Learning Radio Resource Management in 5G Networks: Framework, Opportunities and Challenges

Francesco Davide Calabrese, Li Wang, Eunhanna Ghadimi, Gunnar Peters, Pablo Soldati
Huawei Technologies Sweden AB, System Algorithms Lab, Kista, Sweden
e-mail: {firstname.lastname}@huawei.com

Abstract

With the fifth generation (5G) of mobile broadband systems, Radio Resources Management (RRM) will reach unprecedented levels of complexity. To cope with the higher complexity of RRM functionalities, while retaining the fast execution required in 5G, this manuscript presents a lean 5G RRM architecture that capitalizes on the most recent advances in the field of machine learning in combination with the large amount of data readily available in the network from measurements and system observations. The result is a general-purpose learning framework capable of generating algorithms specialized to RRM functionalities directly from data gathered in the network. The potential of this approach is verified in three study cases and future directions on applications of machine learning to RRM are discussed.

Index Terms

machine learning, radio resource management, 5G, reinforcement learning, algorithm.

I. INTRODUCTION

Radio Resource Management (RRM) in state-of-the-art Radio Access Networks (RANs) comprises a maze of interconnected algorithms, each tailored and optimized for a specific RRM task. With RAN technology continuously evolving (see e.g., the 3GPP Long-Term Evolution (LTE) standard), adding new system features is typically handled by designing new (ad-hoc) RRM algorithms or redesigning existing ones. While this approach has stimulated a rapid development of system features at early stages of standardization, over ten years down the road of system evolution has brought an increasingly fragmented and heterogeneous RRM architecture founded on an ever growing number of parameters. Recently, a set of techniques, known as Self-Organizing Network (SON), were suggested to automate the parameter tuning \[4\]. While intended to address
the growth of control parameters, SON ultimately resulted in an additional layer of rules (and parameters) on top of the legacy design.

The next generation of mobile broadband (5G) systems will gather a variety of new services and features with stringent and rather diverse requirements in terms of peak data rate, reliability, latency and coverage [1]. High data rate services, such as enhanced Mobile Broadband (eMBB), will require seamlessly multi-connectivity across different Radio Access Technologies (RATs) operating over a wide range of frequency bands, from sub-6GHz to millimeter waves (mm-Wave). Ultra-Reliable Low-Latency Communications (URLLC) and massive Machine Type Communications (mMTC), targeting a wide range of services related to the Internet-of-Things (IoT), as well as vehicular communication (e.g., to support autonomous cars), will pose severe requirements to RRM functionalities in terms of latency and reliability.

To meet such wide spectrum of requirements, 5G will integrate a variety of new technology components, e.g. massive MIMO, mm-Wave communication, network slicing, multi-RAT multi-connectivity etc., into an increasingly complex and heterogeneous RAN [1]–[3]. Thus, optimizing RRM to fully cater to 5G systems will come with an unprecedented computational complexity and stringent execution time constraints.

The advent of 5G, however, presents not just an opportunity to deploy cutting-edge communication technology but also a chance for departing from the traditional RRM design. In particular, we propose a logical RRM architecture that capitalizes on the most recent advances in the Machine Learning (ML) field along with the extensive amount of data available in the network from measurements and system observations. In recent years, in fact, ML has experienced an extraordinary growth thanks to new techniques, more powerful computing tools and infrastructures capable of handling larger amounts of data in shorter time. The high parallelization characteristics of some ML techniques make them particularly suitable to address the higher complexity of 5G RRM functionalities, while also enabling the faster execution required in 5G.

Among these techniques, Reinforcement Learning (RL) has emerged as particularly suitable to solve complex control problems, such as RRM in RAN. We therefore propose a general-purpose learning framework that uses RL to generate algorithms specialized to RRM functionalities directly from data gathered in the network. Although RL has been tried successfully in different individual RRM tasks [5]–[7], the application of RL to complex systems (as the 5G RAN), poses several challenges. While highlighting such challenges, we suggest efficient ways to resolve them and demonstrate the effectiveness of the framework in a number of case studies.
II. A BRIEF OVERVIEW OF LEARNING METHODS

Machine Learning deals with the task of inferring a function (or pattern) from a set of noisy data (known as the training set) generated by an unknown true function. The ML branches of interest, in this paper, are supervised learning and reinforcement learning.

