# An Intelligent Two-agent Self-configuration Approach for Radio Resource Management

K. Collados, J.L. Gorricho, J. Serrat
Dept. of Network Engineering
Polytechnic University of Catalonia (UPC)
Barcelona, Spain
juanluis@entel.upc.edu

H. Zheng[1], K. Xu[2]
Key Lab. of Universal Wireless Comm.
[1]BUPT and [2]Tsinghua University
Beijing, China

***Abstract*—in this paper we propose the use of a two-agent learning scheme for the management of radio resources on cellular access networks. The management is materialized by the implementation of a self-configuration system governing the setup of several parameters on each base station. The two agents have independent goals; one is trying to maximize the quality of service and the other the economic benefit. Thanks to the combined use of the fuzzy logic technique and reinforcement learning, both agents will work in a complementary mode, achieving both goals simultaneously.**

***Keywords—self-configuration; reinforcement-learning; fuzzy-logic;***

## I. INTRODUCTION

The management of radio resources for cellular access networks has traditionally followed a static approach [1], [2], [3]. Nevertheless, key issues like minimizing the energy consumption, sharing the infrastructure among different providers for the 4G mobile systems, or establishing a trading mechanism of radio resources among different providers are future optimization challenges requesting a different approach. Furthermore, present and future deployment of heterogeneous wireless networks [4], [5] must definitively evolve to a cooperative global network for unquestionable reasons.

In this scenario, a relevant number of proposals, classified as self-configuration approaches [6], are trying to build efficient systems applying different strategies for the management of radio resources. Most of these approaches are based on the use of machine learning techniques, i.e. supervised learning [7], unsupervised learning [8], [9] and reinforcement learning [10], [11], [12]. Our proposal focuses on applying reinforcement learning and fuzzy logic, with the novelty of considering the combination of two agents on each base station, and consequently, considering a sophisticated set of inputs and the necessary coordination of the two agents' actions to achieve their respective goals.

## II. STATE OF THE ART

One common approach in the literature, targeting the implementation of self-configuration systems, consist in using reinforcement learning to dynamically employ several radio access technologies. For example, the authors from [4] apply reinforcement learning, in particular, the actor-critic technique, to dynamically choose between UMTS and LTE to set up new connections or to modify the existing ones. The algorithm is implemented according to the measures of the current user satisfaction for the ongoing connections on both technologies. A similar strategy is followed in [5] choosing between UMTS and WLAN. And a third example can be found in [13] considering three different radio access technologies: GERAN, UMTS and WLAN; in this last case the research is developed working with neural networks whose weights are learned from applying a combination of fuzzy logic and reinforcement learning.

Other approaches applying the reinforcement learning technique try to tune some of the operating parameters of the base stations to improve the coverage and system capacity. The authors from [12] and [14] combine the Q-Learning technique with the fuzzy logic to deduce the best antenna tilt at the base station, in order to improve the system capacity. The same learning techniques are applied in [11], but in this case tuning the transmitted power from the base stations to improve the system capacity thanks to minimizing the SINR. The novelty of our contribution in comparison to these studies comes from the definition of a self-configuration system with two distinctive learning agents with independent goals.

## III. THE TWO-AGENT SYSTEM

The working scenario for the present proposal assumes an infrastructure of base stations with several providers giving service in a completely uncoordinated manner; nevertheless, all of them are sharing the radio resources at each base station. A minimum number of radio resources are guaranteed per provider and base station, but the remaining resources at each base station constitutes a pool of available resources to be taken by any of the service providers.

The proposed self-configuration system, located at each base station and individual for each provider, is made up of two agents. Each of these two agents is meant to maximize a specific goal; in particular, one of them will maximize the quality of service (QoS agent) and the other will maximize the providers' economic benefit (Profit agent). Figure 1 is a schematic of how these two agents interact between themselves and with their environment. As we can see, both agents retrieve information from the environment, the QoS agent reports to the Profit agent, and finally, only the Profit agent takes actions to modify the operating parameters of the base station.

At each base station, the parameters to be setup by the self-configuration system will be the following:

- The *distribution of channels* allocated to setup new connections and those reserved for handover purposes.
- The *total amount of channels* owned by the provider at that base station. Each provider will decide to request new channels or release some of the already owned ones.
- The *coverage area* of the base station, configurable by tuning the power control mechanism.

