Optimal Radio Resource Management in 5G NR featuring Network Slicing

Karim Boutiba*, Miloud Bagaa†, and Adlen Ksentini*
* EURECOM, France. firstname.lastname@eurecom.fr
† Université du Québec
à Trois-Rivières, QC, Canada. miloud.bagaa@uqtr.ca

***Abstract*—3GPP 5G New Radio (NR) has introduced several new features that the network slicing concept can leverage to guarantee the heterogeneous requirements in terms of throughput and delays of expected 5G network services. Mainly, these features are (i) Mixed-numerology to control the time slot duration, hence guaranteeing low latency requirements; (ii) Bandwidth Parts (BWP) to control the number of radio resources allocated to users, hence satisfying different throughput requirements. However, efficient radio resource management is already complex, and adding these new dimensions will further increase this complexity. In this paper, we first propose modeling radio resource management in 5G NR featuring network slicing through a Mixed Integer Linear Program (MILP). For our best knowledge, this is the first MILP modeling of the radio resource management featuring network slicing taking into account (i) Mixed-numerology, (ii) both latency and throughput requirements (iii) multiple slice attach per UE (iv) Inter-Numerology Interference (INI). After showing that solving the problem takes an exponential time, we considered a new approach to solve it in a polynomial time, which is highly required when scheduling radio resources. The new approach consists of formalizing this problem using a Deep Reinforcement Learning (DRL)-based solver. We evaluate the use of RL to solve the problem for different network configurations and compared its performance with the optimal solution obtained by solving the MILP problem.**

***Index Terms*—5G NR, Network Slicing, Radio Resource Scheduling, Numerology, Bandwidth Parts.**

I. Introduction

Even though 5G is still not fully exploited commercially, the research community started to define the basis of 6G. Many features envisioned in 6G leverage the 5G capabilities, such as network slicing. The latter was introduced as a critical feature of 5G to support the emerging 5G services, such as Augmented and Virtual Reality (AR/VR), autonomous driving, industry 4.0, etc., which presented new challenges to the mobile network, such as low latency communications, high data rate, and increased reliability.

Network slicing allows creating virtual networks on top of the same infrastructure. Each virtual network is considered as a network slice and tailored to satisfy the service or application needs. Considering the concept of network slicing, 3GPP has classified 5G services into three distinct classes according to their communication service requirement: (i) enhanced Mobile BroadBand (eMBB);(ii) ultra-Reliable Low Latency (uRLLC); (iii) massive Machine Type Communications (mMTC). Network slicing can help to reduce CAPEX/OPEX as one physical infrastructure is shared efficiently to fulfill the heterogeneous communication service requirements of emerging network services. It is worth mentioning that an application or a network service can rely on more than one network slice. An example is the case of autonomous drones for surveillance, which require a eMBB service to stream a high-quality video stream and a uRLLC service to control the drone. Therefore, from the side of User Equipment (UE), it can be part to parallel network slices, up to 8 different network slices according to [1].

On the Radio Access Network (RAN) side, enforcing network slicing is more challenging as the radio resources are very limited and the new features introduced by 5G NR make the management of radio resources more complex. Particularly, mixed-numerology is one of the main features of 5G NR. It aims to control the physical resource block (PRB) shape in time and frequency. Using a numerology (μ), the time duration of the PRB is scaled down by a factor 2^{μ}, and as an effect, the frequency size is scaled up by 2^{μ}. Hence, using higher numerology decreases the RAN latency, but it increases (1) the energy consumption since UEs and gNB execute different RAN functions 2^{μ} more times per time unit; (2) the processing efforts as the RAN functions need to be executed in less than $1/\ 2^{\mu}$ ms. Moreover, a UE can not use more than one numerology at a given time slot due to physical layer constraints. Besides, the coexistence of mixed numerology in the same carrier introduces non-orthogonality to the system and causes inter-numerology interference (INI) as subcarriers associated with different numerology will no longer be orthogonal to each other. In this context, the 5G NR radio management system should cautiously select the numerology per UE while serving different network slices. While the eMBB slices need more bandwidth to satisfy the high throughput requirements, the uRLLC slices require a shorter time slot duration to ensure a lower latency requirement, and mMTC slices need better frequency management to sustain a massive number of connected devices. By adding the network slice dimension, the radio resource management in 5G NR becomes more complex compared to the classical resource management (used in 4G and 5G without network slicing). Indeed, it should consider not only (i) time-domain scheduling, (ii) frequency-domain scheduling, but also (iii) numerology selection and (iv) multiple slice management per UE. Several works from the state of the art tackled the radio resource management problem in 5G NR. The authors of [2] proposed a mixed-integer linear program to distribute the available

bandwidth amongst the UEs for different channel conditions and taking into account the INI. However, the numerology was fixed per UE, and all the UEs had the same requirements. Authors in [3] introduced an optimization method for resource and numerology allocation in multi-UE scenarios. They have modeled the problem as a multi-scenario max-min Knapsack problem, which has been solved by an integer programming solution. But, they considered a small bandwidth (10.08 MHz) and a small number of UEs (12 UEs). In [4], the optimization problem is formulated as an integer linear program. To reduce the computational complexity of the optimization, the authors proposed a linear relaxation of the problem. However, they ignored the latency, which is the main criteria in uRLLC services, and did not consider the coexistence of the different types of 5G services.

In this paper, we address the challenging problem of resource management in 5G NR featuring network slicing by first modeling it as a mixed-integer linear program taking into account: i) multiple numerology in the same bandwidth while avoiding the INI; ii) multiple slices attached per UE; iii) different throughput and latency requirements per slice. Then, we model the problem as a Mixed Integer Linear Program (MILP), we analyze its complexity, and we prove that the problem is NP-hard; i.e., solving it will take a considerable amount of time, which is not tolerable when scheduling radio resources in real-time. Reinforcement Learning (RL) that can be seen as a learning, heuristic search strategy when it is applied to optimization problems. Accordingly, we formalize this problem in the RL framework and propose a Deep Reinforcement Learning (DRL)-based approach, where a first version has been presented in [5]. The approach aims to select the numerology to be used and the number of resources allocated per UE at each time slot during a time window while taking into account the channel quality of the UE. Specifically, we designed the DRL-scheduler to be independent of the number of UEs in the system. We have modeled the DRL state to make the solution scalable for larger bandwidths covering both FR1 and FR2 frequency bands, with a bandwidth up to 400 MHz, which correspond to the usage of the mmWave bands.

The main contributions of the paper are summarized as follows:

- Model the radio resource management problem in 5G NR featuring network slicing using MILP and prove it is a NP-hard problem. The novelty of the problem formulation compared to the state of the art models, such as [6], [7], [8], [9], consists in selecting both the numerology and number of PRBs dynamically in each time slot to satisfy slices throughput and latency requirements while supporting multiple slices per UE.
- Devise a solution to solve the problem in a polynomial time. The solution is based on Machine Learning (ML), namely DRL. The novelty of the DRL approach is its ability to: (i) solve unseen instances of the problem in reasonable time, (ii) scale with the bandwidth and the number of UEs and (iii) dynamically select the numerology per UE while scheduling the PRBs in time and frequency.
- Extensive simulation campaign to compare the two approaches: (i) linear programming approach (ii) DRL approach, and evaluate their performances to guarantee the network slice requirements. We observed that approach (i) takes exponential time regarding the number of UEs while approach (ii) reduces the execution time down to 1ms. Furthermore, both approaches are able to meet the requirements of the slices in terms of throughput and latency considering different network configurations.

II. Background

In 4G, radio resources are assigned to UEs every 1 ms intervals; known as TTI (Time Transmission Interval). A low-latency-demanding service has to spend at least one millisecond in the queue to get the required radio resources, which may not be tolerable by uRLLC services that require a RAN latency of less than 1ms. In this context, 5G NR numerology come to make radio resource allocation more flexible. 5G NR numerology reshape radio units in time and frequency. It reduces the TTI to 2, 4, 8, 16 times smaller than the 4G's 1 ms. In 5G NR, each numerology μ is defined by a Sub-Carrier Spacing (SCS) and Cyclic Prefix (CP) [10]. 5G NR Release-15 [11] specifies five main numerology (μ) and defines an SCS of $15 * 2^{\mu}$ kHz and a slot duration of $1/2^{\mu}$ ms, allowing to reduce the access latency considerably at the RAN. Figure 1 illustrates the shape of PRBs under numerology 0,1,2 and 3.

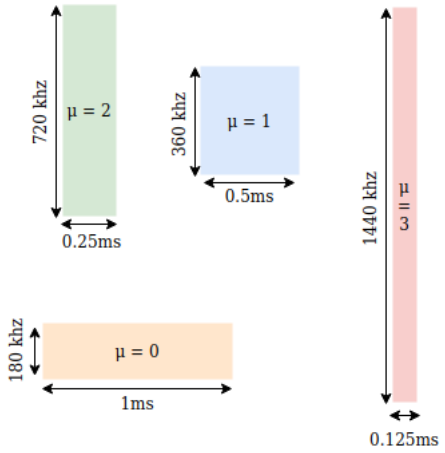


Figure 1: Numerology illustration

Bandwidth Parts (BWP) are subsets of contiguous PRBs allocated per UE, i.e., the UE expects to use resources only in a specific part of the bandwidth. BWPs aim to enable flexible assignment and configuration of PRBs. Each BWP has its own numerology, enabling more efficient spectrum sharing among the heterogeneous services in 5G RAN and hence among slices. For instance, dynamically controlling the numerology and size of a BWP will define the delay and the throughput of the service, respectively, at the RAN level.

