IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 October 2022, accepted 14 November 2022, date of publication 24 November 2022,
date of current version 6 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3224808

RESEARCH ARTICLE

Deep Reinforcement Learning-Based Scheduling for Multiband Massive MIMO

VICTOR HUGO L. LOPES[1,6], (Graduate Student Member, IEEE), CLEVERSON VELOSO NAHUM[2], RYAN M. DREIFUERST[3], PEDRO BATISTA[4], ALDEBARO KLAUTAU[2], (Senior Member, IEEE), KLEBER VIEIRA CARDOSO[1], AND ROBERT W. HEATH JR.[5] (Fellow, IEEE)

[1]Institute of Informatics, Federal University of Goiás, Goiânia 74690-900, Brazil
[2]Department of Computer and Telecommunication Engineering, Federal University of Pará, Belém 66075-110, Brazil
[3]Wireless Networking and Communications Group, The University of Texas at Austin, Austin, TX 78712, USA
[4]Ericsson Research, 164 83 Stockholm, Sweden
[5]Department of Electronics and Computer Engineering, North Carolina State University, Raleigh, NC 27606, USA
[6]Federal Institute of Education, Science, and Technology of Goiás, Inhumas 75402 556, Brazil

Corresponding author: Victor Hugo L. Lopes (victor.lopes@ifg.edu.br)

This paper was partially financed by the Innovation Center, Ericsson Telecom. S.A., Brazil.

ABSTRACT Fifth-generation (5G) cellular communication systems have embraced massive multiple-input-multiple-output (MIMO) in the low- and mid-band frequencies. In a multiband system, the base station can serve different users in each band, while the user equipment can operate only in a single band simultaneously. This paper considers a massive MIMO system where channels are dynamically allocated in different frequency bands. We treat multiband massive MIMO as a scheduling and resource allocation problem and propose deep reinforcement learning (DRL) agents to perform user scheduling. The DRL agents use buffer and channel information to compose their observation space, and the agent's reward function maximizes the transmitted throughput and minimizes the packet loss rate. We compare the proposed DRL algorithms with traditional baselines, such as maximum throughput and proportional fairness. The results show that the DRL models outperformed baselines obtaining a 20% higher network sum rate and an 84% smaller packet loss rate. Moreover, we compare different DRL algorithms focusing on training time to assess the online implementation of the DRL agents, showing that the best agent needs about 50K training steps to converge.

INDEX TERMS Multiband scheduling, MIMO, DRL-based scheduling, mmWave.

I. INTRODUCTION

Massive MIMO remains a key technology in the fifth generation of cellular networks and beyond. It enables good coverage in the network through the use of low- and mid-band frequencies below 7 GHz. At the same time, it offers high spectral efficiency through the use of multi-user MIMO communication to spatially share the channel among different users. The main challenge in applying massive MIMO in the sub 7 GHz frequencies is that the available bands for use are fragmented. This problem can be solved by aggregating frequencies on different carriers. Unfortunately, aggregation requires a more complicated radiofrequency (RF) design at the user equipment (UE) to transmit/receive and process multiple bands simultaneously. An alternative is a multiband architecture, where UEs are limited to using a single carrier simultaneously. The main challenge, in this case, is the dynamic scheduling of UEs to bands in such a way as to reduce the overhead associated with measuring channels while achieving a good assignment of UEs to bands that results in high system efficiency.

Scheduling and resource allocation (SRA) is a key component of most wireless communication systems [1], [2], [3], [4]. For example, in a massive MIMO system, resources that might be scheduled include time-frequency resource blocks and spatial layers [3]. The assignment of users to resources is challenging as it involves combinatorial optimization involving the configuration of other parameters (power, coding,

The associate editor coordinating the review of this manuscript and approving it for publication was Pietro Savazzi.

modulation, and beamformers). There may also be competing objectives that need to be, including rate, fairness, energy, and delay. In general, SRA leads to complicated joint multi-optimization problems [3].

Early approaches for SRA would select users independent of their channels using, e. g., *round robin* [1], [2]. This strategy, though, does not account for the impact of path loss and fading on user links. Since 2000, much work has been devoted to *channel-aware* (or *opportunistic*) SRA [2]. For example, a proportional fair scheduler assigns resources such that the total network throughput is maximized and a minimum level of service is assured for all users [5]. Nowadays, SRA systems adopt cross-layer optimization enabling the usage of different network layers' information for improved performance [1], [6], [7]. This extra flexibility on SRA has been explored in several papers [8], [9]. However, the flexibility brings new difficulty levels due to a large number of operational parameters. The lack of efficient optimization procedures for some SRA problems has motivated research on data-based learning methods.

This paper mainly concerns a combinatorial optimization problem, which is an area traditionally centered on heuristics or dynamic programming. Machine learning has recently expanded the solutions to manage combinatorial optimization problems [10], [11], [12].

One approach for adopting machine learning in combinatorial and other optimization problems is the *learning-to-optimize* paradigm [13], [14]. This approach relies on the existence of an iterative optimization algorithm that provides the labels for supervised learning. After proper training, a neural network can learn how to map the inputs on the output labels and be faster than the optimization algorithm. Nevertheless, a machine learning model trained with the learning-to-optimize approach is not expected to outperform the iterative algorithm that created the labels.

This paper does not use learning-to-optimize or any other supervised learning paradigm but *reinforcement learning* (RL) [15]. RL is well-suited to resource allocation problems that need to adapt continuously to the environment and cannot be solved by efficient optimization algorithms. Another motivation for adopting RL in this paper is the potential for the RL agent to capture and use information that was not explicitly modeled, as experienced in areas such as computational vision [16], [17], and also communications [18].

In this paper, we formulate the multiband massive MIMO as a combinatorial SRA problem and solve it using RL. RL has a long history in the optimization of communication systems [19], [20], [21], e. g. in SRA [22], [23], rate adaptation [21], [24] and self-organizing networks [25], [26]. For relatively small dimensions, we could model the described SRA problem as a *finite Markov decision process* (MDP) and implement the RL agent using *tabular* methods [15]. Unfortunately, scalability in finite MDP is an issue as the complexity grows exponentially with the number of *states* and *actions* [15].

Some SRA problems of interest have a large number of parameters and cannot be modeled as a *finite* MDP nor solved with *tabular methods* [15]. Fortunately, the combination of RL with deep neural networks in DRL approaches makes it possible to deal with a large number of states and actions. Thus, DRL has been extensively investigated for SRA in recent years [21], [23], [27], [28], [29], [30], [31], [32]. Some of the relevant work in this area are summarized in the following paragraphs.

In [21], a distributed method for downlink inter-cell power control and rate adaptation was proposed that used an artificial neural network trained to estimate the Q-values [15] in a DRL agent. Each cell was controlled by a DRL agent that used local measurements with partial observability, given that the cross-cell state observations were unavailable. There were five discrete actions available to the DRL agent. A simplified simulation scenario was used with one base station and users located at random places without mobility.

In [23], the DRL goal was to obtain resource allocation policies that satisfy different *quality of service* (QoS) objectives, such as packet loss rate minimization, guaranteed bit rate satisfaction, and packet delay reduction. An *Actor-Critic* architecture was trained to achieve the QoS metrics by selecting three discrete actions corresponding to the different scheduling rules. Their results were evaluated using simulated data from single antenna cells (MIMO scenarios are not explored) and with un temporally consistent channels.

Aiming to solve the difficulties imposed by high dimensional discrete action spaces in multi-user MIMO SRA problems, the authors of [27] presented a real-time deep deterministic policy gradient (DDPG) user scheduling algorithm. A continuous action space was defined based on a matrix of the UEs' scores obtained from the users' channel correlation matrix, their previous channel quality indicator (CQI), and past average throughput. Resource blocks (RBs) were assigned to the UEs with the highest scores during each time slot. The algorithm sought to maximize throughput and system fairness, not considering metrics for delay-sensitive services.

A *deep belief architecture* (DBA) is used in [28], where features are extracted to train a deep Q-Learning model for dynamic resource allocation in 5G HetNets. In [30], a multi-agent DRL algorithm was applied to the UE association and resource allocation problem, where it was considered a simplified environment with only a single band and Rayleigh fading. An Actor-Critic method for multi-user scheduling in single-cell downlink massive MIMO systems is proposed in [32], which reduces the complexity of the combinatorial problem faced by tabular-based methods. None of these proposed methods considered multiband scheduling nor employed multi-layer data.

A user scheduling algorithm based on DRL for multi-user MIMO systems focused on coverage and capacity optimization in massive MIMO scenarios was proposed in [29]. An optimization parameter called *group alignment of the users' signal strength* was used with a unified QoS threshold

to be dynamically configured with a pre-trained deep policy gradient-based neural network at each *transmission time interval* (TTI). The scheduling scheme considered a discrete action space composed of 15 discrete levels of *signal-to-interference-plus-noise ratio* (SINR) and discrete levels of signal strength from 20 users.

A DRL-based radio resource scheduler for multiple 5G NR numerology settings was proposed in [31], considering only the MAC layer data for the observation space, divided into three parts: eligibility, data rate, and fairness. The first one was a set of UEs with buffered data for transmission that were not associated with a HARQ process, and the last one represented the resource allocation log. The agent tried to learn a resource allocation method by observing a discrete set of actions in which a UE was defined for allocating the current resource block group (RBG). After the training phase, the agent must be able to choose the UE to be scheduled for each time slot. UEs' buffer latency and packet loss rate were not evaluated in [31], which focused on system throughput and fairness metrics. Additionally, the proposed scheduler did not take advantage of information from the PHY layer.

Despite the prior work and the rich set of techniques, DRL-based SRA is still an open problem. Particular issues demand investigation, such as avoiding having the neural network topology (e. g. its number of neurons in the last layer) depending on the number of UEs. Besides, there are general issues such as scalability and the lack of realistic environments for assessing the techniques. Hence, many published solutions are restricted to tasks with small discrete action space [21], [23], [28], [32] or were developed for relatively simple environments [23], [28], [30], [32].

Indeed, the DRL solutions strongly depend on the adopted communication system model and the choice of states and actions. Moreover, the lack of well-established problem settings and baselines delays the adoption of DRL-based SRA in actual deployments.

Our DLR SRA differs from prior work as follows. We deal with partial observability and assume a multiband MIMO model in which the base station can serve different users in each band. However, the UEs are restricted to operating only in a single band at a time. Furthermore, the base station updates the information about the UE channels only in the frequencies that were adopted to serve these users. This creates an observability problem: if a user is not served using a given frequency for a long time, the base station will work with an outdated estimation of the respective channel. Partial observability was addressed in [21] but for a different and simpler system model. Here we emphasize the MIMO transmission, and the number of possible (discrete) actions in the simulations is 4200. In contrast, only five actions were adopted in [21].

Compared with [31] and [27], which considers the allocation of RBs in the time and frequency domains, our work investigates how the DRL agent can deal with outdated channel information. The agents in [31] and [27] rely on full observability while we simulate distinct carrier frequencies and bandwidths, aiming at realistic scenarios combining sub-7 GHz and mmWaves.

