Journal of Network and Computer Applications 234 (2025) 104068

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Research paper

Optimizing 5G network slicing with DRL: Balancing eMBB, URLLC, and mMTC with OMA, NOMA, and RSMA

Silvestre Malta [a,c,*], Pedro Pinto [a,b], Manuel Fernández-Veiga [c]

[a] Instituto Politécnico de Viana do Castelo, Rua Escola Industrial e Comercial Nun'Álvares, Viana do Castelo, 4900-347, Portugal
[b] INESC TEC, Rua Dr. Roberto Frias, Porto, 4200-465, Portugal
[c] University of Vigo and AtlanTTic Research Center, Vigo, Spain

ARTICLE INFO

Keywords:
5G
eMBB
URLLC
mMTC
Network slicing
OMA
NOMA
RSMA
Deep reinforcement learning
Q-learning
DQN

ABSTRACT

The advent of 5th Generation (5G) networks has introduced the strategy of network slicing as a paradigm shift, enabling the provision of services with distinct Quality of Service (QoS) requirements. The 5th Generation New Radio (5G NR) standard complies with the use cases Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC), which demand a dynamic adaptation of network slicing to meet the diverse traffic needs. This dynamic adaptation presents both a critical challenge and a significant opportunity to improve 5G network efficiency. This paper proposes a Deep Reinforcement Learning (DRL) agent that performs dynamic resource allocation in 5G wireless network slicing according to traffic requirements of the 5G use cases within two scenarios: eMBB with URLLC and eMBB with mMTC. The DRL agent evaluates the performance of different decoding schemes such as Orthogonal Multiple Access (OMA), Non-Orthogonal Multiple Access (NOMA), and Rate Splitting Multiple Access (RSMA) and applies the best decoding scheme in these scenarios under different network conditions. The DRL agent has been tested to maximize the sum rate in scenario eMBB with URLLC and to maximize the number of successfully decoded devices in scenario eMBB with mMTC, both with different combinations of number of devices, power gains and number of allocated frequencies. The results show that the DRL agent dynamically chooses the best decoding scheme and presents an efficiency in maximizing the sum rate and the decoded devices between 84% and 100% for both scenarios evaluated.

1. Introduction

Over the years, mobile communications have evolved and currently 5G mobile technology is a key element in the mobile ecosystem (Gokhan Kalem and Tozan, 2021). 5G NR is part of the global specifications for 5G networks as defined by 3rd Generation Partnership Project (3GPP), and it includes the Radio Access Technology (RAT) that is foundational for the next generation of mobile communications (Lin, 2022). Furthermore, 5G NR offers higher data rates, reduced latency, energy savings, cost reduction, higher system capacity, sleep modes, artificial intelligence capabilities, and more advanced modulation techniques and antenna technologies, such as Massive multiple-input multiple-output (mMIMO) and beamforming (López-Pérez et al., 2022). These technologies allow the network to dynamically adjust and optimize signal transmission and reception to improve spectrum efficiency, coverage, and quality of the user experience.

5G NR is also at the forefront of introducing network virtualization, or 'network slicing'. This technique enables service providers to partition a physical network infrastructure into multiple virtual networks, each with dedicated resources and characteristics (Subedi et al., 2021). A network slice can span across the entire network, from the user end device through the 5G NR, the 5G Core, and into the data network, providing an end-to-end service with specific performance characteristics, securely, and reducing costs.

The 5G NR standard complies with the following use cases defined by 3GPP: eMBB, URLLC, and mMTC, each having different requirements regarding bandwidth, latency, and connectivity. eMBB is a user-centric use-case that is commonly used on mobile phones and mobile PCs/tablets for augmented reality and remote presence, it is suited for high data rates and high reliability for large amounts of data transfer (Abdullah and Ameen, 2021). mMTC is a machine-centric use case that aims to provide connectivity to a massive number of Low-Power Wide-Area (LPWA) devices that are cost and energy-constrained, such as wearables, low-cost sensors, actuators, meters, and tracking devices (Sharma and Wang, 2019). URLLC is a machine-centric use case with strict requirements for ultra-reliability and low latency, while the rate is relatively low compared to mMTC. URLLC is commonly used in

* Corresponding author at: Instituto Politécnico de Viana do Castelo, Rua Escola Industrial e Comercial Nun'Álvares, Viana do Castelo, 4900-347, Portugal.
E-mail address: smalta@estg.ipvc.pt (S. Malta).

https://doi.org/10.1016/j.jnca.2024.104068
Received 28 April 2024; Received in revised form 9 September 2024; Accepted 13 November 2024
Available online 28 November 2024

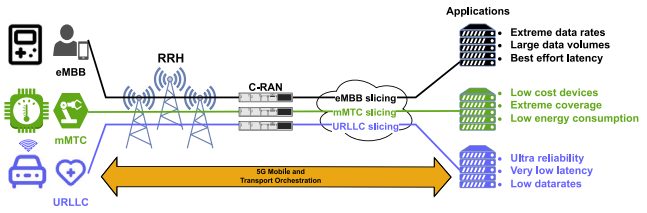


Fig. 1. Overview of a 5G network slicing.

Augmented Reality (AR)/Virtual Reality (VR) technology, autonomous vehicles, cloud robotics, real-time process control, machine coordination, advanced wearables, and real-time collaboration between humans and machines (Chen et al., 2018).

Fig. 1 presents an overview of a 5G network, where network slicing divides the physical network into a set of logical networks to support the eMBB, URLLC, and mMTC use cases. The deployment of Remote Radio Heads (RRHs) for distributed radio access and Cloud Radio Access Networks (CRAN) for centralized control and management of network resources ensures that each slice meets the demands of each use case scenario even in the 5G core network.

To manage multiple access of their clients, 5G network operators dynamically allocate resources and optimize network capacity by implementing OMA, NOMA, and RSMA decoding schemes. In OMA, devices are allocated to orthogonal resources such as time slots, frequency bands, or codes, granting them exclusive access to specific parts of the network's capacity.

With an increasing number of devices, more efficient alternatives such as NOMA or RSMA can be considered. In NOMA, multiple users concurrently use identical time, frequency, or code resources through non-orthogonal encoding, enabling their signals to coexist within a single resource block. Receivers employ Successive Interference Cancellation (SIC) to decode these overlapping signals, facilitating simultaneous transmissions over shared frequency resources. RSMA divides user messages into common and private segments, with the former encoded in one or more shared streams and the latter into individual private streams. Using available channel state information (CSI), these streams are pre-encoded, combined, and broadcast over multiple-input multiple-output (MIMO) or multiple-input single-output (MISO) channels.

Given this context, a challenge arises when network operators need to select dynamically the best decoding scheme to apply in network slicing given the multiple requirements of the 5G use cases. Since CRANs architectures introduce additional layers of abstraction and coordination, increasing the complexity of optimizing resource allocation and coding schemes (Debbabi et al., 2021), NOMA and RSMA are more complex than OMA (Mao et al., 2022). Also, the use cases have diverse and often conflicting requirements that should be dynamically assured in network resource provisioning (or re-provisioning) to adjust to their traffic demands. The challenge to balance the tradeoff between complexity and compliance with traffic requirements should be addressed in scenarios of coexistence of 5G use cases, to maximize the sum rate of the devices or the number of successfully decoded devices.

To orchestrate the resource allocation of different 5G use case combinations with different decoding schemes, previous works considered a single URLLC or mMTC device per timeslot in each transmission block (Liu et al., 2023; Popovski et al., 2018). However, the number of active devices in the use cases URLLC or mMTC can be variable within a specific frequency strip during a transmission block. Moreover, when addressing the requirements of mMTC devices, other studies in the past computed rates based on packets of arbitrary lengths, leading to overly optimistic estimations. More recently, research efforts have been undertaken that assess mobile devices using smaller packet sizes, to address the mMTC use case. Although the authors in Bockelmann et al. (2016) argue that small packet sizes are still not adequately supported in cellular systems designed for human-type communications, this use case should be assessed using finite and short-length packets, for more realistic results.

This paper proposes a DRL agent that optimizes resource allocation by dynamically selecting the optimal decoding scheme for mobile networks with slicing support, given the 5G use case traffic requirements. It uses rate equations to calculate the rewards achieved by individual decoding schemes and defines the policy to determine which decoding scheme should be applied considering dynamic network conditions.

The proposed agent considers to be random the number of active URLLC and mMTC devices during a transmission block within a specific frequency strip. Also, for mMTC devices, the proposed agent uses rate equations suited for finite and short-length packets.

The DRL agent was assessed in the following two scenarios:

- Scenario with URLLC featuring the coexistence of eMBB and URLLC use cases
- Scenario with mMTC featuring the coexistence of eMBB and mMTC use cases

Each test scenario has been assessed using different mobile devices' power gains and results are provided simultaneously for OMA, NOMA, or RSMA decoding schemes enabling the direct comparison and selection of the best operating regimes.

This paper has the following contribution: a DRL agent that optimizes resource allocation by dynamically selecting the best decoding scheme (from OMA, NOMA, and RSMA) in a mobile network using network slicing for the following 5G use case scenarios:

- Scenario with URLLC - where the agent maximizes the sum rate in the coexistence of eMBB an URLLC devices;
- Scenario with mMTC - where the agent maximizes the number of successfully decoded devices given the coexistence of eMBB and mMTC devices.

By focusing on eMBB, URLLC, and mMTC use cases, it is intended to demonstrate the versatility and effectiveness of the DRL-based approach in optimizing 5G network slicing across the full spectrum of service requirements. Our contribution lies in the ability of the DRL agent to dynamically allocate resources and select the most appropriate access scheme (OMA, NOMA, RSMA) based on the specific demands of eMBB, URLLC, and mMTC, thus improving the overall efficiency of the network.

The remainder of this document is organized as follows. Section 2 presents related work. Section 3 discusses the system model and the included decoding schemes. Section 4 presents the resource-sharing model with network slicing for the scenarios with URLLC and mMTC under OMA, NOMA and RSMA decoding schemes. Section 5 introduces the resource optimization through DRL to find the appropriate policies. Section 6 presents the results and analysis of the tested scenarios. Section 7 presents the conclusions.

2. Related work

Recent research has focused on techniques to manage the coexistence of eMBB, URLLC, and mMTC traffic using the same physical resources. Authors in Le et al. (2021) analyzed the URLLC releases and their changes introduced in the downlink transmission multiplexing, allowing reducing latency of a URLLC packet to be transmitted after its arrival for eMBB and URLLC multiplexing. They also suggest that this coexistence can be managed by using traffic prioritization. In Ji et al. (2018) the authors discussed the key requirements for URLLC and suggested improvements regarding scheduling and reliability to ensure coexistence with eMBB.

In Tominaga et al. (2021) the coexistence of eMBB and mMTC in the uplink is analyzed with orthogonal and non-orthogonal network

slicing schemes. It is concluded that as the number of receiving antennas increases, the advantage of NOMA slicing over orthogonal slicing increases in mMTC data rates and mMTC device numbers at given data rates.

To meet the 5G use cases requirements, the authors in Popovski et al. (2018) explored the slicing of radio resources, examining both orthogonal and non-orthogonal approaches, accounting for the diversity of service demands. Their findings indicate that non-orthogonal slicing offers superior performance over orthogonal slicing when it comes to multiplexing 5G services.