Figure 2 shows BWPs, with different numerology, defined in the time domain (x-axis) and frequency domain (y-axis). The BWPs concept with mixed numerology enables a dynamic allocation of numerology and PRBs. A guard-band

is needed between BWPs that overlap in time and have different numerology due to the INI effect. Hence, a UE can benefit from more than one service with different numerology values. Besides, BWPs can participate in reducing UEs' power consumption since the latter can only operate on a part instead of processing the whole bandwidth. The power-saving schemes with UE adaptation to BWP bandwidth introduced in 5G NR standards [12] show 16% - 45% power-saving gain.

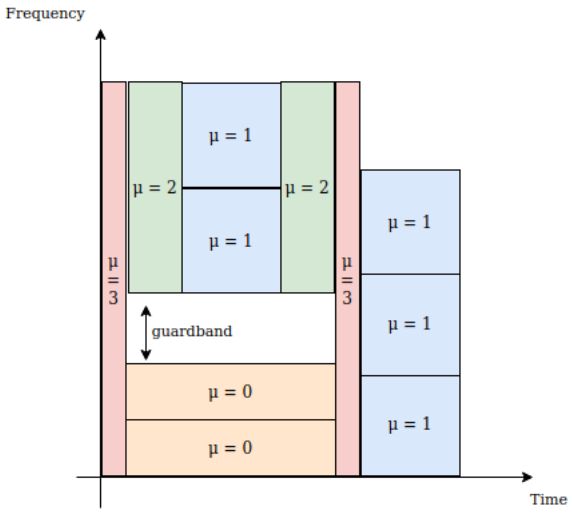


Figure 2: Resource grid illustration

It is worth noting that 5G NR specifications [13] covers the mechanism of configuration and activation of BWPs via RRC and DCI. However, deciding the size and the numerology of BWPs and BWPs scheduling decisions are still open problems.

III. Related Works

The authors in [6] propose a two-level RAN slicing approach based on the O-RAN architecture to allocate the communication and computation RAN resources among uRLLC end-devices. They modeled the resource slicing problem as a single-agent Markov Decision Process (MDP) and designed a deep reinforcement learning algorithm to solve it. However, they considered only numerology 0. Further, they did not provide details on the model scalability in terms of bandwidth and number of UEs; up to 16 UEs and a fixed bandwidth that was not defined in the paper. The work in [7] designed a deep reinforcement learning model able to solve the radio resource scheduling problem in 5G networks under different fixed numerology settings. Nevertheless, the proposed solution did not consider the use of different numerology under the same bandwidth. The authors in [8] formulated a binary non-convex problem that maximizes the aggregate capacity of multiple network slices. The model exploits the channel fading statistic to provide a spectrum allocation that minimizes the INI. The authors leveraged deep reinforcement learning to design a model-free solution computation. However, the presented problem assumed that all UEs belong to one and only one slice and have the same numerology, making the system less flexible. In [9], the authors formulated the resource allocation problem as a non-linear binary program and proved its NP-hardness. They modeled the problem as an MDP. They leveraged the exponential-weight algorithm for exploration and exploitation (EXP3) as well as the multi-agent deep Q-learning (DQL) algorithm to solve the single-agent MDP and the multi-agent MDP, respectively. But, they did not consider multiple numerology and the high dimension of the problem (only eight UEs were considered).

The work in [14] optimized resource allocation with flexible numerology in the frequency domain and variable frame structure in the time domain, with different types of requirements. They prove the NP-hardness of the problem and propose a scalable optimization algorithm based on Linear Programming (LP) and Lagrangian Duality (LD). However, they did not take into account the interference caused by the mixed numerology environment. Besides, the gap between the optimal solution and the proposed solution depends on the required throughput (30% for a 0.5Mbps rate), making the solution unsuitable for high throughput requirements as expected in 5G. In [15], the authors modeled the resource allocation problem for mixed-numerology systems as a multi-objective optimization problem aiming to increase cell throughput, maintain fairness, and minimize the delay and packet loss. They proposed a heuristic-based solution to perform numerology multiplexing as well as resource allocation taking into account the Quality of Service (QoS) and channel quality. Nevertheless, the latency requirement was ignored and the comparison between the optimal solution of their model and the solution of the proposed method was not addressed. Work in [16] designed a random forest-based decision algorithm to accomplish the numerology selection for each service. Then, the numerology selection results will be the basis of system resource scheduling and allocation. However, the solution did not consider the frequency efficiency and did not model the problem formally. Also, they did not compare their solution with the optimal solution of their problem. In [17], the authors studied radio resource allocation for mMTC services based on a mixed-numerology system. They followed an efficient heuristic approach to meet diverse QoS requirements of the Machine-to-Machine (M2M) applications while achieving spectral efficiency. However, they ignored the time domain in the problem definition.

All the above works were investigated only on small instances of the problem (small bandwidths and small number of UEs), and did not take into consideration UEs belonging to multiple slices with different requirements. Furthermore, most of DRL based solutions ignored the mixed numerology environment for the sake of reducing the model complexity. In our work, we consider large instances of the problem with up to 400 MHz of bandwidth, which is the maximum bandwidth in current 5G NR standards. We adapted the state representation of the DRL solution to be scalable regardless of the problem size. In addition, we considered having multiple numerology in the same bandwidth, divided by BWPs, while users can switch numerology to satisfy their heterogeneous needs.

IV. System Model and Problem Formulation

A. System model

We consider a network that consists of a set of UEs $\mathcal{N}$ that compete to access the radio resources. The UEs use different network slices, each characterized by different objectives, characteristics, and Service Level Agreement (SLA). Each UE can be a member of one or multiple network slices simultaneously. The SLA of a slice consists of a set of Key Performance Indicators (KPIs), with each KPI having a target value. In our system, we consider two KPIs: (i) throughput, (ii) latency. For instance, uRLLC slices have a lower target KPI value of latency, while eMBB slices have a higher target KPI value of throughput. The radio resources are divided on the UEs on time window Δ_T and frequency bandwidth Δ_F. Formally, PRBs should be assigned to the UEs within the matrix of shape $\Delta_T \times \Delta_F$ illustrated in Figure 3. Let δT and δF be the minimum allocation units in time and frequency domains, respectively. Formally, $\delta T = 2^{-\mu_{max}}$ ms and $\delta F = 12 * 15 * 2^{\mu max}$ Hz, where μ_{max} is the maximum numerology in the system. Let $\mathcal{N}$ denote the set of UEs in the network. Each UE has a set of time slots to transmit both uplink and downlink traffics. Let $\mathcal{K}$ denote the maximum number of time slots that can be assigned to a UE. To transmit their data, UEs use a set of numerology $\mathcal{M} = \{0, 1, 2, 3, 4\}$. At a given time slot, the UE should use one and only one numerology. However, it can use more than one resource blocks that should be contiguous for the sake of performance.

Table I: Summary of Notations.

Notation	Description
$\mathcal{N}$	The set of UEs existing in the network.
Ω	The maximum number of PRBs assigned to the same UE at the same time slot.
Δ_T	The time slot window of the resource blocks.
Δ_F	The frequency band of the resource blocks.
$\mathcal{M}$	The set of numerology.
μ	A specific numerology. Formally, $\mu \in \mathcal{M}$.
$\mathcal{T}_i^k$	An integer variable that denotes the starting time of the k^{th} time slot of the UE $i \in \mathcal{N}$.
$\mathcal{F}_i^k$	An integer variable that denotes the starting frequency assigned to the UE $i \in \mathcal{N}$ at time slot k.
$\mathcal{X}_{i,\omega}^{k,\mu}$	A decision Boolean variable that shows if a UE i at the time slot k uses ω contiguous PRBs of numerology μ. ω takes values in the range from 1 to Ω.
$\mathcal{Y}_{i,j}^{k,l}$	A decision Boolean variable that shows if the k^{th} and l^{th} time slots of UE i and j overlap.

The size of a resource block in terms of slot duration $\mathcal{E}(\mu)$ (in ms) and frequency $\mathcal{G}(\mu)$ (in Hz) of a numerology μ are defined as follows:

$$\mathcal{E}(\mu) = 2^{-\mu} \tag{1}$$

$$\mathcal{G}(\mu) = 12 \times 15 \times 2^{\mu} \tag{2}$$

As we have explained previously, a UE at a given time slot can have one or multiple contiguous PRBs $(\omega \times \mathcal{G}(\mu))$ that use the same numerology μ. The PRBs of the same UE or different UEs should never overlap. Formally, a UE i never shares the frequency bound with another UE j at the same time.

