Q-Learning for Policy Based SON Management in Wireless Access Networks

Tony Daher, Sana Ben Jemaa
Orange Labs
44 Avenue de la Republique 92320 Chatillon, France
Email:{tony.daher,sana.benjemaa}@orange.com

Laurent Decreusefond
Telecom ParisTech
23 avenue d'Italie, 75013 Paris, France
Email:laurent.decreusefond@mines-telecom.fr

***Abstract*—Self organized networks has been one of the first concrete implementations of autonomic network management concept. Currently, several Self-Organizing-Network (SON) functions are developed by Radio Access Network (RAN) vendors and already deployed in many networks all around the world. These functions have been designed independently to replace different operational tasks. The concern of making these functions work together in a coherent manner has been studied later in particular in SEMAFOUR project where a Policy Based SON Management (PBSM) framework has been proposed to holistically manage a SON enabled network, namely a network with several individual SON functions. Enriching this PBSM framework with cognition capability is the next step towards the realization of the initial promise of SON concept: a unique self-managed network that responds autonomously and efficiently to the operator high level requirements and objectives. This paper proposes a cognitive PBSM system that enhances the SON management decisions by learning from past experience using Q-learning approach. Our approach is evaluated by simulation on a SON enabled Long-Term Evolution Advanced (LTE-A) network with several SON functions. The paper shows that the decisions are enhanced during the learning process and discusses the implementation options of this solution.**

I. Introduction

The automation of (Radio Access Network) RAN management and operation has been brought to the field reality with the standardization of Self Organizing Networks (SON) by the 3rd Generation Partnership Project (3GPP). Today several autonomic functions, also called SON functions, are developed by RAN vendors and deployed in many networks all around the world. A SON function is basically a control loop that autonomously tunes RAN parameters to adapt the RAN to variations of the environment and of the traffic according to operator objectives. Hence time consuming manual optimization tasks can be replaced allowing the reduction of Operational Expenditure (OPEX). Typically, a SON function replaces a given operational task, and is designed independently from other SON functions. The concern of making these functions work together in a coherent manner came in a later step. Even though, the initial promise of the SON concept has been to provide the operator with a Self Organized Network that responds autonomously as a whole to the operator objectives. However, the actual implementation consists in a network with several SON functions operating independently, that we can qualify as a SON enabled network, but not as a Self-Organized Network. The main challenge is then to transform a SON enabled network into a Self-Organized Network.

To this end, a Policy Based SON Management (PBSM) system has been defined in the framework of Semafour European project [1]. The PBSM determines for each SON function the appropriate policy to follow so that all the SON functions achieve together the high level operator objectives. The SON functions are considered to be designed by RAN vendors in a proprietary manner, namely as almost black boxes. The PBSM uses the limited leverage given by the vendor, *i.e.* the SON control parameters, to control the behavior of the SON functions. This paper is a first investigation on cognitive SON management. The objective of the paper is to assess the feasibility and the performance of a Q-learning enabled PBSM. The first section explains in more details the main ideas and challenges in global SON management and motivates the need for cognition. The next section introduces Markov Decision Processes (MDPs) and Reinforcement Learning (RL), particularly Q-learning algorithm that we will apply later to PBSM decision making. Section 4 presents the Q-learning PBSM use case considered in this work and section 5 discusses the simulation results. The considered framework is based on Long-Term Evolution Advanced (LTE-A) SON features and it will be extended in the future to 5G related features. Section 6 concludes the paper and explains the next steps.