Our work differs from previous ones due to the MIMO channel generation process. We pay special attention to this data generation process and use open-source software that enables reproducing our results on other sites.

As in diverse solutions that rely on machine learning, the adopted dataset determines the problem's difficulty. For instance, the adoption of distinct frequency bands imposes requirements on the simulation methodology. One should not use a *drop-based* strategy to generate the channels [33] due to the eventual lack of consistency among the different frequency bands. While some previous works (e. g., [31] and [27]) adopted proprietary simulators, in this paper, we used the well-established QuaDRiGa software [34], [35] to generate the data for training and validating the DRL agents with realistic and consistent channels.

The adopted simulation methodology enabled the composition of an RL environment that can be used to assess SRA techniques. Using this environment and a traditional DRL architecture, this paper also shows results concerning our key assumption: that the agent can deal with partial observability in multiband MIMO environments. However, the DRL agent performance depends on several aspects, such as the network load. It indicates the need for more complete assessment methodologies for DRL-based SRA, such that the tests can lead to robust estimations of the generalization capability of the DRL agent.

In summary, the main contributions of this paper are the following:

- We expand previous work on DRL-based SRA by investigating the issue of partial observability in the context of multiband MIMO systems.
- A RL environment for assessing SRA for multiband MIMO systems, based on a methodology to efficiently train and test DRL agents using offline files generated with the QuaDRiGa simulator, considering a realistic and consistent MIMO scenario with mobile users.
- Results of the SRA performance of a DRL agent in the mentioned environment using the deep Q-Network (DQN) algorithm with a discrete action space and comparison with baseline methods such as round-robin and proportional-fairness. The results indicate that the DRL agent can achieve improved latency while keeping the best throughput obtained by the baselines, even when dealing with partial observability.
- We explored two different scenarios, demonstrating that our proposal can protect the UEs suffering from blockages.
- Concerning the DRL agent, our solution is innovative in how we model the state space and design the reward as a heuristic. Our proposal is cross-layer since it uses both information from the PHY and higher layers.
- We present comparisons and discussions about other DRL algorithms.

- Codes and results are made publicly available at [36], including the DQN agent and the RL environment implemented with the Stable-baselines library, which enables the reproduction of our experiments and tests with new algorithms.

II. COMMUNICATION SYSTEM MODEL

This section introduces the multiband massive MIMO system model considered in this paper. As with any paper that applies machine learning to communications, this section is of critical importance as it outlines the main assumptions in our model, which impact the simulation of the system and the generation of data used by our proposed algorithms. We begin by explaining our vision for multiband massive MIMO communication. Then we explain the key assumptions related to channel modeling and simulation, paying particular attention to the idea of ensuring spatial consistency in the generated data. Next, we describe how performance is evaluated for each communication link in terms of spectral efficiency. Finally, we describe the scheduling and frequency allocation problem under consideration.

A. MULTIBAND MASSIVE MIMO SYSTEM MODEL

In this paper, we pose and solve a specific scheduling problem related to multiband massive MIMO communication. We consider a canonical narrowband massive MIMO system using a digital architecture where there are $N_{\mathrm{t},c}$ antennas at the base station for the c-th frequency band, $N_{\mathrm{u}} = 1$ antenna at the user, and a set of K users in the cell [37]. Only a subset of $\mathcal{K} \in K$ users can be served simultaneously at the same frequency band (e.g., due to a limited number of data streams). We assume a different number $N_{\mathrm{t},c}$ of base station antennas for each carrier c, because the array sizes may differ. For example, arrays in the millimeter wave band can make use of more antenna elements, due to shrinking antenna sizes with the carrier frequency, thereby achieving a similar aperture and antenna gain with low-band counterparts [38].

The limitation on narrowband channels in this paper is simply for simulation convenience. We could consider an OFDMA system and multiple resource blocks in frequency without changing the solution methodology. Because that would increase the computational cost without impacting the comparisons promoted in this paper, we did not use OFDM.

The base station (BS) supports transmission to the users on one of the F discrete frequency bands. Each UE may receive data on only one band at a time. As a result, the BS must perform an assignment of UEs to bands as part of the SRA process. The BS can generally support multiband hardware, enabling it to support many users simultaneously and making the most use of precious spectral resources. We assume that the UE can only support a single band at a time. Using multiple bands on users' devices would require having more sophisticated wideband array designs (due to limited space), more RF components, higher capability data converters, or multiple discrete radios to be operating simultaneously consuming power [39], [40]. Using multiple bands simultaneously would also require channel estimation for all band users, increasing overheads [41]. Note that the different bands are associated with several parameters including the carrier frequency f_c and bandwidth W_c. For example, a low-band carrier may have a bandwidth of 5 MHz, a mid-band carrier with a bandwidth of 20 MHz, and a millimeter wave band with a bandwidth of 100 MHz. As a result, the choice of band impacts the achieved rate through potential differences in the channel as well as in the capability of the system.

We assume that the system uses a TDD transmission protocol for concreteness to obtain channel state information (CSI) at the base station. The BS measures pilots from the users sent on the uplink and then uses that obtained channel state information to do the SRA and downlink beamforming. We could alternatively consider FDD with channel state feedback with some further changes to the system model.

We consider transmissions organized in *blocks* (or frames) of duration T_{c} seconds, as depicted in Fig. 1. In the context of RL, the multi-frame structure can be seen as an *episode*, with a duration of T_{e} seconds, composed of N_{b} blocks. The duration of downlink transmission in a block is $\tau_{\mathrm{d}} T_{\mathrm{s}}$ seconds, where τ_{d} is the number of downlink time *slots* (or *samples*) and T_{s} is the sampling interval. Similarly, τ_{p} is the number of slots-per-block dedicated to pilot sequences, and τ_{u} is the number of slots dedicated to uplink information transmission. We denote discrete time with $t \in \mathbb{Z}^+$ and reset to $t = 1$ at the beginning of each episode. Hence, the relative time corresponding to the t-th slot within an episode is $(t-1)T_{\mathrm{s}}$ seconds. The adopted block structure can represent the standard multi-frame organizations. The number of blocks and slots and the slots' duration are determined by the adopted numerology index, as considered in [31] for 5G NR.

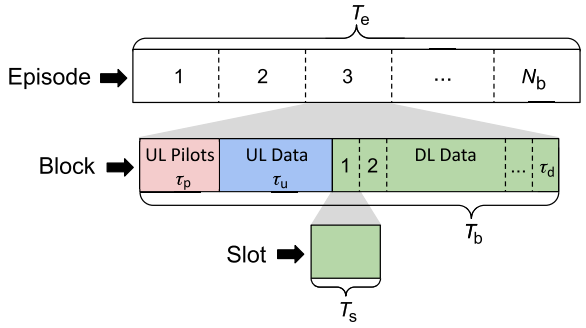


FIGURE 1. Time scales and frame organization: the downlink transmission within a block (or frame) occurs in τ_{d} time slots, where N_{b} blocks form a multi-frame structure, represented as an episode.

Because we focus on *downlink* (DL) configurations, the usage of *uplink* (UL) data (over τ_{u}) is not addressed. Nevertheless, we assume the number of slots per episode is $N_{\mathrm{e}} = N_{\mathrm{b}}(\tau_{\mathrm{p}} + \tau_{\mathrm{u}} + \tau_{\mathrm{d}})$, such that $T_{\mathrm{e}} = N_{\mathrm{e}} T_{\mathrm{s}}$. The duration T_{b} is often chosen according to a definition of *coherence time*. The channel varies within a block according to the adopted channel model described in Section II-C.

B. MULTICELL SYSTEM CONFIGURATION

The power of massive MIMO in cellular systems comes from its ability to provide high spectral efficiency without coordination between cells. To incorporate the interference effects, we consider a hexagonal cellular system of L BSs in the target geographic area to be studied. Each BS is equipped with three sector antennas formed by independently controlled vertical uniform linear arrays (ULA). There is no cooperation among BSs or sectors, and all inference (intercell/intersector) is treated as noise. Thus we allow for interference from antenna back lobes and other base stations. We apply our algorithm and analyze performance for the center cell, as is typical in prior work [27].

When considering Massive MIMO, multiple possible antenna array geometries are available, including linear arrays, planar arrays, and circular arrays. Throughout this paper, we assume each sector has a set of independently controlled ULAs at the base station. In this setting, the elevation beamforming is only controlled by the sector's tilt control, while the combination of ULAs allows for azimuth beamforming. This array balances the flexibility and gain of full-dimensional beamforming with the simplicity and efficiency of more traditional arrays. Operators have traditionally preferred sectorization and tilt control as a means of beamforming [41], [42], so we limit the investigation to one-dimensional geometry. We leave the evaluation of two-dimensional arrays to future work.

We consider a classical hexagonal tessellation of cells. We analyze performance for users in the center cell and consider interference from the neighboring six cells to give a total of $L = 7$ base stations. Each cell has one base station serving three independent sectors for a total of $3L = 21$ serving cells. We place the base stations in a hexagonal grid with an intersite distancing of 282 m. To evaluate performance, we randomly place the UEs within 150 m of the central BS, moving with speeds according to a folded normal distribution $|\mathcal{N}(10, 3)|$m/s, as shown in Fig. 2. At each second, the UE may turn its movement direction with probability $P_{\text{turn}} = 0.2$.

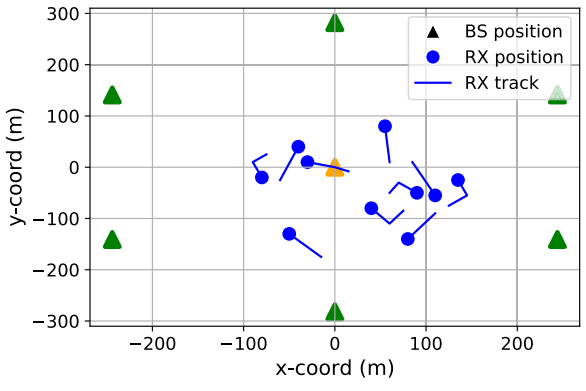


FIGURE 2. Considered multicell system layout, showing an example of UEs patterns.

The interference from neighboring cells is primarily determined by simulating the channels for these interfering cells and assuming they perform perfect interfering azimuth beamforming to the UE, which would be the case that the interfering BSs are all serving users in the same azimuth direction as the intended UE. This essentially is a worst-case assumption on the amount of interference a BS could impose without changing the physical arrays. The blocks (and consequently the time slots) in different cells are assumed to be time-aligned, but we do not model pilot contamination because it is largely inconsequential for our array structure due to the limited pilots necessary without elevation beamforming [43].

C. CHANNEL MODELING

To evaluate an RL algorithm's performance, a channel model with an appropriate level of fidelity is necessary. The model must be sophisticated enough to *preserve correlations* in the channels experienced by different users at different bands during the blocks within the episodes. For example, the commonly employed *drop-based* strategy to generate channels [33], [44] will lead to a lack of consistency among channels at different bands and in different places.