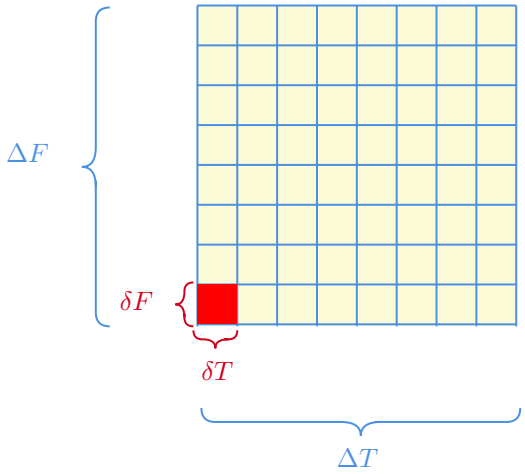


Figure 3: Resource matrix illustration for the 5G NR featuring network slicing resource management Model

B. Problem formulation and optimal solution design

For the sake of readability, all the symbols and optimization variables are summarized in the table I. Let $\mathcal{T}_i^k$ be an integer variable that denotes the starting time of the k^{th} time slot of the UE $i \in \mathcal{N}$. Note that not all the time slots $\{1, 2, \cdots \mathcal{K}\}$ should be assigned to the UE i. If a time slot k is not assigned to the UE i, then $\mathcal{T}_i^k = -1$. Similarly, let $\mathcal{F}_i^k$ be an integer variable that denotes the starting frequency assigned to the UE $i \in \mathcal{N}$ at time slot k. Each UE can have maximum Ω PRBs at the same time slot. Moreover, all the PRBs of the same UE at the same time slot should use the same numerology. If a frequency $\mathcal{F}_i^k$ is not used by the UE i, then $\mathcal{F}_i^k = -1$.

Formally, we have the following two constraints that ensure the time slot and frequency starting value should be higher or equal to -1:

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K}\} : \mathcal{T}_i^k \geq -1 \tag{3}$$

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K}\} : \mathcal{F}_i^k \geq -1 \tag{4}$$

Let $\mathcal{X}_{i,\omega}^{k,\mu}$ a decision Boolean variable that shows if a UE i at the time slot k uses ω contiguous PRBs of the same numerology μ. At a given time slot k, only one numerology can be allocated to UE i. However, many resource blocks can be assigned to the same UE. This constraint is captured thanks to the constraint (5).

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K}\} : \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu} \leq 1 \tag{5}$$

The UE i uses the numerology μ at the time slot k if and only if $\sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu} = 1$.

The time slot duration allocated to UE i transmission must not exceed Δ_T. In this constraint, both the k^{th} starting

time slot (i.e., $\mathcal{T}_i^k$) and time slot duration $\mathcal{E}(\mu)$ using the numerology μ are considered.

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K}\} : \mathcal{T}_i^k + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{E}(\mu) \times \mathcal{X}_{i,\omega}^{k,\mu} \leq \Delta_T \quad (6)$$

From another side, the frequency should not exceed Δ_F. In constraint (7), at each time slot $\mathcal{T}_i^k$, both the starting frequency bound $\mathcal{F}_i^k$ and the amount of frequency bound $(\omega \times \mathcal{G}(\mu))$ are considered.

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K}\} : \mathcal{F}_i^k + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \omega \times \mathcal{G}(\mu) \times \mathcal{X}_{i,\omega}^{k,\mu} \leq \Delta_F \quad (7)$$

$\mathcal{T}_i^k$ should equal to -1 if time slot k is not allocated to UE i.

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K}\} : \mathcal{T}_i^k \leq -1 + \Phi \times \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu} \quad (8)$$

, such that Φ is a big number $\Phi \approx +\infty$.

Based on (3) and (8), $\mathcal{T}_i^k$ equals to -1 if no numerology is used by that time slot
$(\sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu} = 0)$.

Similarly, the frequency $\mathcal{F}_i^k$ should equal -1 if it does not use any numerology at the time slot $\mathcal{T}_i^k$.

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K}\} : \mathcal{F}_i^k \leq -1 + \Phi \times \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu} \quad (9)$$

Based on (4) and (9), $\mathcal{T}_i^k$ equals to -1 if no numerology is used by that time slot.

A UE should not use two numerology at the same time. Also, the time slots should not overlap with each other. In our solution, the time slots of a UE are defined from the smaller to the last. Then, the remaining time slots should have only -1.

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K} - 1\} : \mathcal{T}_i^k + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{E}(\mu) \times \mathcal{X}_{i,\omega}^{k,\mu} \leq \mathcal{T}_i^{k+1} + \Phi \times (1 - \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{X}_{i,\omega}^{k+1,\mu}) \quad (10)$$

Equation 11 ensures that only the first slots assigned for a UE i. In our solution, we fill the first slots before moving to the last ones.

$$\forall i \in \mathcal{N}, \forall k \in \{1, 2, \cdots \mathcal{K} - 1\} : \sum_{\mu \in \mathcal{M}} \mathcal{X}_i^{k+1,\mu} \leq \sum_{\mu \in \mathcal{M}} \mathcal{X}_i^{k,\mu} \quad (11)$$

Equation 12 ensures that two UEs i and j should not use the same frequency resources if their time slots overlap.

$$\forall i, j \in \mathcal{N}, i \neq j, \forall k, l \in \{1, 2, \cdots \mathcal{K} - 1\} : \text{If } \mathcal{T}_j^l \leq \mathcal{T}_i^k \leq \mathcal{T}_j^l + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{E}(\mu) \times \mathcal{X}_{i,\omega}^{k,\mu} : \mathcal{F}_j^l + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \omega \times \mathcal{G}(\mu) \times \mathcal{X}_{j,\omega}^{l,\mu} < \mathcal{F}_i^k$$

OR

$$\mathcal{F}_i^k + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \omega \times \mathcal{G}(\mu) \times \mathcal{X}_{i,\omega}^{k,\mu} < \mathcal{F}_j^l \quad (12)$$

Unfortunately, the above inequality is not linear. In order to convert the optimization problem to a linear integer program, we replace the constraint (12) by the following constraints and variables:

First, we define a Boolean variable $\mathcal{Y}_{i,j}^{k,l}$ that shows if the k^{th} and l^{th} time slots of UE i and j overlap, respectively. Then, we define the following constraints:

$$\begin{aligned}
&\forall i, j \in \mathcal{N}, i \neq j, \forall k, l \in \{1, 2, \cdots \mathcal{K}\} : \\
&\mathcal{T}_j^l \leq \mathcal{T}_i^k + \Phi \times (1 - \mathcal{Y}_{i,j}^{k,l}) \\
&\mathcal{T}_i^k \leq \mathcal{T}_j^l + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{E}(\mu) \times \mathcal{X}_{j,\omega}^{l,\mu} + \Phi \times (1 - \mathcal{Y}_{i,j}^{k,l}) \\
&\mathcal{T}_i^k + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{E}(\mu) \times \mathcal{X}_{i,\omega}^{k,\mu} < \mathcal{T}_j^l + \Phi \times (\mathcal{Y}_{i,j}^{k,l} + \mathcal{Z}_{i,j}^{k,l}) \\
&\mathcal{T}_j^l + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{E}(\mu) \times \mathcal{X}_{j,\omega}^{l,\mu} < \mathcal{T}_i^k + \Phi \times (\mathcal{Y}_{i,j}^{k,l} + 1 \\
&- \mathcal{Z}_{i,j}^{k,l}) \quad (13)
\end{aligned}$$

, such that $\mathcal{Z}_{i,j}^{k,l}$ is a decision Boolean variable that should be fixed by the system.

From (13), $\mathcal{Y}_{i,j}^{k,l} = 1$ iff $\mathcal{T}_j^l \leq \mathcal{T}_i^k \leq \mathcal{T}_j^l + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \mathcal{E}(\mu) \times \mathcal{X}_{j,\omega}^{l,\mu}$. Otherwise, $\mathcal{Y}_{i,j}^{k,l} = 0$, the time slots do not overlap. In case the time slots overlap, we have to ensure that they do not use the same frequency resources.

$$\begin{aligned}
&\forall i, j \in \mathcal{N}, i \neq j, \forall k, l \in \{1, 2, \cdots \mathcal{K}\} : \\
&\mathcal{F}_j^l + \sum_{\mu \in \mathcal{M}} \mathcal{I}_{i,j}^{k,l,\mu} \times G + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \omega \times \mathcal{G}(\mu) \times \mathcal{X}_{j,\omega}^{l,\mu} < \\
&\mathcal{F}_i^k + \Phi \times (1 - \mathcal{Y}_{i,j}^{k,l} + \mathcal{W}_{i,j}^{k,l}) \\
&\mathcal{F}_i^k + \sum_{\mu \in \mathcal{M}} \mathcal{I}_{i,j}^{k,l,\mu} \times G + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1, \cdots \Omega\}} \omega \times \mathcal{G}(\mu) \times \mathcal{X}_{i,\omega}^{k,\mu} < \\
&\mathcal{F}_j^l + \Phi \times (2 - \mathcal{Y}_{i,j}^{k,l} - \mathcal{W}_{i,j}^{k,l}) \quad (14)
\end{aligned}$$

, such that $\mathcal{W}_{i,j}^{k,l}$ is a Boolean variable that should be fixed by the system to ensure that the frequency i and j do not overlap when their time slots do.

