High-level service type analysis and MORL-based network slice configuration for Cell-Free-based 6G Networks

Navideh Ghafouri, *Student Member, IEEE,* John S. Vardakas, *Senior Member, IEEE,* Adlen Ksentini, *Senior Member, IEEE,* and Christos Verikoukis, *Senior Member, IEEE*

***Abstract*—Network slicing has garnered significant attention within the telecommunications community since the introduction of 5G. However, achieving dynamic and intelligent network slice configuration to accommodate diverse service types remains a critical challenge in advanced network orchestration. With the advent of 6G, which is characterized by its highly dynamic and robust nature, there is an urgent need for an intelligent and slice-compatible assignment approach to meet the evolving demands of next-generation networks. In this context, this work introduces an end-to-end network slicing framework that spans from the user to the Centralized Unit, within a system model incorporating an Open Radio Access Network and Cell-Free massive Multiple-Input Multiple-Output architecture. Our contribution begins with a detailed review of the anticipated 6G Key Performance Indicators and their implications for network slicing. We then propose a novel approach that leverages Multi-Objective Reinforcement Learning (MORL) to enable a single intelligent agent to address multiple service requirements through a unified training phase. By replacing multiple specialized agents with a single MORL agent, our approach significantly improves the scalability, reduces the complexity, and enhances the practicality of network slicing orchestration—while maintaining optimal system performance. Numerical results validate the effectiveness of the proposed MORL-based solution. The trained agent not only ensures the Quality of Service for diverse user service requests but also successfully manages the coexistence of conflicting service types. This includes accommodating the stringent requirements of Extremely Reliable and Low-Latency Communications alongside Further-Enhanced Mobile Broadband services within the same network environment.**

***Index Terms*—6G Networks, Open RAN, Network Slicing, Multi-objective Reinforcement Learning, Clustering.**

I. Introduction

With the commercial deployment of 5G underway, the focus of academia and industry has already shifted to the next generation of mobile communications, known as 6G. 6G is envisioned to address the unresolved challenges of 5G while inheriting its novel enablers and enhancing them [1]. Network slicing, a key innovation introduced in 5G, is expected to evolve in 6G to accommodate multiple tenants with diverse requirements. While network slicing supports various service types for a broad range of users, creating and managing these slices remains a challenging and complex task. This complexity arises from the conflicting nature of supported slice types and limitations in network resources. Specifically in the Radio Access Network (RAN) entity clusters and mobile users [2].

Network slicing in 6G must operate across different resource domains, including communication and computational resources. Similar to the concept applied in 5G, each slice must be customized to meet the tenant's requirements and isolated from other slices. 6G is expected to introduce intelligence in various performance and management areas, including RAN slicing and resource allocation procedures [3]. In addition, 6G must support a vast number of users with diverse service types and applications. Meeting the demands of these use cases and services will add significant complexity, layers, and domains to the existing network technologies [4].

While network slicing in 5G literature often involves heuristic and genetic algorithms [5]–[8], Reinforcement Learning (RL) has emerged as a key enabler for network slicing in beyond 5G and 6G networks. RL and Deep RL (DRL)-based techniques enhance network management and resource allocation in the RAN sector [9]. The highly dynamic nature of 6G networks, particularly in the RAN domain, can significantly benefit from RL-based intelligent decision-making processes [3]. RL is considered a promising approach for dynamic resource management in complex environments like 6G networks, mainly because it is a model-free and algorithm-agnostic approach [10].

Meanwhile, Open RAN (O-RAN) enables RL techniques to make real-time decisions and operational adjustments in the network. O-RAN's standardized interfaces and entities provide flexible and intelligent control, making it a promising paradigm for 6G networks' RAN architecture [10]–[12].

To realize the vision of 6G as an intelligent network, comprehensive advancements in both architecture and network management are essential. Initial intelligence refers to a network entity's ability to adjust and reconfigure itself from predefined options in an intelligent, semi-autonomous manner. In contrast, Intelligent Radio (IR) represents a broader and deeper concept. Given that 6G is envisioned as a highly complex, heterogeneous network with diverse system requirements and an exponentially growing number of connected devices, intelligence management across all network components must be self-adaptive [13]. Consequently, this research proposes

N. Ghafouri is with Iquadrat Informatica, Barcelona, Spain (e-mail: n.ghafoori@iquadrat.com@iquadrat.com).

J. S. Vardakas is with Iquadrat Informatica S. L. Barcelona, Spain, and Dept. of Informatics, University of Western Macedonia, Greece, (email:ivardakas@uowm.gr)

A. Ksentini is with Eurecom, France, (email:adlen.ksentini@eurecom.fr)

ISI/ATHINA, Greece, CEID, University of Patras, Greece, Iquadrat Informatica S. L., Barcelona, Spain (e-mail: cveri@upatras.gr)

an IR framework where intelligent algorithms dynamically configure themselves based on the hardware's available capacities. To achieve this, we introduce a MORL-based dynamic slice configuration approach for the O-RAN-enabled Cell-Free massive Multiple-Input-Multiple-Ouput (CF mMIMO) architecture.

Our approach integrates into the network slice management block as a lower layer of the service management and orchestration block, which oversees the entire network. This implies that while higher-level processes manage user-to-network service type mapping and ensure synchronization, the proposed work focuses on optimal, dynamic slice configuration and reconfiguration—an open challenge in the field. Current state-of-the-art methods [11] deploy single agents to assign resource blocks for each service type, requiring twice the number of agents as service types. This is because, in addition to decision-making agents, separate agents are trained for each service type to allocate resource blocks, resulting in multiple neural networks [11]. In contrast, our approach leverages MORL to handle multiple service types with a single agent, improving consistency while reducing the number of agents, neural networks, and training processes. By effectively managing trade-offs between service types, the MORL agent offers a scalable and efficient solution compared to multiple single agents.

Additionally, some service types, such as URLLC, cannot rely solely on resource block allocation due to the impact of link-related parameters like distance. Hence, our approach also considers communication links in the assignment scheme. Specifically, it assigns an end-to-end path comprising Open Radio Units (O-RUs), Open Distributed Units (O-DUs), and interfaces between them, from the user to the Open Centralized Unit (O-CU) while decreasing the complexity. Treating service types as multiple contradictory objectives to optimize simultaneously, the MORL agent is trained to address all objectives and adapt to any specific service type.