Supervised learning infers a function from a set of data pairs comprising an input and a desired (labeled) output, provided by a supervisor. Since the data samples are noisy and the training set may only partly represent the complete dataset, there exist techniques to mitigate the danger of fitting the noise, e.g. regularization and k-fold cross-validation [8].

A supervised learning method, which experiences renewed interest, is Artificial Neural Networks (ANNs). While existing in a variety of architectures, ANNs are generic function approximators which can be tailored to the task at hand through the right selection of their weights. Training an ANN consists in gradually adjusting its weights in the direction that minimizes the expected error loss function between the function represented by the ANN and the actual noisy data samples produced by the original true function. The application of this gradient rule, specialized to ANNs, is known as back-propagation [9].

Reinforcement learning deals with how a software agent learns to behave in a given environment to achieve a given objective, i.e., maximizing a form of reward. As such, it is particularly suitable for control problems, such as those arising in RRM. Hereafter we consider a model-free setup, where the problem is described uniquely in terms of three components: state, action and reward.

The state $s$ is a tuple of values, known as features, that describes the agent in relation to the environment in a way that is relevant to the problem at hand. The action $a$ represents the change that the agent applies to the environment to maximize the given reward. The reward $r$ is a multi-objective scalar function which numerically expresses the agent’s purpose. The interaction, over time, of the agent with the environment is captured by a set of tuples of the form $(s_t, a_t, r_{t+1}, s_{t+1})$, where $t$ is a discrete time counter, describing the transition from a state to the next one as a consequence of applying actions to the environment and receiving rewards.

The first objective of RL is to extract, from the transitions set, a policy $\pi$ that, given a state, returns the action to take in order to maximize the long-term cumulative reward. The RL algorithm thus maps the rewards to the actions, possibly taken far back in time. This notion is known as credit-assignment [9].
The second objective of an RL framework is to rapidly bring the agent from a tabula rasa state, where he does not know how to act, to a condition where it acts as close to optimality as possible. Making as few mistakes as possible in the path to a quasi-optimal behavior is known as regret minimization, a notion closely related to the topic of trading off exploration of the environment (to sample unseen parts of the state-action space at the cost of not choosing the best known action for a state) with exploitation of the knowledge accumulated so far (to maximize the reward at the cost of not trying a new potentially better action). This gradual transition from a pure exploration strategy to an exploitation strategy can be implemented using a variety of techniques, e.g. $\epsilon$-greedy algorithm [9].

III. RL FOR 5G RRM: CHALLENGES AND OPPORTUNITIES

RRM in modern RANs is essentially a large control problem broken down into a set of smaller problems, each pertinent to controlling a set of radio parameters to optimize suitable combinations of Key Performance Indicators (KPIs). Thus, RL offers a powerful alternative to design RRM algorithms in complex and highly dynamic systems such as future 5G RANs.

A. Challenges

Basic formulations of RL have recently been suggested to address specific RRM problems in the wireless communication literature, see e.g. [5]–[7], where the learning algorithm exploits tables to store a running average of a learned functional value for each state-action pair. As the number of possible states and actions grows, the number of data samples required to train this type of algorithm becomes prohibitively large. Moreover, the accumulated experience of such basic RL algorithms cannot be generalized across states that are similar to each other.

The first challenge faced by a control problem of this kind is its large dimensionality. The variety of conditions in the network, paired with the number of configurable parameters, results in an extremely large state-action space. This issue becomes particularly noticeable in 5G systems due to the larger cardinality of the RRM decision domains, e.g. due to massive number of connected devices, larger number of operating bands with wider bandwidths, flexible sub-frame lengths (from 1 ms down to 125 $\mu$s) and sub-carrier spacing, [1]–[3]. Moreover, the stringent requirements for 5G latency-critical applications reduce the execution time of RRM functionalities, e.g. for scheduling resources, to less than 100 $\mu$s, thus increasing the complexity of the task even further.
A crucial challenge for applying RL to RANs is the partial observability of the network state available to the agent, as provided by local measurements taken by users in a cell or by the eNodeB(s) (eNB) controlled by the agent. The degree of observability is also affected by the particular radio technology. For instance, with Massive MIMO and mm-Wave, the baseband chip-set of an eNB might be capable of controlling fewer cells (hence the limited observability) compared to 4G systems, due to the additional complexity brought in by larger antenna arrays and wider system bandwidths.