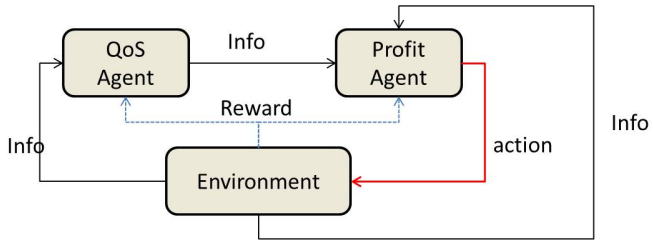


Fig. 1. The two-agent scheme proposal for the self-configuration system

The proposed reinforcement learning system works under the following assumptions:

- The environment to learn from is the base station under control.
- The actions taken by the Profit agent will be the setup of the parameters governing the base station.
- The rewards are quantified measures of the quality of service and the economic benefit achieved by the provider.

A main drawback on applying the traditional reinforcement learning technique comes from working with a discrete number of states and actions [15]. One alternative would be to approximate the value functions: V(s) or Q(s,a) by parameterized mathematical expressions, but this is presently an unclear and too open approach [16]. Another alternative to cope with a continuous input space defining states and actions comes from using the fuzzy logic technique in combination with reinforcement learning. This is the strategy that we will follow in the present study. For both agents on figure 1 we have combined the use of the fuzzy logic and reinforcement learning mechanism, as shown in figures 2 and 3, implementing the actor-critic and the Q-learning techniques [15]. The applied schemes are the same for both agents; but each one having its particular inputs and resulting outputs.

The input values given to the fuzzy logic module for both agents will be the following, for the QoS agent the inputs are:

- The base station blocking rate (unavailability to set up a new service session due to the lack of resource).
- The base station dropping rate (unavailability to hand over a service session due to the lack of resource).

For the Profit agent the inputs are:

- The base station load.
- The amount of owned channels by the provider.
- The *decisions vector* suggested by the QoS agent.
- The current amount of available channels for all providers at that base station.