Moreover, UEs sharing contiguous frequencies and having different numerology should be separated by a guard band G to avoid the INI. $\mathcal{I}_{i,j}^{k,l,\mu}$ is a Boolean variable fixed by the system: if $\mathcal{I}_{i,j}^{k,l,\mu} = 1$ a guard band is needed between UE i

and j during slots k and l when they are not using the same numerology μ (Equation 15).

$$\begin{aligned}&\forall i,j \in \mathcal{N}, i \neq j, \forall k,l \in \{1,2,\cdots \mathcal{K}\}, \forall \mu \in \mathcal{M}: \\ &\mathcal{I}_{i,j}^{k,l,\mu} \geq \sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{X}_{j,\omega}^{l,\mu} - \sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu} \\ &\mathcal{I}_{i,j}^{k,l,\mu} \geq \sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu} - \sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{X}_{j,\omega}^{l,\mu}\end{aligned} \tag{15}$$

Each UE i has a maximum latency Δ_i.

$$\begin{aligned}&\forall i \in \mathcal{N}, \forall k \in \{1,2,\cdots \mathcal{K}\}: \\ &\mathcal{T}_i^k + \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{E}(\mu) \times \mathcal{X}_{i,\omega}^{k,\mu} < \Delta_i\end{aligned} \tag{16}$$

Each UE i has a minimum throughput, which is translated to the number of required resource blocks N_PRB_i taking the Modulation and Coding Scheme (MCS) as input for each UE i during the time window ΔT. It should be noted that a way to derive the number of PRBs using MCS and throughput is presented in [13]. The system that solves our model is responsible for providing this information as input, and can be dynamic between different time windows.

$$\forall i \in \mathcal{N}: \sum_{k \in \{1,\cdots \mathcal{K}\}} \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1,\cdots \Omega\}} \omega \times \mathcal{X}_{i,\omega}^{k,\mu} \geq N\_PRB\_i \tag{17}$$

In our solution, we consider network slicing, such as one UE can have multiple network slices. To do so, a (physical) UE is divided into multiple logical UEs. Each logical UE represents a slice belonging to a UE. Therefore, a UE with multiple slices will have as many logical UEs as slices. For simplicity, we consider logical UEs like UEs with additional constraints (18 and 19), with logical UEs belonging to the same UE forming a group. Formally, a UE $i \in \mathcal{N}$ is a logical representation of a slice belonging to UE $k \in \mathcal{N}_{ph}$ where $\mathcal{N}_{ph}$ denotes the set of physical UEs. For instance, two logical presentations of a UE $i,j \in \mathcal{N}$ and $i \neq j$ can belong to the same physical UE $k \in \mathcal{N}_{ph}$. To prevent a UE $i \in \mathcal{N}$ from using different numerology simultaneously, we have considered G, whereby the same logical presentations of the same physical UE create a group $g \in G$. Constraints 18 and 19 jointly ensure that the logical presentations of the same physical UE do not use different numerology when their time slots overlap.

$$\begin{aligned}&\forall g \in \mathcal{G}, \forall i,j \in g, i \neq j, \forall k,l \in \{1,2,\cdots \mathcal{K}\}, \forall \mu \in \mathcal{M}: \\ &\sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{X}_{j,\omega}^{l,\mu} - \sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu} \leq 1 - \mathcal{Y}_{i,j}^{k,l} \\ &\mathcal{Y}_{i,j}^{k,l} - 1 \leq \sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{X}_{j,\omega}^{l,\mu} - \sum_{\omega \in \{1,\cdots \Omega\}} \mathcal{X}_{i,\omega}^{k,\mu}\end{aligned} \tag{18}$$

$$\begin{aligned}&\forall g \in \mathcal{G}, \forall i,j \in g, i \neq j, \forall k,l \in \{1,2,\cdots \mathcal{K}\}: \\ &\mathcal{T}_i^k - \mathcal{T}_j^l \leq \Phi \times (1 - \mathcal{Y}_{i,j}^{k,l}) \\ &\Phi \times (\mathcal{Y}_{i,j}^{k,l} - 1) \leq \mathcal{T}_i^k - \mathcal{T}_j^l\end{aligned} \tag{19}$$

Finally, the objective is to maximize the bandwidth utilization. In the optimization model, we do not consider priorities between UEs. However, the priority can be easily considered by a slight update of the equation 20.

$$\mathbf{max} \sum_{k=1}^{k=\mathcal{K}} \sum_{\mu \in \mathcal{M}} \sum_{\omega \in \{1,\cdots \Omega\}} \omega \times \mathcal{X}_{i,\omega}^{k,\mu} \tag{20}$$

Now, we are able to formulate the final optimization problem as follows:

Eq. (20)

S.t,

Eq.: (3), (4), (5), (6), (7), (8), (9), (10), (11), (13), (14), (15), (16), (17), (18), (19).

Theorem 1. *Radio resource allocation problem in 5G NR is NP-hard problem.*

Proof. Let P1 denotes the problem of filling the radio resource matrix with BWPs $i \in \{1..n\}$. Let w_i denotes the frequency occupied by each BWP i. Each BWP i is dedicated to a UE j and has a start time t. For the sake of simplicity and without loss of generality, we assume that only one numerology can be used, and the BWPs that exceed UE j latency SLA can be omitted. Let v_i denotes the size of the BWP i. Meanwhile, let P2 denotes the knapsack problem, which is defined as follows: Given a set of objects, each of which has a weight and value, the problem consists in determining the number of each item that should be put in the knapsack so that the total weight does not exceed a specific weight and the total value is maximized. The optimization of knapsack problem is well known in the literature that it is NP-hard problem. To prove that P1 is NP-hard, it is sufficient to proof that the knapsack problem P2 would be reduced to P1 in a polynomial time. If P2 is reduced to P1 in a polynomial time and P1 is not NP-hard, then P2 is also not NP-hard, which is a contradiction. P2 can be reduced in a polynomial time to P1 by formulating P2 as follows: The total weight that can be handled by the knapsack is ΔF and ΔT (two dimensions) and the set of items that should be put in knapsack is the set of BWPs i, such that the sum of v_i is maximized. □

V. Deep Reinforcement Learning based Radio Scheduler (DRL-RS)

A. DRL-RS general overview

Despite seeing great improvement throughout the years in the ability to find optimal solutions for bigger and more complex problems, the time and the computational power needed to find a solution remain an important bottleneck, making this type of algorithms impractical to solve real-world problems. RL intends to learn the skills required to solve the problem more holistically rather than trying to find an optimal solution for one specific configuration of the problem. Where the other combinatorial optimization solving methods will need to solve the problem from scratch every time, the RL approach leverages what it has learned before and the skills it has acquired to quickly provide a solution to a new instance of the problem. Radio resource management

problem in 5G NR can be seen as a sequential decision-making problem where an agent, namely DRL-RS, allocates PRBs and selects numerology for each UE and time slot during ΔT time window. DRL-RS considers the environment as a $\Delta F \times \Delta T$ matrix and aims to fill it by allocating PRBs to a dynamic number of UEs in the system. DRL-RS takes the Modulation and MCS to use during ΔT as input to ensure the link adaptation for dynamic channels.

As each UE can belong to multiple slices, DRL-RS should satisfy the throughput and the latency of the Service Level Agreement (SLA) for each slice. The throughput SLA is the minimum throughput that needs to be achieved by a UE for that slice, while the latency SLA is the maximum latency that a UE can not exceed when serving that slice. To allow a UE to belong to multiple slices, we introduce the concept of virtual UEs. Formally, each UE consists of a set of virtual UEs that belong to only one slice. Virtual UEs belonging to the same UE are organized in groups. The time slots of members of the same group should neither overlap in time nor use different numerology. From this point, we mention by UE a real or virtual UE indifferently. In this scope, DRL-RS loops over the active UEs (i.e., UEs having data in their transmission queues and their SLA is not satisfied yet) until the resource matrix is filled or all the UE's SLAs are met. In each iteration, the scheduler chooses a time slot t, a numerology μ, and a number of resources N for the current UE. The three parameters form a rectangle of shape $2^{\mu_{max}-\mu} * (N * 2^{\mu})$ which will be stacked in the resource matrix at time slot t.

We illustrate how DRL-RS works through a simple example illustrated in Figure 4. We consider a $1.44MHz \times 1ms$ matrix shared between 4 UEs. Each UE has a latency SLA of 0.125ms, 1ms, 0.25ms, 0.5ms, and a throughput SLA that can be met with 1, 3, 1, 2 PRBs, respectively. The x-axis represents time slots where the unit is related to the maximum numerology in the system (i.e., 3 in this example) and the y-axis represents frequency PRBs where the unit is related to the minimum numerology in the system (i.e., 0 in this example). In the first iteration, for the first UE, DRL-RS allocates 1 PRBs with numerology 3 at t = 0. The selection of numerology 3 allows the first UE to respect its latency SLA. Then, DRL-RS allocates 2 PRBs for the 2^{nd} UE at t = 4 with numerology 1. This choice will allow the third and fourth UEs to respect their latency SLA, which shows that DRL-RS gives the current UE an insight into the state of other UEs. After that, the third UE takes 1 PRB using numerology 3 at $t = 1$, while the 4^{th} UE takes 1 PRB at $t = 2$ using numerology 2. The stop condition is not met yet; hence, DRL-RS continues looping over the active UEs that did not meet their SLAs, i.e., the 2^{nd} and 4^{th} UEs. The 2^{nd} UE takes one more PRBs at $t = 4$ with numerology 1. Finally, the 4^{th} UE takes one PRB at $t = 2$ using numerology 2. At this step, all UEs' SLAs are met, and thus DRL-RS finished the resource allocation process and successfully fulfilled all the slices' SLAs.