We use the QuaDRiGa [34], [35] tool suite to generate spatially- and frequency-consistent channels for the UEs considering their mobilities previously described. This widely used statistical channel simulator generates spatially correlated MIMO channels from statistical models–including experimentally validated channel models. We use the 3GPP 38.901 UMi [45], [46] statistical models based on a dual-slope path loss with significant inter-parameter correlations. We augment the QuaDRiGa simulator with additional support for multiband consistency to ensure that scattering clusters are consistent across bands. Our simulations follow the 3GPP specifications, including {12, 19} clusters and {20, 20} rays per cluster, for line-of-sight (LOS) and non-line-of-sight (NLOS), respectively [47]. The entire process of generating channel coefficients is meticulously presented in [47]. However, the essential process combines generating random values corresponding to the per-ray paths with specified distributions, correlating the values, and applying path loss and shadowing effects. The simulation code is available at [48]. Now we describe the system simulation setting and appropriate parameters. Once all paths are defined, according to the multicell system configuration described in Section II-B, we generate channel coefficients according to the 3GPP UMi scenario [45], [46], and correlate the coefficients across UEs in space, time, and frequency band. According to the frame organization described in Fig. 1, the sampling interval is $T_s = 1$ ms, in which blocks with different downlink durations can be represented, with episodes that can last up to $T_e = 2$ s. In each sample, the channel is characterized as line-of-sight (LOS) or non-line-of-sight (NLOS), depending on the distance and scattering clusters. The overall distributions for the coefficients are distinct between LOS and NLOS channels. In the LOS case, the large-scale path loss model

follows a dual-slope model [46] and is given by

$$PL_{\text{UMi-LOS}} = \begin{cases} PL_1 & d_{2D} \leq d_{BP} \\ PL_2 & d_{2D} > d_{BP}, \end{cases} \quad (1)$$

where d_{2D} is the two-dimensional distance between the BS and the UE defined by 3GPP [46], and d_{BP} is the breakpoint distance, defined by

$$d_{BP} = BP_{SF}(h_{BS} - h_{Env})(h_{UT} - h_{Env})f_c, \quad (2)$$

where BP_{SF} is the breakpoint scaling factor, $f_c \ \forall c \in \mathcal{F}$ is the frequency in GHz, h_{BS}, h_{Env}, and h_{UT} are the BS, environment, and UE heights, respectively. PL_1 and PL_2 can be defined by:

$$\begin{aligned} PL_1 &= 21\log_{10}(d_{3D}) + 32.4 + 20\log_{10}(f_c) \\ &\quad + d_{3D}\, d_{BU}, \\ PL_2 &= PL_1(d_{BP}) + 40\log_{10}(\frac{d_{3D}}{d_{BP}}), \end{aligned} \quad (3)$$

where d_{3D} is the three-dimensional distance between the BS and the UE [46] and d_{BU} is the distance between the BS and the UE.

For an NLOS channel, the path loss is [46]:

$$PL_{\text{UMi-NLOS}} = \max(PL_{\text{UMi-LOS}}, PL'_{\text{UMi-NLOS}}) \quad (4)$$

where

$$\begin{aligned} PL'_{\text{UMi-NLOS}} &= 35.3\log_{10}(d_{3D}) + 22.4 \\ &\quad + 21.3\log_{10}(f_c) - 0.3(h_{UT} - 1.5). \end{aligned} \quad (5)$$

Once the channels have been generated, we calculate the Reference Signal Received Power (RSRP), which is used in cellular systems for UE assignment and resource allocation.

We then calculate the RSRP, for a set of N clusters and M rays per cluster with a pathloss PL and shadow fading SF for a BS b and UE u pair with a transmission power P_t, as [47, Section 8.1]

$$\text{RSRP}_{b,u} = PL\ SF|\alpha_0|^2 + \sum_{n=1}^{N}\sum_{m=1}^{M}|\alpha_{n,m}|^2 P_t. \quad (6)$$

The calculation of α_0, which is the LOS path contribution, depends on the Ricean K-factor K_R, the antenna beam patterns for the receiver $\mathbf{F}_u$ and transmitter $\mathbf{F}_b$, and the initial phase Φ_{LOS} given by

$$\alpha_0 = \sqrt{\frac{K_R}{K_R+1}}\mathbf{F}_u^T \begin{bmatrix} \exp(j\Phi_{LOS}) & 0 \\ 0 & -\exp(j\Phi_{LOS}) \end{bmatrix} \mathbf{F}_b. \quad (7)$$

The beam pattern vector $\mathbf{F}_u$ is defined by the two components

$$\mathbf{F}_u = \begin{bmatrix} F_{u,\theta}(\theta_{LOS,EOA}, \phi_{LOS,AOA}) \\ F_{u,\phi}(\theta_{LOS,EOA}, \phi_{LOS,AOA}) \end{bmatrix}. \quad (8)$$

In the case of the transmitter side, all angle-of-arrival (AOA) and elevation-of-arrival (EOA) values are replaced with the angle-of-departure (AOD) and elevation-of-departure (EOD) equivalents. For the NLOS paths, $\alpha_{n,m}$ is calculated according to

$$\alpha_{n,m} = \sqrt{\frac{P_n}{M(K_R+1)}}\mathbf{F}_{u,n,m}^T \mathbf{C}\, \mathbf{F}_{b,n,m}. \quad (9)$$

Here, $\mathbf{C}$ is the cross-polarization matrix defined by the initial phases for each ray (m) cluster (n) polarization (xy) combination $\Phi_{n,m}^{xy}$ as

$$\mathbf{C} = \begin{bmatrix} \exp(j\Phi_{n,m}^{\theta\theta}) & \sqrt{\kappa_{n,m}^{-1}}\exp(j\Phi_{n,m}^{\theta\phi}) \\ \sqrt{\kappa_{n,m}^{-1}}\exp(j\Phi_{n,m}^{\phi\theta}) & \exp(j\Phi_{n,m}^{\phi\phi}) \end{bmatrix}. \quad (10)$$

The NLOS $\mathbf{F}_{u,n,m}$ is defined in the same way as the LOS case but is considered for each ray's angular component $\theta_{n,m,EOA}$ and $\phi_{n,m,EOA}$, with similar extensions to the transmitter.

The RSRP is calculated for each sector-to-UE pair. However, the UE only reports the strongest cell (the nominal serving cell) and the top-6 strongest interfering cells during measurement reports to the base station [49]. Due to sectorization, the top-6 interfering cells cover the majority of the interference since all other cells will not be aligned in the direction of the UE. Although back lobes exist on the sector antennas, there is very limited interference due to the large front-to-back ratio and the tilt mechanism causing back lobes to be projected upwards.

D. SPECTRAL EFFICIENCY

It is assumed that the bandwidths associated with the distinct carrier frequencies differ, and the bit rates are calculated as follows. There are well-established capacity-like formulas for massive MIMO that allow us to conveniently estimate the SE in bits/s/Hz [33], [37], [50]. We use a capacity bound to assume that, when considering a bandwidth W_c for a given frequency band, the downlink bit rate $R_{b,u} = SE_{b,u}W_c$ of user u in target cell b can be obtained via the spectral efficiency

$$SE_{b,u}^{DL} = \frac{\tau_d}{\tau_c}\log_2\left(1 + \frac{RSRP_{b,u}}{I_{b,u}^{\text{inter}} + \sigma^2}\right), \quad (11)$$

where

$$I_{b,u}^{\text{inter}} = \sum \max_{l\neq b}{}^{(6)}(RSRP_{l,u}) \quad (12)$$

is the corresponding intercell interference, σ^2 is the noise power, and $\max^{(k)}(\mathbf{x})$ is the set of the k largest elements of $\mathbf{x}$. These equations reflect the assumptions of optimal precoding from all BS to all UEs. The software for spectral efficiency processing is available in the shared source code [36].

E. USER SCHEDULING AND FREQUENCY ALLOCATION

Now we describe the scheduling operation in more detail. The scheduler of the target cell $b = 1$ is illustrated in Fig. 3, where the notation $h_{b,u}[f_c, t]$ for the channels represents that the BS receives information about its channel with user $u \in \mathcal{K}$ over the available frequency bands $f_c \in \{f_1, \ldots, f_F\}$.

In every slot $t \in \tau_d$ within a block $m \in N_b$ (Fig. 1), the scheduler must decide which users should be allocated

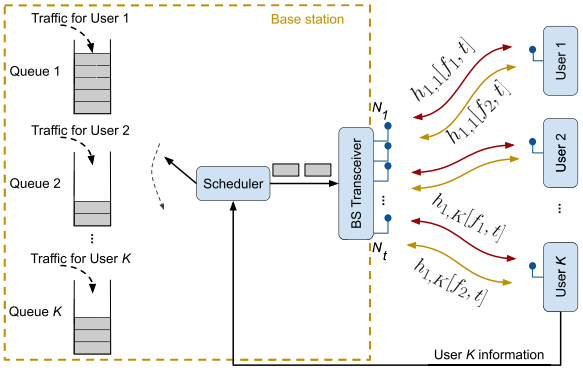


FIGURE 3. Multiband SRA for MU-MIMO using $F = 2$ frequencies from set $\mathcal{F} = \{f_1, f_2\}$.

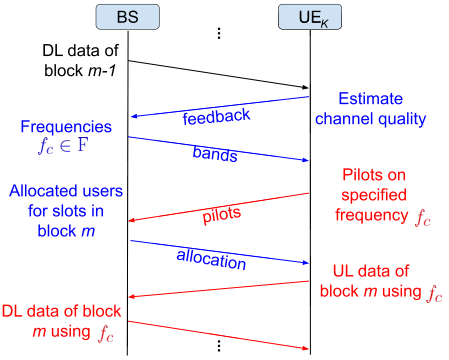


FIGURE 5. Interaction between BS and UE, with blue arrows indicating control channel usage. Red arrows indicate transmission using the frequency f_c specified by the scheduler for block m.

in every slot t for each available frequency band $f_c \in \mathcal{F}$, creating disjoint subsets $\mathcal{K}_{m,f_c,t} \subset \mathcal{K}$ of users in the same slot to be served. As in previous work [51], we consider a single time-frequency resource block per slot, per carrier frequency,[1] which can be spatially multiplexed among different users, as depicted in Fig. 4.

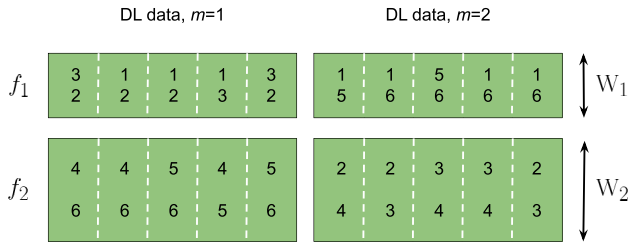


FIGURE 4. Example illustrating the users scheduled according to our multi-user multiband resource allocation model with $\mathcal{F} = \{f_1, f_2\}$, $\mathcal{K} = \{1, 2, \cdots, 6\}$ and $W_2 > W_1$, considering two consecutive blocks $m = 1$ and $m = 2$ with 5 slots each.