Additionally, the coexistence of multiple RATs and the densification of RAN expected in the 5G networks results in a dense multi-agent system that poses stronger challenges in terms of coordination. This requires to extend the single-agent RL solutions to a multi-agent setting by integrating inter-agent coordination mechanisms that enable to exchange information about the observed (local) state and reward.

Finally, a challenge of practical nature is to control the learning process so as to prevent a prolonged degradation of the system performance.

B. Opportunities

While the challenges are noticeable, 5G comes also with opportunities for a radically different RRM architecture. One such opportunity is the increased variety of ways in which the radio environment is sensed (e.g., through massive antenna arrays) and the extensive amount of data (e.g., by user-centric uplink beaconing) characterizing the RAN environment. Such data is made available either in raw form, such as signatures of received power from reference signals at multiple antennae, or in processed forms, such as expressions of signal to noise ratio, bit and block error rates and other KPIs typically measured in the network. Additional knowledge is available to the network related to user mobility and traffic patterns, or more generally to how/when/what the user does or requests in the network.

Another opportunity is presented by the constantly growing computational capabilities of the network which, paired with the growing availability of data, offers the natural setup for the application of machine learning techniques, and particularly RL, in the RRM context.

Therefore, the standardization of 5G RAN offers the opportunity to rethink the design of the RRM by shifting the engineering effort from designing numerous, single purpose, RRM algorithms – as done in legacy RRM architectures – to designing and improving a single, general purpose, learning framework, capable of generating control policies specialized for individual
Figure 1: Logical RRM architecture for RL applied to RANs with a centralized RL trainer node and the distributed agent nodes.

RRM tasks executed by network entities. An additional benefit, which further motivates the adoption of this approach, is that improving the single learning framework results in improved performance across all the underlying RRM algorithms generated by the framework.

IV. LEARNING ARCHITECTURE FOR 5G RRM

To embrace the opportunities made available by 5G, we propose a logical architecture, cf. Figure 1, to enable an advanced learning framework for 5G RRM. This architecture defines a logical layer overlying the 5G technology components for which provides solutions tailored to the corresponding RRM functionalities.

Mapping Figure 1 to ML parlance, the RAN represents the environment. The state, in its entirety, can be represented by a set of features characterizing the agent in relation to the network, e.g., the type and capabilities of terminals, traffic type, the type and number of cells, different radio measurements and KPIs (e.g., cell coverage, cell capacity, packet delay) etc. In practice, each RRM functionality is associated with a more compact state represented by a subset of relevant features. The actions specific to an RRM functionality (e.g., power control) are represented by parameters adjustments (e.g., power-up, power-down, power-hold). The reward may represent a function (not necessarily linear) of conventional KPIs or system requirements used in wireless networking, e.g. a harmonic mean of the perceived user rates can balance between coverage and capacity (cf. [15]).
The trainer and the agent are the key (logical) components of the architecture. The trainer applies a single learning algorithm to generate control policies specialized for different RRM functionalities. The agent executes the policies issued by the trainer to interact with the network. While these entities are traditionally co-located, to overcome the design challenges identified in Section III, we propose that the trainer, where policies are derived offline, and the agent, where policies are executed in real-time, to reside in different network entities, as shown in Figure 1.

The major achievement of the trainer-agent functional split is to cope with the complexity and large dimensionality of 5G RRM problem, while enabling agile solutions executable within the stringent time constraints posed by 5G requirements. Rather than avoiding the complexity by simplifying the algorithms at the expense of performance, our architecture shifts the complexity associated to computations and storage of transitional data to a centralized node (the trainer) and retains all the intelligence and fast execution in the distributed nodes (the agents). This further encourages techniques for greater data efficiency (e.g., transfer learning; cf. Sec. VI). The trainer could therefore reside in a data center, e.g., a server farm, controlling a wide network area, or, in a smaller scale, a cloud RAN.