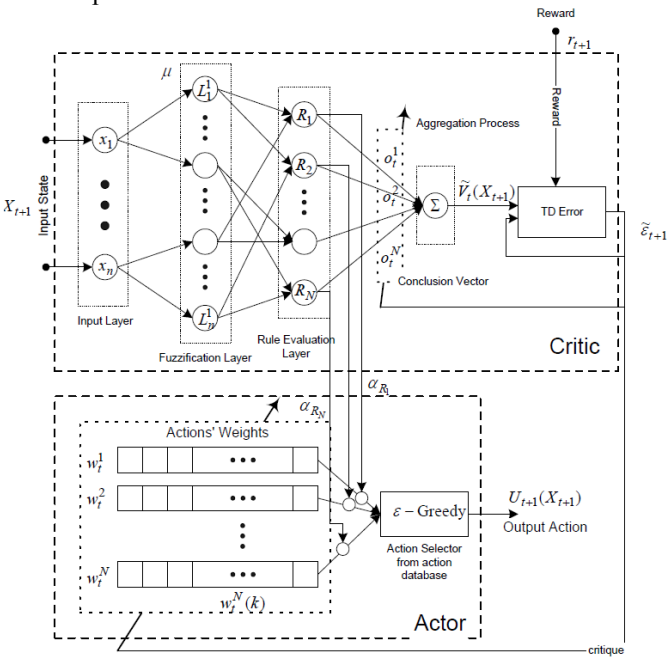


Fig. 2. Actor-critic technique

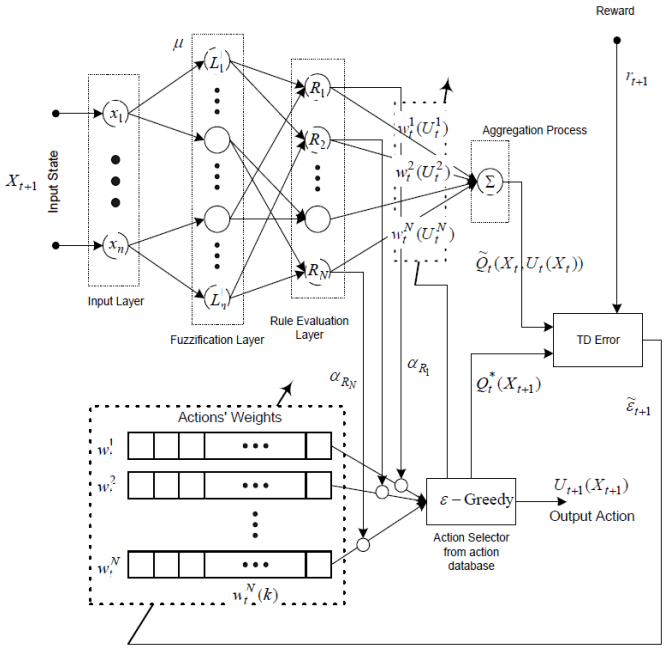


Fig. 3. Q-learning technique

The actor-critic technique, see figure 2, deduces the value of the state V(s) as a linear combination of the outputs from all the rules defined by the fuzzy logic module. Nevertheless, the contribution from any individual rule will be something to be learned by the reinforcement learning mechanism, i.e. the weight $\sigma^n$ to be given to each rule. The Q-learning technique, see figure 3, deduces the value of the pair: (state,action) as Q(s,a), in this case, as a linear combination of the outputs from

all the rules defined by the fuzzy logic module. Again, the contribution $w^n$ from any individual rule is something to be learned by reinforcement learning. Finally, as shown in figures 2 and 3, the actions to be taken are multi-dimensional, three-dimensional in our case, as we will explain later. For comparative purposes, a non-learning strategy has also been implemented for both agents, using internally the same schemes as shown in figures 2 and 3; the only difference is that in this case all rules and actions have the same weight equal to one.

## IV. QoS Agent Design Details

The purpose of this agent is to achieve a given quality of service (QoS) regarding two basic parameters: the blocking rate and the dropping rate. Based on these two parameters the reward function used by the reinforcement learning mechanism is formulated as follows:

$$\text{Reward} = (T_B - B) + \beta * (T_D - D)$$

Where $(T_B - B)$ is the difference between the actual blocking rate and a given target value $T_B$ (5% in all our simulations) and $(T_D - D)$ is the difference between the actual dropping rate and a given target value (1% in all our simulations). The blocking rate is less critical than the dropping rate on estimating the quality of service; consequently, a factor $\beta = 4$ is added to emphasize the dropping rate penalty in front of the other.

The two inputs, the blocking and dropping rates, are labeled according to six different fuzzification categories as shown in figure 4. Depending on the value of the input we obtain the degree of membership to each category; but, as shown in figure 4, independently of the input value, only two categories at most can have a positive output.

We have defined 5 rules within the fuzzy logic module for the QoS agent. For a better understanding of these rules we show them in figure 5 as a bi-dimensional representation, where we highlight the fact that the inputs for the 5 rules come from the 6 + 6 categories defined in figure 4. The rules are named as: *acceptable, high blocking, high dropping, extreme blocking* and *extreme dropping*. The definition of all 5 rules is made as the product of the addition of the degrees of membership to the corresponding categories involved in each rule. For instance, the rule named *acceptable* is defined as the product of the addition of the degree of membership to the 3 categories from the blocking rate input multiplied by the addition of the degree of membership to the 3 categories from the dropping rate input.

The resultant rules outputs are processed by the actor-critic or the Q-learning algorithms to obtain a three-dimensional vector named a *decisions vector* (the QoS agent action in fact), which will be the output delivered to the Profit agent as a set of decision parameters suggested by the QoS agent. This three-dimensional vector will have the following components:

- Increase/decrease the amount of reserved channels for handover (HO)
- Release or acquire a new channel (CH)
- Increase/decrease the coverage area (CA)

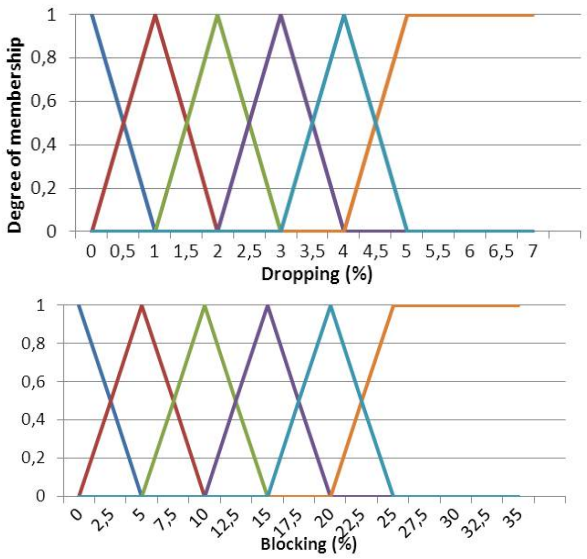


Fig. 4. Dropping and Blocking rates labeling (categories)