In a nutshell, this research work develops a novel end-to-end intelligent network slicing approach from the user to the O-CU realized through the integration of resource block allocation, O-RU clustering, and O-DU assignment. Our proposed approach surpasses the state-of-the-art by uniquely integrating all of the following features:

- slices are configured and reconfigured dynamically for users' requests,
- this approach is capable of providing diverse service types, including coexisting FeMMB and URLLC,
- it is consistent and synchronized as the result of deploying one single multi-objective policy,
- our approach is computationally efficient, as only one single agent and one training phase are applied,
- it improves Energy Efficiency (EE) by activating only the required network nodes for each user,
- and finally, it is robust, sample efficient, scalable, and practical for real-world scenarios, as demonstrated by our extensive evaluation process.

The rest of the article is organized as follows: Section II reviews the related work. Section III provides a detailed overview of the system model and service types. Section IV presents the proposed MORL-based scheme, while Section V evaluates the numerical results. Finally, Section VI concludes the paper.

II. Related Works

In our previous research work [11], we proposed a network-slicing-based resource-management and orchestration approach designed explicitly for O-RAN-enabled 6G networks. This approach employs a centralized decision-making level that utilizes Multi-Agent RL (MARL) to map service types to user requests. The agents at this level maintain network consistency and aim to maximize capacity through optimal assignments, enabling the network to serve more users. The second level consists of single RL agents trained to allocate bandwidth resource blocks to users and realize the required service types. However, allocating bandwidth resource blocks is insufficient for the realization of all types of services. This is because factors such as RAN communication links and entities, grouping and clustering strategies, and other RAN topology specifications significantly affect the QoS required for services.

In a similar effort, the authors of [14] propose a dual-level approach for slicing communication and computation resources for Ultra-Reliable and Low-Latency Communications (URLLC) users in an O-RAN-based system model. The proposed approach is also applied in two levels of decision-making and deploys a Double Deep Q-Network (DDQN) algorithm; however, only one service type is considered. In [15], a combination of Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) is applied to the Unicauca IP Flow Version 2 dataset. This approach is trained to classify five service types: super enhanced Mobile BroadBand (super-eMBB),massive Machine-Type Communications (massive-MTC), super URLLC, super-precision, and super-immersive. However, training on a static dataset lacks the flexibility required for the dynamic nature of 6G. In [16], the authors integrate Joint Communication and Sensing (JCAS) with network slicing by formulating a Nonlinear Integer Programming (NLIP) problem for joint transmission and resource allocation in sensing and communication applications. Using a DQN algorithm, they minimize latency but focus exclusively on a specific application group.

The network-management approach presented in [10] considers that the mobile network operator can change the weights of the RL reward function so that the agent adopts each service type and fulfills the agreed QoS for that slice type. This work uses Transfer Learning (TL) to address the agent's convergence challenges after the modifications. However, hosting multiple conflicting services that share resources in an isolated manner introduces multiple conflicting objectives for the agent. Motivated by this, in this work, we adopt Multi-Objective RL (MORL) algorithms to train the agent across all considered service types, enhancing the likelihood of successful resource assignment. This approach minimizes training complexity while enabling the learned policy to adapt to diverse preferences, allowing the agent to handle multiple service types dynamically.

MORL-based approaches have been previously used in telecommunications for various applications. For instance, research work [17] employs MORL to address the challenge of offloading an application consisting of dependent tasks in Multi-access Edge Computing (MEC). In [18], a MORL framework for load balancing, and benefits from meta-RL to learn a general policy that can adapt to new trade-offs between objectives. Finally, [19] applies MORL with two objectives for resource allocation and energy efficiency in Cloud-RANs (C-RANs).

In this work, we use a MORL agent to assign communication links, resource blocks, and network entities to each user requesting a specific service type. This service-type realization creates slices and clusters of network entities for each request. The next section introduces the considered system model and service types.

III. System Model and Service Types

This section begins by presenting the system model, illustrated in Fig. 1. The proposed model leverages O-RAN and CF mMIMO to enable the intelligent approach. The Near Real-Time Intelligent Controller (Near-RT RIC) in the O-RAN architecture provides an appropriate control loop (ranging from $10ms$ to $1s$) for the agent to deliver a selected service type [20]. The MORL agent is integrated into Near-RT RIC within the xApp framework to enhance its functionality. Deploying CF mMIMO allows for potential communication links between all the O-RAN entities (i.e., O-CU, O-DU, and O-RU).

This work focuses on configuring slices for pre-selected service types. Below, we present the considered service types and the key performance metrics associated with each:

1) FeMBB: Further-Enhanced Mobile BroadBand.
   FeMBB supports use cases such as video streaming, virtual/augmented reality, and holographic verticals. Key Performance Indicators (KPIs) for this service include a high data rate (exceeding 1 Tb/s at the system level) and Spectral Efficiency (SE).
2) umMTC: ultra-massive Machine-Type Communications.
   umMTC underpins the Internet of Things (IoT), the Internet of Everything (IoE), and smart cities. Key KPIs include ultra-low latency ($10-100$ μs) and high Energy Efficiency (EE).
3) ERLLC: Extremely Reliable and Low-Latency Communications.
   ERLLC is critical for applications such as fully automated driving and industrial Internet. In addition to ultra-low latency, high mobility support (greater than $1,000$ km/h) is a crucial KPI for this service.
4) LDHMC: Long-Distance and High-Mobility Communications.
   LDHMC enables 6G's deep-sea and space connectivity ambitions. Mobility plays a key role in this service type.
5) ELPC: Extremely Low-Power Communications.
   ELPC supports applications in e-health by connecting nanodevices, nanosensors, and nanorobots. KPIs for this service include EE and connectivity density.

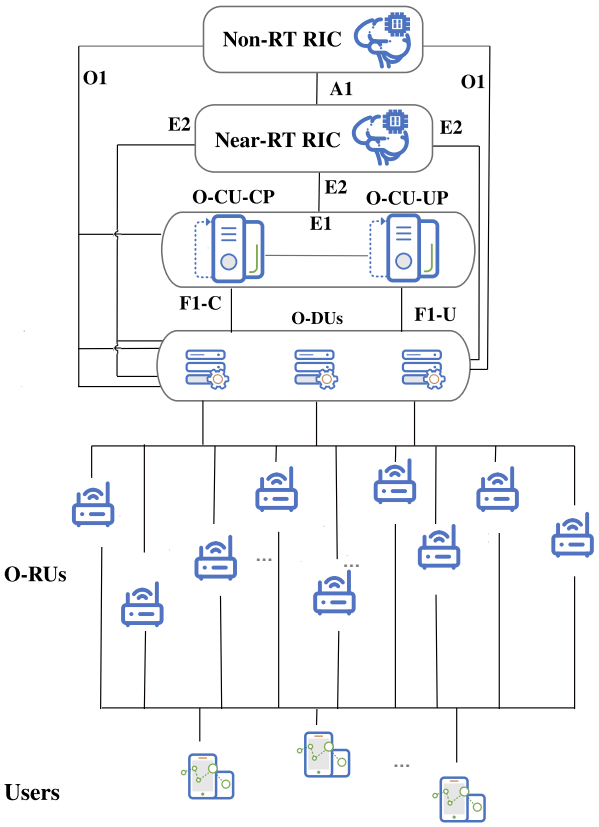

Fig. 1. The system model: O-RAN and CF mMIMO

To enable these service types, we first analyze the network KPIs that are critical for their realization. This analysis provides valuable insights into the envisioned KPI values and their implications for 6G. The following section reviews the KPIs considered in this work and their characteristics in 6G.