II. Towards Cognitive SON Management

The automation of operational RAN management tasks, initiated by the definition of SON functions, was first intended to reduce operational costs. The experience in the last few years has shown that enabling a network with SON enhances the operational efficiency, reflected by OPEX gain, but it also enhances the network performances compared to a manually operated network. Obviously, SONs have gained the operators' trust and SON solutions are deployed in several countries all over the world. On the one hand, distributed SON solutions provided by the RAN vendors as part of the RAN release are becoming more commonly adopted by the operators. On the other hand, SON solutions that can be deployed in a centralized manner (so called centralized SON), in the Network Management System (NMS) of the operator independently from the RAN vendor network, are gaining momentum. It has been shown in the Infonetics report [2] that the deployment

978-3-901882-89-0 @2017 IFIP

of SON solutions as well as the use of advanced network optimization tools has kept the OPEX at the same level even while the network is required to handle more radio access technologies (2G, 3G, 4G), more traffic, and a higher diversity of users and services. As the network complexity is drastically increasing, due to network heterogeneity, the traffic growth and the high user expectation, the automation of RAN operation has become a necessity. Hence, automation is gaining momentum, and it is commonly agreed today that SON is required for an efficient RAN operation. It is also expected to fully realize the initial promise of SON concept: to provide the operator with a Self Organized Network, capable of reaching as a whole the operator high level objectives. These high level objectives reflect the operator strategy in terms of e.g. network capacity, user satisfaction... As stated before, PBSM framework defined in Semafour project aims at building a Self Organized Network out of a RAN enabled with different SON functions. This PBSM system is supposed to be independent from the SON functions, *i.e.*, considering SON functions as black boxes and acting only on the SON configuration parameters. These parameters are supposed to be provided by RAN vendors to orient the behavior of the SON, whereas the internal SON optimization algorithm remains proprietary [1].

The PBSM approach proposed in Semafour relies on SON function models, that consists in modeling for each SON function the impact of the SON Configuration parameter Values (SCV) on the targeted Key Performance Indicators (KPIs). This modeling is performed through exhaustive simulations [3] and enhanced through network feedback [4]. We propose in this paper a different approach which does not model the individual SON functions behavior, but learns the most efficient global decision to be made for the the whole SON system directly from the real network, hence enabling PBSM with cognition capabilities.

Empowering radio components with cognition capability is a concept which has been first introduced in the radio communication field by Mitola III [5] since 1999 under the *Cognitive Radio* paradigm. The cognition cycle involves Observe, Orient, Plan, Learn, Decide, and Act phases. Cognitive radios are radio components that mimic the human cognitive behavior. They are aware of the user/network needs and can adapt their functioning accordingly, also learning from past experience. In this work we focus on the learning phase of the cognitive cycle. For this purpose, we propose a RL PBSM, and particularly, a Q-learning based PBSM. In fact, what makes the the Q-learning an interesting method in our case, is that it does not require a knowledge of the environment, and learn directly from experience. RL has already been studied for several SON functions. To name a few, load balancing and traffic steering [6], interference mitigation techniques [7], power saving [8] etc. It was shown that SON algorithms based on RL lead to higher SON efficiency, hence improving network performance related to the objectives of the considered functionality. RL turned out to be also a reasonable solution for SON coordination in [9] and [10].

This paper presents the first results on learning empowered PBSM, as a component of a global cognitive RAN management system, to pave the way towards building cognitive management for the next generation 5G RAN. Extensive research is currently ongoing in several 5G-PPP phase 1 projects, for example, 5G-NORMA [11], METIS II [12], and FANTASTIC-5G [13], to define the basics of the Radio Access Network (RAN) architecture, as well as new data link and physical layer concepts for the air interface. In addition, the corresponding standardization activities have started in 3GPP [14]. 5G RAN will be enriched by a plethora of new features and technologies such as Device-to-Device (D2D), Vehicular-to-Anything (V2X), Massive-MIMO (M-MIMO), Internet of Things (IoT), Machine Type Communication or new spectrum. 5G will also be vertical driven [15] with a multitude of different requirements corresponding to the specific needs of the vertical industries (e.g. Automotive, e-Health, Multimedia and entertainment,...) in addition to the classic operator objectives (e.g. network capacity and operational efficiency). 5G is characterized by an unprecedented level of flexibility. But high flexibility translates into a tremendous number of possible network configurations. The role of the management system is to choose for each state the optimal configuration. In this context, the management system should be intelligent and continuously enhance its decisions by learning from its past experience.