B. DRL-RS design

In the balance of this section, we model the radio resource management problem in the RL framework by designing the state, the reward and the action of the DRL-RS agent.

1) State: The state S^i observed by the model when serving UE i is composed of 4 vectors, which are F, T_i, O_i and E_i, respectively. Vectors F and T_i contain the frequency boundary (the red line in Figure 4) and the numerology used by UE i for each time slot t, respectively. If a UE is not using any numerology at time t, $T_{i,t}$ is set to -1. Meanwhile, vectors O_i and E_i contain the information about SLA of the current UE and the other UEs, respectively. O_i contains two values, O_i^{thg} measuring the throughput SLA and O_i^{lat} measuring the latency SLA. O_i^{thg} indicates the rate of the achieved throughput over the throughput SLA. The bigger O_i^{thg} is, the better performances of UE i becomes in terms of throughput. Since a UE can achieve a throughput higher than its throughput SLA, O_i^{thg} can have a value bigger than 1. For instance, to decrease the state space size, we can limit O_i^{thg} with a maximum value O_{max}^{thg}. On the other hand, O_i^{lat} indicates the rate of the PRBs used before the latency SLA over all the allocated PRBs for UE i. The bigger O_i^{lat} is, the better performance of the UE i becomes in terms of latency. Note that a UE will respect the latency requirements if all the allocated PRBs are scheduled before the latency SLA, which is equivalent to $O_i^{lat} = 1$. Formally,

$$O_i^{thg} = \min\{\frac{achieved\ throughput}{throughput\ SLA}, O_{max}^{thg}\} \text{ with } O_{max}^{thg} > 1$$
$$O_i^{lat} = \frac{Number\ of\ allocated\ PRBs\ before\ latency\ SLA}{Total\ number\ of\ allocated\ PRBs}$$

E_i contains three values: N_i^{thg}, N_i^{lat} and Min_i^{thg}. N_i^{thg} and N_i^{lat} count the number of UEs excluding UE i that have met their throughput and latency SLAs, respectively. Min_i^{thg} is the smallest throughput SLA achieved by other UEs.

The state's design considers the solution's scalability regarding the number of UEs, the SLA requirements, and the bandwidth size.

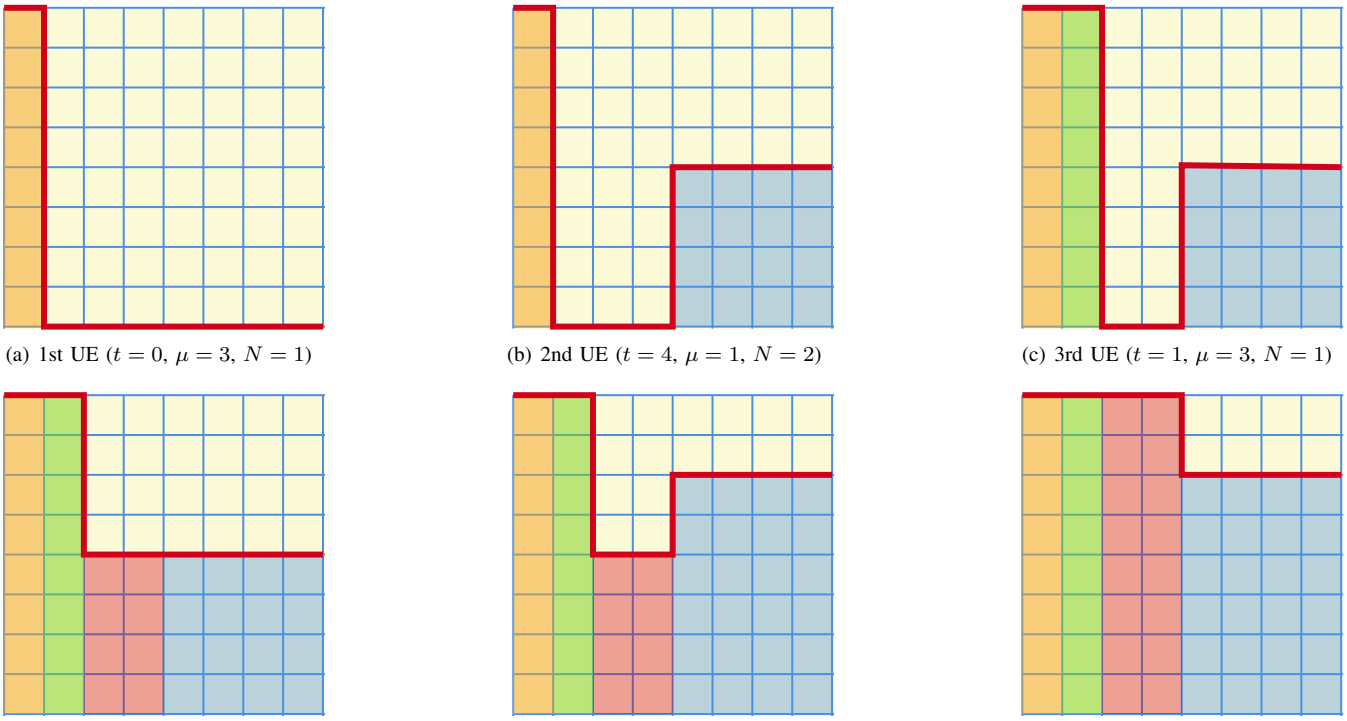
2) Action: The agent take an action parameterized by the tuple (t, μ, N), where t is a time slot, μ is the numerology to use at the given time slot t, and N is the number of resources to allocate at t. Accordingly, the agent allocates a rectangular shape of ($2^{\mu_{max}-\mu}$,$(N * 2^{\mu})$) at time slot t and put it on the top of the frequency boundary line. Since the agent can output unfeasible actions, we added a pre-processing step in order to compute an action space A that contains only the possible actions at the current state S_t^i.

3) Reward: We have adopted an episodic approach; i.e., an episode is over when max T steps are reached, the resource matrix is full, or all UEs SLAs are met.

The agent gets the reward r defined as follow:

$$r^t = \begin{cases} \alpha * (O_{i,t}^{thg} - O_{i,t-1}^{thg}) + (1-\alpha) * \mathcal{S} & \text{if not done} \\ \mathcal{K} & \text{if done and SLAs are met} \\ \mathcal{P} & \text{otherwise} \end{cases}$$

Indeed, while the episode is still in progress and for each step, the agent takes a reward equivalent to the improvement made by the current action at step t since the previous step $t-1$.



(a) 1st UE ($t = 0$, $\mu = 3$, $N = 1$)
(b) 2nd UE ($t = 4$, $\mu = 1$, $N = 2$)
(c) 3rd UE ($t = 1$, $\mu = 3$, $N = 1$)
(d) 4th UE ($t = 2$, $\mu = 2$, $N = 1$)
(e) 2nd UE ($t = 4$, $\mu = 1$, $N = 1$)
(f) 4th UE ($t = 2$, $\mu = 2$, $N = 1$), all SLAs are met

Figure 4: Example of resource allocation using DRL-RS

We formulate this improvement by the term $(O^{thg}_{i,t} - O^{thg}_{i,t-1})$. Moreover, to minimize the number of steps needed to finish an episode, we added a penalty $\mathcal{S}$ for each step. Also, we added weights α and $(1 - \alpha)$ to the two previous terms in order to control their contribution in the reward term. Once the episode is done, the agent takes a high positive reward $\mathcal{K}$ if the SLAs are met for all UEs, or a penalty $\mathcal{P}$ else.

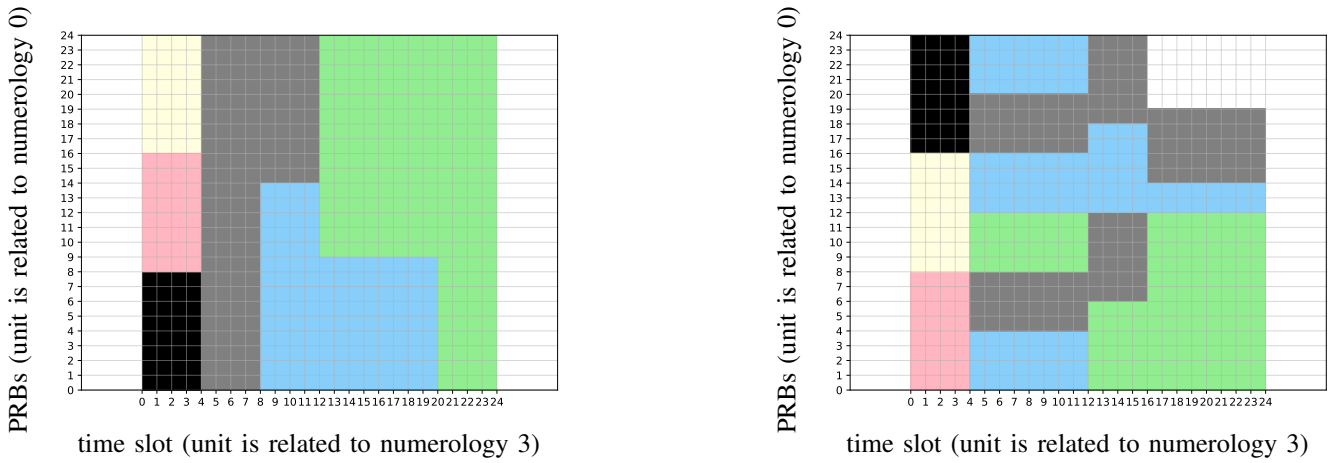
C. DRL-RS detailed description

Among the most efficient off-policy DRL algorithms for continuous state space and discrete actions, we may cite: Deep Q-Network (DQN) [18] and Asynchronous Advantage Actor-Critic (A3C) [19]. The DQN-based method has a replay memory that is sampled and used to optimize the model at every time step while A3C is only optimized when the episode finishes. This makes DQN learning faster in a single actor environment. For this reason, DRL-RS leverages the DQN algorithm.