Few studies have used Reinforcement Learning (RL) to improve and control network access through network slicing. In Bega et al. (2020), the authors used RL to maximize monetization by obtaining the highest possible profit from the deployed infrastructure. Using an on-demand algorithm that updates policies as new requests arrive, the algorithm is claimed to meet the QoS requirements of each network slice while maximizing revenue. A DRL model was used to train two Deep Neural Networks (DNNs), one to estimate the revenue for each state and the other for the rejection action. Two types of slices were used: Inelastic associated with URLLC services where users require a certain fixed throughput demand which needs to be satisfied at all times and elastics related to eMBB and mMTC which only need guarantees on the average throughput, requiring that the expected average throughput over long time scales is above a certain threshold. The state space of the system contains inelastic and elastic slices and the next events that indicate arrivals or departures. Authors in Bakri et al. (2021) present a Deep Q Network (DQN) model to minimize penalty costs caused by violations of Service-Level Agreement (SLA) in 5G Radio Access Network (5G-RAN). Online and target DNNs were used to represent the state space, which included the number of slices requested for each type of device eMBB, URLLC and mMTC, as the type of request for the last slice. In this model, the space of actions is binary, indicating whether or not the new arrival slice request should be accepted.

Approaches for orchestrating resources in the context of network slicing have been presented. Authors in Rezazadeh et al. (2020) introduced a DRL model to manage the computing resources in small cells connected to a central unit to reduce the latency, energy consumption, and costs related to the creation of Virtual Network Functions (VNFs). Two DNNs that are activated using Rectified Linear Unit (ReLU) have been used to model the state space. This state space consists of the number of requested services per slice, the computing resources allocated to each VNF in the service chains, and the number of instantiated VNFs. The model was established based on the observation of traffic fluctuations to determine whether to increase or decrease the computing resources allocated to each VNF. The research (Liu et al., 2020) introduced a decentralized method using Deep Deterministic Policy Gradient (DDPG) for efficient end-to-end resource orchestration, focusing on minimizing overhead and delay. The approach also accounting SLA violations and resource limitations in the edge slices. The system involves a coordinator agent that manages resource orchestration policies and multiple orchestration agents. These decentralized agents estimate the resource demands of the network slices and allocate resources locally. The agents operate within a state space modeled on the status of network slices in a queue and performance information provided by the coordinator and orchestration agents. The action space is defined as the resources that will be assigned to network slices in the base stations and edge servers.

Recent discussions on resource management highlight the importance of resource scheduling methods, particularly emphasizing the concept of network slicing. Authors in Comşa et al. (2018) proposed a framework for packet scheduling in Radio Access Network (RAN) slices, where radio resources are shared during each transmission interval. A set of DQN algorithms were used to achieve the optimal action value. The state space consisted of the active users during each transmission time interval, the arrival rates in the data queues, the channel quality indicator, and the network services' performance requirements. The number of resource blocks allocated per transmission interval has been used to model actions. Li et al. (2019) introduced a Q-Learning-based scheduling strategy to orchestrate the operation of VNFs that make up a service function chain, to reduce delay. The strategy interprets delay as the time difference between the first and the last VNF in the chain. The Q-Learning agent utilized a state space derived from the status of the network function virtual infrastructure. The agent's decisions determined the VNF to be executed at a given time t. The assessment of this proposed strategy took into account a system comprising four VNF nodes and five network services sensitive to delay, each with unique setting parameters.

Strategies for allocating network slicing resources which employ RL models, have been documented in the resource management literature. To reduce end-to-end latency, the authors of Khumalo et al. (2021) proposed a resource allocation method based on Q-Learning. The state space was defined using variables such as user requests, request arrival rate, percentage of allocated resources, percentage of unused allocated resources, minimum allocation requirements, and the maximum allowed delay. To determine the actions, the agent had to decide whether to allocate or not allocate compute resources. The authors of Santos et al. (2021) employ the method of RSMA for the transmission of URLLC traffic. In this scenario, the URLLC device divides its message into two segments, each transmitted with a fraction of the total power. The Base Station (BS) recovers the original message using SIC. The orthogonal and non-orthogonal network slicing approaches were evaluated in the presence of eMBB devices. The objective was to distribute network resources between different user profiles with varying requirements. The sum rate pair performance of eMBB and URLLC devices operating in OMA, NOMA, and RSMA has been compared. The results show that, despite strict reliability requirements, RSMA is capable of achieving higher rates when the power splitting factor is properly configured. They concluded that OMA presents the highest rate pair values until the rate for eMBB is around 7 bits/s/Hz, from here RSMA outperforms OMA and NOMA. They also concluded that the sum rate of URLLC remains almost constant in RSMA and NOMA as the rate of eMBB increases, which makes this method a good option when the objective is to achieve higher eMBB rates. The authors in Alsenwi et al. (2021) examine resource slicing for eMBB and URLLC. They introduce an optimization problem to enhance eMBB data rates, adhering to URLLC reliability constraints, by leveraging an optimization-assisted DRL framework. The proposed methodology is validated through simulation results, showing its capability to satisfy stringent URLLC reliability standards while keeping eMBB reliability above 90%. The study (Ahsan et al., 2022) presents a deep State Action Reward State Action (SARSA) learning method for the optimization of uplink resource distribution in scenarios assisted by NOMA in URLLC. The suggested algorithm, which incorporates real-time feedback, ensures dependable resource allocation under fluctuating network conditions. Simulations indicate that it outperforms conventional Q-Learning algorithms in terms of convergence and performance, underlining the utility of NOMA in URLLC applications. In Filali et al. (2022), the authors have introduced a two-level RAN slicing mechanism for software-defined networking (SDN)-enabled 5G networks, focusing on efficient resource allocation. The SDN controller handles resource block (RB) allocation to gNodeBs on a large time scale, while gNodeBs manage dynamic allocation to end-users on a short time scale, with the ability to borrow RBs from neighboring gNodeBs. This approach minimizes signaling overhead and ensures timely resource distribution, crucial for meeting the stringent requirements of URLLC and eMBB services. By modeling allocation problems as single-agent and multi-agent Markov Decision Processs (MDPs) and using EXP3 and DQN, the method improves resource allocation efficiency, outperforming benchmarks in data rate and latency. The authors in Setayesh et al. (2022) proposed a hierarchical DRL framework for RAN slicing in OFDMA-based networks, focusing on efficient resource allocation for eMBB and URLLC services. The

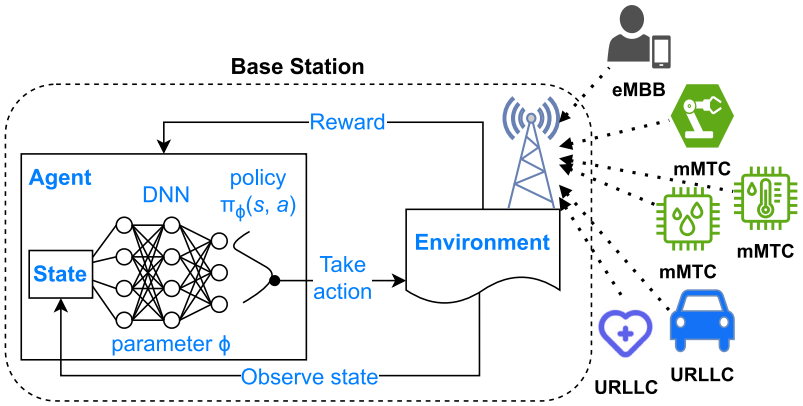


Fig. 2. System model.

framework employs a DRL algorithm for long-term slice configuration and attention-based DNNs for short-term resource allocation. This approach, using numerology, mini-slot transmission, and combined scheduling techniques, adapts well to network dynamics and surpasses existing methods in throughput and SLA satisfaction.

Although current research provides valuable perspectives on the possibilities of network slicing and DRL for improving 5G, the existing approaches should scale to meet the diverse and dynamic demands of eMBB, URLLC, and mMTC traffic simultaneously, especially in large-scale network deployments. Research works such as Liu et al. (2023), Popovski et al. (2018) consider a single URLLC device per timeslot in each transmission block. However, the number of active devices in URLLC use cases can be variable within a specific frequency strip during a transmission block.

3. System model

Fig. 2 presents the system model, where multiple services use the same radio resources for uplink transmission to a BS. This BS embeds a DRL agent, actively observing the state of the environment, receiving rewards, and autonomously determining the optimal actions to take.

More specifically, it is assumed that resource sharing is possible in OMA (Fig. 3(a)), wherein each traffic class uses separate physical resources without overlap. Multiple devices within the same service are allowed to use these resources, i.e., a URLLC device can be assigned individual resources (represented by a light red color), and multiple URLLC devices can also be allocated to the same resources (represented by a dark red color). In contrast, with NOMA and RSMA (Fig. 3(b)) different classes of traffic can coexist over the same physical resources. Assuming that eMBB devices are assigned to one or more radio resources at specific frequencies, whereas URLLC devices can occupy multiple radio frequencies and a single mini-slot due to their tight latency requirement. mMTC devices are assigned a designated frequency resource. We further assume that eMBB devices have full CSI, where URLLC and mMTC devices do not know CSI, since pilot-assisted channel estimation is difficult for low delay and short-packet communications, respectively. Consequently, the eMBB device can adapt its transmission power according to the channel, but power control is not possible for the other devices. Finally, the number of URLLC and mMTC devices in the system, n_U and n_M, respectively, is assumed to be random, with arrival rates that follow Poisson(λ_U) and Poisson(λ_M) distributions.

3.1. Channel model

At a frequency f, let $h_f^B \in \mathbb{C}$, denote the complex channel coefficients for an eMBB user, while $h_{u,f}^U \in \mathbb{C}$ and $h_{m,f}^M \in \mathbb{C}$ represent the channel gain for URLLC user $u = 1, \ldots, n_U$ and mMTC user $m = 1, \ldots, n_M$. Assuming Rayleigh block fading channels, with $h_f^B \sim$

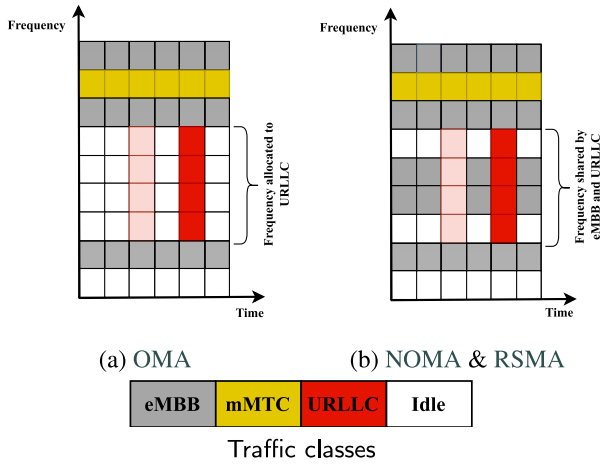


Fig. 3. Physical resources sharing.