A. Mobility

Mobility refers to the continuous network coverage for users in motion while maintaining predefined QoS. In 5G, the maximum user speed was considered to be 500 km/h [21], whereas in 6G, it is envisioned to reach 1000 km/h [11]. Although physical layer technologies primarily influence mobility performance, certain architectural considerations can also contribute to its improvement. One such consideration is the decoupling of the User Plane (U-plane) and Control Plane (C-plane). This separation enhances coverage for mobile users by reducing the overhead of handover signaling and ensuring a more stable connection with base stations [21]. Additionally, balancing high system capacity and transmission reliability is crucial for mobile users. While higher frequency bands and larger bandwidth increase system transmission capacity, they also suffer from greater path loss, necessitating more handovers. By utilizing both C-plane and U-plane, network resources can be split between low-frequency bands for reliable C-plane signaling and high-frequency bands for high-capacity U-plane data transmission.

Another architectural enhancement involves connecting multiple remote RUs to each baseband unit, which reduces the handover time and minimizes the number of failed handovers [21]. Furthermore, fast-moving users need access to more than two cells. In this regard, CF mMIMO provides significant advantages by eliminating cell boundaries. As a result, the serving access point remains unchanged, effectively resolving handover-related issues [22].

B. Area Traffic Capacity and Connectivity Density

Area traffic capacity and connectivity density reflect the varying population of users served by the network across different covered areas. In real scenarios, user distribution and traffic demand are often non-uniform. Consequently, focusing solely on the network's overall performance may lead to sub-optimal resource utilization. For instance, some areas with low connectivity density may have surplus transmission capacity, while areas with high connectivity density may experience insufficient resources [23]. To achieve optimal performance for these two metrics, the network must provide services tailored to the specific demand of each area. This demand-aware approach also enhances the network's EE. While area traffic capacity and connectivity density are closely related, they represent distinct concepts. Area traffic capacity measures the total traffic throughput served per geographic area (bps/m^2), while connectivity density represents the total number of connected and/or accessible devices per unit area ($Device/km^2$) [24]. The envisioned values for each KPI are $1\ Gps/m^2$ and $107\ d/km^2$, respectively, in 6G [11].

C. Spectral and Energy Efficiency

Peak SE is the maximum data-rate under ideal conditions normalized by channel bandwidth (in $bit/s/Hz$), where the maximum data-rate is the received data bits assuming error-free conditions, assuming that all available radio resources in the corresponding link direction are fully utilized by a single mobile station [24]. This KPI is commonly used to evaluate the performance of mobile broadband service types, alongside the data-rate. Network EE measures the network's ability to minimize energy consumption in relation to the provided traffic capacity. This KPI can be measured in both cases of transmitting data and in idle periods (sleep mode). In the case of data transmission, EE is linked to SE, while in sleep mode, it is evaluated by the sleeping duration or ratio. According to [25], network SE increases monotonically with traffic load, but EE depends on the power consumption of the Base Station (BS) in both sleep and active mode.

D. Latency and Reliability

Latency, the key KPI for ERLLC, has been envisioned to range between 10 and $100\ \ s$ [11]. A deeper examination of this KPI reveals that latency must be broken down into smaller components, including delays within various entities and processes involved in data transmission. U-plane latency is an application-layer KPI, whereas computation-related latency—comprising execution latency, service latency, and processing latency—lacks a standardized calculation method [24]. In addition to minimizing latency, ERLLC applications require reliable and available network access, as reducing latency directly contributes to enhanced reliability. While radio link quality metrics and low-level multiplexing techniques significantly influence network reliability, efficient high-level resource allocation algorithms can further improve it considerably [26].

E. Data-Rate (system level and user level)

Tightly related to SE and throughput, data-rate is another important KPI for mobile broadband services. While peak data-rate refers to the maximum achievable data-rate under ideal conditions, data-rate is also measured with another metric called user-experienced data-rate. This KPI is calculated as a 5 percent point of the Cumulative Distribution Function (CDF) of the user throughput; the latter is defined as the number of correctly received bits during active time [24]. Meeting the required system- and user-level data rates for FeMBB users is challenging, as a user's channel condition significantly impacts this KPI. This challenge becomes even more complex when the network simultaneously supports both FeMBB and ERLLC services. In such cases, the system must maximize the data rate for some users while ensuring that latency constraints are met for others.

F. KPIs in the System Model

While improving network KPIs often relies on physical layer techniques, architectural choices and high-level management play a critical role in establishing the foundation for these enhancements. In this work, we combine O-RAN and CF mMIMO as our system model, inspired by their complementary benefits. The CF aspect of the RAN eliminates cell boundaries, resolving handover issues, while O-RAN enhances the architecture through its open and programmable C-plane and U-plane at the O-CU. This makes the system inherently mobility-friendly. Additionally, splitting CPU responsibilities across multiple layers—including Non-RT RIC, Near-RT RIC, O-CU, O-DU, and O-RU—coupled with interconnections at every level, ensures scalability and practicality [12].

Our user-centered approach in the CF part [12] initiates slice assignment based on a user request, ensuring services are tailored to user demands. This approach provides a slice of the resource blocks, communication links, and a cluster of O-RUs and O-DUs to a user, customized to the service type requested. By integrating insights from our previous work [11], a user request triggers high-level centralized decision-making, slice selection, and subsequent slice configuration and realization. This ensures an optimal ratio of users to provided services, maximizing network capacity utilization. Essentially, on-demand service capability is achieved through user-based network capacity assignment, inherently considering area traffic capacity and connectivity density in system capability calculations [23].

While deploying a request-oriented approach for slice realization selects the best path and links for each user through the RAN, which results in an optimal assignment; it also improves

the EE of the system enormously. This is because the O-DUs, O-RUs, and generally all the entities not engaged in the ongoing transmissions can stay in sleep mode. As explained in [12], in our considered system model, the physical links exist between all the O-RUs and O-DUs (since we deploy CF RAN), but only the selected ones activate for each request.

