y_i is 1, and contains the knowledge of the agent since it determines how often one of the two possible actions are chosen. Notice that p_i depends on the input $\mathbf{x}^{(i)}$ and the weighting vector $\mathbf{w}^{(i)}$. Thus, the learning of the agent can be condensed in the weighting vector so that each time step t the agent learns by updating its weights as

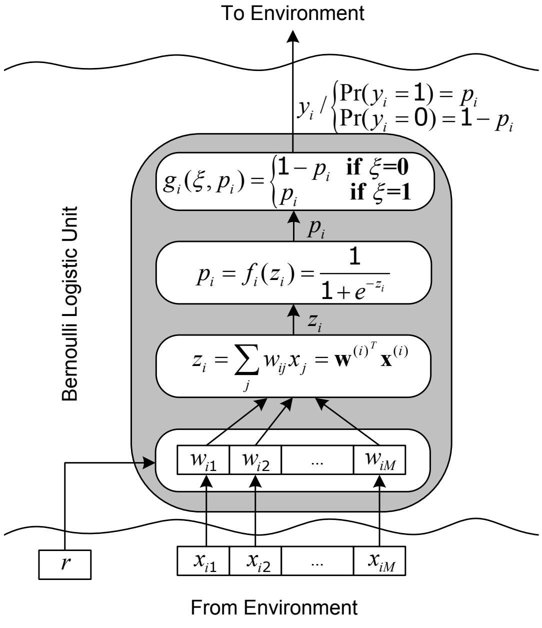
To Environment
$y_i \,/\, \begin{cases}\Pr(y_i=1)=p_i\\ \Pr(y_i=0)=1-p_i\end{cases}$
Bernoulli Logistic Unit
$g_i(\xi,p_i)=\begin{cases}1-p_i & \textbf{if } \xi\textbf{=0}\\ p_i & \textbf{if } \xi\textbf{=1}\end{cases}$
p_i
$p_i = f_i(z_i) = \frac{1}{1+e^{-z_i}}$
z_i
$z_i = \sum_j w_{ij}x_j = \mathbf{w}^{(i)^T}\mathbf{x}^{(i)}$
w_{i1}
w_{i2}
...
w_{iM}
r
x_{i1}
x_{i2}
...
x_{iM}
From Environment

Fig. 1. REINFORCE algorithm procedure.

$$\Delta w_{ij}(t) = \alpha(t)\cdot[r(t)-\bar{r}(t-1)]\cdot \\ \cdot [y_i(t-1)-p_i(t-1)]\cdot x_{ij}(t-1). \tag{3}$$

Parameter $\alpha(t) > 0$ is called the learning rate and $r(t)$ is the reward returned by the environment. $\bar{r}(t)$ is the reinforcement baseline or average reward, which is obtained from the reinforcement signal as in the following:

$$\bar{r}(t) = \beta r(t) + (1-\beta)\bar{r}(t-1), \tag{4}$$

where $0 < \beta \leqslant 1$. Low β assures enough memory of past rewards. On the other hand, decreasing the learning rate α with the RL steps improves the convergence speed of the algorithm [6]. Thus, the learning rate is linearly decreased as $\alpha(t) = \alpha(t-1) - \Delta$ where Δ should be small enough to assure a smooth transition between steps. Moreover, the maximum number of steps must be set so that negative values of α are avoided.

In general, several RL agents can be considered. An interesting theorem for this methodology [10] is that the average update vector $E\left\{\Delta\mathbf{W}|\mathbf{W}\right\}$ is proportional to $\nabla_{\mathbf{W}}E\left\{r|\mathbf{W}\right\}$, the gradient of the average reward, where $\mathbf{W} = \left[\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots \mathbf{w}^{(i)}\right]$. Then, this theorem assures that by updating the weights following (3), the weight matrix $\mathbf{W}$ found when the algorithm converges (i.e., $E\left\{\Delta\mathbf{W}|\mathbf{W}\right\} = 0$) maximizes the average reward $E\left\{r|\mathbf{W}\right\}$. Hence, we build a DSM framework and propose a RL-based strategy that benefits from this property of the REINFORCE methodology.