$\mathcal{CN}(0, \gamma_B)$, $h_{u,f}^U \sim \mathcal{CN}(0, \gamma_U)$, and $h_{m,f}^M \sim \mathcal{CN}(0, \gamma_M)$. The received signal at BS in a given resource block indexed by $i = (t, f)$ is therefore

$$y[i] = h_f^B s_B[i] + \sum_{u=1}^{n_U} h_{u,f}^U s_{U,u}[i] + z[i], \tag{1}$$

where the first term is due to the transmission of symbol $s_B[i]$ by the eMBB user, the second term corresponds to the superposition of signals $s_{U,u}$ transmitted by URLLC users, and $z[i]$ is the noise (Liu et al., 2023). Likewise, for the coexistence of eMBB and mMTC users transmitting the symbols $s_{M,m}[i]$, the received signal is given by

$$\tilde{y}[i] = h_f^B s_B[i] + \sum_{m=1}^{n_M} h_{m,f}^M s_{M,m}[i] + z[i]. \tag{2}$$

3.2. Signal model for eMBB

It is considered that the eMBB device operates in the Finite Block Length (FBL) regime. The goal of the system is to maximize the transmission rate $\mathsf{R}_{B,f}$ subject to a target probability of bit error ϵ_B, under a long-term average power constraint P. Due to the assumption of perfect CSI, $\mathsf{P} := P(g_f^B)$ can be adapted to the instantaneous channel gains so that predetermined signal-to-noise ratio (SNR) s is met. In Gaussian point-to-point channels, the achievable R under the FBL is given by the first-order normal approximation (Polyanskiy et al., 2010)

$$\mathsf{R}(n, \mathsf{s}, \epsilon) = \mathsf{C}(\mathsf{s}) - \sqrt{\frac{\mathsf{V}(\mathsf{s})}{n}} Q^{-1}(\epsilon), \quad \mathsf{s} \geq 0, \epsilon \in [0, 1] \tag{3}$$

in bits per channel use (or bits/s/Hz), where n is the codeword blocklength, ϵ is the bit error probability, $\mathsf{C}(x) := \log_2(1+x)$ is the Shannon capacity, the function

$$\mathsf{V}(x) = 1 - \frac{1}{(1+x)^2}, \quad x \geq 0 \tag{4}$$

is the channel dispersion, and $Q^{-1}(\cdot)$ is the inverse of the Gaussian error function $Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{t^2}{2}} \mathrm{d}t$, $x \in \mathbb{R}$. The expression for the bit error probability given a channel with SNR s is therefore

$$\epsilon = \mathsf{P}_e(\mathsf{s}, r, n) = Q\left(\sqrt{\frac{n}{\mathsf{V}(\mathsf{s})}}\left(\mathsf{C}(\mathsf{s}) - r\right)\log_{\mathrm{e}} 2\right). \tag{5}$$

It has been shown in Zhu et al. (2023) that the error probability $\mathsf{P}_e(x, \mathsf{s}, n)$ is jointly convex in (s, r), which opens the way to solving other interesting optimization problems involving ϵ as an objective function or constraint. However, this is out of the scope of this paper.

The optimization problem for eMBB can be formulated as

$$\begin{aligned} \max_{P} \quad & \mathsf{R}_{B,f}(\epsilon) := \mathsf{R}(n, g_f^B \mathsf{P}, \epsilon) \\ \text{subject to} \quad & \mathbb{E}[\mathsf{P}] \leq 1, \end{aligned} \tag{6}$$

where the bit error probability ϵ and the blocklength n are fixed. Thus, it is intended to maximize the rate subject to the long-term average power constraint (6). The optimal solution to this problem is known (Caire et al., 1999), and it consists of channel inversion, with the optimal transmission power equal to $\mathsf{P}^*(g_f^B) = \frac{c}{g_f^B}$, where $c := \frac{\gamma_B}{\Gamma(0, a/\gamma_B)}$, and $\Gamma(0, x) = \int_x^{+\infty} \frac{\mathrm{e}^{-t}}{t}\mathrm{d}t = E_1(x)$ is a special case of the incomplete Gamma function or, equivalently, of the exponential integral function $E_1(x)$. In this signal model, $\frac{a}{\gamma_B} = \log \frac{1}{1-\epsilon_B} \approx \epsilon_B$ is set, where log denotes the natural logarithm and ϵ_B is the desired error probability for the eMBB users. A typical value is $\epsilon_B = 10^{-2}$ or $\epsilon_B = 10^{-3}$ (see Liu et al. (2023)). Hence, $E_1\left(\frac{a}{\gamma_B}\right) \approx E_1(\epsilon_B) \approx 8.8332$. Using $\mathsf{P}^* = \gamma_B/(\Gamma(0, \epsilon_B) g_{B,f})$, the optimal transmission rate is

$$\mathsf{R}^*_{B,f}(\epsilon) = \mathsf{R}(n, g_f^B \mathsf{P}^*, \epsilon) = \mathsf{R}(n, \gamma_B/\Gamma(0, \epsilon_B), \epsilon). \tag{7}$$

and, as it does not depend on the frequency f due to channel inversion, it is dropped the subscript f henceforth.

3.3. Signal model for URLLC

URLLC devices transmit messages using $0 < t \leq F$ frequency resource blocks. It is assumed further that CSI is unavailable, and a normalized transmission power of 1 is used. However, BSs estimate the CSI of the devices and use SIC for message decoding. Since URLLC users exhibit intermittent behavior, each being active with probability p_U per timeslot, the number k_U of active devices during a transmission block within a specific frequency strip is random. Note that these assumptions extend the URLLC configuration described in Popovski et al. (2018), which considers only the presence of a single URLLC device.

Under SIC, if device j is decoded after device i (denoted $j \succ i$), and the channel gain of device i at frequency f is $G^U_{i,f}$, its signal to interference plus noise ratio (SINR) at frequency f is given by

$$\mathsf{SINR}^U_{i,f} = \frac{G^U_{i,f}}{\sigma^2 + \sum_{j \in [k_U]: j \succ i} G^U_{j,f}}, \tag{8}$$

where σ^2 is the noise power. Consequently, the FBL transmission rate is

$$\mathsf{R}_{U,i}(f) := \mathsf{R}(n, \mathsf{SINR}^U_{i,f}, \epsilon). \tag{9}$$

The SIC decoding order is taken as the decreasing order of the received powers, so device i is decoded before device j at frequency f whenever $G^U_{i,f} > G^U_{j,f}$. Therefore, using this, the received rate for URLLC device i is

$$\mathsf{R}_{U,i} := \sum_{f=1}^{t} \mathsf{R}_{U,i}(f) = \sum_{f=1}^{t} \mathsf{R}(n, \mathsf{SINR}^U_{i,f}, \epsilon), \tag{10}$$

where $\mathsf{SINR}^U_{i,f}$ is given by (8). The sum rate for URLLC devices is

$$\mathsf{R}^{\mathsf{sum-rate}}_U := \sum_{i=1}^{k_U} \mathsf{R}_{U,i} = \sum_{i=1}^{k_U} \sum_{f=1}^{t} \mathsf{R}(n, \mathsf{SINR}^U_{i,f}, \epsilon). \tag{11}$$

The optimization problem for URLLC consists of maximizing the sum rate, which depends on the number of frequency slots ft and the unknown number of active devices k_U.

3.4. Signal model for mMTC

Since the CSI for mMTC is unknown, the transmission power is also normalized at 1. mMTC devices request a rate of R_M bits/s/Hz, for which specific frequency resources must be assigned. Regarding other services, the BS employs SIC for decoding mMTC signals, where the order of decoding is based on the channel gains of the devices. Define i as the mMTC device possessing the i-the highest channel gain, with the channel gains sorted in a decreasing sequence $G_1 \geq G_2 \geq \cdots \geq G_{n_M}$. Namely, the decoding order follows the sequence $1, 2, \ldots, n_M$. Since there is no interference from other services (see the next Section), SINR for a random device i has the expression

$$\mathsf{SINR}_i = \frac{G_i}{\sigma^2 + \sum_{j \in [n_M]: j > i} G_j}, \tag{12}$$

where σ^2 is the noise power. If $\mathsf{R}(n, \mathsf{SINR}_i, \epsilon) \geq \mathsf{R}_M$ holds, then the mMTC device can be decoded with bit error probability less than ϵ and then canceled (if ϵ is low enough, it is neglected the event of chained decoding failures due to imperfect cancellation). Should decoding be unsuccessful, the process is considered to end immediately. Denote by A_M the count of devices decoded successfully,

$$A_M = \sum_{i=1}^{n_M} \mathbf{1}\left(\mathsf{R}(n, \mathsf{SINR}_i, \epsilon) \geq \mathsf{R}_M\right), \tag{13}$$

and the error probability matches the ratio of successfully decoded devices A_M to all active devices, $\mathsf{P}_M = 1 - \mathbb{E}[A_M]/\lambda_M$, where $\mathbb{E}[\cdot]$ denotes the expectation. The mMTC optimization problem can be formulated as

$$\max\{\lambda_M(\epsilon_M) : \mathsf{P}_M \leq \epsilon_M\}, \tag{14}$$

where it is included explicitly the dependence of the maximal arrival rate on the service error probability. Such optimization considers the FBL n and the decoding bit error probability for each device, though these have been omitted in (14) for simplicity. Also, as a remark, our model does not consider the queue dynamics at the devices (i.e., queue stability, cf. Robaglia et al. (2023)), only the decoding error probability.

4. Resource-sharing model with network slicing

The resource sharing model with network slicing for joint resource allocation is studied in two scenarios: scenario with URLLC and scenario with mMTC. The resource-sharing model is defined for both scenarios under OMA, NOMA, and RSMA.

4.1. Scenario with URLLC

Without loss of generality, the system supports a single eMBB user and multiple URLLC devices simultaneously. The analysis of the different slicing combinations is as follows.

OMA forms our baseline scheme, where each type of device is assigned disjoint frequency resources, $F - f_U$ and f_U, for eMBB and URLLC, respectively. The achievable eMBB sum rate while ensuring that the reliability requirements (ϵ_B, ϵ_U) are met is

$$\mathsf{R}^{\mathsf{sum,OMA}}_B = (F - f_U)\mathsf{R}^{\mathsf{OMA}}_B \quad \text{(bits/s/Hz)} \tag{15}$$

where $\mathsf{R}^{\mathsf{OMA}}_B = \mathsf{R}^*_{B,f}$ is calculated with (7). The rate for URLLC can be computed using (11). Therefore, the aggregated sum rate under OMA results from $\mathsf{R}^{\mathsf{sum,OMA}}_{BU} = \mathsf{R}^{\mathsf{sum,OMA}}_B + \mathsf{R}^{\mathsf{sum-rate}}_U$.

Assuming a NOMA transmission scheme in the uplink, eMBB and URLLC users will share all the F frequency resources available. Since

URLLC devices are sensitive to latency, the BS prioritizes decoding URLLC devices first. If this is accomplished, the eMBB device decoding starts, otherwise it is aborted due to the inability to cancel the interference. The resulting sum rate with such a decoding scheme is

$$\mathsf{R}_B^{\mathsf{sum,NOMA}} := \sum_{f=1}^{F} \mathsf{R}^*_{B,f}. \tag{16}$$

To fulfill the reliability criteria, the primary objective is to attain the maximum rate R_U, while the error probabilities can be evaluated independently. To see this, for eMBB there is an error probability equal to

$$\mathsf{P}_B = 1 - \mathbb{E}[A_B], \tag{17}$$

the probability of a failed decoding, where $A_B \in \{0,1\}$ means the number of correctly decoded eMBB devices. For URLLC the error probability is determined by the decoding order. Letting $a_{k,k'} \in \{0,1\}$ be the variable indicating that user k is decoded before user k', and noting that $a_{k,k'} + a_{k',k} = 1$, the SINR of URLLC device i follows

$$\mathsf{SINR}^U_{i,f} = \frac{G^U_{i,f}}{\sigma^2 + \sum_{k:a_{i,k}=1}^{n_U} a_{i,k} G^U_{k,f} + G^B_f}. \tag{18}$$

Consequently, the sum rate of URLLC devices takes the form of

$$\mathsf{R}_U^{\mathsf{sum,NOMA}} := \sum_{i=1}^{K_U} \sum_{f=1}^{f_U} \mathsf{R}(n, \mathsf{SINR}^U_{i,f}, \epsilon) \quad \text{bits/s/Hz}. \tag{19}$$

Notice that the only difference between $\mathsf{R}_U^{\mathsf{sum-rate}}$ and (19) is just the calculation of SINR, as there is an additional interference term in the latter.