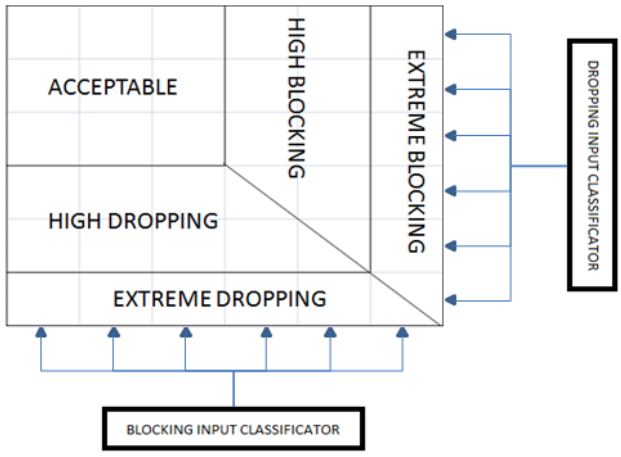


Fig. 5. Two-dimensional representation of the 5 rules for the QoS agent

Each rule output is multiplied by the learned action weight, resulting in, let us say, an action eligibility weight, see figures 2 and 3, although, in the end, we add the contribution of all actions to produce the definitive action. The actions (*decisions vector* as named above) are predefined for each rule as shown in figure 6. Each row of the table specifies a "set of values" due to each rule. The "set of values" is multiplied by its action eligibility weight and finally, all of them from all rules are added to deduce the definitive three-dimensional vector to be delivered to the Profit agent.

| | | | | | | |
|---|---|---|---|---|---|---|
| Acceptable | $HO_1$ | $CH_1$ | $CA_1$ | 0 | 0 | 0.75 |
| High Dropping | $HO_2$ | $CH_2$ | $CA_2$ | -0.5 | 0.2 | -0.8 |
| Extreme Dropping | $HO_3$ | $CH_3$ | $CA_3$ | -1 | 0.35 | -1.2 |
| High Blocking | $HO_4$ | $CH_4$ | $CA_4$ | 0.5 | 0.2 | -0.8 |
| Extreme Blocking | $HO_5$ | $CH_5$ | $CA_5$ | 1 | 0.35 | -1.2 |

Fig. 6. QoS agent actions (decisions vector)

## V. PROFIT AGENT DESIGN DETAILS

The purpose of this agent is to achieve the maximum economic benefit for the provider, to accomplish this objective the reward function is defined as:

$$\text{Reward} = \text{Load} * \text{Price} * \text{Owned\_Channels} - \text{Cost} * \text{Owned\_Channels}$$

where *Price* and *Cost* are constant values, *Load* accounts for the average percentage of channels in use, and *Owned_Channels* accounts for the amount of channels kept at that time by that particular provider. Accordingly, the inputs for the Profit agent are classified into two sets:

- Basic inputs:
  - The cell load
  - The three dimensional *decisions vector* from the QoS agent
- Control inputs (only used to turn on/off a rule):
  - Availability of spare channels to be acquired
  - Possibility to reduce/increase the coverage area
  - Possibility to reduce/increase the amount of reserved channels for handover

We have defined different categories depending on the considered basic inputs as shown in figures 7, 8, 9 and 10. The inputs: HO, CH and CA of figures 8, 9 and 10 are the three components of the decisions vector output from the QoS agent. And depending on the type of input we have considered a different set of categories: 5 for HO, 1 for CH and 2 for CA. For the HO input we have considered 5 categories because when the QoS agent is requesting to increase/decrease the amount of reserved channels for handover, we understand that we can indirectly satisfy this request modifying the coverage area or acquiring a new channel.