III. Dynamic Spectrum Management System Model

Fig. 2(a) depicts the proposed DSM framework for a multicell OFDMA network. It is composed of a centralized *DSM*

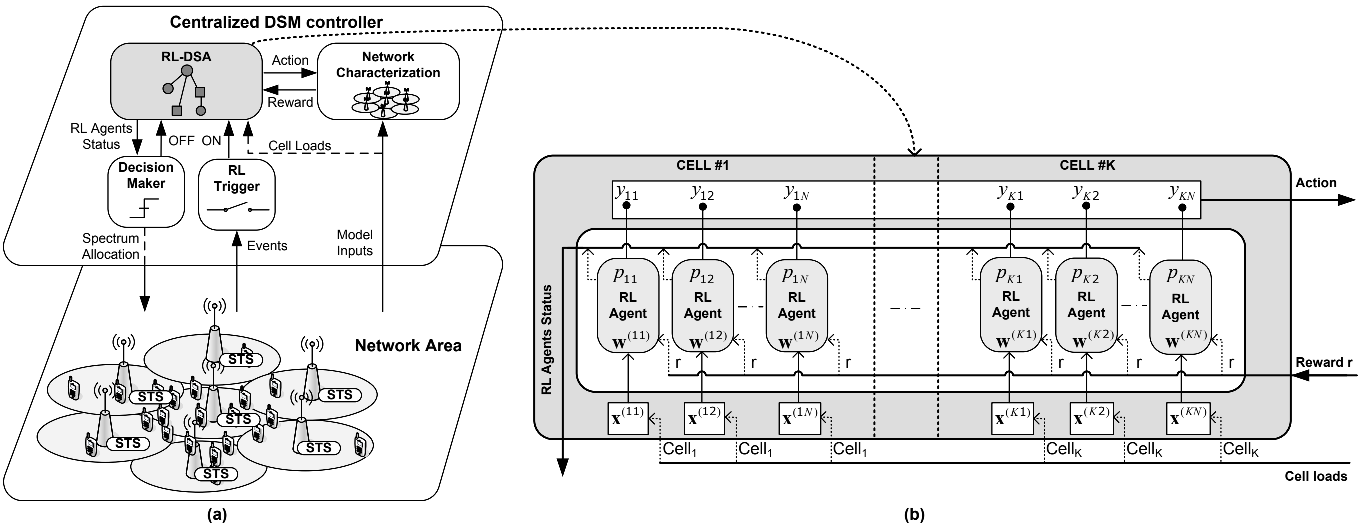

Fig. 2. Proposed DSM Model based on RL. (a) System Model, (b) RL-DSA Model.

controller providing the proper cell-by-cell chunk assignment based on a RL-DSA (Dynamic Spectrum Assignment) algorithm. It is located in an entity of the network able to control a set of cells (e.g., the aGW in the case of 3G LTE). Each cell has a Short-Term Scheduler (STS) in charge of scheduling in the short-term the users' transmissions into the available resources (i.e., chunks assigned by the DSM controller) in a temporal frame-by-frame basis.

The network variable status is observed and analyzed by a *RL Trigger* entity to detect the instants when the current spectrum assignment is no longer valid and thus trigger the RL-DSA algorithm. Metrics such as the load per cell, the intercell interference patterns and the QoS indicators are combined to decide the execution of the RL algorithm.

RL explores the action space in order to learn the assignment that produces the best reward. This implies that several assignments should be tested before the algorithm converges to a specific one. Thus, the RL actions are applied to a *Network Characterization* entity based on real measurements that mimics the behavior of the real network. Inputs to this entity are the deployment of base stations, the load per cell and the average pathloss of the users from serving and neighboring cells. These measurements can actually be obtained from real networks. The RL algorithm converges to a solution based on the interaction with the network characterization entity that returns the reward for a given action.

Finally, a *Decision Maker* analyses the status of the RL algorithm to decide when it has converged and the final chunk-to-cell assignment. It also implements the procedures to redeploy the new assignment in the real system.