Assuming a RSMA strategy, this class will be decoded first, the rate-splitting principle will only be applied to URLLC devices. The order in which a specific device is decoded is identical to that described in Section 3.3. As explained, only one eMBB user and k URLLC devices are present in the system, with the eMBB device always decoded after the URLLC signal. The decoding procedure is terminated if one user's decoding fails, and the other users' undecoded messages are lost.

The calculation of the eMBB rate is done in the same method as in NOMA, i.e., using (16). Instead, for URLLC, we suppose that the message from device $i = 1, \dots, k$) is divided into two parts: $s_{U,i,f} = (s^1_{U,i,f}, s^2_{U,i,f})$, such that a fraction $0 < \beta < 1$ of the transmitted power is used for the first sub-message, and the remaining $1 - \beta$ for the other sub-message. The received signal at the BS at frequency f is thus

$$y = h^B_f s_{B,f} + \sqrt{\beta} \sum_{i=1}^{k} h^U_{i,f} s^1_{U,i,f} + \sqrt{1-\beta} \sum_{i=1}^{k} h^U_{i,f} s^2_{U,i,f} + z. \tag{20}$$

In addition, assuming that the $2k$ URLLC signals are decoded in a known order, $\boldsymbol{\pi}$, and these signals are recovered in decreasing order based on their receiving power, i.e. the received signals can be arranged so that if $\beta_j h^U_{i,f} |s^j_{U,i,f}|^2 > \beta_{j'} h^U_{i',f} |s^{j'}_{U,i',f,}|^2$, then the sub-message $s^j_{U,i,f}$ is decoded before $s^{j'}_{U,i',f}$, a condition that will be denoted as $\pi^{(f)}_{i,j} \prec \pi^{(f)}_{i',j'}$. Here, $j = 1, 2$, and $\beta_1 = \beta$, $\beta_2 = 1 - \beta$. Once the permutation $\boldsymbol{\pi}$ has been fixed, the SINR of $s^j_{U,i,f}$ is

$$\mathsf{SINR}^U_{i,f,j} = \frac{\beta_j |h^U_{i,f}|^2}{\sigma^2 + \sum_{\pi^{(f)}_{i',j'} \succ \pi^{(f)}_{i,j}} |h^U_{i',f}|^2 + |h^B_f|^2}, \tag{21}$$

and the corresponding rate is computed from this with

$$\mathsf{R}^U_{i,f,j} = \mathsf{R}(n, \mathsf{SINR}^U_{i,f,j}, \epsilon). \tag{22}$$

From this, the sum rate for device i is

$$\mathsf{R}^{\mathsf{sum,RSMA}}_{U,i} = \sum_{f=1}^{F} \mathsf{R}^U_{i,f,1} + \mathsf{R}^U_{i,f,2} \text{ bits/s/Hz}, \tag{23}$$

giving a sum rate for the n_U URLLC devices, equal to

$$\mathsf{R}^{\mathsf{sum,RSMA}}_{U} = \sum_{i=1}^{n_U} \mathsf{R}^{\mathsf{sum,RSMA}}_{U,i}. \tag{24}$$

Thus, formally, the uplink RSMA system is essentially the same as in a NOMA system with $k = 2n_U + 1$ devices. As a result, the receiver must perform SIC decoding on k layers, which introduces a significant level of complexity.

4.2. Scenario with mMTC

The case of eMBB and mMTC users contending for resource blocks can be modeled similarly to the scenario with URLLC.

As for OMA, i.e. the baseline, the resources are shared through time sharing. The system is devoted to servicing eMBB traffic for a fraction α of the time, the achievable transmission rate is simply $\alpha \mathsf{R}^{\mathsf{OMA}}_B(\epsilon)$ bits/s/Hz, using (7), for a target error probability ϵ. The remaining $1-\alpha$ time is exclusively allocated to mMTC devices. Consequently, the computation of the arrival rate for mMTC devices λ_M can be performed using the method described in Section 3.4. It is equal to the rate $\mathsf{R}^{\mathsf{OMA}}_M/(1-\alpha)$, so λ_M can be calculated using the equation

$$\lambda_M = \lambda^{\mathsf{OMA}}_M \frac{\mathsf{R}^{\mathsf{OMA}}_M}{1-\alpha}, \tag{25}$$

which can be derived from (14). However, as an advantage of using RL for solving the sum rate optimization problems, we can directly estimate λ_M from the simulations, i.e., its value can also be learned from the environment.

Concerning NOMA, neither of the two possible orderings for decoding can be assumed as optimal for all the channel realizations (i.e., eMBB first vs. mMTC first) because the requirement for mMTC is to maximize the fraction of successfully decoded data streams. Following the arguments in Popovski et al. (2018), Liu et al. (2023), we choose to recover the mMTC packets first. To that end, the mMTC devices are sorted in descending order, and the BS attempts to decode them individually. So, if an mMTC device could not be decoded, the BS then attempts with the eMBB traffic. Upon correct decoding of the latter, the BS continues with the mMTC devices. However, if the recovery of the eMBB signal fails, the decoding procedure is stopped as it is impossible to cancel its interference.

For the eMBB device, it is advisable to select a target SINR_B, since this ensures that the rate $\mathsf{R}'_B = (n, \mathsf{SINR}_B, \epsilon)$ (bits/s/Hz) can be achieved. The analysis of decoding errors for mMTC is more complicated, since these suffer interference from the other class and by the presence of inactive users. To simplify the derivation, we assume that mMTC devices are always hampered by eMBB's interference and that this is the only cause of a decoding error. Using this hypothesis

$$\mathsf{SINR}^M_i = \frac{G_i}{\sigma^2 + \sum_{j=i+1}^{k_M} G_j + \mathsf{SINR}_B}, \tag{26}$$

so that this device i is decodable when $\log_2(1 + \mathsf{SINR}^M_i) > r_B$, and the BS will proceed to subsequent mMTC devices. Otherwise, the process fails and finishes.

The error probability of mMTC is again $1 - \mathbb{E}[A_M]/\lambda_M$. Likewise, the error probability of an eMBB device is given by $1 - \mathbb{E}[A_B]$. To determine the achievable pairs $(\mathsf{R}_B, \mathsf{R}_M)$, the problem can be formulated as follows

$$\lambda^{\mathsf{NOMA}}_M(\mathsf{R}_B) = \sup\{\lambda_M \geq 0 : 1 - \mathbb{E}[A_M]/\lambda_M \leq \epsilon_M,$$
$$1 - \mathbb{E}[A_B] \leq \epsilon_B\}. \tag{27}$$

for some SINR_B. Notice that the optimization is in the throughput, namely, the queue stability is not addressed.

Under RSMA, the decoding procedure is the same as for NOMA: if the mMTC device and eMBB cannot achieve the target rate due to high interference from each other, the decoding procedure halts because it falls into a deadlock. Using RSMA in the coexistence scenario of eMBB and mMTC can alleviate this effect, considering that rate-splitting is capable of partial cancellation of interference. Specifically, an eMBB message is divided into two streams, so the BS can decode one stream at a lower rate and then proceed to the other devices.

The procedure begins by sorting the mMTC devices based on their channel gains in descending order, determining the decoding order for these devices. At the BS, decoding starts with mMTC devices, in a prescribed order, and continues to the eMBB's low-rate sub-message as soon as the next mMTC cannot be decoded. Subsequently, the decoding process for these is resumed and, whenever it fails again, the second eMBB sub-message is attempted. Suppose R_{B_1}, R_{B_2} are the corresponding rates for the submessages; hence, $\mathsf{R}_{B_1} + \mathsf{R}_{B_2} < \mathsf{R}_B$, this indicates a failure for the eMBB user, whereas $\mathsf{R}_{B_1} + \mathsf{R}_{B_2} \geq \mathsf{R}_B$ means that mMTC goes on.

The received signal can be written as

$$y = \sqrt{\beta} h^B s_{B_1} + \sqrt{1-\beta} h^B s_{B_2} + \sum_{m=1}^{k_M} h_m^M s_m + z. \tag{28}$$

The target SNR for the eMBB, assuming that s_{B_1} is decoded first, is

$$\mathsf{SINR}_{B_1} = \frac{\beta \mathsf{SINR}_B}{\sigma^2 + (1-\beta)\mathsf{SINR}_B + \sum_{m=m_1}^{k_M} G_m}, \tag{29}$$

where m_1 denotes the position of the mMTC device in the sequence to be decoded subsequent to s_{B_1}. Therefore, the rate is given by

$$\mathsf{R}_{B_1}^{\mathsf{RSMA}} = \mathsf{R}(n, \mathsf{SINR}_{B_1}, \epsilon) \quad \text{(bits/s/Hz)}. \tag{30}$$

In the same way, the SINR of s_{B_2} can be calculated as

$$\mathsf{SINR}_{B_2} = \frac{(1-\beta)\mathsf{SINR}_B}{\sigma^2 + \sum_{m=m_2}^{k_M} G_m}, \tag{31}$$

where m_2 represents the number of mMTC devices that will be decoded after s_{B_2}. The corresponding rate is

$$\mathsf{R}_{B_2}^{\mathsf{RSMA}} = \mathsf{R}(n, \mathsf{SINR}_{B_2}, \epsilon) \quad \text{(bits/s/Hz)}. \tag{32}$$

For the mMTC devices m the SINR is

$$\mathsf{SINR}_m^M = \begin{cases} \frac{G_m}{\sigma^2 + \mathsf{SINR}_B + \sum_{i=m+1}^{k_M} G_m}, \\ \quad \text{if } m \text{ is decoded before } s_{B_1} \\ \frac{G_m}{\sigma^2 + (1-\beta)\mathsf{SINR}_B + \sum_{i=m+1}^{k_M} G_m} \\ \quad \text{if } m \text{ is decoded between } s_{B_1} \text{ and } s_{B_2} \\ \frac{G_m}{\sigma^2 + \sum_{i=m+1}^{k_M} G_m} \\ \quad \text{if } m \text{ is decoded after } s_{B_2}. \end{cases} \tag{33}$$

Let A_M and A_B represent the count of decoded mMTC and eMBB devices, respectively. Condition $A_B = 1$ is satisfied only if the sum of R_{B_1} and R_{B_2} is greater than or equal to R_B; otherwise, $A_B = 0$. For mMTC devices, they can be decoded if $\mathsf{R}(n, \mathsf{SINR}_m^M, \epsilon) \geq r_M$. Therefore, to determine all the feasible $(\mathsf{R}_B, \lambda_M)$ combinations, the problem can be expressed as in (27).

5. Resource optimization through deep Q-learning

As part of the model, neural networks, known as DQNs, approximate the Q-values associated with different state–action pairs, playing an important role in learning the optimal policy. This optimization is accomplished by the Q-Learning process. Q-Learning is an RL algorithm that is value-based and its main objective is to learn the optimal action-value function $Q(s, a)$, which denotes the expected return when taking action a in state s (Watkins and Dayan, 1992). The neural network's ability to capture complex environment patterns and relationships allows our model to be generalized across diverse scenarios. In addition, a replay buffer mechanism, an important component of training stability, is applied to the system to store and randomly sample past experiences. This enhances the model's learning efficiency and robustness, ensuring a more effective exploration of the state–action space and accelerating convergence towards an optimal solution.