To address latency, defined in computing as the time between a request and the algorithm's response [24], our slice realization approach is embedded in the Near-RT RIC. This provides a faster control loop compared to the Non-RT RIC, enabling end-to-end RAN orchestration from Non-RT RIC to the user. By replacing the current MORL-based approach in the lower layer described in [11], we achieve a more responsive system.

Reliability, traditionally measured at the packet transmission level, is defined in this work as the percentage of user requests successfully fulfilled by the network [24]. While prior research focused primarily on the coexistence of eMBB and URLLC in 6G and mMIMO-based RAN [27], [28], our intelligent approach incorporates all predefined service types during training. This enables the provision of end-to-end slices tailored to each type after training. Managing five service types posed challenges in our previous work [11], motivating a deeper study to improve consistency and reduce complexity.

The next section provides a detailed explanation of the newly proposed approach.

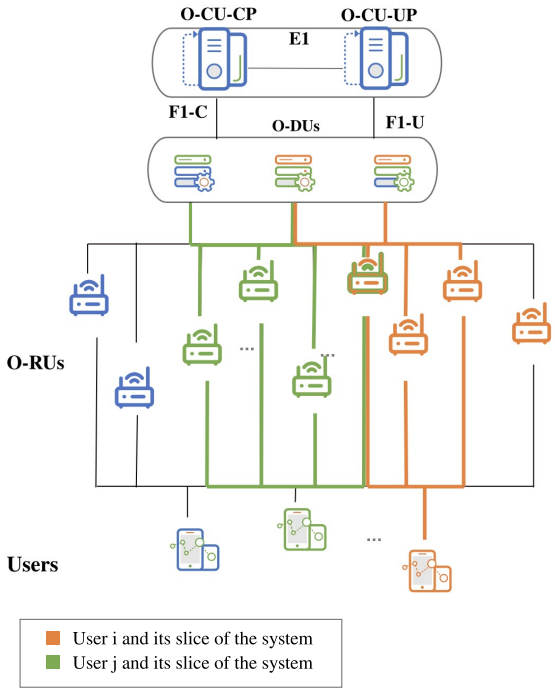

Fig. 2. A user-centric network slice in O-RAN

IV. The Proposed MORL-Based Scheme

A. The General Idea

Unlike traditional connection-oriented communication systems, intelligence and sensing are integral components of 6G networks. These intelligent network entities enable task-oriented communications driven by user requests. Here, a "user" encompasses a wide range of entities, including vehicles, devices, and individuals. The network must provide customized, user-centered services [29]. A user-centric network, leveraging widely available edge resources, achieves scalability and robustness while remaining adaptable to individual user configurations.

While dynamic service type mapping has been studied in the literature [11], this work considers all effective parameters for dynamically delivering the predefined QoS for each service type. A single agent evaluates effective metrics and KPIs for each service type and executes a sequence of actions tailored to that slice type. Consequently, the agent manages resource allocation, O-RU clustering, and O-DU assignment, culminating in a user-centric, dynamic slice configuration. Fig. 2 illustrates the user-centric slice, where clusters of O-RUs and O-DUs service users by activating appropriate connection links. Each color in this figure represents a potential cluster of O-RUs and O-DUs assigned to service a user. As shown in Fig. 2, an O-RU or O-DU may serve multiple users. However, when treating each o-RAN entity as a set of resource blocks, no single block is shared across users, ensuring slice isolation.

This approach eliminates the need for separate agents for each service type, reducing the number of agents required to match the number of tenants the network serves. To achieve this, a MORL agent is implemented instead of multiple single RL agents. This MORL agent learns policies over competing objectives without prior knowledge of their relative importance during training. Once trained, it can execute optimal policies for any given objective [30].

In MORL, the Markov Decision Process (MDP) is extended to the Multi-Objective Markov Decision Process (MOMDP). This includes a vector reward function (instead of one single value), a space of preferences (objectives or service types), and preference functions. If the preference space contains only one value, the MOMDP reduces to a standard MDP. Furthermore, this work leverages an Envelope Multi-Objective Q-learning (Envelope MOQ-learning) algorithm [30]. The proposed Envelope MOQ-learning algorithm is designed to learn policies across multiple preferences simultaneously. Unlike scalarized Q-learning, which simplifies rewards into single values, the Envelope MOQ-learning algorithm uses vectorized value functions. It updates parameters based on the convex envelope of the solution frontier, offering a more robust optimization. The extended Bellman operator handles multi-objective value functions, with the update defined as:

$$(TQ)(s,a,\omega) := r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[(HQ)(s',\omega) \right] \quad (1)$$

where H is an optimality filter for the MOQ function that selects the multi-objective value corresponding to the supremum. In the extended Bellman optimality operator, ω is the m-dimensional preference vector. The return of arg_Q in H depends on which ω is chosen for scalarization.

The learning algorithm minimizes a combined loss function consisting of two parts of L^A and L^B, which ensure proximity to expected rewards and guide the solution toward greater

utility, respectively. The final loss function is:

$$L(\theta) = (1 - \lambda)L^A(\theta) + \lambda L^B(\theta) \quad (2)$$

where λ gradually increases from 0 to 1, shifting focus between loss terms through Homotopy optimization [31]. Additionally, the algorithm uses Hindsight Experience Replay [32] to improve sample efficiency and policy gradient methods to adapt the agent's policy when preferences are unknown.

To implement the Envelope MOQ-learning algorithm within our system model, we represent the system as a binary tree, which acts as the interactive environment for the algorithm. This structured representation allows for systematic interactions between the algorithm and the model, enabling efficient learning and decision-making processes. As outlined in Algorithm 1, the agent is limited to two possible actions at each step. In detail, our MORL agent models O-DUs and O-RUs as nodes, while the interfaces between them are represented as edges (Algorithm 1: Lines 2, 3). The system's binary tree has a depth equal to the total number of O-DUs and O-RUs (Algorithm 1: Line 4). At each level of the tree, selecting the right subtree indicates that the node at that level is included in the chosen path, whereas selecting the left subtree signifies that the corresponding O-DU or O-RU is excluded from the path under consideration (Algorithm 1: Lines 5–10). Consequently, during each time step within an episode, the agent decides whether to include a particular node in the path. Each final node in the binary tree contains an array of reward values corresponding to the parameters relevant to each service type or objective. This array includes one value for each service type, constructed to account for the important metrics specific to that type. When a user requests a particular service type, the algorithm assigns a path spanning all levels of the tree that represents the most appropriate slice for the request in the current network instance (Algorithm 1: Line 11). While Algorithm 1 provides an overview of the slice configuration process, detailed explanations are presented in the following section.