IV. RL-DSA Algorithm

Let consider N available chunks in a downlink OFDMA cellular system to distribute over K cells. Chunks are numbered from 1 to N and cells are numbered from 1 to K. Once it has been triggered, the purpose of the RL-DSA algorithm is to decide which chunks should be assigned to a cell and which not. To this end, the RL-DSA algorithm seeks chunk-to-cell assignments that maximize spectral efficiency per cell provided that the QoS per cell is assured. In general, this is a Multiple-objective Optimization Problem (MOP), since solving the problem in a single cell may suppose not solving it in others. To cope with this problem, a RL feed-forward network composed of N RL agents per cell based on the REINFORCE algorithm is proposed (Fig. 2(b)).

Only one input per RL agent is considered ($M = 1$). Inputs $\mathbf{x}^{(kn)}$ reflect the load status of a cell (i.e., the percentage of users in the k-th cell with respect to the total number of users in the system) so that the algorithm is able to adapt to homogeneous and heterogeneous spatial distributions of the load. Additionally, it is considered that the n-th chunk is assigned to the k-th cell of the network characterization entity if the output y_{kn} in each RL step is 1. Thus, the action chosen by the RL system to interact with the network characterization entity is a binary assignment vector $A = [y_{11}, y_{12}, ..., y_{KN}]$ that contains the chunk-to-cell assignment given by the algorithm. For each assignment, the network characterization entity returns a reward r that is used by the respective RL agents to learn (update their weights) and decide their next outputs. Finally, as a convergence criterion, the decision maker stops the RL algorithm when the variation of all internal probabilities between two consecutive steps is below a given threshold ε during S steps or a maximum number of steps MAX_STEPS is reached. After that, in order to take the final assignment to the real network, it is decided that a specific chunk n is assigned to a cell k if p_{kn} (the internal probability of kn-th RL agent) is greater than 0.5. Otherwise the chunk is not assigned.

The execution cycle of a single RL agent in this system is depicted in Fig. 3 where two phases are differentiated. At the beginning of each step t there is a learning phase, where the reward $r(t)$ is captured and the weights and average reward are updated following (3) and (4) respectively. Then, the action phase for the next assignment is performed, where input $\mathbf{x}^{(kn)}(t)$ is captured and the output $y_{kn}(t)$ is computed from the procedure depicted in Fig. 1. Initially, full assignment is set, i.e., $y_{kn}(0) = 1 \; \forall n, k$. Finally, the initial average reward value is set to the first received reward.

A. Exploitation and exploration tradeoff

One of the characteristic features of RL is the tradeoff between *exploration* and *exploitation*. RL agents try to learn from interaction the best actions that better solve a problem by choosing the actions that give more reward (in this sense they *exploit* the learning). However, the agents should select with a small probability p_{explore} actions that are not considered suitable in order to discover new actions that may lead to better solutions in future (in this case, agents *explore* the environment). Obviously, exploration and exploitation go in different directions in the consecution of the best reward in the short-term, but RL joins both features to build a robust self-adaptive methodology that learns from the environment and reaches great reward in the long-term. Thus, internal probabilities of the RL agents in Fig. 2(b) are always maintained within the interval $[p_{\text{explore}}, 1 - p_{\text{explore}}]$ to assure a minimum exploratory probability p_{explore} on a RL agent even if its internal probability tends to 0 or 1.

B. Reward signal formulation

The REINFORCE algorithm makes the RL system evolve in such a way that the reward is maximized. Then, a reward signal that captures the final maximization objective and hence the performance of the cellular system in terms of spectral efficiency and QoS has to be defined. In this work, ${P_k}^{T_{th}}$ denotes the average user dissatisfaction per cell k and it reflects the percentage of time that the received throughput per user during a certain period is below a target throughput T_{th}. On the other hand, the spectral efficiency per cell is defined as

$$\eta_k = \text{Total cell throughput / Cell Bandwidth}, \qquad (5)$$

in bits/s/Hz. The reward signal $r_k(t)$ per cell is defined as:

$$r_k(t) = (1 - {P_k}^{T_{th}}(t))\eta_k(t). \qquad (6)$$

Notice that the maximum reward per cell is obtained when the dissatisfaction probability tends to 0 and the spectral efficiency increases. In order to maximize the overall expected reward for all cells, the reward signal is the aggregation of the rewards per cell as

$$r(t) = \sum_k r_k(t). \qquad (7)$$

Thus, the optimization problem that the RL algorithm intends to solve is maximizing the expected average reward in the long-term. That is,

$$\max \left\{ \lim_{t \to \infty} \mathrm{E}_t \left[r(t) \right] \right\}, \qquad (8)$$

where $\mathrm{E}_t[\cdot]$ denotes the expectation over time.

V. Results

Results presented in this paper focus on the validation of the proposed algorithm by averaging the outcome of several trials with static users' distributions. Even with these static network conditions, where no load variations are considered within a trial, the RL-DSA algorithm remains dynamic since it adapts the initial spectrum assignment (full assignment) to another one that improves the received reward.

A system simulator has been developed to simulate the behavior of an OFDMA-based multicell network in downlink whose parameters are summarized in Table I. The power devoted to every chunk is constant and does not change during

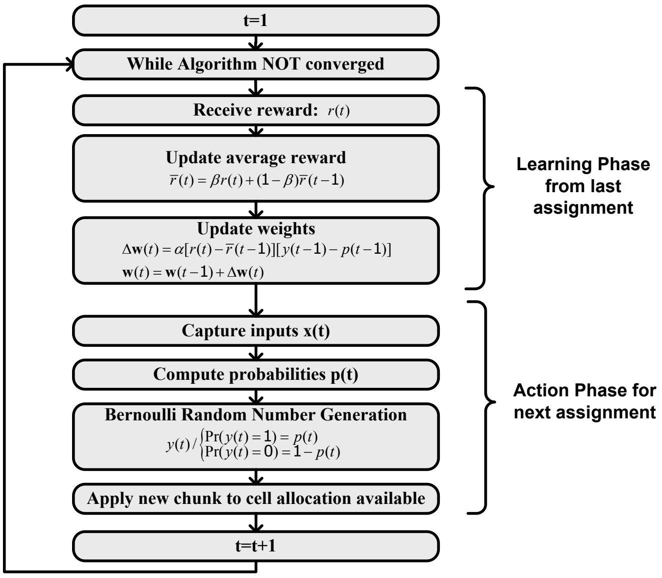
t=1
While Algorithm NOT converged
Receive reward: $r(t)$
Update average reward
$\overline{r}(t) = \beta r(t) + (1-\beta)\overline{r}(t-1)$
Update weights
$\Delta\mathbf{w}(t) = \alpha[r(t) - \overline{r}(t-1)][y(t-1) - p(t-1)]$
$\mathbf{w}(t) = \mathbf{w}(t-1) + \Delta\mathbf{w}(t)$
Capture inputs x(t)
Compute probabilities p(t)
Bernoulli Random Number Generation
$y(t) / \begin{cases} \Pr(y(t)=1) = p(t) \\ \Pr(y(t)=0) = 1 - p(t) \end{cases}$
Apply new chunk to cell allocation available
t=t+1
Learning Phase from last assignment
Action Phase for next assignment

Fig. 3. Execution procedure of a single RL agent.

TABLE I
SIMULATION PARAMETERS

Number of cells	$K = 3$
Cell Radius	$R = 500$ meters
Antenna patterns	Omnidirectional
Frame duration	2 ms
Carrier frequency	2 GHz
Number of subcarriers per chunk	25
Chunk bandwidth	$W/N = 375$ KHz
Power per chunk	$P = 35$ dBm
Path loss in dB at d km	$128.1 + 37.6log10(d)$ [12]
Shadowing standard deviation	8 dB [12]
Small Scale Fading model	ITU Ped. A [12]
UE thermal noise	-174 dBm/Hz
UE noise factor	9 dB
UE speed	0 km/h (static)
User's satisfaction throughput	$T_{th} = 128$ kbps
Maximum theoretical spectral efficiency	$\eta_{max} = 4$ bits/s/Hz
BER	10^{-3}
Short Term Scheduling method	Proportional Fair [13]
Averaging window	50 frames
RL parameters $[\alpha, \Delta, \beta]$	$\left[10, 10^{-5}, 0.1\right]$
Exploratory probability	$p_{explore} = 0.1\%$
RL convergence criterion $[\varepsilon, S]$	$\left[10^{-4}, 5000\right]$
MAX_STEPS	100000