The core concept in RL is MDP, which is defined by a tuple (S, A, P, R, γ). In this tuple, S represents the set of states, A represents the set of actions, P is the state transition probability function: $P(s'|s, a)$ denotes the probability of transition from state s to state s' when action a is taken. R is the reward function: $R(s, a, s')$ indicates the immediate reward obtained after transitioning from state s to state s' by taking action a. γ is the discount factor that defines the relative importance of future rewards (Gao et al., 2022).

The equation below (c.f. Sutton and Barto (2018)) presents the update rule at each time step t for the Q-value

$$Q(S_t, A_t) = (1-\alpha)Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a)), \tag{34}$$

where the learning rate, denoted as α, is a value between 0 and 1. The discount factor, denoted as γ, also falls within the range of 0 to 1 that determines the importance of future rewards, where a γ value of 0 only considers immediate rewards and a higher γ aims for higher long-term rewards. The variable R represents the reward.

DQN (Ramaswamy and Hullermeier, 2022) is a variant of Q-Learning, that uses DNNs to approximate the optimal action-value function in RL. The transitions are stored in a data buffer called the experience replay buffer based on their arrival time. When the buffer becomes full, newer data points replace older ones in the buffer. It has employed a ϵ-greedy policy for both exploration and sample generation. When using a neural network for function approximation, the state s serves as input to the network, while the $q(s, a)$ values for all actions in the state s serve as the output.

For both slicing approaches, a DQN agent, has been built by employing a neural network specified by Keras (Chollet et al., 2015). This network is responsible for mapping the state s of the environment to the Q-value of the agent for each available action a. Once the agent has been trained, it can utilize the predicted Q-values to determine the specific action executed in response to a particular state in the environment. A sequential model has been used with 3 hidden layers of neurons. The first hidden layer is dense, consisting of 192 ReLU neurons, where the network's shape is identical to the input layer, corresponding to the dimensionality of the state information s of each slicing environment. The second and third hidden layers are also dense with 256 and 64 ReLU neurons respectively. The output layer has dimensionality corresponding to the number of possible actions of each slicing approach with a linear activation to specify a separate neuron for each of the outcomes. The model has been compiled with mean squared error to compute the cost function and with Adam as routine optimizer with a learning rate of 0.001.

As the online update with a neural network with millions of weights does not perform efficiently, the agent selects a batch of experiences, equal to the batch size from the Replay Buffer and updates the Q-values. Also, a behavior policy was defined to explore the environment and store the samples $(s, a, r, s', done)$ in a buffer. The samples are generated using an exploratory behavior policy, while the deterministic target policy using Q-values is improved. Two networks are used in the training process: a target and a primary network. The target network is responsible for obtaining the maximum q-value of the next state, represented as $\max_a \hat{q}(s_{t+1}, a; w_t^-)$. On the other hand, the primary network has weights w_t and is updated through backpropagation of the temporal difference error across the network. This is represented by $\hat{q}(s_t, a, w_t)$, where w_t represents the weights of the neural network that is being learned during the DQN learning process.

The updated equation is given by

$$w_{t+1} = w_t + \alpha[R_{t+1} + \gamma \cdot \max_a \hat{q}(S_{t+1}, a, w_t^-) - \hat{q}(S_t, A_t, w_t)]\nabla \hat{q}(S_t, A_t, w_t). \tag{35}$$

To determine the action to be taken at timestep t, the agent randomly selects a value v between 0 and 1. This value, along with the hyperparameters ϵ, epsilon_decay, and $\epsilon_{\min}$, determines whether the agent chooses an exploratory or an exploitative action. At the start of the training, the value of ϵ is initially high. However, after each

episode, it gradually decreases based on the epsilon_decay parameter until it reaches the minimum value of $\epsilon_{\min}$. This will influence the decision to take an exploratory or exploitative action. In the initial episodes, the majority of actions will involve exploration. However, as the episodes progress, the agent will gradually decrease the number of exploratory actions it takes. This implies that, in the early episodes, the agent will primarily select random actions to learn through exploration. However, towards the end, the agent will primarily rely on the knowledge acquired by the model through memory replay. During the exploitation process, the state is provided as input to the prediction method of the model. This method returns the activation output for each possible action.

This study involved the creation of datasets to enable simulations for scenarios with URLLC and mMTC slicing optimizations. Each dataset comprises 100 timesteps, that accommodate a flexible configuration of frequencies and devices per frequency to capture various operational scenarios. The datasets consist of comprehensive data on the channel gains of each device at different frequencies. These gains were computed using block Rayleigh fading channels, a methodology expounded upon in Section 3, ensuring the incorporation of realistic propagation effects and enabling comprehensive evaluations of system performance. Significantly, the datasets associated with URLLC simulations were developed with two unique average power gains: scenario with URLLC mode A highlighting eMBB devices with mean values of 10 dB and URLLC devices with mean values of 20 dB, and scenario with URLLC mode B where eMBB devices had a mean of 20 dB and URLLC devices had a mean of 10 dB. Furthermore, for the scenario with mMTC simulations, datasets were developed with two sub-scenarios, specifically scenario mode A where eMBB devices had a mean of 20 dB and mMTC devices had a mean of 5 dB, and mode B where eMBB devices had a mean of 5 dB and mMTC devices had a mean of 20 dB.

5.1. Scenario with URLLC

The primary goal of the scenario with URLLC is to carry out a combined policy and resource allocation for the strategy of sharing spectrum (OMA, NOMA, RSMA) and the frequency slots F_U assigned to the (unidentified) URLLC devices. Therefore, the goal is to maximize

$$\max_{x\in\{\mathsf{OMA},\mathsf{NOMA},\mathsf{RSMA}\}} \mathsf{R}_U^{\mathsf{sum},x}$$
$$\text{subject to } \mathsf{R}_U^{\mathsf{sum},x} \geq \underline{R}$$
$$F_U \leq F. \tag{36}$$

To address the optimization problem mentioned above, two possible approaches can be explored. The first involves assigning a predetermined number of frequency slots to URLLC. However, this approach may not yield the optimal solution. On the other hand, the second approach involves conducting an exhaustive search providing the optimal solution. However, this approach comes with increased complexity. To successfully address this optimization problem using DRL, it must be defined the system state space and decision actions. Specifically, when considering scenario with URLLC slicing, the state space S can be described as follows

$$S = \{(k_u, p, f) : k_u \geq 0, p \in \{\mathsf{OMA},\mathsf{NOMA},\mathsf{RSMA}\},$$
$$0 \leq f \leq F\}, \tag{37}$$

where in (k_u, p, f), k_u denotes the number of URLLC devices, p denotes the transmission and decoding strategy (OMA, NOMA, or RSMA), and f is the slicing policy, i.e., the number of slots for URLLC.

The action space $\mathcal{A}$ for this slicing is formed by:

- Increase/decrease the number of allocated frequency slots to URLLC by 1, without violating the problem constraints.
- Change the transmission and decoding policy within the set {OMA, NOMA, RSMA}, according to the current state.

The slicing problem's reward function is defined as the objective function $\mathsf{R}_{Ut}^{\mathsf{sum},x} + \mathsf{R}_{Bt}^{\mathsf{sum},x}$, which aims to maximize the sum rate for both the URLLC devices and the eMBB device. Here, the decoding policy used at the time step t is denoted by x, which can take values from the set {OMA, NOMA, RSMA}.

5.2. Scenario with mMTC

In the scenario with mMTC, the optimization problem is to maximize the expected arrival rate λ_M^{x} of mMTC subject to the system constraints: a maximum error probability for decoding the eMBB device, and a maximum bit error probability for the recovery of each finite-length message for the mMTC device. Again, the blocklength n and the bit error probability in decoding are set as fixed parameters, and p refers to the transmission/decoding policy in use (OMA, NOMA, or RSMA). Which of these is optimal (for maximizing λ_M will depend in a complex manner on the optimization parameters included in the model.

The state space S can be described as follows:

$$S = \{(n_m, p, P_m, A_m) : n_m \geq 0,$$
$$p \in \{\mathsf{OMA},\mathsf{NOMA},\mathsf{RSMA}\},$$
$$P_m \leq \epsilon_m, A_m \geq 0\}, \tag{38}$$

where in (n_m, p, P_m, A_m), n_m denotes the number of mMTC active devices to decode, p denotes the transmission and decoding strategy (OMA, NOMA, or RSMA), P_m is the expectation, and A_m is the number of successfully mMTC decoded above a fixed target rate.

The action space $\mathcal{A}$ for this slicing is formed by:

- Increase/Decrease the number of admitted mMTC devices by 1, without violating the problem constraints.
- Change the transmission and decoding policy within the set {OMA, NOMA, RSMA}, according to the current state.

The reward function for the slicing problem can be represented by the objective function $\mathsf{R}_M^{\mathsf{sum},x}(t) + \mathsf{R}_B^{\mathsf{sum},x}(t) + n_{\mathrm{M}}(t) + n_{\mathrm{B}}(t)$. This function maximizes the sum rate and number of successfully decoded devices. Here, x belongs to the set {OMA, NOMA, RSMA} and represents the decoding policy. Additionally, r and n represent, respectively, the sum rate obtained and the number of decoded devices by mMTC and eMBB at each time step t.

6. Results and analysis

To simulate the resource optimization using the DRL agent, the baseline values were calculated for every simulation using the three decoding strategies OMA, NOMA, and RSMA. The results from the simulations involving scenarios with URLLC and mMTC, along with a discussion, are presented.

6.1. Scenario with URLLC

Table 1 presents the hyperparameters used for simulations in the scenario with URLLC and in the scenario with mMTC.

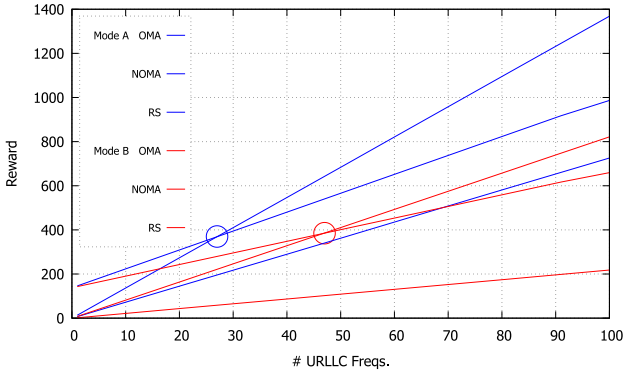
Table 2 presents the network parameters used during the scenario with URLLC slicing simulations. n is the size of the finite blocklength, σ^2 represents the variance, mode A and B are the average signal powers that eMBB and URLLC devices used in this scenario, γ_B, γ_U are the average channel gains for eMBB and URLLC, ϵ_B, ϵ_U are the reliability requirements of eMBB and URLLC, the power factor used by RSMA decoding scheme for the first part of the message, the next values, depending on the simulation setup, can be set between [1−100], F is the number of frequencies available on the BS, F_u is maximum number of frequencies that can be allocated to URLLC devices and URLLC devices represent the number of URLLC devices present in the simulation.

Table 1
DRL simulation parameters.