In the following section we introduce MDPs and RL, particularly the Q-learning algorithm that we will apply later to PBSM decision making.

III. Markov Decision Processes and Reinforcement Learning

A MDP is a mathematical framework that permits to model a decision making process. It is defined by the tuple $M = (S, A, p, r)$ where:

- S is the set of states of the system
- A is the set of possible action
- p is the state transition probability $p : S \times A \rightarrow Prob(S)$ (For each state and action we specify a probability distribution over next states)
- r is the reward function $r : S \times A \rightarrow \mathbf{R}$

An important assumption in the MDP framework is that the current state s_t and action a_t are a sufficient statistics for the next state s_{t+1}. This assumption translates through the following transition model:

$$P_{ss'}{}^{a} = Pr\{s_{t+1} = s' | s_t = s, a_t = s\} \quad (1)$$

Another important assumption is the stationarity i.e. the rewards and transitions do not change over time. The objective of MDPs is to find a policy for the decision maker or agent. A policy π is a mapping from the states space to actions space:

$$\pi : S \rightarrow A \quad (2)$$

In other words, a policy tells the agent what action to take when in a certain state s. An optimal policy is a policy π^* that maximizes a long term reward that is defined as:

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \tag{3}$$

Where $\gamma < 1$ is a discount factor that reflects the weight of future rewards at time t. In other words the reward R_t is a cumulative function of the instantaneous rewards perceived at each iteration.

One way to find the optimal policy is through RL. As defined in [16], RL is learning what to do, how to map situations to actions, so as to maximize a numerical reward signal. The process is portrayed in figure 1:

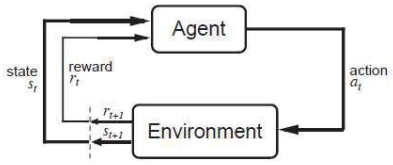

Fig. 1: Reinforcement Learning [16]

A RL process is an MDP if it satisfies the assumptions mentioned before (1) and (2). RL algorithms are based on estimates of the value and the action-value of a state, under a policy π. The value of a state reflects how good it is to be in a certain state in terms of the expected rewards in the future iterations, following a policy π. It is defined as:

$$V^{\pi}(s) = \mathbf{E}_{\pi}\{R_t | s_t = s\} = \mathbf{E}_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\} \tag{4}$$

The optimal value function is hence:

$$V^*(s) = \max_{\pi} V^{\pi}(s) \tag{5}$$

Respectively, the action value (Q-function) reflects how good it is to be in an action state pair, in terms of the expected rewards in the future iterations, following a policy π:

$$\begin{aligned} Q^{\pi}(s,a) &= \mathbf{E}_{\pi}\{R_t | s_t = s, a_t = a\} \\ &= \mathbf{E}_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\} \end{aligned} \tag{6}$$

And the optimal action value function is:

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{7}$$

There exists several methods to solve the RL problem. In our work, we chose the temporal difference approach and more precisely the Q-learning [17]. As stated in section II, Q-learning has the property of learning the best policy without any a priori knowledge of its environment, making it thus a suitable approach for our problem. The algorithm is described in the following.

Q-learning Algorithm

Initialize Q(s,a) arbitrarily
Initialize s
for t=1,...,T
- pick action a for state s according to ϵ-greedy policy:

$$a = \begin{cases} \underset{a \in A}{argmax}\{Q(s,a)\} & \text{with probability } 1-\epsilon \\ rand(A) & \text{with probability } \epsilon \end{cases} \tag{8}$$

- observe new state s'
- observe perceived reward $r(s,a)$
- update Q function as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)] \tag{9}$$

Until convergence

It can be shown that this algorithm converges to the optimal policy Q^* with probability 1 so long as all actions are repeatedly sampled in all states and if the step size α is decremented properly [17].

In the next section we introduce our PBSM based on Q-learning. We then present and analyze our results.