The agents, instead, become light-weight nodes (e.g., eNBs) that execute RRM functionalities in real-time, e.g., by making a series of matrix multiplications (possibly with dedicated hardware) when the policy is represented by ANNs, rather than evaluating online a potentially complex algorithm. Thus, the agent, may reside in an eNB controlling a cluster of cells or at higher hierarchical RAN levels to control wider network areas, e.g., at a Mobility Management Entity (MME) etc.

V. LEARNING FRAMEWORK FOR 5G RRM

The core of the proposed learning framework is a general-purpose algorithm to produce control policies specialized to individual RRM functionalities. The algorithm capitalizes on three major components: Neural-Fitted Q-Iteration (NFQ), ensemble and transfer learning. While future advances in the ML field may bring new algorithm components without altering the architecture, our current proposal answers several challenges listed in Section III-A. Hereafter, we describe the design of such algorithm for offline policy generation at the trainer and discuss real-time execution at the agent.
A. RRM trainer design

Figure 2a shows a systemization of the functional blocks of the trainer. Each agent comprises a functional RRM module wherein the latest generated policies for one or more RRM functionalities controlled by the agent are maintained. A training module is used to generate or update the RRM policies based on the general-purpose training algorithm described next.

To derive RRM policies at the trainer, we adopt a Q-learning approach [10]—an attractive RL algorithm—which aims at learning an action-value function (i.e., the Q-function) to estimate a long-term reward.

Given a policy \( \pi \), the Q-function is the expected utility of taking an action \( a \) in a given state...
$s$ and following the policy $\pi$ afterwards. That is

$$Q_\pi(s_t, a_t) = \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots],$$

where $\gamma \in [0, 1)$ is a discounting factor causing the value of rewards to decay exponentially over time, thus making the optimization horizon finite. While small values of $\gamma$ adjust the preference for short-term rewards, larger values allows to optimize for longer-term rewards thus improving the KPIs of the system.

Maximizing the long-term reward is equivalent to searching a policy that maximizes the Q-function. Given a proper definition of the state, most control problems (including those arising in RRM) can be modeled as Markov decision processes, where the optimum Q-function can be found by solving the Bellman equation \cite{bellman}:

$$Q(s_t, a_t) = r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}).$$

Contrary to the traditional approach of relying on prohibitively large lookup tables of Q-values associated with each state-action pair (see, e.g., \cite{garg2021}, \cite{garg2021a}), we apply Q-learning via a functional approximation of the Q-function \cite{bellman} to efficiently deal with high dimensionality issues of RRM problems (cf. Section III-A). Specifically, instead of updating the Q-values found at individual entries of the lookup table, the parameters of a functional approximation are optimized so as to minimize the error between the predicted Q-value $Q(s_t, a_t)$ and the target Q-value $r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$. This design choice enables a better generalization across states, including previously unseen states, thereby accelerating the learning.

Although different functional approximators, e.g., ANNs, decision forests, etc., are suitable for the proposed learning framework, we design the training algorithm based on ANNs due to their excellent generalization capabilities which makes them an effective choice for non-linear function approximation.

There exist a variety of ways in which ANNs can be used in an RL framework. Arguably, Neural-Fitted Q Iteration algorithm \cite{schulman2015}–\cite{schulman2015a} is a powerful tool combining ANNs and Q-learning. Figure 2b illustrates the NFQ policy generation procedure comprising: an outer loop, i.e., the Q-iterations, where the inputs $(s_t, a_t)$ and the corresponding outputs $(r_{t+1} + \gamma \max_{a_{t+1}} Q_{k-1}(s_{t+1}, a_{t+1}))$ are generated based on the current ANN ($Q_{k-1}$); and an inner loop, i.e., the training iterations, where the weights of the ANN are adjusted to fit the newly generated input-output pairs ($Q_{k-1} \rightarrow Q_k$). The nature of ANNs enables to exploit the massively parallel computation model of modern hardware (such as GPUs for general-purpose computing) to accelerate the training phase.
To improve knowledge generalization we make use of ensemble learning where multiple ANNs with distinct structures and configurations (e.g., number of layers, neurons per layer, etc.) are independently trained to learn the same (unknown) Q-function (see Figure 2b). As long as each ANN in the ensemble is outperforming a random policy and the ANNs are uncorrelated with each other, aggregating multiple ANNs at the agent via a properly designed voting strategy (e.g., majority voting) results in a superior policy.