simulation. The Signal to Interference plus Noise (SINR) ratio per each chunk is calculated as

$$\gamma_{m,n} = \frac{P_k G_{k,m} S_{k,m} F_{k,m,n}}{\sum\limits_{j \in \Phi_n j \neq k} \left(P_j G_{j,m} S_{j,m} F_{j,m,n}\right) + \Upsilon_N}, \qquad (9)$$

where $\gamma_{m,n}$ represents the SINR in the n-th chunk for the m-th user, index k represents the serving cell and j any interfering cell. Φ_n is the set of cells using the n-th chunk. P_k, $G_{k,m}$, $S_{k,m}$ and $F_{k,m,n}$ denote respectively the transmitted chunk power, the distance dependant channel gain, the shadowing, and the fast frequency selective fading component that depends on the chunk n. Finally, Υ_N denotes the total noise power including receiver noise figure.

As a frame-by-frame Short-Term Scheduling (STS) a Proportional Fair scheduler has been used [13]. The achievable rate $R_{m,n}$ that the m-th user can obtain in the n-th chunk in a frame is [14]

$$R_{m,n} = \frac{W}{N} \log_2 \left(1 - \frac{1.5\gamma_{m,n}}{\ln(5BER)}\right), \qquad (10)$$

where W/N is the chunk bandwidth and *BER* stands for the target Bit Error Rate. The maximum theoretical spectral efficiency η_{max} depends on the modulation scheme and is limited to 4 bits/s/Hz. For RL-DSA simulations the same simulator has been employed for the network characterization entity in Fig. 2(a) in order to predict the behavior of the real network.

A different number of users from 18 to 81 are distributed homogeneously within a cell. Users remain static during simulations and their buffers are always full so their traffic model represents a service that demands as much capacity as possible. They are satisfied if the received throughput during last second is above 128 kbps. 100 different user distributions have been tested and averaged for each number of users.

Under abovementioned conditions the RL-DSA algorithm seeks for suitable spectrum assignments. Performance results are presented in terms of spectral efficiency and user's average dissatisfaction probability by comparing the proposed RL-DSA algorithm with fixed frequency reuse factors FRF1 (full assignment) and FRF3 (1/3 of the bandwidth is assigned per cell), and a dynamic heuristic strategy named here DSA-heur [11]. The scenario is composed of 3 cells and a maximum of 6 available chunks. This is an adequate scenario since, with this deployment, FRF3 constitutes an upper bound in terms of spectral efficiency because it does not introduce intercell

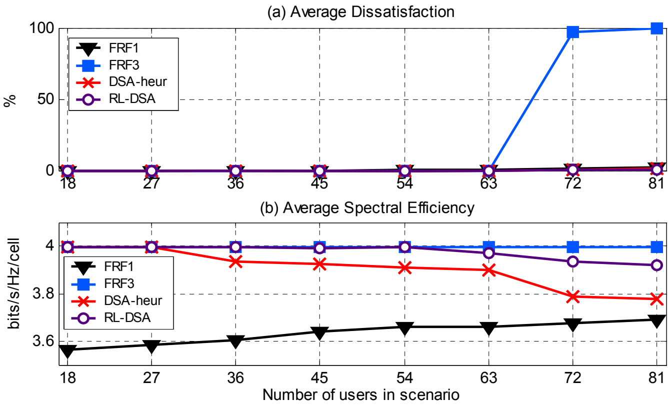

Fig. 4. Performance results in homogeneous spatial distribution. (a) Average user's dissatisfaction probability, (b) Average spectral efficiency.