DRL-RS executes two steps: decision making and updating the Q-Networks. In DQN, two networks are used: a local Q-Network and a target Q-Network. The latter is the same as the local network except that its parameters are updated every τ steps. They are combined to help the convergence and stabilization of the learning.

- **Decision making:** The DRL-RS agent observes a state S^i_t for UE i and feeds it to the local Q-Network. In DQN, the Q-Network outputs the Q value of each couple (S^i_t ,a) where a is an integer that identifies the action. In order to evaluate the action, the agent needs to derive the three parameters (t, μ, N) from a. To do so, we partitioned the integer value a, using the method "the partitioning of an integer into different parts" introduced in [20], as follows:
$t = a \div (\mu_{max} * N_{max})$ where N_{max} is the maximum number of resources that can be allocated to one UE and μ_{max} is the maximum numerology in the system. Note that $\div$ is the division of integer division and $\mod$ is the mod of integer division.
$\mu = a \mod (\mu_{max} * N_{max}) \div N_{max}$
$N = (a \mod (\mu_{max} * N_{max}) \mod N_{max}) + 1.$

  Then, the agent removes the unfeasible actions from the action space (e.g., allocating resources that overlap with other existing resources) by setting the Q value of each unfeasible tuple (t, μ, N), indexed by a, to a negative value. The DQN algorithm will not explore the unfeasible actions since DQN explores actions having high Q values. After that, we apply an ϵ-greedy approach to choose an action. This means that the agent will choose a random action over the feasable actions with ϵ probability and the best action over the action distribution with a 1- ϵ probability. The value of ϵ decays over time during the learning phase. It allows pushing the agent to explore the environment at the beginning of the training phase and better exploit the learned decisions over time.
- **Updating the Q-Networks:** At each step, the current state, the action, the next state, and the reward are stored in a buffer known as the replay buffer. The local Q-Network is updated using a random sample from the



(a) O-RS solution

(b) DRL-RS solution

Figure 5: Resource matrix illustration

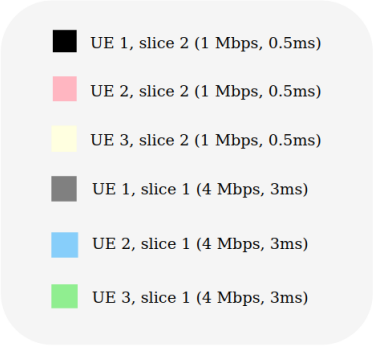


Figure 6: Virtual UEs scheduled in the resource matrix illustration example (Figure 4)

Table II: Illustration example parameters

Parameter	Value
Bandwidth	5 MHz
Maximum numerology	3
Time window	3 ms
Number of UEs	3
Number of slice per UE	2
1st slice throughput requirement	4 Mbps
1st slice latency requirement	3 ms
2nd slice throughput requirement	1 Mbps
2nd slice latency requirement	0.5 ms
MCS	26

replay buffer, which reduces the correlation between the agent's experiences and increases the stability of the learning. Using mean square error (MSE) and ADAM optimizer [21], the parameters of the local Q-Network are optimized at every step by considering the local and target values, while the parameters of the target Q-Network are smoothly updated at each step using a parameter τ.

VI. Performance Evaluation

In this section, we will introduce the simulation environment and parameters used for O-RS and DRL-RS. We note O-RS the exact solver of the MILP model. Then, we illustrate an example of the resource matrix filled with different approaches on a small instance of the problem for the sake of illustration clarity. After that, we solve bigger instances of the problem with different network configurations. We compare the different approaches in terms of efficiency, scalability, and execution time.

A. Simulation parameters

To evaluate different contributions in a 5G environment, we used the mixed-numerology 5G Simulator developed in [22] that relies on 6.1.4.2 of TS 38.214 [13] specifications to compute the Transport Block Size (TBS). The simulator takes the scheduling policy regardless of the source of the policy. For instance, we can plug O-RS and DRL-RS into the simulator and compare their results. All tests are performed on a machine with 64 CPUs, an Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz (2.7 GHz with Turbo Boost technology), and 128 GBs of RAM.

1) O-RS: The MILP problem is solved using Gurobi version 9.1.2. The absolute optimality gap is set to 10^{-8}, and the time limit to 50000 seconds. We recall that the absolute gap is the absolute difference between the best possible objective value and the objective value of the best feasible solution.

2) DRL-RS: The simulation environment is implemented using Python. We used PyTorch to implement the DRL-RS agent. We have trained the DRL-RS agent using 500 independent episodes. We have fixed the maximum number of steps at each episode T by 100. We have varied the reward parameters and chosen the values that stabilize the convergence of the model. The different considered parameters are presented in Table III. The model converges after 200 episodes.

For each Bandwidth size and ΔT, we trained an instance of the model using MCS 16, 3 UEs. Each UE is attached to one eMBB and one uRLLC slice with a fixed SLA. The sum of throughput SLA of UEs nearly equals the maximum achievable throughput in the corresponding bandwidth.

Table III: DRL-RS parameters

Parameter	Value
$\mathcal{P}$	-0.02
$\mathcal{S}$	-0.01
$\mathcal{K}$	5000
α	0.5
O_{max}^{thg}	2
Number of hidden layers	2
Hidden layer size	64 nodes
discount factor γ	0.99
Batch size	128
Learning rate	$5 * 10^{-4}$
Replay buffer size	10^9
Soft update coefficient τ	0.001
Optimizer	ADAM [21]
ϵ-start	1
ϵ-decay	0.99
ϵ-end	0.01

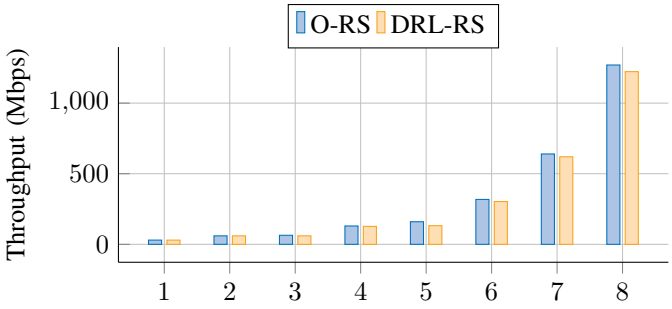


Figure 7: Aggregated throughput comparison between O-RS and DRL-RS

B. Resource grid illustration

We solve the problem instance described in Table II using O-RS and DRL-RS. We selected a small matrix for the sake of illustration clarity. Figure 5 compares the resource allocation policies generated by each approach. Figures 5(a), 5(b) illustrate the radio resource allocation matrix filled by O-RS and DRL-RS, respectively. Each virtual UE has a color, for instance we have 6 virtual UEs. We recall that each slice associated with a UE is a virtual UE. The mapping between colors and virtual UEs is illustrated in Figure 6. We observe that, for the two approaches, the black, pink and yellow UEs respect the 0.5 ms latency (resources are allocated before 0.5 ms thanks to the use of numerology 1 by different approaches) and each one uses 4 PRBs (carry 2984 bits of data per time window with MCS 26, we compute the estimated amount of data sent by second: 2984*1000/3 = 0.99 Mbps) required by the 2nd slice in Table II. Gray, blue and green UEs are the virtual UEs belonging to the 1st slice, each one respects the 3 ms latency and the 4 Mbps throughput (each UE uses more than 16 PRBs, which carry 11832 with MCS 26 per time window, and hence, the throughput per second is estimated to be more than 11832*1000/3 = 3.94 Mbps). We conclude that O-RS and DRL-RS generate different allocation policies and all these policies fulfill the slices SLA in terms of throughput and latency.

We observe that O-RS and DRL-RS dynamically selected the numerology 0 and 1 to fill the matrix. These decisions are due to the fact that DRL-RS and O-RS are designed to be able to select a numerology at each timestamp taking into consideration different constraints. For instance, DRL-RS and O-RS selected numerology 1 to satisfy the 0.5 ms latency for the second slice, and numerology 0 to satisfy the first slice which does not require higher numerology to satisfy its latency SLA.