We assume that $K_{\max}$ is the maximum number of users that can be served in a slot t using band f_c, i. e., the maximum cardinality $|\mathcal{K}_{m,f_c,t}|$ is $K_{\max}$. An example of such scheduling is depicted in Fig. 4 for two blocks, $m = 1$ and 2, with $K_{\max} = 2$, where $\mathcal{K}_{1,f_1,2} = \{1, 2\}$ and $\mathcal{K}_{2,f_1,2} = \{4, 6\}$. User 6 is served at four slots, as indicated by $\mathcal{K}_{1,f_2,1} = \mathcal{K}_{1,f_2,2} = \{4, 6\}$ and $\mathcal{K}_{1,f_2,3} = \mathcal{K}_{1,f_2,5} = \{5, 6\}$, but ends up not being served in block $m = 2$ in f_2. Without loss of generality, we will assume that $K_{\max}$ users are always scheduled in each frequency band. The BS downlink transmission uses the same power P_{DL} per band, such that its total power is FP_{DL}.

It is assumed that all active users can send feedback through a control channel, indicating their channel quality through the CSI for band f_c. The BS also uses this control channel to inform, before the transmission of DL data for block m, the sets $\mathcal{K}_{m,f_c}$ and $\mathcal{K}_{m,f_c,t}$ for t within block m. This procedure is depicted in Fig. 5.

[1]Note that a 4G LTE / 5G NR scheduler assigns resources over a time-frequency *grid* [52] organized as a hierarchy of resource *blocks*, time-slots, etc.

We assume that the scheduler should be able to operate either with full knowledge of the CSI of all UEs associated with the BS, at all available frequencies or with partial knowledge (i.e., outdated). For the first case, we assume ideal *full observability* [53], as considered in [27], in which the scheduler at the BS would have complete CSI for all K active users in all F frequency bands for each block m. In the case of a *partial observability* scenario, we assume that a fresh CSI on frequency band f_c is obtained only for the users in the set $\mathcal{K}_{m-1,f_c}$ of the previous block. In both cases, the scheduler process is executed during the channel coherence time.

F. DATA TRAFFIC AND INTERFERENCE

The users' data traffic is modeled as Poisson processes with time-varying mean $\lambda_u[t]$ packets for user u. The incoming traffic is buffered and when a buffer is full, its packets are *tail-dropped*, and the same happens if the packet reaches a maximum age ξ. The buffer of user u has a size equivalent to S_u packets, with constant packet size. For faster implementation, we do not generate packets but account for the *buffer occupancy* of user u as $q_u \in \mathbb{N}$. The BS knows for each of the K buffers: its occupancy, and the age of the oldest packet,[2] and the number of dropped packets.

Therefore the simulations, and consequently the RL agent, are not restricted to scenarios such as *full buffer* or *infinite backlogs*, often assumed to enable analytical results. The simulations can incorporate *burst traffic and limited buffer size* [3].

III. DEEP REINFORCEMENT LEARNING SYSTEM MODEL

The following subsections describe the details related to our DRL-based approach to scheduling. First, we present the basic RL configuration, i.e., states, actions, and rewards. Afterward, we comment on the specific DRL agent employed. Finally, we introduce the RL datasets and simulation platform involved in the experiments.

[2]To know the age of the oldest, the BS is required to store the time of arrival of all packets.

A. STATES, ACTIONS, AND REWARDS

From a machine learning perspective, our DRL-based scheduler, here named DRL-SRA (Deep Reinforcement Learning Scheduling and Resource Allocation) is illustrated in Fig. 6. The agent observes the environment (state) and performs an action, which sets the users per frequency band (resources). The action changes the state in which a reward can be calculated. The RL *state* consists of information from both channels and queues, generically denoted as CSI and queue state information (QSI), respectively, similar to [54]. Thus, the scheduler is *cross-layer* since it considers information from layers other than the PHY, such as the buffer occupancy of active users and the age of the packets. Among the distinct time scales in which a modern SRA algorithm can operate [55], we deal in this paper with *short-term* (also called radio scheduler and "channel-aware" scheduling) distributed SRA, with a time scale of a few milliseconds.

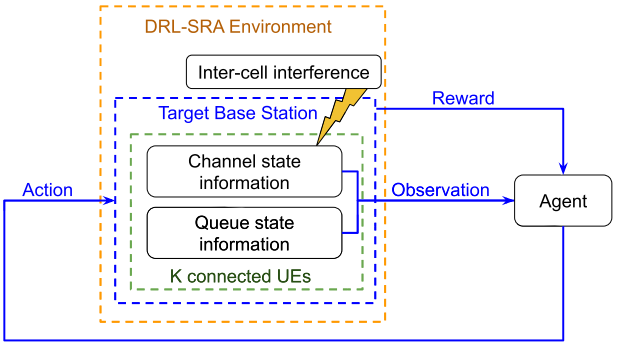


FIGURE 6. DRL-SRA overview: inputs are the CSI and QSI, and outputs are the users that should be served and their respective frequency bands.

We assume the number of downlink data streams is limited by the uplink pilot dimension rather than by the number $N_{\mathrm{t},c}$ of BS antennas, such that $|\mathcal{K}_{m,f_c,t}| < N_{\mathrm{t},c}, \forall c$. Nevertheless, the total number $K > |\mathcal{K}_{m,f_c,t}|$ of connected users is larger than the number of served ones. In this case, the system operates in the so-called *scheduling regime* [37] that makes cross-layer scheduling more important [56]. The BS processes the received channel information (CSI) and updates a matrix C of dimension $K \times F$ with estimated spectral efficiencies. The matrix C is updated once per block, as illustrated in Fig. 5, but when partial observability is implemented, not all elements of C are modified. Similarly, the BS receives the QSI with the status of all buffers.

The RL *action* $\mathbf{a}_t$ is represented by a matrix of integers with dimension $F \times K_{\max}$. The action space is discrete, and because we always choose $K_{\max}$ users, its dimension is

$$N_{\mathrm{a}} = \prod_{i=0}^{F-1} \binom{K - iK_{\max}}{K_{\max}}. \tag{13}$$

For example, assuming $K = 10$, $K_{\max} = 3$ and $F = 2$, we have 4200 possible actions.

Concerning the agent input, we adopt the parameters described in Table 1, which consider information from both QSI and CSI (Table 1). Thus, the observed state s_t groups information from all K connected users at time t in the matrix of dimension $K \times (F + 2)$, and is given by:

$$s_t = (\mathbf{o}_t, \mathbf{g}_t, \mathbf{se}_t), \tag{14}$$

where the vectors $\mathbf{o}_t$, $\mathbf{g}_t$ and $\mathbf{se}_t$ have as elements the values in slot t for users $u = 1, \ldots, K$ of buffer occupancy $\mathbf{o}_t^u \in [0, 1]$, age of oldest packet (relative) $\mathbf{g}_t^u \in [0, 1]$ and SE $\mathbf{se}_t^{f,u}$ for frequency f, respectively. The relative age $\mathbf{g}_t^u$ is obtained by taking the age of the oldest packet in the buffer of user u at time t and dividing it by the maximum age ξ a packet can stay in the buffer before being dropped.

TABLE 1. RL state definition with the last two columns corresponding to the dimension and update frequency.

Description	Category	Dim.	Update
Buffer occupancy $\mathbf{o}_t^u$	QSI	$1 \times K$	slot
Oldest packet age $\mathbf{g}_t^u$	QSI	$1 \times K$	slot
Spectral efficiency $\mathbf{se}_t^{f,u}$	CSI	$F \times K$	block

The state described in Eq. (14) leads to a continuous state space, and our RL agent employs a neural network to deal with it.

The RL *reward* r_t is calculated after the agent takes an action, given a certain state observed at instant t. As observed in [57], [58], [59], and [60], the solution to the SRA problems with a DRL framework strongly depends on the design of meaningful reward functions correlated to the scheduler objectives. DRL is quite flexible concerning the metrics adopted in the reward formulation, but they should guide agent training and often require considerable reward engineering based on trial and error [61].

In this paper, the reward function is a heuristic adopted because it led to good results compared to other alternatives. It is given by

$$r_t = \frac{T_t}{B_{t-1}} - \frac{D_t}{T_t}, \tag{15}$$

where $B_{t-1} = \sum_{u=1}^{K} \mathbf{o}_{t-1}^u$ is the sum of the buffer occupancies rate. Similarly, T_t and D_t are the sum among users of all transmitted and dropped bits in time t.

B. THE DRL AGENT

We adopt a DRL agent that estimates the action-value function $Q(s_t, a_t)$ using iterative updates [15]. More specifically, we adopt a Deep Q-Network algorithm (DQN) [62], [63], an off-policy algorithm that uses the function approximator $Q(s_t, a_t; \theta_i)$ with weights (parameters) θ_i as a Q-network of the i-th iteration [64]. DQN uses a technique named *experience replay*, based on a set $\mathcal{D} = \{d_1, \ldots, d_{\mathrm{N}}\}$ known as *replay memory* composed of the agent's experiences $d_t = (s_t, \mathbf{a}_t, r_t, s_{t+1})$ in each time-step (i.e., a slot t) within multiple episodes, where s_t is the state of the observation, $\mathbf{a}_t$ is the action selected by the agent, and r_t is the reward. Once the replay memory is collected, mini-batch rounds are performed

during the experience replay. The agent applies an ϵ-greedy strategy to select the actions to be taken. As promoted in [63], the DQN training algorithm can randomly select a set of past experiences from the replay memory buffer to use for weight updates. Each step of experience can be used in many weight updates. Moreover, by randomizing the samples of the replay memory, any possible strong correlation between the data samples can be avoided, improving convergence as detected in [63].

C. RL DATASETS AND SIMULATION PLATFORM

Due to the computational cost of obtaining and processing the data from the wireless communication system for training and validation, we pre-compute the channels, as done in [26], wrapping up a dataset composed of the MIMO channels generated by QuaDRiGa, a MATLAB-based open-source simulator, as described in Sec. II-C. In addition, the dataset also incorporates the users' incoming traffic, according to Sec. II-F, allowing the evaluation of different load patterns for network users. The channels in the dataset are used to obtain parameters such as SE in a post-processing stage by using a Python-based simulation platform, which is available to facilitate reproducing the experiments presented in this paper.[3] The simulation platform was implemented with the DRL library *Stable Baselines* [65], a fork of the OpenAI Baselines project [66], which uses TensorFlow.

This simulation platform was used to train and test the DRL-SRA agent performance, considering the previously described communication and DRL subsystems. When the DRL-SRA agent is in the training stage, Algorithm 1 is used in order to train the agent, following the architecture described in Fig. 6. The DRL-SRA agent will take the action $\mathbf{a}_t$ according to a greedy strategy (ensuring adequate exploration of the state space), in which the system can compute the reward according to Eq. (15) and the figures of merit within each training episode. Each training step is used to build the *replay memory* $\mathcal{D}$ used by the DQN algorithm as described in Sec. III-B. Once the replay memory is filled, it is used in the training rounds, in which Q-values are updated [63]. During the testing stage, the DRL-SRA agent will take action for the state observation s_t only based on prediction through the model policy. The communication subsystem is similar to Algorithm 1 implementation.