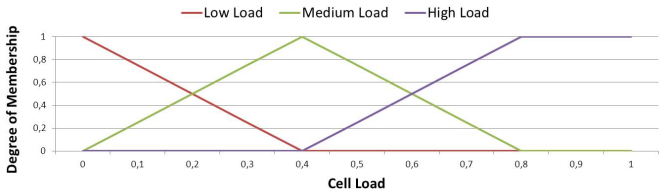


Fig. 7. Cell load categories

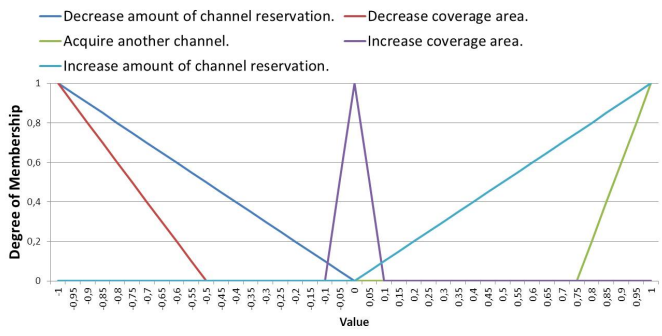


Fig. 8. Handover (HO) categories

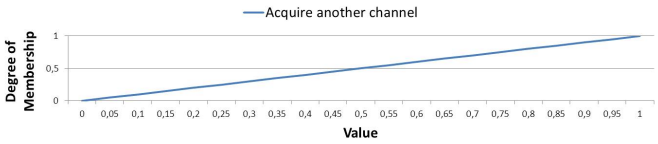


Fig. 9. Acquire channel (CH) category

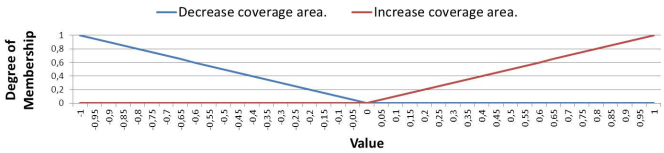


Fig. 10. Coverage Area (CA) categories

We have defined 6 rules within the fuzzy logic module for the Profit agent, which are:

- Increase the amount of channel reservation.
- Decrease the amount of channel reservation.
- Increase the coverage area.
- Decrease the coverage area.
- Acquire a channel.
- Release a channel.

Nevertheless, the above rules are deactivated under given circumstances, so we turn on/off the rules to avoid learning from actions that cannot be taken. The thresholds used to activate each of the six rules are set as follows:

- The *increase the amount of channel reservation* rule is active only if the amount of channels devoted to handover is below 80% of the total amount of owned channels by the provider.
- The *decrease the amount of channel reservation* rule is active only if the amount of channels devoted to handover is above 20% of the total amount of owned channels by the provider.
- The *increase the coverage area* rule is active only if the radius of the coverage area is below one time the distance between two base stations.
- The *decrease the coverage area* rule is active only if the radius of the coverage area is above 75% of the distance between two base stations.
- The *acquire a channel* rule is active only if there are spare channels at the base station.
- The *release a channel* rule is active only if the provider owns more than 3 channels.

The contribution of all categories, enumerated in figures 7 to 10, to define the 6 rules of the Profit agent, is shown in figure 11. Finally, the action to be taken by the Profit agent will be a three-dimensional action so as to:

- Increase/decrease the amount of reserved channels for handover.
- Increase/decrease the coverage area.
- Acquire (if available) or release a channel.

## VI. Simulation Results

The working scenario has been simulated defining a rectangular geographical area to be covered by 9 base stations. For the users' mobility a completely random pattern has been applied, the original location and speed are chosen at random, and the speed is kept along the time but with small random variations. Those mobile terminals reaching the geographical area bounds are mirrored in their trajectories to always keep them within the coverage area of the 9 base stations. The traffic demand generated by all users follows the Poison statistics for the call setup requests and the session duration.

Each provider has its own independent set of users with slightly different statistical profiles related to the amount of users and traffic demand. The simulation results we present here have been retrieved from the base station located in the center of the geographical area under study. On running the simulations, the two-agent system at each base station and for each provider will decide, on its own, to increase or decrease its coverage area, to request or release a communication channel, and to increase or decrease the channels devoted to handover purposes. As previously mentioned, a non-learning algorithm has also been implemented for benchmarking purposes. The non-learning algorithm implies the use of a fuzzy logic module with constant weights, all equal to one, for all rules, instead of using reinforcement learning to learn its weights.