Once the NFQ iterations of the outer loop are completed for all ANNs, a new policy, represented by the ensemble of Q-functions, is transferred to agents in forms of ANNs weights. With transfer learning, cf. Section VI, the same ensemble of ANNs is used across agents. Upon updating ANN weights, each agent (e.g., the eNB) executes a new, potentially better performing, policy.

B. RRM agent design

Upon receiving from the trainer an updated control policy for an RRM task in the form of new ANNs connection weights, cf. Figure 3a, the agent interacts with the underlying cellular network by executing the new policy.

Figure 3b shows how the agent selects an action for an RRM task, given the latest policy and according to the desired exploration and exploitation trade-off. An explorative action can be selected either via an implicit scheme, e.g. Boltzmann exploration [14], or an explicit method, e.g., $\epsilon$-greedy algorithm [9]. An exploitative action is instead selected by running majority voting on the ensemble of ANNs.

Upon taking an action for a given RRM task, the agent gathers network observations in the form of L1/L2 measurements and other KPIs. Measurements are processed into features relevant for characterizing the state associated with the RRM task, and transitions of the form $(s_t, a_t, r_{t+1}, s_{t+1})$ are transferred to the trainer, cf. Figure 3a.

In addition to accelerating the training, massively-parallel dedicated hardware can also be used to accelerate the execution of ANNs. Therefore, the proposed design has the added benefit of enabling extremely fast execution of complex RRM tasks, potentially within the stringent time requirements of 5G.

VI. LEARNING EFFICIENCY

Wireless networks are always-running critical systems. Learning (i.e., generalizing) in an efficient way is, therefore, crucial to minimize the impact of the exploration while retaining all
the advantages of a self-learned control policy. Hereafter we discuss four approaches to making learning as efficient as possible.

A. Transfer learning

Transfer learning is one of the most effective techniques in improving the efficiency of the learning as it allows us to "crowd-source" the data across the whole network: the experiences gathered in one part of the network can be reused in any other part of the network and vice-versa.

Two basic forms of transfer learning are parameter transfer and instance transfer. The former consists in sharing the same policy (i.e., using the same ANN weights) across different agents, thus significantly reducing the number of free parameters. The latter consists in sharing experience (i.e., transitions) among agents in the network. The architecture shown in Figure 1 enables this
strategy by gathering transitions from multiple agents at the trainer and using such collective experience to derive policies.

A variant of the instance transfer approach consists in initially providing the trainer with artificial samples generated from a network simulator which allows to jump-start the learning. Given that the simulated models and the reality are different, exploration is still required at the agent in order to adapt the policy to the real world.

B. High-quality data collection

$\epsilon$-greedy is an undirected type of exploration. While effective despite its simplicity, there exist more advanced exploration schemes capable of directing the exploration towards more informative parts of the state-action space, thus improving the quality of the collected data. Similarly, we can speed up the learning by embedding human expert knowledge in the policy and confining the exploration within the most relevant regions of the state-action space.

C. Compact state representation

The third strategy to facilitate the learning task is to reduce the number of features characterizing the state in order to reduce the dimensionality of the learning space. This requires the identification of the most informative features to describe the state of the agent in relation to the RRM problem of interest.

D. Efficient data usage

Finally, given that we have gathered informative data samples and have identified the most informative state features, we have to extract as much knowledge as possible (that is, we have to achieve the best possible generalization) out of those data. As such, an effective credit assignment algorithm, such as NFQ-Iteration in conjunction with offline batch training and ensemble learning is preferable to a more basic approach like table-based Q-learning.