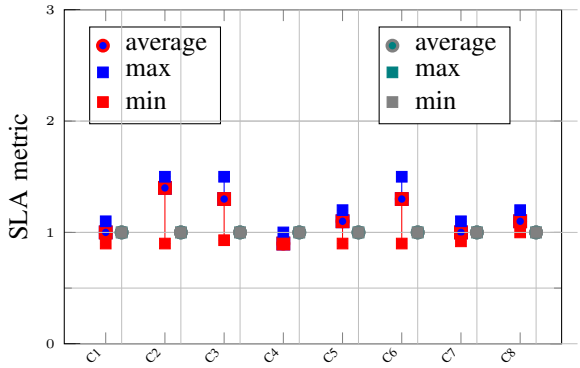


Figure 8: Throughput and Latency SLA fulfillment by DRL-RS

C. Comparison with the State of The Art

In the balance of this subsection, we compare DRL-RS results with the results of [14]. We recall that [14] proposed a scalable optimization algorithm based on Linear Programming (LP) and Lagrangian Duality (LD) (noted as LP+LD). We adapted our simulation parameters to reproduce the same experiment using a scheduling window of 2 MHz of bandwidth and 2 ms of duration. We assumed a maximum numerology of 3. We considered 10 UEs with a target throughput of 450 kbps (this is due to the limitation of LP+LD). We varied the target latency. Figure 9 depicts the achieved throughput using different latency targets. The x-axis represents the latency targets while the y-axis represents the achieved throughput in kbps. We notice that DRL-RS is able to achieve slightly higher throughput than LP+LD solution. We are not able to compare with bigger instances of the problem because [14] did not consider it. In contrast, DRL-RS considers bigger instances, which is one of the critical contributions of our work compared to state of the art solutions.

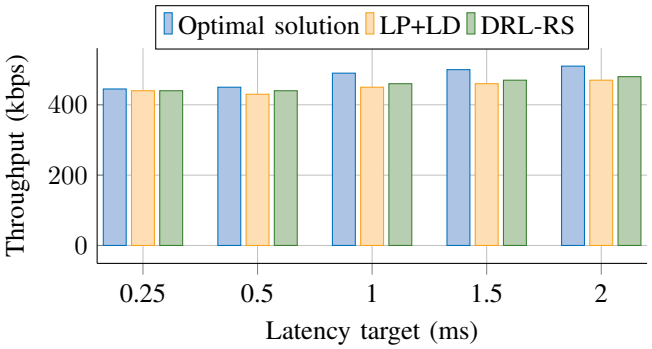


Figure 9: Average throughput per user comparison between DRL-RS and the state of The Art

Table IV: 5G NR Configurations

Index	Bandwidth (MHz)	ΔT (ms)	Number of UEs	Number of slice per UE	Slices SLA list (Mbps, ms)	MCS
1	20	3	3	2	(9,10),(0.2,1)	16
2	20	3	10	2	(5,10),(0.2,1)	26
3	40	1	10	2	(5.5,10),(0.2,0.5)	16
4	40	1	10	2	(13,10),(0.7,0.25)	26
5	100	3	3	3	(40,10),(1,2),(0.2,1)	16
6	100	3	6	3	(40,10),(2,2),(0.7,1)	26
7	400	1	5	2	(110,10),(1,0.125)	16
8	400	1	20	1	(50,10)	26

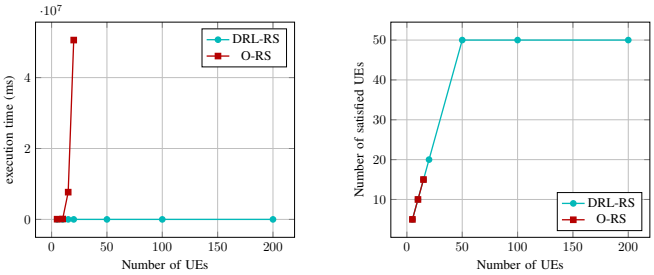
D. Efficiency and Scalability Evaluation

In this subsection, we will evaluate the effeciency and the scalability of DRL-RS. Figure 7 compares the aggregated UE throughput achieved in each configuration by O-RS and DRL-RS. We notice that the achieved throughput by DRL-RS is close to the optimal achievable throughput obtained by O-RS, which means that DRL-RS is able to fill the resource matrix efficiently while satisfying the throughput and latency SLA for all the slices (Figures 8). We observe a small gap between the performance of DRL-RS and O-RS in higher bandwidths (greater than 100 MHz). This gap is caused by the wasted resources that DRL-RS may generate due to the nature of DRL algorithms, which approximate continuous state using neural networks. Indeed, these resources cannot be used by any UE due to shape and numerology constraints.

Figure 8 depicts the achieved SLA by DRL-RS for each configuration listed in Table IV. The y-axis represents the metrics used for measuring the SLA: $\max_i O_i^{thg}$, $\min_i O_i^{thg}$, average$_i O_i^{thg}$ for the throughput SLA (blue and red in the figure) and $\max_i O_i^{lat}$, $\min_i O_i^{lat}$, average$_i O_i^{lat}$ for the latency SLA (green and grey in the figure). We remind that these values are described in Section V-B1. The SLA is assumed respected when the value of these metrics is higher than 1 with a tolerance of 5% due to the QNetworks estimation errors. The x-axis represents the configuration indexes summarized in Table IV.

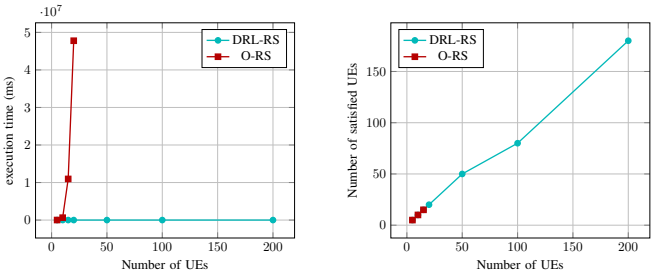
We adopted different bandwidth sizes from 20 MHz to 400 MHz, representing the maximum bandwidth defined by 5G NR specifications and corresponding to the usage of the mmWave bands. Moreover, in order to support the dynamic nature of the RAN environment, we varied the number of UEs, the number of slices per UE (1, 2, or 3 slices per UE), slices SLA and the MCS assigned to each UE (16 and 26 for medium and high channel quality, respectively). Figure 8 reveals that the slice' SLAs are met for each configuration. We notice that the maximum and the minimum achieved SLA for throughput are greater than 1 ($\max_i O_i^{thg} > 1$) and 0.94 ($\min_i O_i^{thg} > 0.94$), respectively. This means that all the UEs have achieved their throughput SLA targets. We also notice that the latency SLAs are respected for all the configurations (i.e., $\min_i O_i^{lat} = 1$).

Figures 10(a), 11(a), 12(a), 13(a), 14(a) compare the execution time needed by O-RS, DRL-RS to find a feasible solution for a bandwidth of 20MHz (configuration 1), 40 MHz (configuration 3), 100MHz (configuration 5) and 400MHz (configuration 7 and 8), respectively. The x-axis represents the number of UEs to be scheduled and the y-axis represents the execution time in milliseconds. The number of UEs is



(a) The execution time (b) Performance Evaluation

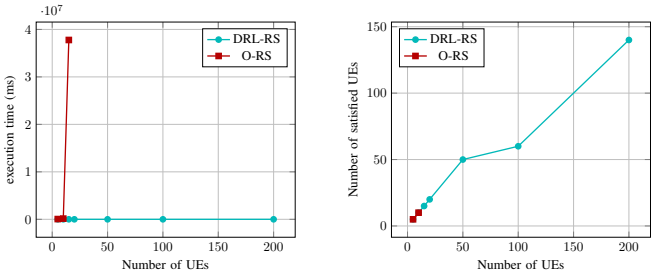
Figure 10: 20 MHz matrix



(a) The execution time (b) Performance Evaluation

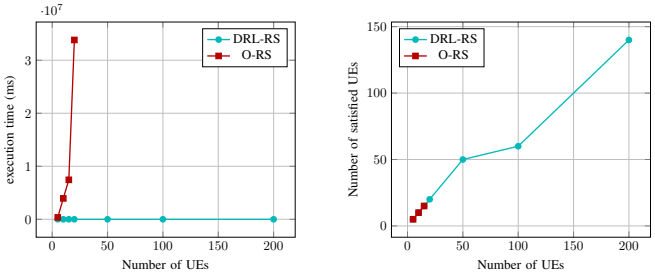
Figure 11: 40 MHz matrix

defined by β while the throughput SLA for each UE will be $Initial\ SLA \div \beta$, where $\beta = 5 * k$, with k a positive integer; and $Initial\ SLA$ is the maximum possible throughput SLA in the given configuration. We observe that O-RS's execution time is exponential by report to the number of UEs while DRL-RS take much less time to be executed. We notice that O-RS is not able to solve the problem instances with more than 30 UEs before the fixed time limit (50000 s). Figure 15 zoom out the execution time for Figure 14(a) to visualize the difference in execution time more clearly for smaller instances (number of UEs less than 30 UEs for O-RS plot). We notice that DRL-RS takes much less time to find a feasible solution (less than 1ms). Hence, we can say that DRL-RS is suitable for real-time scheduling.