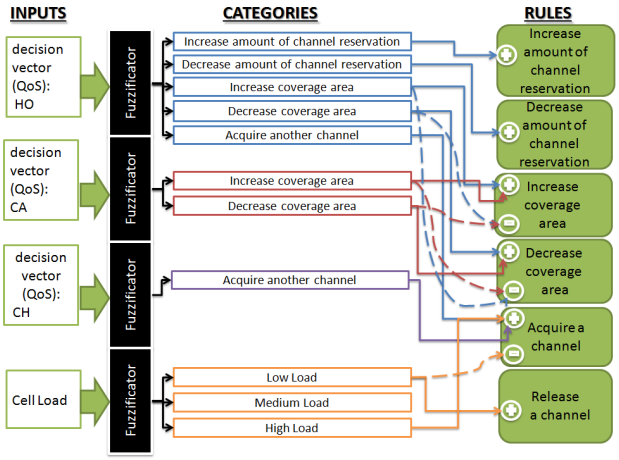


Fig. 11. Schematic of inputs/categories/rules for the profit agent

Due to the two-agent scheme proposal, we will focus on testing the system performance for any possible combination of the learning techniques adopted in each agent. The features under evaluation will be: the quality of service, the economic benefit and the averaged load.

### A. Single provider scenario

All the following figures and tables will show a compact notation, where the combination of two capital letters refers to, the first one on the left, the algorithm used at the profit agent; the second one on the right, the algorithm used at the QoS agent. The capital letters mean: A for Actor-Critic; Q for Q-Learning and N if no learning technique is applied.

Figure 12 shows the achieved blocking and dropping rates as a function of the learning techniques used for each agent. Applying the Actor-Critic technique on both agents, the dropping and blocking rates are kept below the target values. On the other hand, the remaining combinations overtake the target blocking rate of 5%, except for the NA combination. The NA combination seems to be quite successful, but, although not seen on Figure 12, the dropping rate also overtakes the target value of 1%. The worst results come from not using any learning technique at the QoS agent.

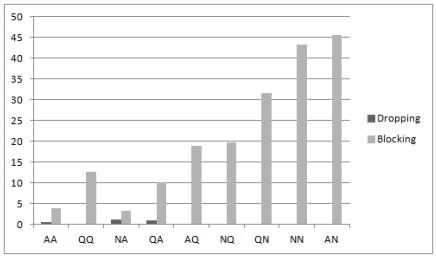


Fig. 12. Percentage of blocking and dropping rates

Figure 13 shows the achieved income and derived costs. This time, using the Q-Learning technique on both agents provides the best income results, whereas the cost penalty is kept almost identical for all combinations of algorithms due to a similar usage of available channels in all cases.

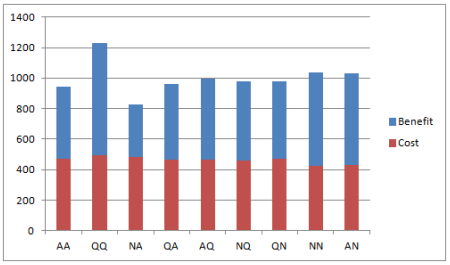


Fig. 13. Absolute operational Income and Cost

Figure 14 shows the usage of channels. As we can see, a medium load is assured in all cases; basically due to the *cell load* input to the profit agent, as shown in Figure 11 for the last two rules. Using the Q-Learning technique at the profit agent we obtain the best results, excluding the AN combination, which is not representative as its blocking rate, see figure 12, makes it useless. Considering that the load input is a parameter of the profit agent *reward* function, a greater load of the system is understood as a positive achievement for this agent trying to maximize the economic benefit.

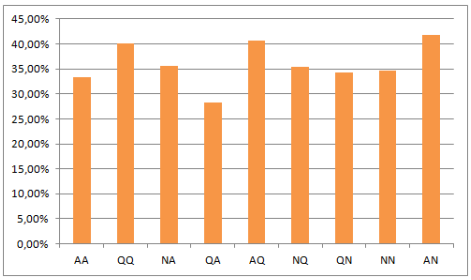


Fig. 14. Averaged load

Analyzing the results shown in figures 12 to 14, we can say that the combinations of learning techniques for both agents can be divided into two main categories. The first one results from those combinations where the Actor-Critic algorithm is used at the QoS agent, this category is more focused on achieving a high quality of service, instead of a better profit for the provider. The second category comes from those combinations using the Q-learning technique at the QoS agent, this category is more focused on achieving the maximum economic benefit for the provider, regardless of not satisfying the targets expected for the QoS. Finally, when not applying any learning technique at the QoS agent it always produces the worst results.