VII. Evaluation study cases

We demonstrate the generality of the learning framework with three different RRM problems relevant to 5G. While 5G has not yet been standardized, to avoid ambiguity and clearly ascribe the observed gains to the learned policies (rather than arbitrary simulation assumptions), we choose to evaluate them in a sub-6GHz LTE-compliant event-driven system simulator that provides clear baselines for comparison with state-of-the-art RRM functionalities.
A. Edgeless connectivity

The 3GPP 5G New Radio (NR) study item considers an NR cell as a gigabit-NodeB (gNB) connected to multiple transmission/reception points (TRPs), each capable to provide NR synchronization signals carrying the same cell ID. While a 5G NR cell may conceptually resemble Single-Frequency Network (SFN) operation in LTE-Advanced, its aim is to enable advanced user-centric features, such as edgeless connectivity by means of joint transmission/reception, coordinated scheduling, etc.

Transmit/receive diversity can effectively remove cell edges by (implicitly) creating areas between TRPs wherein users enjoy joint transmission/reception. Defining such areas requires advanced TRP selection methods based, e.g., on signal-to-interference ratio (SIR) thresholds. While static thresholds can guarantee user-specific QoS, dynamically adapting the thresholds to the time-varying user/system-load distribution can enhance both coverage and capacity from a network perspective. Since a proper model of these complex system dynamics is impractical, we apply the general-purpose RRM learning framework of Section V to learn near-optimal SIR thresholds. Specifically, we process readily available L1/L2 measurements into features relevant to characterize the state associated with the task: first and second order statistics of RB utilization per TRP, geographical user distribution, current value of the SIR threshold. We then use the cell
harmonic mean of user perceived rates as system reward.

Figure 4 shows simulation results based on real network traffic in Singapore with small and big packet users with Poisson distribution arrivals. Compared to using fixed network-common SIR threshold configuration, with the proposed algorithm TRPs cooperatively learn to adapt their thresholds to time-varying load distributions and significantly improve both coverage and capacity.

B. Load balancing in ultra-dense HetNets

RAN densification is one of the key enablers to meet the ever increasing demands for higher data rates and coverage in future cellular networks [1]. To this end, the 5G ecosystem will integrate a sub-6GHz coverage layer with a capacity layer of ultra-dense patches of small cells operating at a wide range of frequencies (from sub-6GHz to mm-Wave frequencies) [1]. In such heterogeneous environment, agile load balancing will be a crucial RRM functionality.

With a sub-6GHz layer of ultra-dense small cells, traditional semi-static Cell Range Extension (CRE) offsets, such as the Cell Individual Offset (CIO) designed for LTE-Advanced, will be ineffective and create frequent handovers between and within network layers. While the increased dimensionality of this problem compared to 4G systems makes an explicit model difficult and
traditional optimization techniques impractical, we can demonstrate that the learning framework proposed in the paper can effectively control load across network layers.

As first step towards more advanced load balancing algorithms, we design an RL agent for adapting the CIO value for an ultra-dense small cell layer underling an LTE-Advanced coverage layer. At each small cell, an RL agent adjusts the CIO via quantized values (e.g. ±1 dB, or 0 dB) and informs its UEs. Figure 5 shows that the RL based load balancing scheme, with the cell-edge UE perceived data rate as the reward, is able to strike a better balance between coverage and capacity compared to the traditional schemes with different fixed network-common CIO configurations.

C. Distributed downlink power control

In addition to proper load balancing, efficiently managing inter-cell interference will be crucial in ultra-dense small cell deployments. We therefore studied downlink power control and rate adaptation by using our RL framework with a set of distributed agents, each controlling one cell. To compress the dimensionality of the learning space, the state of the network is partially

Figure 6: The convergence of RL based power control policy in a 2-agent use-case.
observed via few carefully selected features such as cells power budget, average SINR, sum user rates, etc. While features are extracted from measurements within a cell, a global network-wide reward is locally reconstructed by means of inter-agent communication which facilitates the derivation of suitable cooperative power control actions (and therefore the overall policy).

Figure 6 shows the convergence of the RL based algorithm on a 2-cell/2-agents example with full-buffer traffic and randomly distributed users. Agents take turns every 100 milliseconds to control the power budget of their own cell. The network harmonic mean throughput and the network sum-log throughput are considered as reward functions. Starting from an initial exploration probability of 90%, agents extensively explore the state-action space while gradually decreasing the exploration probability until reaching the final value of 10% within 40 seconds. The figure shows that agents successfully learn power control policies optimizing the associated rewards. Extensive simulations with uneven loads and burst data traffic show that compared to a baseline with fixed power budget, our scheme achieves in average 90% power saving as well as gains of 94% and 22% for 5%-tile and the median users throughput, respectively [15].