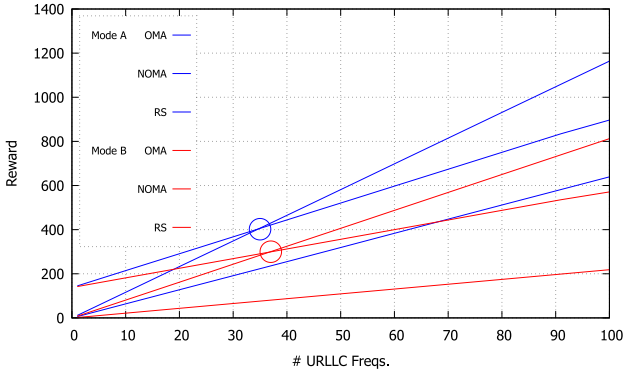
Parameter	Value
Episodes	{10 000, 20 000, 30 000}
Timesteps per episode	100
start ϵ	1
end ϵ	0.05
ϵ decay stops at the episode	1000
First hidden layer	192 ReLU neurons
Second hidden layer	256 ReLU neurons
Third hidden layer	64 ReLU neurons
Output layer	linear activation
Experience replay size	500
Optimizer	Adam
Learning rate	0.001

Table 2
Network parameters for scenario with URLLC simulations.

Parameter		Value
n (bits)		1000
σ^2		1
Average signal powers (dBm):	mode A	eMBB 10 | URLLC 20
	mode B	eMBB 20 | URLLC 10
γ_B, γ_U (dB)		10
Reliabilities ϵ_B, ϵ_U		10^{-3}, 10^{-4}
RSMA power factor of URLLC messages		0.7
F		[1 – 100]
F_u		[1 – 100]
URLLC devices		[1 – 100]



(a) Baseline mean reward values from 1 to 100 URLLC frequencies with 1 URLLC device



(b) Baseline mean reward values from 1 to 100 URLLC frequencies with 100 URLLC device

Fig. 4. Baseline mean rewards.

Figs. 4(a) and 4(b) present the baseline mean reward values for 1 and 100 URLLC frequencies with one and 100 URLLC devices. The scenario with URLLC mode A is presented in blue and the mode B in red. The number of frequencies allocated to URLLC devices gradually increases from 1 to 100. The baseline results indicate that OMA outperforms NOMA and RSMA in terms of rewards when a smaller number of frequencies are assigned to URLLC. Nevertheless, as the number of frequencies allocated to URLLC increases, RSMA yields higher rewards. However, the rewards of NOMA are lower compared to OMA and RSMA. The turning point, where RSMA improves upon OMA, is indicated by circles. It is important to note that, as described in (15), when the OMA decoding scheme is used, eMBB and URLLC devices do not share the frequencies available on the BS. The device for eMBB will use the remaining frequencies not used by URLLC. And as for NOMA and RSMA decoding policies, eMBB and URLLC devices share all F frequency resources.

Table 3 presents an outline of the simulation results. The agent was tested under two power gains sub-scenarios, varying the total frequencies available (denoted by F) in the BS with fixed/random frequencies for URLLC (denoted by F_u) with fixed/random number of URLLC devices. The baseline and DRL decoding schemes applied, illustrate how many times each decoding scheme was respectively used during the 100 time steps of each simulation. Baseline cumulative reward indicates the maximum possible reward obtained in each simulation condition. The DRL cumulative reward indicates the rewards achieved by the agent and the DRL cumulative percentage reward shows the percentage of reward obtained by the agent compared to the baseline values. The training episodes denote the number of episodes used during the training sessions.

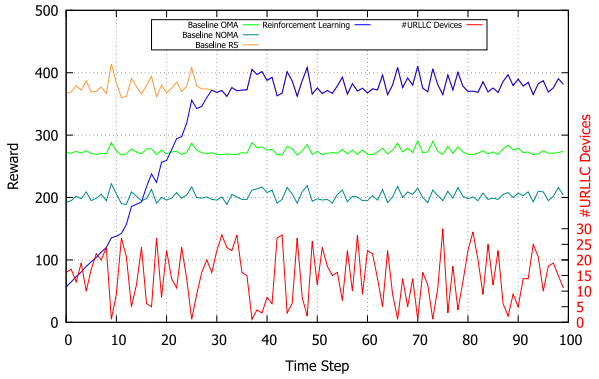
Fig. 5(a) presents the rewards obtained during simulation URLLC-1. The power gains of the devices were from mode A. A total of 100 timesteps were simulated, and during each timestep, a random number of URLLC devices arrived at the base station. The number of arrivals ranged from 1 to 30 and the number of available frequencies at the BS was set to 30. The plotted rewards depict the baseline maximum performance for OMA, NOMA, and RSMA alongside the rewards achieved by the DRL agent. These results confirm the discernible pattern in the agent's decision-making process. Initially, spanning the first 30 timesteps, the agent consistently selects actions to increase the number of frequencies allocated to URLLC, indicative of its learning strategy to accommodate varying URLLC device arrivals. Subsequently, beyond timestep 30, a plateau is observed in the number of URLLC frequencies allocated, suggesting the attainment of the maximum feasible allocation. Furthermore, the decoding policy transitions towards RSMA, signifying the agent's adaptive behavior in response to changing environmental dynamics.

As depicted in Figs. 4(a) and 4(b), the effectiveness of OMA decoding outperforms other decoding choices when the number of frequencies assigned to URLLC is kept below a specific limit. To evaluate the agent's flexibility in choosing suitable decoding strategies based on the frequencies allocated for URLLC, a simulation is depicted in Fig. 5(b) that represents simulation URLLC-3, with a constant number of 100 URLLC devices and a random quantity of URLLC frequencies that change at each time step. The results indicate that the agent is skilled at adaptively changing its decoding approach according to the assigned URLLC frequencies. In this particular simulation, as the number of URLLC frequencies approaches 20, the rewards obtained from both OMA and RSMA decoding choices show a remarkable similarity, suggesting that the selection between these decoding alternatives has a minimal influence on the total rewards. The results from the data depicted by the blue circles show that for URLLC frequencies below 15, the primary choice for decoding is OMA. Noteworthy is that in this simulation, there is no direct increase or decrease in URLLC frequencies as a specific action. Instead, the agent's decision-making process is based entirely on evaluating the current situation and choosing the best decoding action, demonstrating its adeptness when evaluated with different URLLC frequency assignments.

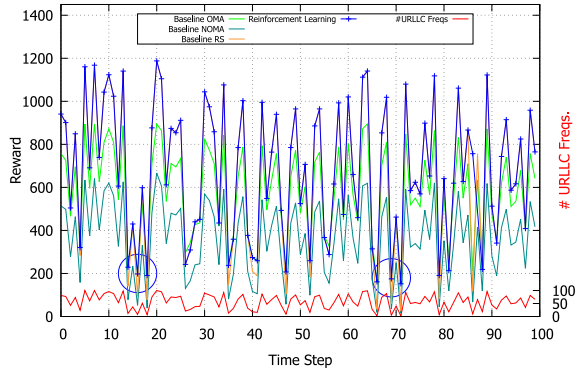
Table 3
DRL training in scenario with URLLC simulations.

# Sim.	Mode	F	F_u	# URLLC devices	Baseline # Decoding schemes applied			DRL # Decoding schemes applied			Baseline cumulative reward	DRL cumulative reward	DRL cumulative reward (%)	Training episodes
					OMA	NOMA	RSMA	OMA	NOMA	RSMA				
URLLC-1*	*A*	30	30	[1–30]	0	0	100	11	0	89	37 941.31	32 712.76	86.21	20 000
URLLC-2	*A*	10	1	[1–30]	100	0	0	100	0	0	2309.18	2309.18	100	10 000
URLLC-3	*A*	100	[1–100]	100	24	0	76	12	0	88	67 767.61	67 318.76	99.33	10 000
URLLC-4*	*B*	30	30	[1–30]	0	0	100	14	0	86	24 891.80	20 914.69	84.02	20 000
URLLC-5	*B*	10	1	[1–30]	100	0	0	100	0	0	1842.98	1842.98	100	10 000
URLLC-6	*B*	100	[1–100]	100	19	0	81	9	1	90	47 853.72	46 277.56	96.70	10 000

The agent dynamically increments or decrements the number of frequencies allocated to URLLC.



(a) Simulation URLLC-1



(b) Simulation URLLC-3

Fig. 5. Rewards obtained for scenario with URLLC.

The outcomes of the DRL agent training reveal a consistent trend: assigned the task of adjusting the number of URLLC frequencies, the agent's cumulative reward percentage stabilizes at approximately 85%. This behavior is linked to the agent's initial evaluation phase, which aims to identify the optimal URLLC frequency count, leading to lower rewards in the early stages. As training progresses, the agent opts for the maximum URLLC frequency count to optimize the rewards. In contrast, when tested in scenarios with randomized URLLC frequency counts, the agent demonstrates proficiency in selecting the suitable decoding scheme to maximize rewards, demonstrating its adaptability and ability to make informed decisions under varying circumstances.

6.2. Scenario with mMTC

Table 4 presents the network parameters used during the scenario with mMTC slicing simulations. The parameters are similar to those of Table 2 being set to the mMTC devices. The difference relies on the values for the target eMBB SINR mode *A* and *B* that the eMBB device

Table 4
Network parameters for scenario with mMTC simulations.

Parameter		Value
n (bits)		1000
σ^2		1
Average signal powers (dBm):	mode *A*	eMBB 20 | mMTC 5
	mode *B*	eMBB 5 | mMTC 20
Target eMBB SINR (dBm)	mode *A*	20
	mode *B*	5
γ_B, γ_M (dB)		10
Reliabilities ϵ_B, ϵ_M		10^{-3}, 10^{-1}
RSMA power factor (1st part of msg)		0.7
OMA fraction of time allocated to eMBB		0.2
Fixed mMTC target rate (bps)		0.1
F		1
mMTC devices		[1 – 1000]

used, the OMA fraction of time allocated to eMBB at each time step and the fixed mMTC target rate.

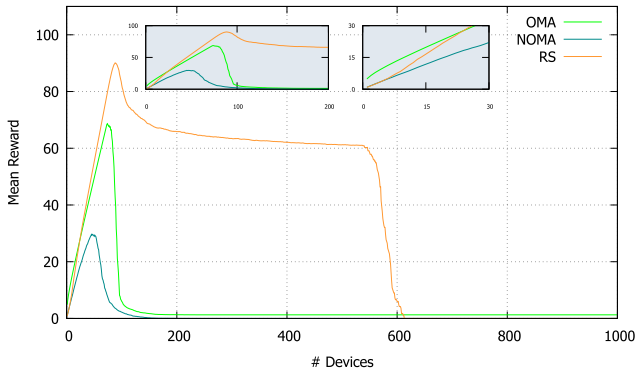
Fig. 6(a) and Fig. 6(b) illustrate the baseline computations of the average rewards achieved using OMA, NOMA, and RSMA over 100 time steps with power gains of scenario with mMTC mode *A* and mode *B* respectively. These calculations cover a variable number of mMTC devices, from 1 to 1000. In the baseline representation of mode *A*, a clear pattern becomes evident as the quantity of mMTC devices grows: RSMA demonstrates resilience by retaining its ability to decode mMTC devices even as the number of devices increases, albeit with a gradual decrease in rewards, maintaining its effectiveness up to around 600 devices. On the other hand, the rewards achieved by OMA and NOMA show a noticeable reduction when the number of mMTC devices exceeds specific thresholds, with OMA experiencing a drop at approximately 70 devices and NOMA at around 30 devices. A comparison between OMA and RSMA demonstrates that OMA surpasses RSMA in reward performance until approximately 25 mMTC devices need to be decoded. This suggests a turning point where RSMA becomes more beneficial in managing a larger number of mMTC devices.