### *B. Two coexistings providers scenario*

To have a deeper insight of the proposed approach, a second scenario with two coexisting providers has also been tested. In this scenario each individual provider works on its own and has its own users. Both providers share a pool of available channels at each base station. The results shown on the following tables are restricted to the "AA" and "QQ" combinations. "AA" was chosen for its high performance on QoS, and "QQ" for its performance for the economic benefit. In this new scenario the reward functions have been modified. For one provider we will maintain the same QoS targets (blocking = 5% and dropping = 1%) but the *price* paid by the end user for the use of a channel will be two-times the value set for the scenario where we had a single provider. For the other provider the *price* for the use of a channel will not be modified and the QoS targets will be less restrictive (blocking = 15% and dropping = 5%). The users in this scenario are subscribers of a single provider. In other words, the users cannot choose the provider they want to connect to. The rules and actions taken by the profit agent for each provider are the same as they were shown in Figure 11, the only difference from the previous study case comes from both providers sharing a common pool of available channels at any base station. On the following tables “Op1” stands for the provider with more restrictive QoS targets. The other provider will be labeled as "Op2". In both cases the acronym is followed by the combination of learning techniques used on both agents. All four possible combinations have been tested and are shown on tables I to IV.

TABLE I. AA-AA TEST RESULTS

| | **Op1-AA (%)** | **Op2-AA (%)** |
|---|---|---|
| **Coverage area** | 19,26 | 12.84 |
| **Owned channels** | 51.52 | 42.09 |
| **Average load** | 44.94 | 47.85 |
| **Channels in handover** | 45.99 | 46.69 |
| **Blocking** | 20.37 | 16.09 |
| **Dropping** | 0.05 | 2.31 |

TABLE II. AA-QQ TEST RESULTS

| | **Op1-AA (%)** | **Op2-QQ (%)** |
|---|---|---|
| **Coverage area** | 15.27 | 12.35 |
| **Owned channels** | 52.90 | 44.69 |
| **Average load** | 39.13 | 48.25 |
| **Channels in handover** | 50.87 | 44.83 |
| **Blocking** | 10.56 | 8.57 |
| **Dropping** | 0.11 | 0.73 |

TABLE III. QQ-AA TEST RESULTS

| | **Op1-QQ (%)** | **Op2-AA (%)** |
|---|---|---|
| **Coverage area** | 12.06 | 14.03 |
| **Owned channels** | 40.64 | 54.49 |
| **Average load** | 49.29 | 45.34 |
| **Channels in handover** | 51.17 | 43.34 |
| **Blocking** | 22.97 | 10.57 |
| **Dropping** | 0.01 | 0.00 |

TABLE IV. QQ-QQ TEST RESULTS

| | **Op1-QQ (%)** | **Op2-QQ (%)** |
|---|---|---|
| **Coverage area** | 13.07 | 10.82 |
| **Owned channels** | 44.31 | 49.50 |
| **Average load** | 45.80 | 42.33 |
| **Channels in handover** | 46.18 | 42.50 |
| **Blocking** | 21.61 | 7.76 |
| **Dropping** | 0.00 | 2.95 |

As we can see on tables I to IV, in spite of having two providers sharing the same amount of available channels as in the previous study case, both of them try to keep their base stations loaded at a *medium* value. Each single provider has less available channels at its disposal; hence, to increase their economic benefit they must increase their load keeping it in a medium value as defined in Figure 7. Nevertheless, the reservation of channels for handover is maintained around 50%, as it was the case for the single provider. It may seem a high value, but necessary to satisfy the dropping rate target.

## VII. CONCLUSIONS

In this paper we have proposed and evaluated the use of a two-agent scheme for the management of radio resources on a cellular access network. In order to do that, we have implemented a self-configuration system governing a set of working parameters on all the base stations to maximize the quality of service and the economic benefit. The simulation results have shown that the two-agent approach is suitable to handle both goals simultaneously, outperforming any alternative using non-learning approaches. From this research we have to point out the impact of the specific learning technique adopted at any of the two agents. The Actor-critic algorithm seems to give the best QoS results when applied at both agents, whereas the Q-Learning algorithm will provide the best income results. Nevertheless, independently of considering a single provider, two providers sharing a pool of resources, or even applying a variable traffic demand or a variable amount of available resources, those algorithms constantly readapt the weights of the applied rules, from the fuzzy logic module, to the present circumstances, trying to maximize their long-term rewards.