IV. PBSM Based on Q-Learning

Q-learning algorithm can be used to find an optimal policy for the PBSM. We consider a section of a heterogeneous LTE(-A) mobile network of N cells, where the following decentralized SON functions are deployed:

- **Mobility Load Balancing (MLB)**: Deployed on each macro cell. It iteratively tunes the cell individual offset parameter (CIO) between each neighbor macro cell pairs in order to balance the load between macro cells.
- **Cell Range Expansion (CRE)**: This function is similar to the MLB, excepts that it tunes the CIO of pico cells in order to balance the load between the slave pico cell and the macro cell where it is deployed.
- **Enhanced Inter Cell Interference Coordination (eICIC)**: Protects pico cell edge users that are attached to a pico cell because of CRE's offset. These users suffer high downlink interference from the macro cell (because the power received from the macro is higher than the one received from the pico). The eICIC protects these edge users by tuning the number of Almost Blank Sub-frames (ABS) in a frame of the macro cell (in LTE, a frame consists of 10 sub-frames, each having a duration of 1ms). ABS include only control and cell-specific reference signals, transmitted at reduced power. The pico cell edge users will take advantage of these sub-frames, either by the use of a proportional fairness scheduler or by informing the pico cell of the ABS pattern, to transmit in better channel conditions.

Each function can be configured through different SCV sets. Let C be the set of all possible SCV sets combinations in the network for all the deployed SON functions. The PBSM is hence faced with $|C|$ different possible configurations in the network. Let $l_n \in [0,1]$ be the load indicator of a cell $n \in N$.

Furthermore we consider 3 load states for each cell: low load, mid load and high load. A cell n is in a load state L_n according to the following rules:

$$L_n = \begin{cases} \text{low load} & \text{if } l_n < T_{low} \\ \text{mid load} & \text{if } l_n \in [T_{low}, T_{high}] \\ \text{high load} & \text{if } l_n > T_{high} \end{cases} \quad (10)$$

Where T_{low} and T_{high} are load thresholds thoroughly defined by the users through the observation of the system's dynamics. Now we define the considered MDP:

- **The set of actions**: The objective is to chose which configuration maximizes a reward function. Hence the set of actions is the set of all SCV combinations C.
- **State space** $S = C \times L^N$: A state is defined by the applied configuration in the overall SON deployed functions and the Load distribution in the network.
- **Reward**: The considered KPIs for the reward are:
  - $LD_{i,c,t}$ is the load of cell i
  - $T_{i,c,t}$ is the average user throughput in cell i
  - ${T_{i,c,t}}^e$ is the average pico cell edge user throughput of pico cell i

  Moreover, $\hat{LD}_{c,t}$, $\hat{T}_{c,t}$ and $\hat{T}^e_{c,t}$ are respectively the average load, average user throughput and average pico cell edge user throughput in the whole considered network section. $w_i \in [0, 1]$ are weights that sum up to 1. They reflect the operator's priority to optimize the corresponding KPI. The instantaneous reward is hence defined as:

$$r_t = \omega_1(1 - \sigma_t) + \omega_2\hat{T}_t + \omega_3\hat{T}^e_t \quad (11)$$

Where the load variance σ_t is:

$$\sigma_{c,t} = \frac{\sum_{i=0}^{|N|} (LD_{i,c,t} - \hat{LD}_{c,t})^2}{|N|}$$

The objective is to find a policy that maximizes the long term reward defined in equations 3 and 11 i.e. the best SCV sets combination $c \in C$ for each state $s \in S$. To do so we apply the Q-learning algorithm introduced above.