As depicted in Fig. 6(b), it is observed that RSMA and NOMA have similar rewards, suggesting that there is no notable distinction between the two decoding approaches. Despite comparable rewards, NOMA still outperforms RSMA in terms of decoding performance when dealing with a peak of 90 mMTC devices. In line with the scenario with mode *A*, when there are fewer mMTC devices to decode (up to 15), OMA is confirmed to be the most effective in terms of rewards, with NOMA following closely behind, and then RSMA. This underscores the effectiveness of OMA in scenarios with a lower number of mMTC devices.

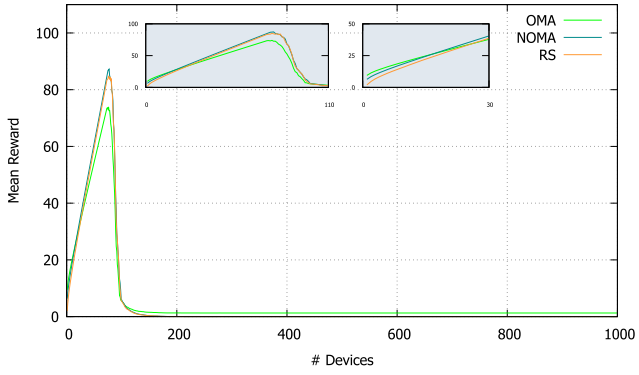
The outcomes depicted in Table 5 showcase a variety of simulations conducted using different experimental setups. These simulations evaluated parameters similar to those in Table 3. The distinctions lie in

Table 5
DRL training in scenario with mMTC simulations.

# Sim.	Mode	# inc/dec in action	Max mMTC devices	Baseline # Decoding schemes applied			DRL # Decoding schemes applied			Baseline cumulative reward	Baseline cumulative reward	DRL cumulative reward (%)	Training episodes
				OMA	NOMA	RSMA	OMA	NOMA	RSMA				
mMTC-1	*A*	20	200	0	0	100	0	0	100	9248.08	7824.40	84.6	10 000
mMTC-2	*A*	5	200	0	0	100	0	0	100	9248.08	8123.88	87.84	15 000
mMTC-3	*A*	1	200	0	0	100	0	0	100	9248.08	8403.54	90.86	30 000
mMTC-4	*A*	2	10	100	0	0	100	0	0	1522.74	1468.78	96.45	10 000
mMTC-5	*A*	1	10	100	0	0	100	0	0	1522.74	1522.74	100	10 000
mMTC-6	*B*	1	200	0	94	6	1	98	1	9951.84	8348.87	83.89	30 000
mMTC-7	*B*	1	10	100	0	0	100	0	0	2021.72	1970.60	97.47	10 000



(a) with mode *A* power gains



(b) with mode *B* power gains

Fig. 6. Baseline mean reward values from 1 to 1000 mMTC devices.

parameters as the number of devices increased or decreased by the DRL agent (# inc/dec in action) to be admitted to the BS; and the highest permissible number of mMTC devices that can be housed in the BS to optimize the rate per device.

Fig. 7(a) illustrates the rewards achieved by the DRL agent during its training in simulation mMTC-3, emulating the admission of up to 200 mMTC devices on the BS. From the results, as the number of mMTC devices to be admitted in the simulations increases, the agent faces greater challenges, because more episodes are needed in the training phase. Nevertheless, it develops an almost optimal policy for determining the appropriate number of mMTC devices to admit and for selecting the correct decoding strategy. A reduced number of increments or decrements in mMTC devices also corresponds to the increased training episodes needed to attain satisfactory results.

Fig. 7(b) depicts the rewards obtained from simulation mMTC-5, involving 10 mMTC devices with power gains from mode *A*. With 10 mMTC devices, the DRL agent successfully generalizes the strategy by choosing OMA as the best decoding choice and accurately deciding on the number of mMTC devices to accept. As a result, having fewer devices improves the agent's ability to generalize the accurate strategy.

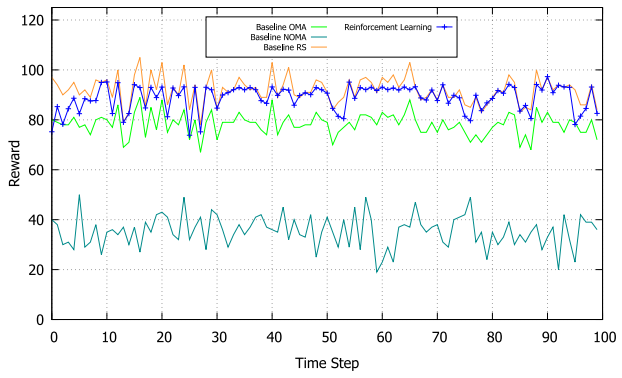
When dealing with 200 mMTC devices, the DRL agent faces difficulties in estimating how many devices to accurately decode. Nevertheless, when choosing the best decoding strategy, the agent shows an effective generalization over different time steps. When comparing the results of simulations involving up to 200 mMTC decoding devices, improved outcomes are obtained when increasing or decreasing the number of devices by one. However, this improvement comes at the cost of needing a higher number of training episodes to achieve these overall rewards. On the other hand, when dealing with 10 mMTC decoding devices, the agent demonstrates its generalization capabilities due to the less complex task of determining the accurate number of devices to decode. A notable observation is seen in the scenario with mMTC mode *A*, where OMA consistently emerges as the most effective decoding choice, leading the agent to achieve 100% of cumulative reward. In contrast, in a similar simulation under the power gain of mode *B*, where all the decoding choices produce identical results, the agent has more difficulty selecting the appropriate decoding option. However, no significant discrepancies are noted in the final result because the difference between the rewards obtained by each is minimal. Within the context of the scenario with mMTC, the transmission powers chosen for both types of traffic are rather different, so NOMA in the power domain can (under favorable channel conditions) discriminate considerable precision the superposed signals when a substantial degree of sharing exists; whereas RSMA, doing a power split between the sub-messages has the effect of equalizing the received powers.

6.3. Discussion

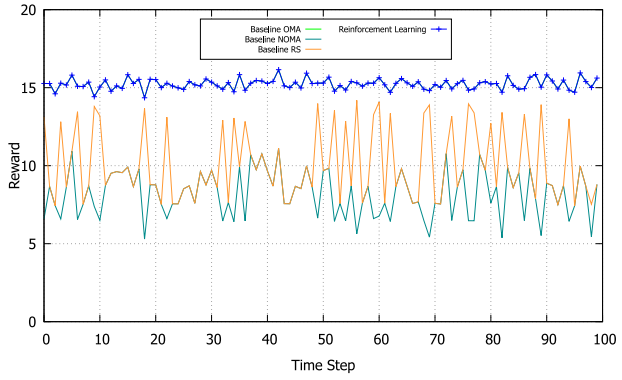
The efficiency of the DRL agent in enhancing resource allocation for the scenario with URLLC and with mMTC has been evaluated by simulations, where the agent was compared with a baseline.

Regarding the scenario with URLLC, the DRL agent presented a minimum efficacy of 84.02% and achieved a maximum of 100%. In addition, baseline simulations showed that OMA outperforms NOMA and RSMA in scenarios where the number of frequencies allocated to URLLC is limited. However, as the number of URLLC frequencies increases, RSMA starts to yield higher rewards. The DRL agent was trained with multiple hyperparameters and scenarios, and its performance showed adaptability. It learned to dynamically allocate URLLC frequencies and choose the appropriate decoding scheme based on the current environment. It has been shown that OMA is more effective in scenarios with a smaller number of frequencies URLLC, demonstrating its effectiveness in scenarios with limited resources. The policies obtained by the DRL agent, allow an adaptive decoding scheme: As the number of URLLC frequencies increased, the agent gradually shifted towards RSMA, presenting an adaptive behavior to optimize rewards.

Regarding the scenario with mMTC, the agent's performance ranged from a low of 83.89% to a peak efficacy of 100%. Also, the baseline simulations revealed that RSMA exhibits resilience in decoding mMTC devices even with increasing device numbers, maintaining effectiveness



(a) Simulation mMTC-3



(b) Simulation mMTC-5

Fig. 7. Rewards obtained for scenario with mMTC.

up to a threshold with power gains from mode *A*, while NOMA tend to be more effective when power gains used are from mode *B*. The DRL agent, when trained with mMTC devices ranging from 10 to 200, developed near-optimal policies for admitting devices and selecting decoding schemes. The agent demonstrated improved generalization with fewer devices, accurately choosing OMA as the best decoding scheme for 10 mMTC devices. While incrementing/decrementing devices by one led to higher results in terms of cumulative reward, it required more training episodes. Simpler tasks, such as handling 10 mMTC devices, resulted in maximum cumulative rewards with OMA.

For both scenarios, the DRL agent showcased the potential for adaptive resource allocation under dynamic network conditions, improving spectral efficiency. Through training, the agent learned to select the most effective decoding strategy for two distinct scenarios, balancing the requirements of the devices URLLC and mMTC.

7. Conclusions

5G networks support the following use cases based on 3GPP specification: eMBB, URLLC, and mMTC, each with its bandwidth, latency, and connectivity requirements. By implementing OMA, NOMA, and RSMA decoding schemes, 5G network operators dynamically allocate resources and optimize network capacity. Given these circumstances, it becomes a complex task for network operators to dynamically determine the most suitable decoding method for the network slicing, considering the diverse requirements of the 5G use cases.

This paper proposed a DRL agent to optimize resource allocation and decoding schemes in network slicing scenarios with eMBB, URLLC, and mMTC. The DRL agent evaluates the performance of different decoding schemes such as OMA, NOMA, and RSMA and applies the best decoding scheme, in these scenarios with dynamic network conditions. The DRL agent has been tested in two scenarios, eMBB with URLLC and eMBB with mMTC. In each scenario, the network conditions were tested by altering the device count, device power gains, and the allocation of frequencies. The outcomes from these evaluations indicated that the efficiency of selecting the decoding schemes OMA, NOMA, and RSMA varied between roughly 84% and 100% across both scenarios.

Future research could explore the integration of all three use cases 5G (eMBB, URLLC, mMTC) simultaneously while accounting for additional network constraints, enhancing the complexity of network slicing scenarios. Incorporating alternative machine learning techniques beyond DRL, such as hybrid or supervised learning approaches, could improve decision-making efficiency and address the significant challenge of computational overhead, especially in large-scale networks. Techniques like model compression, hardware acceleration, and parallel processing should be investigated to optimize computational efficiency. Additionally, incorporating user behavior models could refine the network's ability to adapt resources based on real-time demand. Real-world implementations are necessary to validate the findings, addressing practical challenges such as integrating DRL-based solutions with existing network infrastructure. This may require enhancements in SDN and network function virtualization (NFV) for seamless operation. The DRL agent's real-time adaptability and scalability could be further enhanced by exploring edge computing and distributed processing. As the framework progresses to 6th Generation (6G) networks, incorporating novel services such as extended reality and quantum networking, guaranteeing compatibility and performance within these advanced environments will be crucial for future research and development.