## ACKNOWLEDGMENT

This work has been done in the framework of the EVANS project (PIRSES-GA-2010-269323), as well as with the support of projects TEC2012-38574-C02-02 from Ministerio de Ciencia y Educacion, and the FLAMINGO project (318488) of the 7th FP of the European Commission.

## REFERENCES


[1] Zwi Altman, Hervé Dubreil, Ridha Nasri et al. "Understanding UMTS Radio Network Modelling, Planning and Automated Optimisation", John Wiley & Sons, Ltd., 2006.

[2] Antti Tölli, Petteri Hakalin and Harri Holma, "Performance Evaluation of Common Radio Resource Management (CRRM)", ICC 2002.

[3] J.Pérez-Romero, 0. Sallent, R. Agustí, P. Karlssot et al. "Common Radio Resources Management: Functional Models and Implementation Requeriements", PIMRC 2005.

[4] Nemanja Vucevic, Jordi Pérez-Romero, Oriol Sallent, Ramon Agustí "Reinforcement learning for joint radio resource management in LTE-UMTS scenarios" Computer Networks, Elsevier, 2011.

[5] Zhiyong Feng, Li Tan et al. "Reinforcement Learning Based Dynamic Network Self-Optimization for Heterogenous Networks", PACRIM, 2009.

[6] Osianoh Glenn Aliu, Ali Imran et al. "A Survey of Self Organisation in Future Cellular Networks", IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 336-361, 2013.

[7] R. M. Khanafer, B. Solana, J. Triola, R. Barco, L. Moltsen, Z. Altman, and P. Lazaro, "Automated diagnosis for UMTS networks using bayesian network approach," IEEE Transactions on Vehicular Technology, vol. 57, no. 4, pp. 2451–2461, 2008.

[8] L. Badia, M. Boaretto, and M. Zorzi, "Neural self-organization for the packet scheduling in wireless networks," in Proc. WCNC Wireless Communications and Networking Conf. 2004 IEEE, vol. 3, 2004, pp. 1927–1932.

[9] J. Laiho, K. Raivio, P. Lehtimaki, K. Hatonen, and O. Simula, "Advanced analysis methods for 3G cellular networks," IEEE Transactions on Wireless Communications, vol. 4, no. 3, pp. 930–942, 2005.

[10] F. Bernardo, R. Agusti, J. Perez-Romero, and O. Sallent, "A self-organized spectrum assignment strategy in next generation OFDMA networks providing secondary spectrum access," in Proc. IEEE Int. Conf. Communications ICC '09, 2009, pp. 1–5.

[11] M. Dirani and Z. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in OFDMA cellular networks," in Proc. 8th Int Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) Symp, 2010, pp. 170–176.

[12] R. Razavi, S. Klein, and H. Claussen, "Self-optimization of capacity and coverage in LTE networks using a fuzzy reinforcement learning approach," in Proc. IEEE 21st Int Personal Indoor and Mobile Radio Communications (PIMRC) Symp, 2010, pp. 1865–1870.

[13] L. Giupponi, R. Agustí et al., "A novel approach for joint radio resource management based on fuzzy neural methodology", IEEE T. on vehicular technology, v.57, n.3, 2008.

[14] Jingyu Li, Jie Zeng et al. "Self-Optimization of Coverage and Capacity in LTE Networks Based on Central Control and Decentralized Fuzzy Q-Learning", IJDSN, Volume 2012, Art. 878595, Hindawi Publishing Corp. 2012

[15] S. Sutton, G. Barto, "Reinforcement Learning: An Introduction", Cambridge MA USA MIT Press, 1998.

[16] S. Whiteson, P. Stone, "Evolutionary Function Approximation for Reinforcement Learning", The Journal of Machine Learning Research, vol. 7, 2006.

[17] A. L. Stefan, M. Ramkumar, R. H. Nielsen, N. R. Prasad, R. Prasad, "A QoS aware reinforcement learning algorithm for macro-femto interference in dynamic environments" ICUMT, 2011.