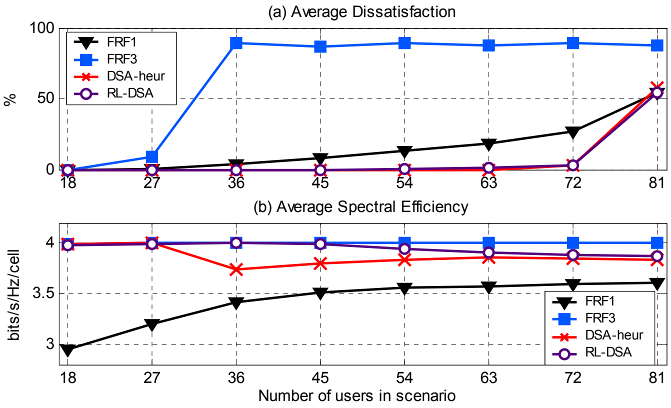

Fig. 5. Performance results in heterogeneous spatial distribution. (a) Average user's dissatisfaction probability, (b) Average spectral efficiency.

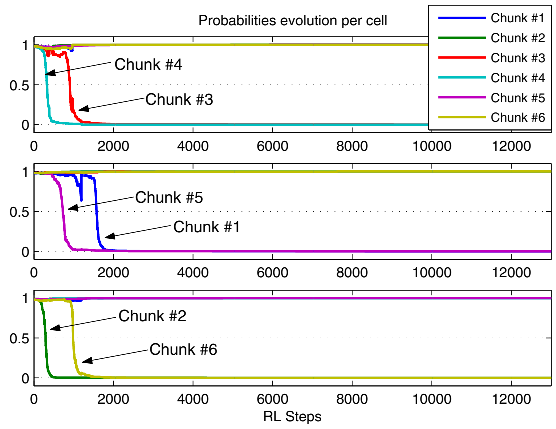

Fig. 6. RL internal probabilities evolution per cell.

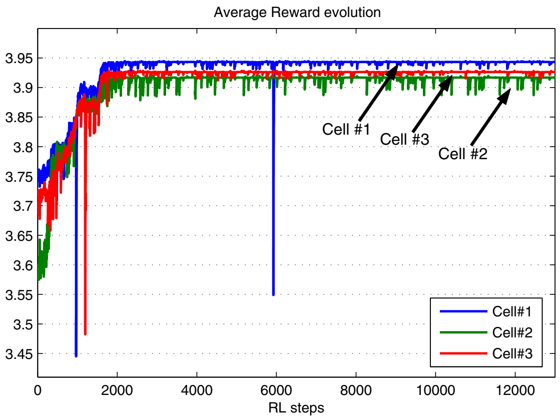

Fig. 7. Average reward evolution per cell.

interference (two different chunks are given to each cell). In contrast, FRF1 is the strategy that introduces the worst intercell interference (the six chunks are given to all cells), whereas DSA-heur represents another adaptive strategy in between.

Fig. 4 and Fig. 5 show the performance comparison for a homogeneous (33%, 33%, 33%) and a heterogeneous (84%, 8%, 8%) spatial distribution of the users among cells respectively, where values in brackets denote the percentage of the users in the scenario in each cell. RL-DSA shows the best tradeoff between spectral efficiency and dissatisfaction probability. In both scenarios, RL-DSA practically obtains maximal spectral efficiency for low loads (up to 45 users) maintaining the dissatisfaction probability negligible. For higher loads, RL-DSA is still the strategy that better combines spectral efficiency and QoS fulfillment, since, on one hand, it improves the spectral efficiency with respect to FRF1 and DSA-heur maintaining the same dissatisfaction and, on the other hand is able to improve the dissatisfaction up to 100% with respect to FRF3 that lacks of enough capacity per cell when the system gets highly loaded.