CRediT authorship contribution statement

Silvestre Malta: Writing – original draft, Validation, Methodology, Investigation. **Pedro Pinto:** Supervision. **Manuel Fernández-Veiga:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the grant PID2020-113795RB-C33 funded by MICIU/AEI/10.13039/501100011033 (COMPROMISE project), and "TRUFFLES: TRUsted Framework for Federated LEarning Systems, within the strategic cybersecurity projects (INCIBE, Spain), funded by the Recovery, Transformation and Resilience Plan (European Union, Next Generation).

Data availability

The authors are unable or have chosen not to specify which data has been used.

References

Abdullah, D.M., Ameen, S.Y., 2021. Enhanced mobile broadband (EMBB): A review. J. Inf. Technol. Inform. 1 (1), 13–19.

Ahsan, W., Yi, W., Liu, Y., Nallanathan, A., 2022. A reliable reinforcement learning for resource allocation in uplink NOMA-URLLC networks. IEEE Trans. Wireless Commun. 21 (8), 5989–6002. http://dx.doi.org/10.1109/twc.2022.3144618.

Alsenwi, M., Tran, N.H., Bennis, M., Pandey, S.R., Bairagi, A.K., Hong, C.S., 2021. Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: A deep reinforcement learning based approach. IEEE Trans. Wireless Commun. 20 (7), 4585–4600. http://dx.doi.org/10.1109/TWC.2021.3060514.

Bakri, S., Brik, B., Ksentini, A., 2021. On using reinforcement learning for network slice admission control in 5G: Offline vs. online. Int. J. Commun. Syst. 34 (7), e4757. http://dx.doi.org/10.1002/dac.4757, e4757 dac.4757. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/dac.4757. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4757.

Bega, D., Gramaglia, M., Banchs, A., Sciancalepore, V., Costa-Pérez, X., 2020. A machine learning approach to 5G infrastructure market optimization. IEEE Trans. Mob. Comput. 19 (3), 498–512. http://dx.doi.org/10.1109/TMC.2019.2896950.

Bockelmann, C., Pratas, N., Nikopour, H., Au, K., Svensson, T., Stefanovic, C., Popovski, P., Dekorsy, A., 2016. Massive machine-type communications in 5g: physical and MAC-layer solutions. IEEE Commun. Mag. 54 (9), 59–65. http://dx.doi.org/10.1109/MCOM.2016.7565189.

Caire, G., Taricco, G., Biglieri, E., 1999. Optimum power control over fading channels. IEEE Trans. Inform. Theory 45 (5), 1468–1489. http://dx.doi.org/10.1109/18.771147.

Chen, H., Abbas, R., Cheng, P., Shirvanimoghaddam, M., Hardjawana, W., Bao, W., Li, Y., Vucetic, B., 2018. Ultra-reliable low latency cellular networks: Use cases, challenges and approaches. IEEE Commun. Mag. 56 (12), 119–125. http://dx.doi.org/10.1109/MCOM.2018.1701178.

Chollet, F., et al., 2015. Keras. https://keras.io.

Comşa, I.-S., Zhang, S., Aydin, M.E., Kuonen, P., Lu, Y., Trestian, R., Ghinea, G.a., 2018. Towards 5G: A reinforcement learning-based scheduling solution for data traffic management. IEEE Trans. Netw. Serv. Manag. 15 (4), 1661–1675. http://dx.doi.org/10.1109/TNSM.2018.2863563.

Debbabi, F., Jmal, R., Chaari Fourati, L., 2021. 5G network slicing: Fundamental concepts, architectures, algorithmics, projects practices, and open issues. Concurr. Comput.: Pract. Exper. 33 (20), e6352. http://dx.doi.org/10.1002/cpe.6352, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6352. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6352.

Filali, A., Mlika, Z., Cherkaoui, S., Kobbane, A., 2022. Dynamic SDN-based radio access network slicing with deep reinforcement learning for URLLC and eMBB services. IEEE Trans. Netw. Sci. Eng. 9, 2174–2187. http://dx.doi.org/10.1109/TNSE.2022.3157274.

Gao, T., Chen, B., Mi, Q., 2022. A survey of Markov model in reinforcement learning. In: 2022 International Conference on Artificial Intelligence in Information and Communication. ICAIIC, IEEE, pp. 284–287.

Gokhan Kalem, B.S., Tozan, H., 2021. Technology forecasting in the mobile telecommunication industry: A case study towards the 5G era. Eng. Manage. J. 33 (1), 15–29. http://dx.doi.org/10.1080/10429247.2020.1764833, arXiv:http://dx.doi.org/10.1080/10429247.2020.1764833.

Ji, H., Park, S., Yeo, J., Kim, Y., Lee, J., Shim, B., 2018. Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects. IEEE Wirel. Commun. 25 (3), 124–130. http://dx.doi.org/10.1109/MWC.2018.1700294.

Khumalo, N.N., Oyerinde, O.O., Mfupe, L., 2021. Reinforcement learning-based resource management model for fog radio access network architectures in 5G. IEEE Access 9, 12706–12716. http://dx.doi.org/10.1109/ACCESS.2021.3051695.

Le, T.-K., Salim, U., Kaltenberger, F., 2021. An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17. IEEE Access 9, 433–444. http://dx.doi.org/10.1109/ACCESS.2020.3046773.

Li, J., Shi, W., Zhang, N., Shen, X.S., 2019. Reinforcement learning based VNF scheduling with end-to-end delay guarantee. In: 2019 IEEE/CIC International Conference on Communications in China. ICCC, IEEE, pp. 572–577.

Lin, X., 2022. An overview of 5G advanced evolution in 3GPP release 18. IEEE Commun. Stand. Mag. 6 (3), 77–83. http://dx.doi.org/10.1109/MCOMSTD.0001.2200001.

Liu, Y., Clerckx, B., Popovski, P., 2023. Network slicing for eMBB, URLLC, and mMTC: An uplink rate-splitting multiple access approach. IEEE Trans. Wireless Commun. 1–14. http://dx.doi.org/10.1109/TWC.2023.3295804.

Liu, Q., Han, T., Moges, E., 2020. EdgeSlice: Slicing wireless edge computing network with decentralized deep reinforcement learning. In: 2020 IEEE 40th International Conference on Distributed Computing Systems. ICDCS, pp. 234–244. http://dx.doi.org/10.1109/ICDCS47774.2020.00028.

López-Pérez, D., De Domenico, A., Piovesan, N., Xinli, G., Bao, H., Qitao, S., Debbah, M., 2022. A survey on 5G radio access network energy efficiency: Massive MIMO, lean carrier design, sleep modes, and machine learning. IEEE Commun. Surv. Tutor. 24 (1), 653–697. http://dx.doi.org/10.1109/COMST.2022.3142532.

Mao, Y., Dizdar, O., Clerckx, B., Schober, R., Popovski, P., Poor, H.V., 2022. Rate-splitting multiple access: Fundamentals, survey, and future research trends. IEEE Commun. Surv. Tutor. 24 (4), 2073–2126. http://dx.doi.org/10.1109/COMST.2022.3191937.

Polyanskiy, Y., Poor, H.V., Verdu, S., 2010. Channel coding rate in the finite blocklength regime. IEEE Trans. Inform. Theory 56 (5), 2307–2359. http://dx.doi.org/10.1109/tit.2010.2043769.

Popovski, P., Trillingsgaard, K.F., Simeone, O., Durisi, G., 2018. 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. IEEE Access 6, 55765–55779. http://dx.doi.org/10.1109/access.2018.2872781.

Ramaswamy, A., Hullermeier, E., 2022. Deep Q-learning: Theoretical insights from an asymptotic analysis. IEEE Trans. Artif. Intell. 3 (2), 139–151. http://dx.doi.org/10.1109/tai.2021.3111142.

Rezazadeh, F., Chergui, H., Alonso, L., Verikoukis, C., 2020. Continuous multi-objective zero-touch network slicing via twin delayed DDPG and openai gym. In: GLOBECOM 2020 - 2020 IEEE Global Communications Conference. pp. 1–6. http://dx.doi.org/10.1109/GLOBECOM42002.2020.9322237.

Robaglia, B.t.-M., Coupechoux, M., Tsilimantos, D., 2023. Deep reinforcement learning for uplink scheduling in NOMA-urllc networks. http://dx.doi.org/10.48550/ARXIV.2308.14523, arXiv preprint arXiv:2308.14523. arXiv arXiv:2308.14523.

Santos, E.J.D., Souza, R.D., Rebelatto, J.L., 2021. Rate-splitting multiple access for URLLC uplink in physical layer network slicing with eMBB. IEEE Access 9, 163178–163187. http://dx.doi.org/10.1109/ACCESS.2021.3134207.

Setayesh, M., Bahrami, S., Wong, V., 2022. Resource slicing for eMBB and URLLC services in radio access network using hierarchical deep learning. IEEE Trans. Wireless Commun. 1. http://dx.doi.org/10.1109/TWC.2022.3171264.

Sharma, S.K., Wang, X., 2019. Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions. IEEE Commun. Surv. Tutor. 22 (1), 426–471.

Subedi, P., Alsadoon, A., Prasad, P.W.C., Rehman, S., Giweli, N., Imran, M., Arif, S., 2021. Network slicing: a next generation 5G perspective. EURASIP J. Wireless Commun. Networking 2021 (1), 102. http://dx.doi.org/10.1186/s13638-021-01983-7.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: an Introduction. MIT Press.

Tominaga, E.N., Alves, H., López, O.L.A., Souza, R.D., Rebelatto, J.L., Latva-aho, M., 2021. Network slicing for eMBB and mMTC with NOMA and space diversity reception. In: 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring). pp. 1–6. http://dx.doi.org/10.1109/VTC2021-Spring51267.2021.9448974.

Watkins, C.J.C.H., Dayan, P., 1992. Q-learning. Mach. Learn. 8 (3), 279–292. http://dx.doi.org/10.1007/BF00992698.

Zhu, Y., Hu, Y., Yuan, X., Gursoy, M.C., Poor, H.V., Schmeink, A., 2023. Joint convexity of error probability in blocklength and transmit power in the finite blocklength regime. IEEE Trans. Wireless Commun. 22 (4), 2409–2423. http://dx.doi.org/10.1109/TWC.2022.3211454.

Silvestre Malta received a B.S. degree in Computer Engineering from the Instituto Politécnico de Leiria, Portugal, and an M.S. degree in Information Systems from the Instituto Politécnico de Viana do Castelo, Portugal. Currently, he is pursuing a PhD degree in Computer Science from Vigo University, Spain. Since 2004, he has been a Network Engineer in several companies, and since 2010 he has been an Invited Assistant Professor at Instituto Politécnico de Viana do Castelo, Portugal. His research interests include IA/ML applied in the area of computer networks.

Pedro Pinto received an MSc degree (2007) in Communication Networks and Services, from the University of Porto, Portugal. He received his PhD degree (2015) in Telecommunications jointly from the Universities of Minho, Aveiro, and Porto, Portugal. Currently, he is an Assistant Professor at Instituto Politécnico de Viana do Castelo, Portugal, and a researcher affiliated with the INESC TEC research institution. His research interests include the areas of computer networks, data privacy, and cybersecurity.

Manuel Fernández-Veiga received his PhD degree in telecommunications engineering from the University of Vigo (Spain) in 2001, where he is now an associate professor and a researcher affiliated with the Atlantic research center within the Information & Computing Lab. He has authored or co-authored over 80 papers in peer-reviewed international conferences and journals. His main research interests include wireless communications, distributed algorithms, and quantum Internet