Algorithm 1 The selection of the network slice

1 **Input =** Objective (mapped request)
2 **Nodes =** DUs , RUs
3 **Edges =** User-RU links , RU-DU links
4 **Model =** A Binary tree of Nodes and Edges
5 **Based on the selected objective:**
6 For (the depth of the tree):
7 If (Node is selected):
8 Include the Node and the Edge ending to the selected Node to the slice path
9 Else :
10 Go to the next level
11 **Output =** Slice (selected path in the tree)

B. Implementation Technicalities

As previously described, the system architecture in this work incorporates CF mMIMO, enabling each user to potentially connect to all O-RUs. Furthermore, there is a physical connection between all O-RUs and O-DUs. However, this does not imply that every user is connected to all O-RUs or that all O-RUs are connected to all O-DUs simultaneously. Instead, when the agent determines the optimal path through the O-RUs and O-DUs, only the relevant links are activated for that specific user. This approach ensures the system remains energy-efficient by minimizing unnecessary active connections.

Fig. 3 provides an alternative representation of the system model as a binary tree. The depth of the tree corresponds to the total number of O-RUs and O-DUs. Each final node in the tree represents a binary number ranging from $0, 1, \ldots, 2^n$ with n as the tree's depth. he binary representation consists of n n digits, with each digit corresponding to an entity. The higher-valued positions in the binary sequence (starting from the left) represent O-DUs, while the lower-valued positions correspond to O-RUs. A value of 1 in a position indicates that the corresponding O-DU or O-RU is included in the selected path, while a 0 means it is excluded. At the 2^n th level of the binary tree, each node is associated with a reward vector, where each element reflects the path's value for a particular parameter or objective. Each service type corresponds to one objective, represented by specific parameters that quantify the value of a path for that service type. Thus, as shown in Fig. 3, the tree's depth n defines 2^n reward vectors, each consisting of m elements, where m is the total number of service types.

To train the MORL agent, we implement the Envelope MOQ-learning algorithm proposed in [30]. The novelty of the algorithm is based on the incorporation of the concept of vectorized value functions, enabling the agent to optimize across multiple objectives simultaneously. Instead of focusing on a single preference during value updates, the algorithm leverages the convex envelope of the solution frontier to update parameters. By extending the Bellman equation, the agent learns a single parametric representation of the optimal policies for all preferences. During the training phase, the agent operates without prior knowledge of the relative importance of different service types, while it learns optimal policies across the entire space of service types. After training, the agent's policy dynamically adapts to any chosen service type based on user demands. The next section delves into the details of the implementation and simulation processes, showcasing the results of this approach.

V. Simulations and Numerical Evaluations

In what follows, we implement the Envelope MOQ-learning algorithm and a self-designed environment. Simulations use Python 3.9, and the environment is developed by deploying the OpenAI gym library.

A. Simulation and training details

As discussed in Section IV.B, the system model in our architecture is represented as a binary tree with depths of eight, as it considers three O-DUs and five O-RUs. The O-RUs may connect to a random number of users ranging from 0 to 3, while a similar arrangement is applied to the O-DUs while serving assigned to the user request. This mapping is performed at a higher level by the high-level decision-making

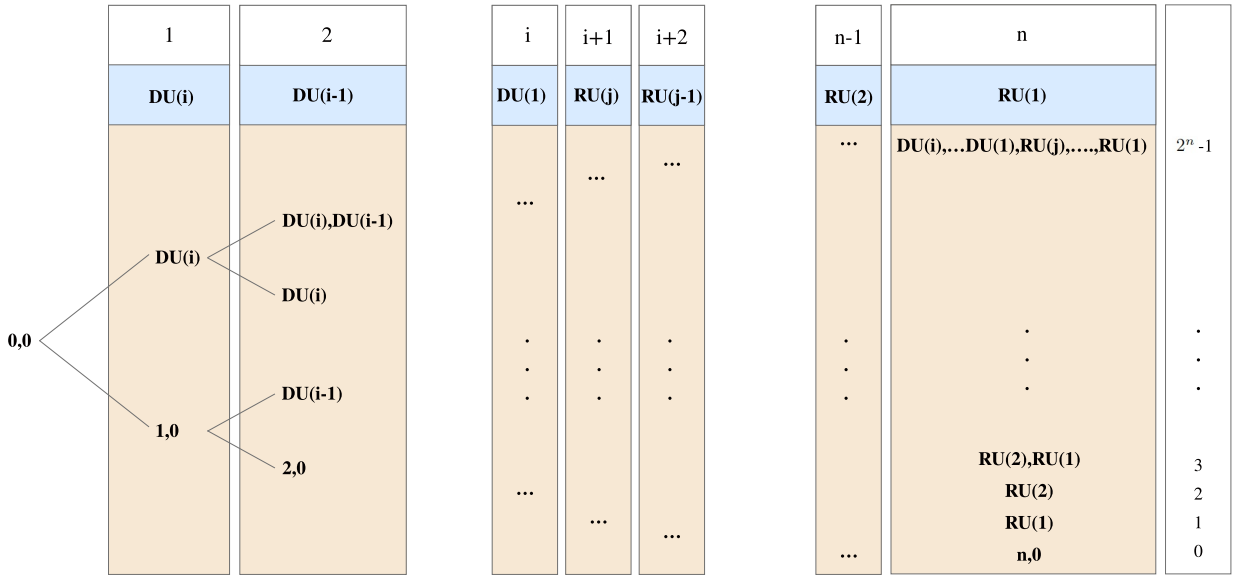

Fig. 3. Binary tree representation of the system model in Low-part

MARL agents [11]. The episode concludes once the agent assigns a path through the O-RUs and O-DUs to the user corresponding to the requested service type. Consequently, the binary tree has 255 leaves at its $8th$ level, representing all possible paths. Each state in the simulation consists of a double denoting potential positions in the tree, while the possible actions are moving left or right within the binary tree. The reward at each leaf (or path) is a tuple of four elements.

The simulation considers the following service types: FeMBB, umMTC, ERLLC, LDHMC, and ELPC. Although LDHMC is included in the broader system model, it is excluded from this level of management since its primary defining KPI is mobility [11]. This exclusion allows the simulation to focus on the remaining four service types, with one element in the reward array assigned to each type.