The employed DQN agent is based on a fully-connected deep neural network with two hidden layers, with rectified linear units (ReLU) as activation functions. In contrast, the output layer adopts a linear activation function. The input consists of an array of 40 elements representing the state. The two hidden layers have 256 neurons each. The output dimension coincides with the number of actions and is given by $N_a = 5040$. Hence, a forward step corresponds to multiplying the input vector by a matrix of dimension 256×41 (to take the bias into account) and calculating the ReLU activation for the resulting array of dimension 256. The other two layers repeat the operation using matrices of dimensions 256×257 and 5040×257, with ReLU and linear activations, respectively. This forward step corresponds to 5.46 GFLOPS. Both A2C and PPO1 agents use two neural networks that share the weights and have a similar topology to the model adopted by the DQN agent. Hence, the forward step by A2C and PPO1 agents corresponds to 5.61 GFLOPS.

In terms of asymptotic complexity, when the number of actions $N_a \gg h_j, \forall j$, where h_j is the number of neurons in layer j, the proposed method has $\Theta(h_J \times N_a)$ complexity, where J is the last layer index. The Round Robin scheduler has complexity $\Theta(K_{max} \times F)$, and both Proportional Fairness and Maximum Throughput have complexity $\Theta(K \times F)$.

[3] https://github.com/LABORA-INF-UFG/DRL-SRA-Gym-SB

Algorithm 1: Training the DRL-SRA Agent.

Input: DQN agent configuration, number of episodes E, episode size N_e, number of active UEs K, K_{max}, and the frequency bands F.
Output: Q-values.

```
Initialize replay memory D;
for each episode e = 1, ..., E do
    for each slot t = 1, ..., N_e do
        while replay memory D is not complete do
            Get input traffic;
            Take an action a = (K_t, f_t);
            for each frequency band f_c in action a do
                Compute the spectral efficiency se_t^{f,u};
            Calculate effective rates R[k, t];
            Calculate reward r_t;
            Update queues;
            Update agent information;
            Store transition in D;
        Update Q-values;
        Reset D;
Output results;
```

IV. SIMULATION RESULTS AND DISCUSSION

The main goal of the simulations is to assess how the proposed agent can learn an adequate allocation policy to maximize the aggregated network throughput (sum rate) while ensuring control of delay and packet loss. For comparison, three traditional schedulers in the literature [67] were adopted: Round Robin (RR), Proportional Fair (PF), and Maximum Throughput (MT). These baseline schedulers present different objectives and performances, especially in terms of sum-rate maximization and fairness scheduling, being adequate to DRL-SRA agents' flexibility evaluations. We are also interested in evaluating the training cost of the models in terms of model convergence. As will be described, two different types of simulations were considered, representing different scenarios.

A. SIMULATION I

To evaluate the impact of different network traffic, simulations were carried out with two different traffic datasets, representing two values of average traffic load per user, i.e., the average total packet arrival per UE per second (or simply MIR - Mean Incoming Rate). These average values follow the data traffic patterns described in Sec. II-F: 110 Mbps per user (MIR-110 Mbps) and 150 Mbps per user (MIR-150 Mbps), labeled as *lower* and *higher* traffic scenarios, respectively. As described previously, the simulations consider two distinct scenarios regarding the observability of the UE states, representing full or partial observability. For the first case, both the spectral efficiencies and the expected data rates of each UE in each band are available in the BS. In the second case, only the latest known information is available, which is updated only when the UE is scheduled in that band. The parameters used in this simulation are summarized in Table 2. The DQN algorithm was instantiated with a feedforward multilayer perceptron (MLP) network with two hidden layers of 256 perceptrons each, using layer normalization with the same configuration for both behavior and target policy. In the output layer, a linear activation function[4] is used, where x is the values coming from the last layer of the hidden layer, W^T is the transpose of the neural network weights matrix, and b is the bias.

TABLE 2. Simulation setup.

Description	Value
$\mathcal{F}$	{2, 28} GHz
W	{20, 100} MHz
K	10
$K_{\max}$	2
Buffer size (per UE)	240 Kb
Slot duration (T_s)	1 ms
Blocks per episode	200
Learning rate	1.22e-4
Discount factor (γ)	0.90
Mean UE incoming rate (MIR)	{110, 150} Mbps
DQN policy	MLP 2 x 256
Hidden layers activation function	ReLU
Linear activation function	$y = xW^T + b$

Figures 7–9 present the performance of the scheduler agents under three network metrics: throughput, buffer delay, and packet loss. The figures exhibit the average value of these metrics computed over all the UEs associated with the BS. The x-axis shows the amount of training time steps performed, while the y-axis presents the average value obtained in a validation run for each metric. Figures in the left (a) correspond to the *lower* traffic scenario and those in the right (b) to the *higher* traffic scenario. The dashed lines represent the partial observability scenario, while the solid lines represent the full observability scenario.

[4]https://pytorch.org/docs/stable/generated/torch.nn.Linear.html

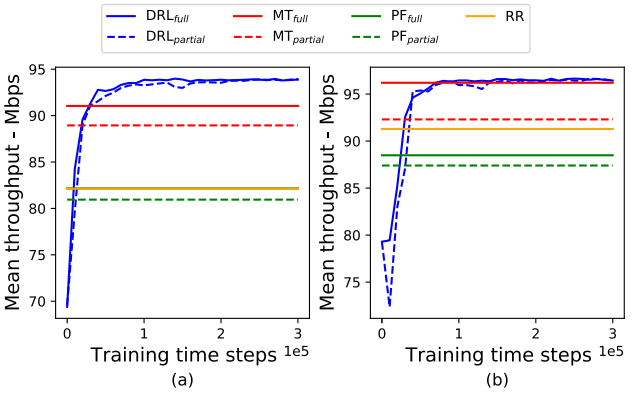


FIGURE 7. Results for the mean throughput per UE: (a) low traffic and (b) high traffic.

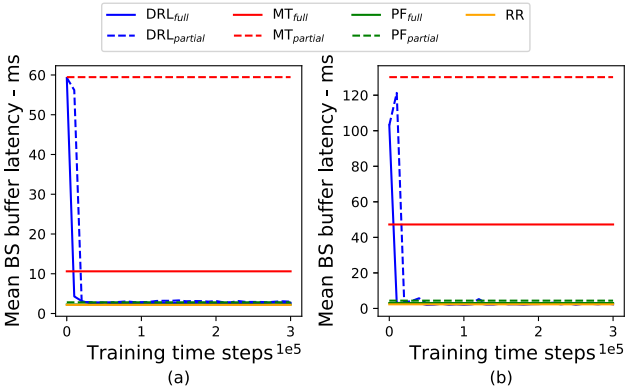


FIGURE 8. Results for the mean buffer delay: (a) low traffic and (b) high traffic.

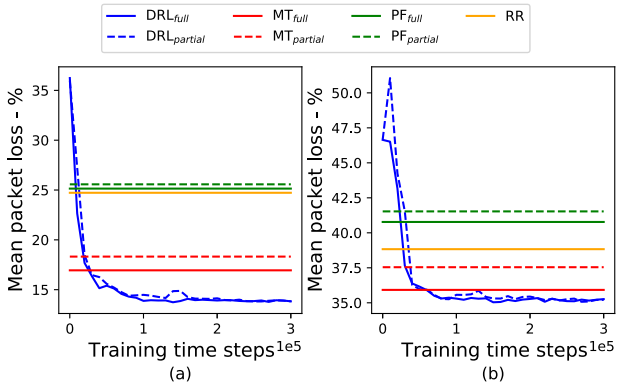


FIGURE 9. Results for mean packet loss: (a) low traffic and (b) high traffic.

Regarding average throughput (Fig. 7), the DRL-SRA agent outperforms all baseline schedulers in all network traffic load and observability scenarios. The DRL-SRA agent model convergence occurs around $50K$ training time steps in both network traffic load scenarios. It is important to note that baseline agents that use the knowledge of the UEs' channels in the decision process (i.e., PF and MT) are severely impacted when not using full observability. On the contrary, the DRL-SRA agent can present similar performance for

both full and partial observations after the convergence of the models. In this sense, the DRL-SRA agent achieves a throughput 5.3% higher than the MT agent in partial observability and between 12.6% and 13.8% higher than the other agents. In the *higher* traffic load scenario (Fig. 7-b), despite the large impact of increasing network load, i.e., due to the higher impact of the packet loss (the channel capacity is upper bounded), the DRL-SRA agent continues to be able to deliver the highest throughput, especially in a partial observability scenario. As illustrated in the figure the DRL-SRA agent consistently outperforms RR and PF agents in this metric after convergence.

In terms of average buffer delay, the DRL-SRA agent has a performance similar to RR and PF agents, which are the agents with the best results in this metric compared with the MT agent, as shown in Fig. 8. This illustrates the DRL-SRA agent's ability to pursue more elaborated and complex behaviors than a conventional scheduler. The MT agent exhibits the largest buffer delays, notably higher than the other agents in the *higher* traffic load scenario. This happens due to the MT's unfair allocation policy, in which UEs with low channel capacity and/or demand (i.e., low data rates due to low packet arrival rate) tend to have their resource allocation delayed until their buffer occupation achieves a size large enough to sustain a higher throughput for a certain time. In the partial observability scenarios, the MT agent presents an even more expressive deterioration of this metric, while the proposed agent demonstrates greater robustness and stability.

It is important to point out that the DRL-SRA agent is not trained to control delays. The observed performance can be assigned to the allocation policy, where maximizing throughput and minimizing packet loss led to lower latencies. Thus, these results demonstrate that the throughput performance obtained is not due to the cost of damming user packets with low buffer occupancy, as done by MT.

Regarding the average packet loss (Fig. 9), we observe that the proposed DRL-SRA agent is able to outperform all baseline agents in all scenarios as soon as the learning algorithm of the DRL agent converges. This result was expected since the packet loss metric is correlated to the throughput in the evaluated environment. The performance of RR and PF agents confirms this observation.

Additionally, when considering the t-test analysis between the scenarios with partial and full observability, we can see in Table 3 that there is no relevant statistical difference between the two scenarios among all the metrics considered here. In this sense, the proposed method's ability to learn even only having access to outdated information from the channels was highlighted.

TABLE 3. T-test results for simulation I.

	Low traffic			High traffic		
	t	P	Dif	t	P	Dif
Throughput	0.34	0.347	0.42	0.46	0.643	0.61
Delay	0.80	0.422	1.39	0.65	0.515	3.96
Loss	0.34	0.277	0.38	0.44	0.353	0.39

TABLE 4. Simulation II setup.