We consider the scenario represented in figure 2. The algorithm is tested on an LTE(-A) system level simulator based on the 3GPP specifications. It is a semi-dynamic simulator that performs correlated snapshots with a time resolution of 1 second. We take into account path loss and shadowing. Users arrive in the network according to a Poisson arrival and have pre-defined mobility parameters. We consider only down-link: users arrive and request to download a file. They are either successfully served and leave the network or they are dropped. The traffic is considered stationary, unequally distributed between the macro cells. High loaded macro cells (Macro 2 & 3) contain each an extra traffic hotspot served by a pico cell (Pico 2 & 3). In this network section there are 3 macro cell borders. Load balancing is only required on 2 of them (between Macros 1 and 2 and Macros 1 and 3), we have 2 CIO pairs to be tuned. We hence consider 2 MLB for the considered borders (the MLB between the equally loaded cells (Macro 2 & 3) is considered to be turned off), 1 CRE and 1 eICIC deployed on each of the pico cells. We have a total of 6 instances of 3 different SON functionalities. Each function can be configured with SCV sets so as to achieve certain performances. Among the many various possible configurations, the ones we consider in our study are summed up in table 1.

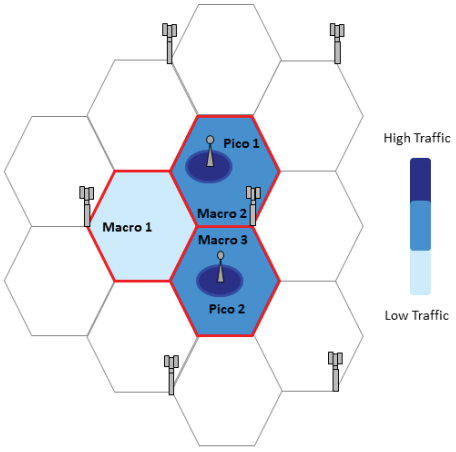

Fig. 2: Heterogeneous Network Model With Unbalanced Traffic Distribution

SON	SCV Sets
MLB	Off: Function is turned off SCV1: Soft configuration SCV2: Aggressive configuration
CRE	SCV1 Soft configuration SCV2 Aggressive configuration
eICIC	Off Function is turned off SCV1 Reduces throughput gap between macro and pico cell edge users

TABLE I: SCV sets behavior description

By soft configuration we mean a configuration that reduces the load gap (without allowing the CIO of the cells to reach high levels) and by aggressive we designate a configuration that seeks to equilibrate the load as much as possible (by allowing higher levels of CIO). We have thus 144 possible combinations of SCV sets in this section of the network i.e. $|C| = 144$ $((3 \times 3) \times (2 \times 2) \times (2 \times 2))$. In order to reduce the complexity of the algorithm, we consider intuitively 2 cell classes: macro cell class and pico cell class. So the SCV combination becomes per cell class and not per cell (i.e. SON functions on the macro cells have the same configuration. Same for the functions on the pico cells), leaving us with only 12 possible SON configurations described in Table II.

Figure 3 represents a hierarchical view of the system: the RL agent applies a configuration c to the SON instances, who consequently apply parameter changes in the network

Combination	Description
1	MLB: Off , CRE: SCV1, eICIC: Off
2	MLB: SCV1, CRE: SCV2, eICIC: Off
3	MLB: SCV2, CRE: SCV1, eICIC: Off
4	MLB: Off , CRE: SCV2, eICIC: Off
5	MLB: SCV1, CRE: SCV1, eICIC: Off
6	MLB: SCV2, CRE: SCV2, eICIC: Off
7	MLB: Off , CRE: SCV1, eICIC: On
8	MLB: SCV1, CRE: SCV2, eICIC: On
9	MLB: SCV2, CRE: SCV1, eICIC: On
10	MLB: Off , CRE: SCV2, eICIC: On
11	MLB: SCV1, CRE: SCV1, eICIC: On
12	MLB: SCV2, CRE: SCV2, eICIC: On