The reward array elements are defined as follows:

1) *Available resource blocks*: Best suited for FeMBB, this metric reflects the high data-rate and high spectral efficiency (SE) required by this service type.
2) *Channel quality metric*: Assigned to umMTC, this metric combines user distance from an O-RU and the number of users already served by the O-RU. It aligns with the latency and connectivity density KPIs of this service type.
3) *Distance metric*: Used for ERLLC, which focuses on real-time, emergency-related use cases where minimizing distance improves reliability and responsiveness.
4) *System balance*: Linked to ELPC, this metric emphasizes distributed management. Maintaining the balance between in-use and available resources ensures that all parts of the network can efficiently serve new users. This approach improves connectivity density, area traffic capacity, and overall network availability.

The available resources of a path (the first element of the leaf reward vector) are calculated as the sum of available bandwidth in the engaged O-DUs, and the data-rate that engaged O-RUs can provide [12]. To further refine this metric, the available bandwidth of an O-DU is weighted by a coefficient inversely proportional to its distance from the O-RUs. This weighting helps the agent avoid selecting O-DUs and O-RUs that are further apart, even if they offer higher bandwidth, ensuring path uniqueness for the agent. While system bandwidth is shared equally among O-DUs, the available bandwidth for each O-DU is affected by the number of users it is already serving. The network balance metric represents the ratio of available resources from all engaged nodes (O-DUs and O-RUs) to the total available resources across all nodes. Other metrics, such as channel quality and distance, follow the definitions established in [12]. Thus, the state space, action space, and reward space in the system are defined as follows:

$$state\ space = \begin{bmatrix} \{0,0\}, & \cdots & ,\{0,2^0-1\} \\ \vdots & \ddots & \vdots \\ \{8,0\}, & \cdots & ,\{8,2^8-1\} \end{bmatrix} \tag{3}$$

$$action\ space = \{0,1\} \tag{4}$$

$$\begin{aligned} reward\ space = \{&available\ resources,\\ &channel\ quality\ metric,\\ &distance,\\ &network\ balance\} \end{aligned} \tag{5}$$

The binary tree in our system model represents a combination of the state space and 255 reward arrays at the final level. While the state space (tree structure) remains fixed across all episodes, the last level of the tree is regenerated at the start of each episode by resetting the environment.

Table I shows the simulation parameters with fixed values, while the unfixed parameters, which vary dynamically in each episode, include:

1) The number of users already being served by the O-RUs.
2) The number of O-RUs actively connected to the O-DUs.

TABLE I
SIMULATION FIXED PARAMETERS

Parameter	Name	Value
P_p	Pilot power	$200mW$
P_t	Transmission powr	$200mW$
α	Path loss exponent	$3.76dB$
$(\sigma)^2$	Variance of AWGN	$-94dBM$
d	ORU-UE distance	$Random(1, 10)$
B_{DU}	BU bandwidth	$18000MHz$
B_{RU}	RU bandwidth	$6000MHz$

TABLE II
LOGICALLY DELETED LEAVES

Path number	Binary representation	Network nodes
0	$[0, 0, 0, 0, 0, 0, 0, 0]$	$-$
1	$[0, 0, 0, 0, 0, 0, 0, 1]$	RU_1
...	...	No DUs included
31	$[0, 0, 0, 1, 1, 1, 1, 1]$	$RU_1, ..., RU_5$
32	$[0, 0, 1, 0, 0, 0, 0, 0]$	DU_1
64	$[0, 1, 0, 0, 0, 0, 0, 0]$	DU_2
96	$[0, 1, 1, 0, 0, 0, 0, 0]$	DU_1, DU_2
128	$[1, 0, 0, 0, 0, 0, 0, 0]$	DU_3
160	$[1, 0, 1, 0, 0, 0, 0, 0]$	DU_3, DU_1
192	$[1, 1, 0, 0, 0, 0, 0, 0]$	DU_3, DU_2
224	$[1, 1, 1, 0, 0, 0, 0, 0]$	DU_3, DU_2, DU_1

3) The distance between the existing users and the new user initiating the request.

These parameters are randomly assigned in every episode, ensuring that the metrics and the environment dynamically evolve for each simulation instance. To stabilize the training process and improve numerical computations, all metrics are normalized to fit within the range $[0, 1]$. Normalization serves two primary purposes: first, it enhances the stability of training and numerical operations. Second, it ensures that the agent can effectively identify the path that maximizes utility for a given service type.

As a result, all the leaf nodes being in a convex coverage set is a prerequisite for the agent. The convex coverage set of the Pareto frontier contains all possible solutions that maximize the cumulative utility for all possible preferences (four service types) [30]. However, in the real scenario of our network, 39 paths out of 255 paths are not practical and, thus, not optimal. Table II presents these paths non-functional paths. Thus, we assign a negative number to the reward array of these paths to logically eliminate the path. The final reward space is then defined as follows:

$$reward\ space = \{ \begin{cases} -1 \text{ or} \\ [0, 1], \end{cases} , \begin{cases} -1\text{or} \\ [0, 1], \end{cases} , \begin{cases} -1\text{or} \\ [0, 1], \end{cases} , \begin{cases} -1\text{or} \\ [0, 1], \end{cases} \} \quad (6)$$

The Q-network for the Enveloped Q-learning [30] utilizes double Q-learning [33]. Although it can be with similar off-policy algorithms, Q-learning with a target network and prioritized experience reply is the best choice to ensure compatibility with our previously proposed high-level decision-making approach in [11]. The primary difference between a DQN and a multi-objective QN lies in their inputs and outputs during implementation. In MORL, the state representation also includes the parameters of a linear preference function. The output layer dimensions are determined by the product of the action space size and the number of objectives. The implemented Multi-Objective Q-networks (MQNs) are implemented as four fully connected layers consisting of 16, 32, 64, and 32 multiplied by the sum of space and action sizes. After extensive trials with a random combination of hyperparameters, as suggested in [30], the training process uses the Adam optimizer along with the values 0.95, $1e-3$, and 0.5 for $gamma$, $learning\ rate$, and $epsilon$, respectively. The training loss and reward plots are presented in Fig. 4 and Fig. 5.