VIII. Final remarks and future directions

Our vision to achieve flexible and agile RRM in 5G is a clean-slate RRM architecture design based on a general purpose learning framework capable of autonomously generating control policies (i.e., algorithms) to solve complex and highly dimensional RRM tasks directly from data gathered in the network. In contrast to the legacy architectures where RRM consisted of a complex blend of numerous single-purpose algorithms, our vision advocates for designing and improving a single, general purpose, learning framework for different RRM functionalities. Experience (i.e., data) gathered by access nodes in the operator’s network can be reused to either generate RRM policies at network deployment or to update individual policies following changes in the non-stationary wireless environment. This creates the opportunity to more effectively deploy/upgrade network equipment, thereby reducing CAPEX and OPEX for the operator while enhancing the system performance.

While the proposed framework, based on NFQ-Iteration combined with transfer learning and ensemble learning, is valid and capable of producing results for a variety of RRM features, the fast pace of advancements in the field of ML regularly produces new and more powerful techniques that can enrich or replace the proposed framework without affecting the overall architecture. A natural extension of this work is to enhance the learning framework to jointly solve multiple RRM
tasks with a single learned policy. Broadening the task solved by the learning framework would require not only a larger dataset, given the increased dimensionality, to derive the control policy but also, and preferably, more refined learning techniques. A different line of research would address more deeply the issue of coordination across agents to bring them closer to a team-like behavior rather than independent agents. Improving this work in both these directions is at the core of any future investigation considering machine learning for solving RRM problems.

REFERENCES

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, “What will 5G be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G,” *IEEE Communications Magazine*, vol. 2, pp. 74–80, Feb. 2014.

[3] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, “Design considerations for a 5G network architecture,” *IEEE Communications Magazine*, vol. 11, pp. 65–77, Nov. 2014.

[4] A. Imran, A. Zoha, and A. Abu-Dayya, “Challenges in 5G: How to empower SON with Big Data for enabling 5G,” *IEEE Networks*, vol. 6, pp. 27–33, Nov. 2014.

[5] A. Galindo-Serrano and L. Giupponi, “Distributed Q-learning for interference control in OFDMA-based femtocell networks,” in *IEEE Vehicular Technology Conference*, 2010.

[6] M. Bennis and D. Niyato, “A Q-learning based approach to interference avoidance in self-organized femtocell networks,” in *IEEE Globecom Workshops*, 2010.

[7] I. Macaluso, D. Finn, B. Ozgul, and L. A. DaSilva, “Complexity of spectrum activity and benefits of reinforcement learning for dynamic channel selection,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2237–2248, Nov. 2013.

[8] Y. S. Abu-Mustafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from Data - A short Course*. AML Book, 2012.

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

[10] C. Watkins, “Learning from delayed rewards,” Ph.D. dissertation, University of Cambridge, England, 1989.

[11] M. Riedmiller, *Neural fitted Q Iteration - First experiences with a data efficient neural reinforcement learning method*. Springer, 2005.

[12] T. Gabel, C. Lutz, and M. Riedmiller, “Improved neural fitted Q iteration applied to a novel computer gaming and learning benchmark,” in *IEEE Symposium on Adaptive Dynamic Programming And Reinforcement Learning, ADPRL*, April 2011.

[13] M. Riedmiller and H. Braun, “A direct adaptive method for faster back propagation learning: The RPROP algorithm,” in *IEEE Int. Conf. on Neural Networks*, 1993.

[14] C. Gehring and D. Precup, “Smart exploration in reinforcement learning using absolute temporal difference errors,” in *the 12th International Conference on Autonomous Agents and Multiagent Systems*, 2013.

[15] E. Ghadimi, F. D. Calabrese, G. Peters, and P. Soldati, “A reinforcement learning approach to power control and rate adaptation in cellular networks,” *IEEE International Conference in Communications (ICC)*, 2017.