Fig. 6 and Fig. 7 illustrate the RL behavior when it searches for a specific spectrum assignment under a specific situation with 81 users randomly spread around the scenario (homogeneous scenario case). Notice how the internal probabilities (Fig. 6) evolve either to $1-p_{\text{explore}}$ or p_{explore} determining the chunk assignment for each cell when the probability is greater than 0.5. Then, cell 1 is assigned chunks 1, 2, 5 and 6, cell 2 is assigned chunks 2, 3, 4 and 6 and cell 3 is assigned chunks 1, 3, 4 and 5. Observe that each assigned chunk to a cell has only interference from one of the other two cells. In this way, different from the other spectrum assignment strategies, the RL-DSA algorithm minimizes the intercell interference for all chunks and allows maximizing the spectral efficiency while maintaining the satisfaction of the users (see Fig. 4 for 81 homogeneously distributed users).

Finally, Fig. 7 shows the evolution of the average reward per cell while the RL-DSA algorithm seeks for the solution. It can be seen that the reward increases for all cells as the RL framework learns from the Network characterization entity. Thus the RL-DSA is able to learn the spectrum assignment that improves the performance of the network. In this process there are some spectrum assignments that return low rewards (low peaks in figure). Even in these situations the RL-DSA still learns since it discards solutions that seem to be not suitable.

VI. Conclusion

In this work the suitability of Reinforcement Learning (RL) for Dynamic Spectrum Management (DSM) in next generation OFDMA-based networks has been studied by introducing a novel RL-based framework and a Dynamic Spectrum Assignment (DSA) algorithm whose foundations reside on an optimal RL methodology called REINFORCE. The different modules of the framework have been discussed and presented. Within this framework, the RL algorithm demonstrates the best tradeoff between spectral efficiency and QoS fulfillment when compared with other fixed and dynamic spectrum assignment strategies under homogeneous and heterogeneous spatial distributions. Future work will address more complex scenarios with a higher number of cells, chunks and users. Also dynamism will be introduced to demonstrate the capabilities of the RL-DSA algorithm when facing temporal and spatial variations of the network load.

References

[1] J. A. Hoffmeyer, "Regulatory and standardization aspects of dsa technologies - global requirements and perspective," in *IEEE New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2005, pp. 700–705.

[2] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEE Signal. Proc. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

[3] T. A. Weiss and F. K. Jondral, "Spectrum pooling: an innovative strategy for the enhancement of spectrum efficiency," *IEEE Commun. Mag.*, vol. 42, no. 3, pp. S8–14, 2004.

[4] R. W. Thomas, D. H. Friend, L. A. Dasilva, and A. B. Mackenzie, "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives," *IEEE Commun. Mag.*, vol. 44, no. 12, pp. 51–57, 2006.

[5] B. Le, R. T.W., and B. C.W., "Cognitive radio realities," *Wireless Communications and Mobile Computing*, vol. 7, pp. 1037–1048, 2007.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, March 1998.

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

[8] S. Singh and D. Bertsekas, "Reinforcement learning for dynamic channel allocation in cellular telephone systems," in *Advances in Neural Information Processing Systems*, vol. 9. The MIT Press, 1997.

[9] N. Lilith and K. Dogancay, "Dynamic channel allocation for mobile cellular traffic using reduced-state reinforcement learning," in *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 4, 2004, pp. 2195–2200 Vol.4.

[10] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. V8, no. 3, pp. 229–256, May 1992.

[11] F. Bernardo, J. Pérez-Romero, O. Sallent, and R. Agustí, "Advanced spectrum management in multicell OFDMA networks enabling cognitive radio usage," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2008, pp. 1927–1932.

[12] 3GPP, "Physical layer aspects for evolved universal terrestrial radio access (UTRA)," 3GPP, Tech. Rep. TR 25.814 v7.1.0, September 2006, release 7.

[13] C. Wengerter, J. Ohlhorst, and A. v. Elbwart, "Fairness and throughput analysis for generalized proportional fair frequency scheduling in OFDMA," in *IEEE 61st Vehicular Technology Conference 2005-Spring*, vol. 3, 2005, pp. 1903–1907.

[14] J. Jang and K. B. Lee, "Transmit power adaptation for multiuser OFDM systems," *IEEE J. Sel. Area. Comm.*, vol. 21, no. 2, pp. 171–178, 2003.