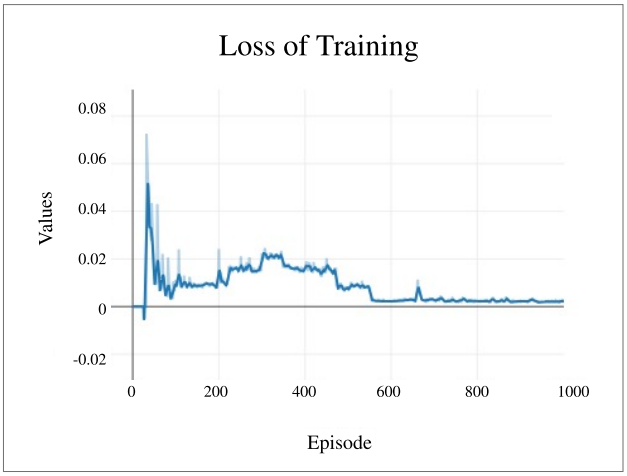

Fig. 4. Loses of training for 1000 episodes

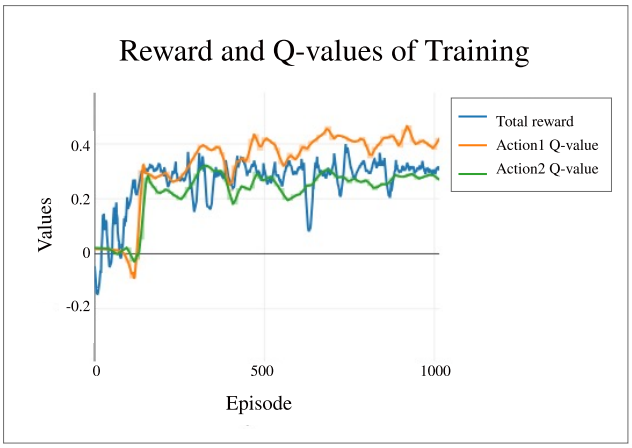

Fig. 5. Normalized Q-values and Reward plots of the training

As Fig.4 shows, the agent was trained for 1000 episodes, converging after approximately 700 episodes. Fig.5 presents the Q-values for each possible action (shown in green and orange curves). The blue plot is the total reward of the episodes. After the training phase, we test the model. As mentioned before, in our dynamic environment, each episode shows a different representation of the environment. The test was conducted on a tree representation of a network with the numerical values shown in Table III and Table IV. Also, Table V shows the chosen path for every service type:

TABLE III
RUs´DATA

RUs	Distance of user from RU(m)	Number of already serving users	Distances of already serving users (m)	Data-rate RU can provide for user (Mbps)
1	5	2	$\{8, 1, 9\}$	$7.69841741e+03$
2	2	3	$\{2, 3, 4\}$	$2.71952530e+02$
3	4	3	$\{2, 3, 5\}$	$7.73846452e+03$
4	1	3	$\{5, 4, 2\}$	$9.85962201e+02$
5	4	0	–	$2.82952730e+04$

TABLE IV
DUs´DATA

DUs	Number of RUs already serving	Remaining Bandwidth (MHz)	Distance to RUs 1,2,3,4,5 (m)
1	0	18000	$\{2000, 2700, 2900, 2650, 2950\}$
2	2	14000	$\{2500, 2300, 2400, 2800, 2800\}$
3	1	16000	$\{3000, 3100, 2600, 2750, 2800\}$

B. Analysis of the numerical results

We follow the same approach as in [12] to estimate the data-rate, enabling an approximation of Signal-to-Interference-plus-Noise-Ratio ($SINR$) that is independent of the type of precoding scheme. We assume the $SINR$ for transmissions between O-DUs and O-RUs to be zero, making the provided data rate approximately equal to the available bandwidth.

Data-rate and SE: As shown in Fig. 6, the data-rate results show that the tenant who was assigned the FeMBB service type is able to access 32000 MHz bandwidth of O-DUs, and the three O-RUs can provide approximately 36 $Gbps$ of data-rate; this fact meets the goal of the FeMBB service. The selected O-RUs are serving 6 other users, which results in a total of 12000 MHz of O-RUs' bandwidth in-use for them in our simulation. As a result, the SE of the considered slice is approximately 6 $bit/s/Hz$.

Latency: For the ERLLC, we first monitored the computing latency, defined as the time the agent required to choose the path. The observed execution time was 0.0317 s.

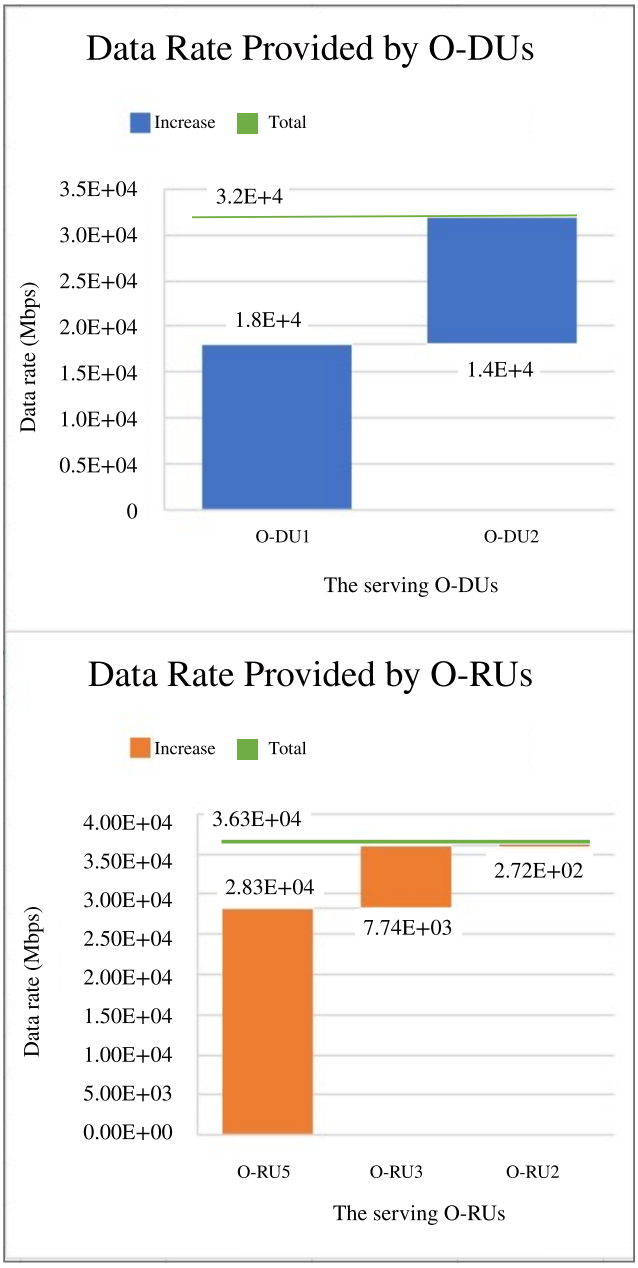

Fig. 6. The user-experienced data-rate in service type FeMMB

Next, we present the Pareto frontier of the model, considering FeMMB and ERLLC as the two conflicting service types in the system.