(a) Execution time comparison (b) Performance Evaluation

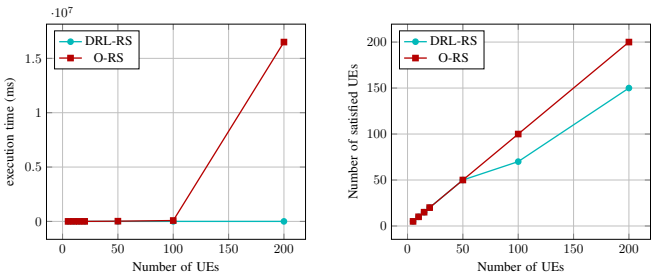
Figure 12: 100 MHz matrix



(a) Execution time comparison (b) Performance Evaluation

Figure 13: 400 MHz matrix

Figures 10(b), 11(b), 12(b), 13(b), 14(b) compare the number of satisfied UEs when applying O-RS and DRL-RS on a bandwidth of 20MHz (configuration 1), 40MHz (configuration 3), 100MHz (configuration 5) and 400MHz (configuration 7 and 8), respectively. The x-axis represents the number of UEs to be scheduled, and the y-axis represents the number of satisfied UEs in terms of both throughput and latency. We observe that DRL-RS is able to fulfill the throughput and latency requirements of up to 50, 180, 130, 140, and 150 UEs out of 200 UEs in configuration 1, 3, 5, 7, and 8, respectively (we recall that configuration requirements are different IV).



(a) Execution time comparison (b) Performance Evaluation

Figure 14: 400 MHz matrix (1 slice attach case)

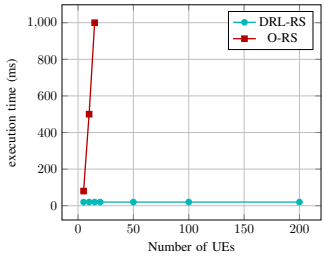


Figure 15: Execution time for the 400 MHz matrix (single slice attach case)

To summarize, O-RS is able to derive the optimal solution for the proposed optimization problem. However, we observed that it takes exponential time regarding the number of UEs. DRL-RS is an approach to reducing the execution time. DRL-RS takes less than 1ms to find a feasible solution in different network configurations. Hence, DRL-RS is more suitable for real-time scheduling. Moreover, the DRL solution was able to select the numerology dynamically, preventing from using higher numerology when it is not needed (since the DRL-RS design allows the selection of numerology for each time slot).

Conclusions

In this paper, we modeled the radio resource allocation problem in 5G NR featuring network slicing and considering a mixed-numerology environment as a Mixed Integer Linear Program. We proved that the problem is NP-hard and proposed a DRL-based approach to solve it for big instances of the problem (i.e., when the number of UEs is higher than 50 and bandwidth size bigger than 40 MHz). Simulation results demonstrated the proposed approach' efficiency and scalability in meeting the desired quality of service requirements. Indeed, DRL-RS can find feasible solutions for the radio resource management problem in a reasonable amount of time which makes it suitable for real-time scheduling. Besides, DRL-RS is energy efficient since it dynamically selects high numerology only when it is required in the context of multiple slices per UE. Future works will investigate the impact of numerology selection on the energy consumption deeply using a real 5G platform based on OpenAirInterface [23].

Acknowledgment

This work was partially supported by the European Union's Horizon Program under 6GBricks project (Grant No. 101096954).

References

[1] 3GPP. "System architecture for the 5G System (5GS)". In: *TS 23.501* Release 15 (2018).

[2] Ljiljana Marijanović et al. "Optimal Resource Allocation with Flexible Numerology". In: *ICCS*. 2018.

[3] Ljiljana Marijanovic, Stefan Schwarz, and Markus Rupp. "Multi-User Resource Allocation for Low Latency Communications Based on Mixed Numerology". In: *VTC2019-Fall*. 2019.

[4] Ljiljana Marijanovic, Stefan Schwarz, and Markus Rupp. "A Novel Optimization Method for Resource Allocation Based on Mixed Numerology". In: *ICC*. 2019.

[5] Boutiba Karim, Bagaa Miloud, and Ksentini Adlen. "Radio Resource Management in Multi-numerology 5G New Radio featuring Network Slicing". In: *ICC*. 2022.

[6] Abderrahime Filali et al. *Communication and Computation O-RAN Resource Slicing for URLLC Services Using Deep Reinforcement Learning*. 2022. arXiv: 2202.06439 [cs.NI].

[7] Faroq Al-Tam, Noélia Correia, and Jonathan Rodriguez. "Learn to Schedule (LEASCH): A Deep Reinforcement Learning Approach for Radio Resource Scheduling in the 5G MAC Layer". In: *IEEE Access* (2020).

[8] Marco Zambianco and Giacomo Verticale. "Spectrum Allocation for Network Slices with Inter-Numerology Interference using Deep Reinforcement Learning". In: *PIMRC*. 2020.

[9] Abderrahime Filali et al. *Dynamic SDN-based Radio Access Network Slicing with Deep Reinforcement Learning for URLLC and eMBB Services*. 2022. arXiv: 2202.06435 [cs.NI].

[10] 3rd Generation Partnership Project (3GPP). "5G; Study on New Radio Access Technology; Radio Interface Protocol Aspects". In: *TR 38.804* Release 14 (2017).

[11] 3rd Generation Partnership Project (3GPP). "Physical channels and modulation". In: *3GPP TS 38.211 version 15.3.0 Release 15* (2018).

[12] 3rd Generation Partnership Project (3GPP). "Study on User Equipment (UE) power saving in NR". In: *3GPP TR 38.840 V16.0.0 Release 16* (2019).

[13] 3GPP. "5G NR; Physical layer procedures for data". In: *TS 38.214* Release 15 (2018).

[14] Lei You et al. "Resource Optimization With Flexible Numerology and Frame Structure for Heterogeneous Services". In: *IEEE Communications Letters* (2018).

[15] Anique Akhtar and Hüseyin Arslan. "Downlink resource allocation and packet scheduling in multi-numerology wireless systems". In: *WCNCW*. 2018.

[16] Jingxuan Zhang et al. "Machine Learning Based Flexible Transmission Time Interval Scheduling for eMBB and uRLLC Coexistence Scenario". In: *IEEE Access* (2019).

[17] Yalcin Sadi, Serhat Erkucuk, and Erdal Panayirci. "Flexible Physical Layer based Resource Allocation for Machine Type Communications Towards 6G". In: *2nd 6G SUMMIT*. 2020.

[18] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: (2013).

[19] Volodymyr Mnih et al. "Asynchronous methods for deep reinforcement learning". In: *International conference on machine learning*. PMLR. 2016, pp. 1928–1937.

[20] Donald E. Knuth. *The Art of Computer Programming: Fascicle 3: Generating All Combinations and Partitions*. Vol. 4. 2005.

[21] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017.

[22] Karim Boutiba et al. "NRflex: Enforcing network slicing in 5G New Radio". In: *Computer Communications* (2021).

[23] Florian Kaltenberger et al. "The OpenAirInterface 5G new radio implementation: Current status and roadmap". In: *WSA 2019, 23rd ITG Workshop on Smart Antennas, Demo Session, 24-26 April 2019, Vienna, Austria*. Ed. by ITG. Copyright ITG. Personal use of this material is permitted. The definitive version of this paper was published in WSA 2019, 23rd ITG Workshop on Smart Antennas, Demo Session, 24-26 April 2019, Vienna, Austria and is available at : Vienna, 2019.

Karim Boutiba received his Engineering degree in Computer Systems from the National School of Computer Science (ESI) , Algiers, Algeria, in 2020. Currently, he is pursuing a Ph.D. in the Communication Systems department at EURECOM, France, under the supervision of Pr. Adlen Ksentini. He is working towards enforcing Network Slicing in 5G networks and beyond. His research interests include Next-Generation Networking and Internet, 5G New Radio, Network Slicing, Open RAN, Optimization algorithms and Reinforcement Learning for 5G networks and beyond.

Miloud Bagaa currently is a professor in the department of electrical and computer engineering of UQTR. He received the engineer's, master's, and Ph.D. degrees from the University of Science and Technology Houari Boumediene, Algiers, Algeria, in 2005, 2008, and 2014, respectively. From 2009 to 2015, he was a researcher with the Research Center on Scientific and Technical Information, Algiers. From 2015 to 2016, he was postdoctoral researcher with the Norwegian University of Science and Technology, Trondheim, Norway. From 2016 to 2019, he was a postdoctoral researcher at Aalto University. From 2019 to 2020, he was a senior researcher at Aalto University. Last but not least, he was a visiting researcher at Aalto University and a senior system cloud specialist at CSC from Oct. 2020 until Nov. 2022.

Adlen Ksentini is a professor in the Communication Systems Department of EURECOM. He is leading the Network softwarization group. He is involved in several EU projects related to Network Slicing and 5G, such as 5G!Drones and MonB5G. He is leading the Network softwarization group activities related to Network Slicing and Edge Computing. He has been involved in several H2020 EU projects on 5G, such as 5G!Pagoda, 5GTransformer, 5G!Drones and MonB5G. Adlen Ksentini research interests are on Network Sofwerization and Network Cloudification focusing on topics related to: network virtualization, Software Defined Networking (SDN), Edge Computing, Network slicing for 5G and beyond networks. He is interested on both system and architectural issues, but also on algorithms problems related to those topics, using Markov Chains, Optimization algorithms and Machine Learning (ML). Adlen Ksentini has received the best paper award from IEEE IWCMC 2016, IEEE ICC 2012, and ACM MSWiM 2005 conferences, and has been awarded the 2017 IEEE Comsoc Fred W. Ellersick (best IEEE communications Magazine's paper).