Description	Value
K	4
$K_{\max}$	1
Blocks per episode (validation)	10000
Blocks per episode (training)	200
Low blockage probability P_{low}	0.25
High blockage probability P_{high}	1 - P_{low}
Expected blockage duration	1/2 sec
Expected blockage frequency	0.43 bl/sec

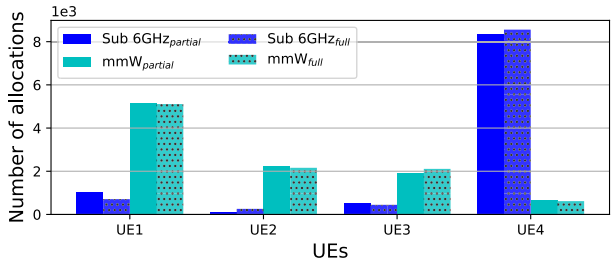
TABLE 5. Simulation II - UEs setup.

	UE-1	UE-2	UE-3	UE-4
Mean incoming rate (Mbps)	210	110	110	210
Blockage probability (mmW)	High	Low	Low	High
Channel capacity (mmW)	High	Low	Low	Low
Channel capacity (sub 6GHz)	Low	Low	Low	High

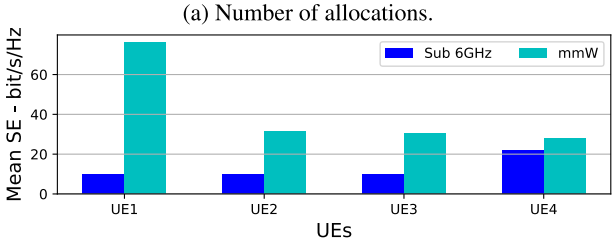
B. SIMULATION II

In order to explore the ability of the DRL-SRA agent to operate in more complex scenarios, a second experiment was carried out according to parameters that differ from the former experiment described in Table 4. A set of 4 connected UEs (i.e., $K = 4$) for model training and validation is determined, where each of these UEs can assume a distinct profile, as described in Table 5. UEs 1 and 4 have a higher average packet arrival rate in comparison with UEs 2 and 3. None of the UEs suffer blockages in the sub 6 GHz band, while blockages can occur in the mmW band, with different probabilities, frequencies of occurrence, and duration. Thus, UEs 1 and 4 are susceptible to a higher probability of blockage in the mmW band in comparison with UEs 2 and 3. The blockage probabilities, their expected duration, and the expected frequency of occurrence were modeled in line with [68], considering an urban scenario with dynamic and static blockers (e.g., pedestrians and buildings, respectively). For simplicity, it is assumed that blockage occurs on both LOS and NLOS links so that the spectral efficiency of the blocked channel tends to zero. Partial and full observability scenarios are also considered, as done in the former experiment. The training was performed with episodes with 200 time steps each (as done in the previous experiment) but with longer validation episodes (10K time steps) using different datasets. The results are presented in the next figures, which are discussed below, showing the averages obtained in the validation episodes.

First, we evaluate the behavior of allocations performed by the DRL-SRA agent. Fig. 10a shows the average number of allocations per UE during the validation episodes. We can see that, in both observability scenarios, the DRL-SRA agent



(a) Number of allocations.



(b) Spectral efficiency.

FIGURE 10. **Comparing the number of allocations and the spectral efficiency received by each UE per frequency band.**

is able to capture both the highest packet arrival rates from UE1 and UE4, as well as the frequencies that provide the best spectral efficiencies (as shown in Fig. 10b). Thus, UE1 is allocated with priority in the mmW band. At the same time, UE4 receives even more allocation priority in the sub 6 GHz band. This demonstrates that the agent can learn the band that can provide the best performance for each UE. Such behavior allowed the protection of the UEs with greater demands. Meanwhile, the allocation policy for UE2 and UE3 becomes very similar, which is also justified by their characteristics of channels and demands. There is a prioritization of allocation in the mmWave band whenever possible.

Fig. 11 presents the average aggregate sum-rate metric, in view of the performance observed in the baseline schedulers, for both observability scenarios. It is observed that

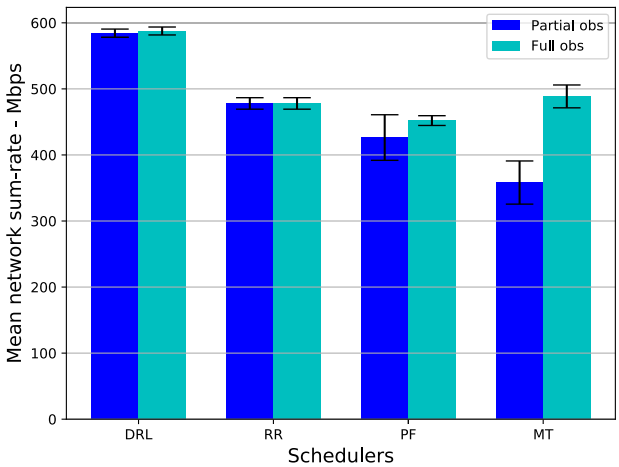


FIGURE 11. **Comparing the network sum-rate results.**

the DRL-SRA agent can deliver a throughput performance superior to the other schedulers, even in a partial observability scenario. In a scenario of partial observability, the lagged information of the channels of the UEs imposes a degradation in the performance of the schedulers that use such information in the decision-making, as in the case of PF and MT. The errors caused by this lag can be even more harmful in scenarios with the possibility of signal blockage. These schedulers can take a long time to notice such situations, negatively impacting throughput and increasing delay and packet losses, for example. In this sense, it is observed that the DRL-SRA agent is able to learn such situations, mainly due to the impact that the blockages cause in obtaining the training rewards. Thus, the DRL-SRA agent can outperform the RR scheduler by about 22% when operating in partial observability and about 20% in relation to the MT when operating in full observability.

Regarding the average delay of packets in the BS buffers, Fig. 12 shows the results. As noted earlier, the RR scheduler can deliver good performances on this metric compared to PF and MT. However, the DRL-SRA agent manages to overcome it, with an average delay of about 62% to 64% lower in the case of partial and full observability, respectively. Again, these metrics demonstrate that the gain obtained in the network sum rate is not obtained at the cost of imposing severe delays for those UEs with low buffer occupancy, as observed in PF and MT, for example. Additionally, the delay in these schedulers can be even worse in partial observability scenarios, again justified by the errors created by the lagged information of the UEs channels.

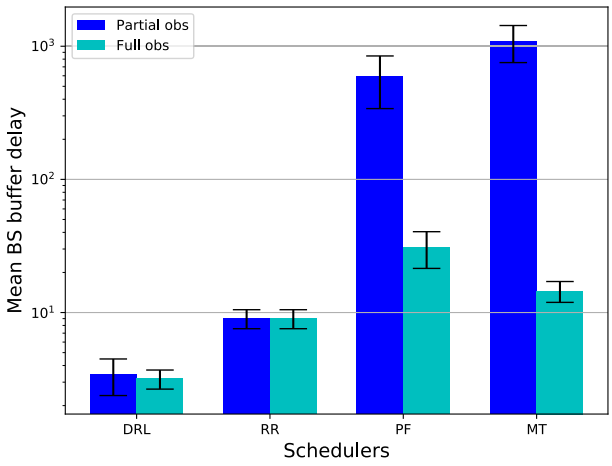
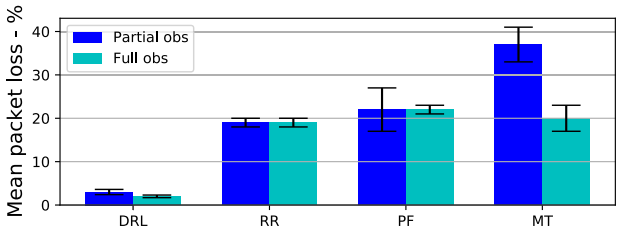
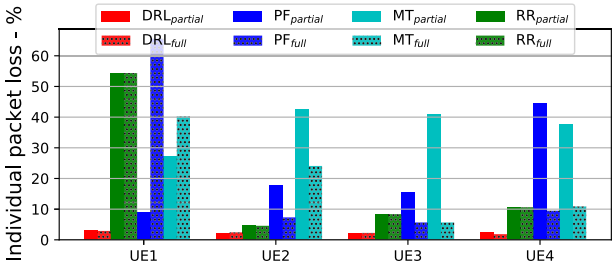


FIGURE 12. **Comparing the BS buffers delay results.**

Observing the packet loss metrics in these scenarios is important, as shown in Figs. 13a and 13b, for the aggregate and individual packet loss, respectively. The DRL-SRA agent manages to operate with a much lower packet loss compared with other schedulers, even in a partial observability scenario, delivering an average packet loss about 84% lower than the



(a) Network packet loss.



(b) Individual packet loss.

FIGURE 13. Comparing the packet loss results.

RR (in partial observability), which has the best performance on this metric among the baseline schedulers. Looking at the individual metric in more detail (Fig. 13b), even though the RR delivers the lowest average packet loss among the baseline schedulers, UE1 is severely impaired in a very unbalanced way among the other UEs. Thus, although it guarantees fair allocation opportunities for all UEs, this illustrates the need for employing scheduling and resource allocation methods that use smarter mechanisms in the decision-making process. In this sense, it is possible to notice that only the DRL-SRA agent can guarantee an equal average packet loss among all UEs, even if the UEs have different capacities and demands.

Finally, regarding the t-test analysis presented in Table 6, it is proved that there is no relevant statistical difference between the scenarios with partial and full observability.

TABLE 6. T-test analysis for the experiment II.

	t	P	Dif
Throughput	2.92	0.006	5.91
Delay	4.96	0.007	2.04
Loss	2.27	0.026	0.008

C. TRAINING ANALYSIS

The results of the two experiments illustrate the potential of the DRL-SRA agent as an important alternative in the challenges of user scheduling and resource allocation in multiband MU-MIMO networks. However, it is important to consider the costs involved in training the agent. Thus, Fig. 14 and Fig. 15 present details about the training of the DRL-SRA agent, considering one of the models used in experiment I (Sec. IV-A) in a scenario of partial observability and high average packet arrival rate. Results are presented both for the use of the DQN algorithm (adopted by the DRL-SRA agent) as well as for the A2C [62] and PPO1 [69] algorithms. Although other algorithms available in the stable-baselines library could be considered, only these ones presented minimally equivalent performance and/or compatibility, while the others being disregarded to favor the visualization of the graphs.

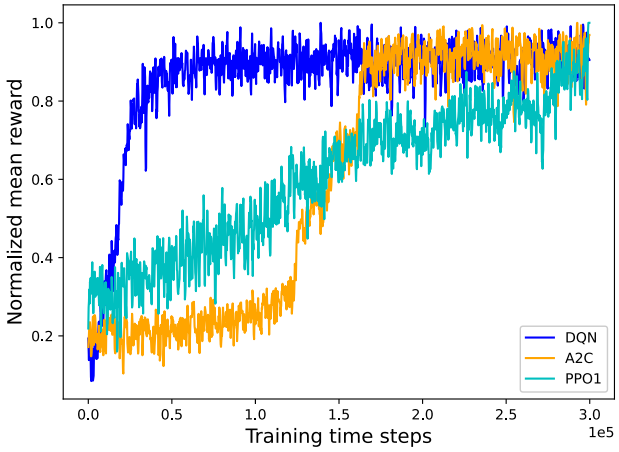


FIGURE 14. Training reward evolution analysis.