TABLE II

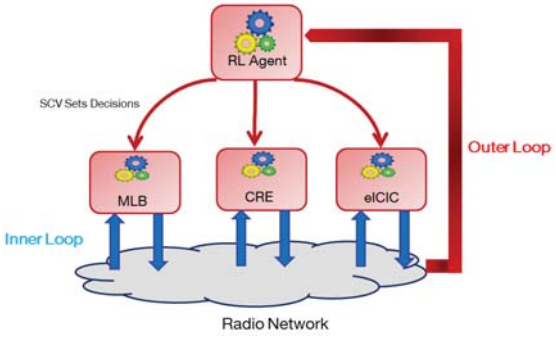

Fig. 3: PBSM based on RL

dynamically according to their configuration and the network status (provided through feedback information by the inner loops). After a certain time (sufficient for the SON functions to converge), the RL agent in its turn receives network KPIs through the outer loop, evaluates the instantaneous reward, then applies the Q-learning algorithm and takes a new decision (configuration) to be applied, and so on.

V. Simulation Results

After running Q-learning for the described scenario, we test the optimal policy provided by the algorithm, which is a mapping between each visited state and the best corresponding action (configuration). Note that the output policy is obtained through learning, by running the Q-learning algorithm, as described in section III. This is the exploration or learning phase. The output policy is then evaluated in the exploitation phase (no exploration). Figure 4 represents the action selection statistics of the output policy during the exploitation phase. It shows that only 2 configurations are picked according to the optimal policy. One of them (configuration 9) is picked almost all the time. This means that the same configuration is optimal for all the visited system states. This is validated in figure 5, where we test successively the 12 configurations. Note that configurations 8 and 9 have very close rewards. This

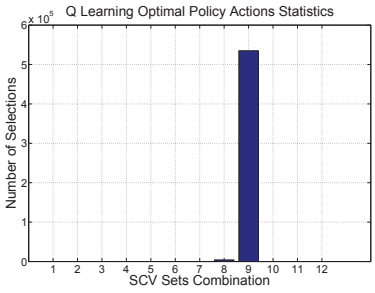

Fig. 4: SCV sets combination selection statistics in exploration phase

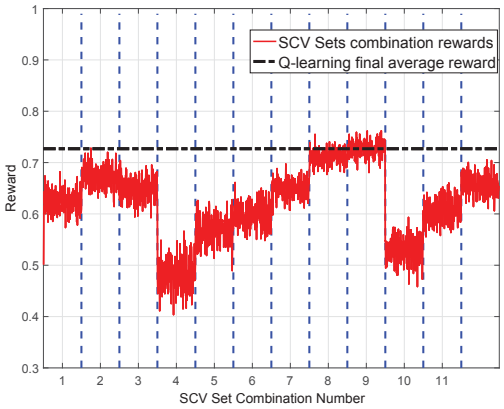

Fig. 5: SCV sets combination exhaustive search

explains why figure 4 shows that configuration 8 is picked, although very rarely. We conclude that Q-learning finds the optimal action, and the policy indeed succeeds in maximizing the perceived reward.

Concerning the convergence time, from figure 6 we notice that the algorithm converges after a number of iterations superior to 150000, which is reasonable for Q learning algorithms. On another hand, figure 3 shows that the learning is performed through an outer loop, that is much slower than the inner loops of the individual SONs. Indeed, Learning through the outer loop is a slow process, because the agent has to wait for all the SON functions to converge before collecting the KPIs. This

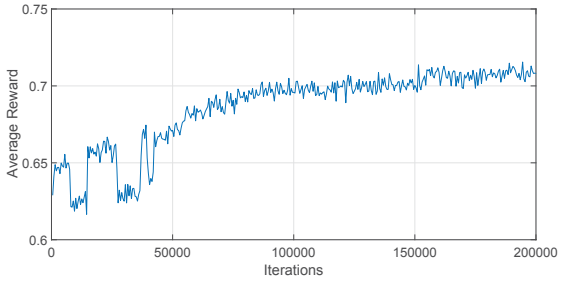

Fig. 6: Average perceived reward in the learning phase

leads to a quite considerable convergence time. For example, if we consider the RL iteration to be around 30min i.e. all the SONs converge in no more than 30min, then the process takes in the order of several hundreds of days to converge (real network time). Because of such convergence time, it would be unreasonable to learn from the real network. A solution would be to perform offline pre-training before starting to learn from the real network. This way, the algorithm does not have to learn from scratch, hence improving the convergence.