Fig. 14 shows the evolution of the training in terms of the average reward obtained (normalized) in relation to the number of used training time steps. The DQN algorithm presents the fastest convergence, occurring with about $50K$ time steps, as already observed in the previous experiment's curves. Given that the A2C algorithm converges only after $160K$ time steps, while the PPO1 needs $300K$ time steps to reach the same level of rewards, the adoption of the DQN algorithm is preferred. Although the convergence time observed by the DQN can impact its adoption for online operation, several techniques already described in the literature can be used to deal with this issue. For example, transfer learning [70] can be employed in order to allow the use of previously trained models and adapted to the current situation of the network in operation.

Comparing the performance of the models trained using each algorithm considered here, Fig. 15 displays the average packet loss and sum rate (top and bottom, respectively). A2C manages to approach the performance obtained by the DQN only after the convergence and the PPO1 presents a lower performance. Even if a longer convergence time can be accepted, the choice of the DQN algorithm is still preferred due to the smaller amount of data required for training the models. As the duration of each episode is extremely short (200 time steps), the training duration can be greatly reduced, generating several positive impacts in the management of the DRL environment.

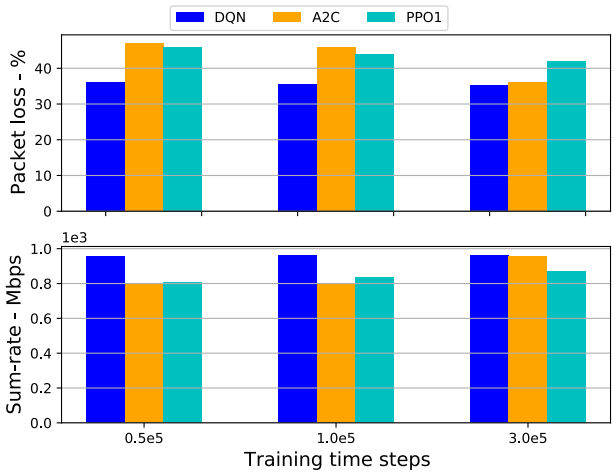


FIGURE 15. Comparing the algorithms metrics.

D. DISCUSSION

The experiments presented demonstrate important aspects regarding the use of schedulers based on DRL for the problem of scheduling and resource allocation in multiband Massive MIMO. To provide a discussion, some points need to be highlighted:

- Trained models can be obtained with low training time steps (50 K 100 K time steps), where datasets do not need to contain very long samples.
- The RL modeling used allows the partial observability not to generate severe negative impacts on the agent's performance. Additionally, in partial observability scenarios, the proposed agent outperforms all the schedulers considered, mainly in the throughput and loss metrics.
- The proposed agent is able to learn the impact of different load, blockage, and spectral efficiency profiles.
- The proposed agent is able to incorporate the behaviors of both schedulers considered. Furthermore, the throughput maximization achieved does not impose an increase in delay.

Recalling that none of the related work considered jointly all the aspects used here, impacting the agent's flexibility and its applicability in more realistic scenarios, as already described.

When comparing the proposed method against the other algorithms (A2C and PPO1), it is observed that the choice of DQN is more reasonable since it converges with a shorter training time, delivers superior performance, and does not represent a higher computational cost (Sec. III-C).

The drawback observed in the proposed agent is the relationship between the space of actions and the size of the output layer of the neural network, generating impacts on scalability. This aspect will be addressed in future work.

V. CONCLUSION

This paper presented the DRL-SRA, a DRL-based agent for scheduling mobile users in a multiband massive MIMO system using a DQN algorithm, with observation data from the PHY and MAC layers. Our agent was trained and tested in a flexible platform, implemented with the stable-baselines library, using a realistic and consistent MIMO scenario with mobile users generated with the QuaDRiGa simulator. Simulations were conducted to evaluate the DRL-SRA agent performance against different parameters, using well-known baseline schedulers for comparison.

The experiments show that the time required to train the models is viable for their adoption in solutions that require online learning models, as in the architecture proposed in [71], which employs a RAN AI controller that has logical interfaces with many of the network functions (both in BS and in the core network), responsible for processing AI solutions in non-real-time, in parallel to the running system. Additionally, the proposed agent is suitable for delay-tolerant and delay-sensitive [72] services without compromising the users' throughput and packet loss.

DRL-based SRA is a powerful tool for the optimization of 5G and 6G networks. The construction of data-driven SRA may be relevant to future generations of wireless network services that will require more flexible and adaptable radio resource management tools supporting a suite of new use cases. DRL may be able to account for practical impairments that are difficult to adjust for using only standard optimization theory. The simulation scenarios, however, must be realistic, and the solutions should take into account practical aspects such as scalability with the number of users. This paper presented a reproducible framework that can facilitate further investigations toward the goal of making DRL-based SRA closer to actual deployment.

REFERENCES

[1] W. Ajib and D. Haccoun, "An overview of scheduling algorithms in MIMO-based fourth-generation wireless systems," *IEEE Netw.*, vol. 19, no. 5, pp. 43–48, Sep. 2005.

[2] A. Asadi and V. Mancuso, "A survey on opportunistic scheduling in wireless communications," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1671–1688, 4th Quart., 2012.

[3] E. Castañeda, A. Silva, A. Gameiro, and M. Kountouris, "An overview on resource allocation techniques for multi-user MIMO systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 239–284, 1st Quart., 2017.

[4] Y. Xu, G. Gui, H. Gacanin, and F. Adachi, "A survey on resource allocation for 5G heterogeneous networks: Current research, future trends, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 668–695, 2nd Quart., 2021.

[5] H. Kim and Y. Han, "A proportional fair scheduling for multicarrier transmission systems," *IEEE Commun. Lett.*, vol. 9, no. 3, pp. 210–212, Mar. 2005.

[6] A. Khalek, C. Caramanis, and R. Heath, "Delay-constrained video transmission: Quality-driven resource allocation and scheduling," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 1, pp. 60–75, Jan. 2015.

[7] A. Vora and K.-D. Kang, "Downlink scheduling and resource allocation for 5G MIMO multicarrier systems," in *Proc. IEEE 5G World Forum (5GWF)*, Jul. 2018, pp. 174–179.

[8] G. Femenias, F. Riera-Palou, X. Mestre, and J. J. Olmos, "Downlink scheduling and resource allocation for 5G MIMO-multicarrier: OFDM vs FBMC/OQAM," *IEEE Access*, vol. 5, pp. 13770–13786, 2017.

[9] B. Maaz, K. Khawam, S. Tohme, S. Lahoud, and J. Nasreddine, "Joint user association, power control and scheduling in multi-cell 5G networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.

[10] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," 2016, *arXiv:1611.09940*.

[11] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour d'horizon," *Eur. J. Oper. Res.*, vol. 290, no. 2, pp. 405–421, 2021.

[12] W. Kool, H. van Hoof, and M. Welling, "Attention, learn to solve routing problems!" 2018, *arXiv:1803.08475*.

[13] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

[14] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, Jun. 2019.

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

[16] R. Zhang, L. Wu, Y. Yang, W. Wu, Y. Chen, and M. Xu, "Multi-camera multi-player tracking with deep player identification in sports video," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107260.

[17] R. Zhang, L. Xu, Z. Yu, Y. Shi, C. Mu, and M. Xu, "Deep-IRTarget: An automatic target detector in infrared imagery using dual-domain feature extraction and allocation," *IEEE Trans. Multimedia*, vol. 24, pp. 1735–1749, 2022.

[18] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

[19] C. Pandana and K. J. R. Liu, "Near-optimal reinforcement learning framework for energy-aware sensor communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 788–797, Apr. 2005.

[20] I. S. Comcsa, S. Zhang, M. Aydin, J. Chen, P. Kuonen, and J. F. Wagen, "Adaptive proportional fair parameterization based LTE scheduling using continuous actor-critic reinforcement learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 4387–4393.

[21] E. Ghadimi, F. D. Calabrese, G. Peters, and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[22] D. V. Djonin and V. Krishnamurthy, "MIMO transmission control in fading channels—A constrained Markov decision process formulation with monotone randomized policies," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5069–5083, Oct. 2007.

[23] I.-S. Comsa, A. De-Domenico, and D. Ktenas, "QoS-driven scheduling in 5G radio access networks—A reinforcement learning approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.

[24] G. Peserico, T. Fedullo, A. Morato, S. Vitturi, and F. Tramarin, "Rate adaptation by reinforcement learning for Wi-Fi industrial networks," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1139–1142.

[25] O.-C. Iacoboaiea, B. Sayrac, S. B. Jemaa, and P. Bianchi, "SoN coordination in heterogeneous networks: A reinforcement learning framework," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 5835–5847, Sep. 2016.

[26] R. M. Dreifuerst, S. Daulton, Y. Qian, P. Varkey, M. Balandat, S. Kasturia, A. Tomar, A. Yazdan, V. Ponnampalam, and R. W. Heath, "Optimizing coverage and capacity in cellular networks using machine learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 8138–8142.

[27] X. Guo, Z. Li, P. Liu, R. Yan, Y. Han, X. Hei, and G. Zhong, "A novel user selection massive MIMO scheduling algorithm via real time DDPG," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[28] F. Tang, Y. Zhou, and N. Kato, "Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5G HetNet," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2773–2782, Dec. 2020.

[29] Y. Yang, Y. Li, K. Li, S. Zhao, R. Chen, J. Wang, and S. Ci, "DECCO: Deep-learning enabled coverage and capacity optimization for massive MIMO systems," *IEEE Access*, vol. 6, pp. 23361–23371, 2018.

[30] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.

[31] F. Al-Tam, N. Correia, and J. Rodriguez, "Learn to schedule (LEASCH): A deep reinforcement learning approach for radio resource scheduling in the 5G MAC layer," *IEEE Access*, vol. 8, pp. 108088–108101, 2020.

[32] L. Chen, F. Sun, K. Li, R. Chen, Y. Yang, and J. Wang, "Deep reinforcement learning for resource allocation in massive MIMO," in *Proc. 29th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2021, pp. 1611–1615.

[33] R. W. Heath, Jr., and A. Lozano, *Foundations of MIMO Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[34] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.

[35] F. Burkhardt, S. Jaeckel, E. Eberlein, and R. Prieto-Cerdeira, "QuaDRiGa: A MIMO channel model for land mobile satellite," in *Proc. 8th Eur. Conf. Antennas Propag. (EuCAP)*, Apr. 2014, pp. 1274–1278.

[36] V. H. L. Lopes, A. Klautau, C. Nahum, and K. Cardoso, "DRL-based scheduling for multiband access massive MIMO—Simulation platform (code and data)," DRL-SRA Project Repository, Labora INF-UFG, Goiânia, Brazil, Jun. 2022. [Online]. Available: https://github.com/LABORA-INF-UFG/DRL-SRA-Gym-SB

[37] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends Signal Process.*, vol. 11, nos. 3–4, pp. 154–655, Nov. 2017, doi: 10.1561/2000000093.

[38] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78729–78757, 2019.

[39] Y. Palaskas, A. Ravi, S. Pellerano, and S. Sandhu, "4 design considerations for integrated MIMO radio transceivers," in *Wireless Technologies*. Boca Raton, FL, USA: CRC Press, 2017, pp. 107–130.

[40] M. Ikram, N. Nguyen-Trong, and A. Abbosh, "Multiband MIMO microwave and millimeter antenna system employing dual-function tapered slot structure," *IEEE Trans. Antennas Propag.*, vol. 67, no. 8, pp. 5705–5710, Aug. 2019.

[41] Y.-H. Nam, M. S. Rahman, Y. Li, G. Xu, E. Onggosanusi, J. Zhang, and J.-Y. Seol, "Full dimension MIMO for LTE-advanced and 5G," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Feb. 2015, pp. 143–148.

[42] R. Maslennikov, A. Trushanin, O. Testov, M. Vechkanov, A. Antipova, T. A. Thomas, A. Ghosh, and E. Visotsky, "Azimuth and elevation sectorization for the stadium environment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 3971–3976.

[43] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell TDD systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640–2651, Aug. 2011.

[44] R. Hasan, M. M. Mowla, and N. Hoque, "Performance estimation of massive MIMO drop-based propagation channel model for mmWave communication," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, 2020, pp. 461–464.

[45] B. Mondal, T. A. Thomas, E. Visotsky, F. W. Vook, A. Ghosh, Y.-H. Nam, Y. Li, J. Zhang, M. Zhang, Q. Luo, Y. Kakishima, and K. Kitao, "3D channel model in 3GPP," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 16–23, Mar. 2015.

[46] Q. Zhu, C.-X. Wang, B. Hua, K. Mao, S. Jiang, and M. Yao, "3GPP TR 38.901 channel model," in *Wiley 5G Ref: The Essential 5G Reference Online*. 2019, pp. 1–35.

[47] 3GPP, *Study on 3D Channel Model for LTE*, 3rd Generation Partnership Project (3GPP), document (TR) 36.873, 06 2017, version 12.5.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2574

[48] R. M. Dreifuerst. (Jun. 2020). *QuaDRiGA Simulation Extensions*. [Online]. Available: https://github.com/Ryandry1st/QuaDRiGA-Simulation-Extensions

[49] 3GPP, *Requirements for support of radio resource management*, 3rd Generation Partnership Project (3GPP), Technical Specification (TS), document 3GPP TS 138.133, 2018, release 15 version 15.3.0. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138100_138199/138133/15.03.00_60/ts_138133v150300p.pdf

[50] T. Marzetta, E. Larsson, and H. Yang, *Fundamentals of Massive MIMO*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[51] J. Choi, N. Lee, S.-N. Hong, and G. Caire, "Joint user scheduling, power allocation, and precoding design for massive MIMO systems: A principal component analysis approach," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 396–400.
[52] D. C. Larsson, "NR physical layer overview," in *5G and Beyond: Fundamentals and Standards*. Springer, 2021, p. 259.
[53] A. Avranas, M. Kountouris, and P. Ciblat, "Deep reinforcement learning for resource constrained multiclass scheduling in wireless networks," 2020, *arXiv:2011.13634*.
[54] J. S. Shekhawat, R. Agrawal, K. G. Shenoy, and R. Shashidhara, "A reinforcement learning framework for QoS-driven radio resource scheduler," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–7.
[55] N. Omidvar, A. Liu, V. Lau, F. Zhang, D. H. K. Tsang, and M. R. Pakravan, "Optimal hierarchical radio resource management for HetNets with flexible backhaul," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4239–4255, Jul. 2018.
[56] V. K. N. Lau, "Asymptotic analysis of SDMA systems with near-orthogonal user scheduling (NEOUS) under imperfect CSIT," *IEEE Trans. Commun.*, vol. 57, no. 3, pp. 747–753, Mar. 2009.
[57] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.
[58] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.
[59] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.
[60] Z. Lu and M. C. Gursoy, "Dynamic channel access and power control via deep reinforcement learning," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2019, pp. 1–5.
[61] C. She, R. Dong, Z. Gu, Z. Hou, Y. Li, W. Hardjawana, C. Yang, L. Song, and B. Vucetic, "Deep learning for ultra-reliable and low-latency communications in 6G networks," *IEEE Netw.*, vol. 34, no. 5, pp. 219–225, Sep./Oct. 2020.
[62] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 48, M. F. Balcan and K. Q. Weinberger, Eds. New York, NY, USA: PMLR, Jun. 2016, pp. 1928–1937. [Online]. Available: http://proceedings.mlr.press/v48/mniha16.html
[63] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[64] V. Mnih, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[65] A. Hill, A. Raffin, M. Ernestus, A. Gleave, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu. (Aug. 2018). *Stable Baselines*. [Online]. Available: https://stable-baselines.readthedocs.io/en/master/index.html
[66] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov. (2017). *OpenAI Baselines*. [Online]. Available: https://github.com/openai/baselines
[67] S. Schwarz, C. Mehlfuhrer, and M. Rupp, "Low complexity approximate maximum throughput scheduling for LTE," in *Proc. Conf. Rec. 44th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2010, pp. 1563–1569.
[68] I. K. Jain, R. Kumar, and S. S. Panwar, "The impact of mobile blockers on millimeter wave cellular systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 854–868, Apr. 2019.
[69] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[70] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jul. 2021.
[71] S. Han, T. Xie, I. Chih-Lin, L. Chai, Z. Liu, Y. Yuan, and C. Cui, "Artificial-intelligence-enabled air interface for 6G: Solutions, challenges, and standardization impacts," *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 73–79, Oct. 2020.
[72] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for radio resource allocation with diverse quality-of-service requirements in 5G," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2309–2324, Apr. 2021.

VICTOR HUGO L. LOPES (Graduate Student Member, IEEE) received the degree in informatics–information systems from the Cefet-Goiás, in 2006, and the master's degree in electrical engineering from the Faculty of Technology, UnB, in 2015. He is currently pursuing the Ph.D. degree in computer science with the Institute of Informatics, Federal University of Goiás (UFG). He has been a Professor with the Federal Institute of Education, Science, and Technology of Goiás (IFG), since 2013. His research interests include radio resource management and machine learning applied to next-generation wireless networks.

CLEVERSON VELOSO NAHUM received the B.Sc. degree in computer engineering from the Federal University of Pará (UFPA), Belém, Pará, Brazil, in 2019, and the master's degree in electrical engineering with emphasis on telecommunications from the Electrical Engineering Graduate Program, UFPA, in 2021, where he is currently pursuing the Ph.D. degree. He is part of the Research and Development Center for Telecommunications, Automation and Electronics (LASSE), since 2016. His current research interests include network slicing, radio resource management, and artificial intelligence applied on mobile communication systems.

RYAN M. DREIFUERST received the B.S. degree in electrical engineering from the Milwaukee School of Engineering with minors in mathematics and physics, the B.S. degree in electrical and communications engineering from the Technische Hochschule Lübeck, the M.S. degree in electrical engineering from The University of Texas at Austin (UT Austin), where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include machine learning and signal processing. Specifically, he has focused on augmenting machine learning with domain knowledge for physical layer processing in wireless communications.

PEDRO BATISTA received the B.S., M.S., and Ph.D. degrees from the Electrical Engineering Graduate Program, Federal University of Pará, Brazil. He is currently a Researcher at Ericsson. His research interests include optimization of future mobile networks, particularly, using machine learning and machine reasoning, and future internet architectures.

ALDEBARO KLAUTAU (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the Federal University of Pará (UFPA), in 1990, the M.Sc. degree in electrical engineering from the Federal University of Santa Catarina (UFSC), in 1993, and the Ph.D. degree in electrical engineering from the University of California at San Diego (UCSD), in 2003. He is currently a Full Professor at the UFPA, where he is the ITU Focal Point and co-ordinates the LASSE Research Group. He is a Researcher of CNPq, Brazil, and the Brazilian Telecommunications Society (SBrT). His research interests include machine learning and signal processing for communications and embedded systems.

KLEBER VIEIRA CARDOSO received the degree in computer science from the Universidade Federal de Goiás (UFG), in 1997, and the M.Sc. and Ph.D. degrees in electrical engineering from the COPPE, Universidade Federal do Rio de Janeiro, in 2002 and 2009, respectively. He is currently an Associate Professor with the Institute of Informatics, UFG, where he has been a Professor and a Researcher, since 2009. He spent his sabbatical at Virginia Tech, USA, in 2015, and the Inria Saclay Research Centre, France, in 2020. He has participated in some international research projects (including two from joint calls BR-EU) and coordinated several national-sponsored research and development projects. His research interests include wireless networks, SDN, virtualization, resource allocation, and performance evaluation.

ROBERT W. HEATH JR. (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in 1996 and 1997, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2002. From 1998 to 2001, he was a Senior Member of the Technical Staff then a Senior Consultant at Iospan Wireless Inc., San Jose, CA, USA, where he worked on the design and implementation of the physical and link layers of the first commercial MIMO-OFDM communication systems. From 2002 to 2020, he was with The University of Texas at Austin, most recently as a Cockrell Family Regents Chair in Engineering and the Director of the UT SAVES. He is currently a Distinguished Professor with North Carolina State University. He is also the President and the CEO of MIMO Wireless Inc. He has authored *Introduction to Wireless Digital Communication* (Prentice Hall, 2017) and *Digital Wireless Communication: Physical Layer Exploration Lab Using the NI USRP* (National Technology and Science Press, 2012), and coauthored *Millimeter Wave Wireless Communications* (Prentice Hall, 2014) and *Foundations of MIMO Communication* (Cambridge University Press, 2018). In 2017, he was selected as a fellow of the National Academy of Inventors. He has been the coauthor of a number award winning conference and journal papers, including recently the 2016 IEEE Communications Society Fred W. Ellersick Prize, the 2016 IEEE Communications and Information Theory Societies Joint Paper Award, the 2017 Marconi Prize Paper Award, and the 2019 IEEE Communications Society Stephen O. Rice Prize. He received the 2017 EURASIP Technical Achievement Award and the 2019 IEEE Kiyo Tomiyasu Award. He was a Distinguished Lecturer and a member of the Board of Governors in the IEEE Signal Processing Society. He is also a Licensed Amateur Radio Operator, a Private Pilot, and a Registered Professional Engineer, TX. He is also the Editor-in-Chief of *IEEE Signal Processing Magazine* and is a Member-at-Large of the IEEE Communications Society Board of Governors.