Furthermore, the considered approach can be extended, with the same logic, to bigger network sections with more SON functions and configurations. In that case, the state and action space will increase considerably, hence increasing the convergence time. Several solutions can be envisaged to tackle this problem e.g. by reducing the states space through fuzzy RL algorithms [18], or by distributing the learning process using a cooperative multi-agent RL approach [19].

VI. Conclusion and Further Works

In this paper we have introduced a PBSM based on RL. More precisely we have used the Q-learning algorithm to find an optimal policy. We have shown that after a training phase, the algorithm succeeds in finding the optimal policy. The PBSM hence configures the deployed SONs in the network according to this policy. This approach does not require any models and leads consequently to more accurate results. We have also discussed the possibilities of extending this approach to larger network sections with more SON functions. However, Q-learning requires a considerable training time. Plus, in order to find the optimal policy, the system explores many actions during the learning phase, which may lead to performance degradation if this phase is applied on the real network. This issue is known in the RL framework as the exploration/exploitation dilemma.

Besides, since it looks like the optimal policy consists in one optimal action in all states (figure 4), then our problem reduces to a special case of MDPs where there is only a single state. Thus, there might be a possibility to study and analyze the problem using a different approach, that still lies in the RL framework: the Multi-Armed Bandits (MAB) [20].

Finally, the extension of this cognitive PBSM approach for 5G networks and use cases is also an important axis for our future research.

References

[1] SEMAFOUR project web page http://fp7-semafour.eu/.

[2] "Son and optimization software annual worldwide and regional market size and forecasts," *Infonetics Research*, November 2, 2015.

[3] S. Hahn and T. Kürner, "Managing and altering mobile radio networks by using son function performance models," in *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*. IEEE, 2014, pp. 214–218.

[4] S. Lohmỳller, L. C. Schmelz, and S. Hahn, "Adaptive son management using kpi measurements," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 2016, pp. 625–631.

[5] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE personal communications*, vol. 6, no. 4, pp. 13–18, 1999.

[6] P. Muñoz, R. Barco, J. M. Ruiz-Avilés, I. de la Bandera, and A. Aguilar, "Fuzzy rule-based reinforcement learning for load balancing techniques in enterprise lte femtocells," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1962–1973, 2013.

[7] M. Dirani and Z. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in ofdma cellular networks," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*. IEEE, 2010, pp. 170–176.

[8] A. De Domenico, V. Savin, D. Ktenas, and A. Maeder, "Backhaul-aware small cell dtx based on fuzzy q-learning in heterogeneous cellular networks," in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[9] O. Iacoboaiea, B. Sayrac, S. B. Jemaa, and P. Bianchi, "Son coordination for parameter conflict resolution: A reinforcement learning framework," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2014 IEEE*. IEEE, 2014, pp. 196–201.

[10] O.-C. Iacoboaiea, B. Sayrac, S. B. Jemaa, and P. Bianchi, "Son coordination in heterogeneous networks: A reinforcement learning framework." 2015.

[11] 5G-NORMA project web page https://5gnorma.5g-ppp.eu/.

[12] METIS II project web page https://metis-ii.5g-ppp.eu/.

[13] FANTASTIC-5G project web page http://fantastic5g.eu/.

[14] Tech. Rep., 3GPP, " Study on New Services and Markets Technology Enablers" 3rd Generation Partnership Project (3GPP), TR 22.891, V14.0.0, 2016.

[15] "View on 5g architecture," 5G PPP Architecture Working Group, Tech. Rep., July 2016.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[17] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[18] H. R. Berenji, "Fuzzy q-learning: a new approach for fuzzy dynamic programming," in *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*. IEEE, 1994, pp. 486–491.

[19] A. Galindo-Serrano and L. Giupponi, "Distributed q-learning for aggregated interference control in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1823–1834, 2010.

[20] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.