As shown in Fig.7, The agent´s policy prediction achieves more than 91 % precise accuracy compared to the real data and converges to the true Pareto front with only an 8.8 % error.

Reliability, connectivity density, and area traffic capacity: Reliability determines the overall experience of the network in terms of accessibility and performance. To validate the reliability of our approach, we monitored the execution time for 10 different test runs by using different tree models. As shown in Fig. 8, the algorithm's latency ranged between 0.0175 and 0.0360 s, with an average execution latency of 0.0226 s. Fig. 9 depicts the data-rate provided by the O-RUs

TABLE V
TEST RESULTS

Service type	Metric	Chosen path	Binary representation	Network nodes
FeMBB	Available resources	$[7, 118]$	$[0, 1, 1, 1, 0, 1, 1, 0]$	DU2,DU1,RU5, RU3,RU2
umMTC	Quality mertric	$[7, 127]$	[0,1,1,1,1,1,1,1]	DU2,DU1,RU5, RU4,RU3,RU2,RU1
ERLLC	Distance	$[7, 126]$	$[0, 1, 1, 1, 1, 1, 1, 0]$	DU2,DU1,RU5, RU4,RU3,RU2
ELPC	Network balance	$[7, 124]$	$[0, 1, 1, 1, 1, 1, 0, 0]$	DU2,DU1,RU5, RU4,RU3

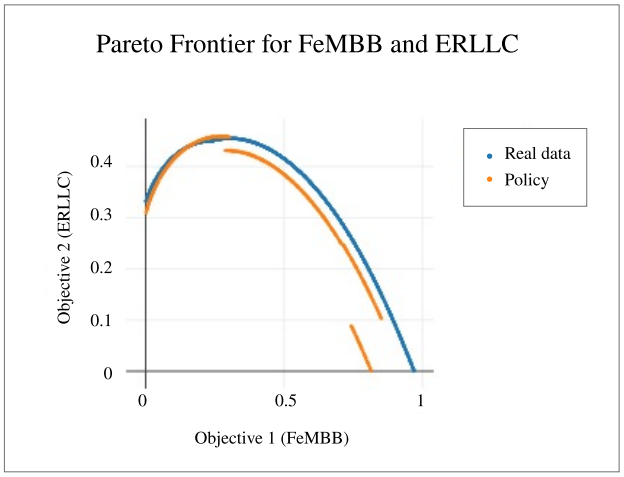

Fig. 7. Pareto frontier of the policy for FemBB and ERLLC

in 10 different network instances. As Fig. 9 demonstrates, all the users experienced a data-rate higher than 1 $Gbps$. It is worth mentioning that the actual data-rate will likely be lower when accounting for a more realistic $SINR$.

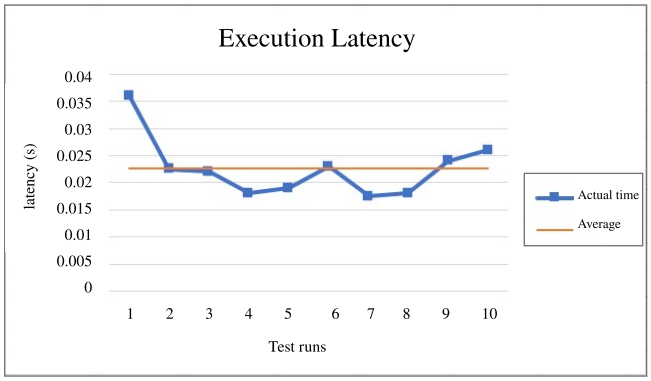

Fig. 8. The Execution latency of our agent in ERLLC service type of 10 test runs.

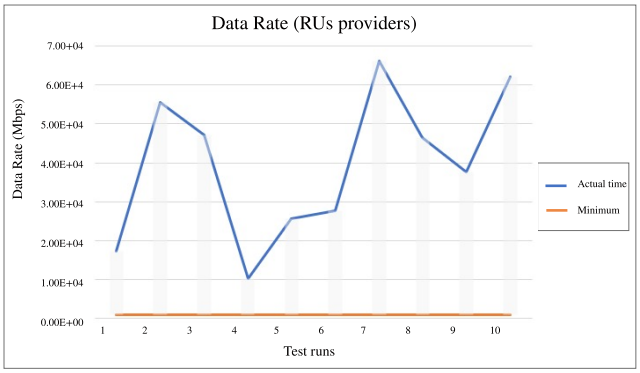

Fig. 9. The user-experienced data rate in service type FeMMB.

All 10 tests, each for 4 preferences, were successfully assigned by the agent. Thus, the slice configuration approach can be considered reliable. However, as these results are based on episodic simulations with a small number of APs, the connectivity density and area traffic capacity could not be accurately measured. Since ensuring equally loaded entities in the network creates available resources across all components (avoiding fully loaded or idle states), the system balance metric was incorporated to improve these two KPIs, as well as network availability.

Mobility and EE: In this study, we aimed to improve mobility and EE by leveraging architecture-related technologies and our low-complexity scheme. A smooth slice reconfiguration in CF-based RAN can significantly improve mobility. Moreover, employing one agent instead of multiple agents and training processes enhances the efficiency of the required energy for computation and memory demands. Furthermore, a dynamic slice configuration approach results in the efficient use of communication links by activating only the engaged links, thereby substantially improving energy efficiency.

C. Complexity and Scalability, Performance in Real Scenarios

Considering dynamic slice configuration and reconfiguration as a complex task in 6G networks, only distributed, scalable, and low-complex approaches are likely to be implemented in real-world 6G networks. This section explains how our considered MORL agent leverages scalability, sample efficiency, and improved policy adaptation to ensure practical applicability in real scenarios.

To evaluate the scalability of the MORL algorithm, we investigate three aspects: handling a large state space (i.e., a large number of O-DUs and O-RUs), accommodating many objectives (i.e., a large number of service types), and managing large optimal policy sets.

While our simulation setup included a total number of 8 O-DUs and O-RUs, in real scenarios, a user will be supported by a larger number of O-DUs and O-RUs. However, the Envelope MOQ-learning algorithm w has demonstrated success in environments with large state spaces, such as the Super Mario Bros game, which features a state space size of approximately $4*240*256*3$ (RGB) frames [30]. In contrast, the state space size of our environment grows according to $2^{n+1}-1$, where n is the total number of O-RUs and O-DUs. A comparison of our simulation state space size with a video game's state space size indicates that our simulation's state space size can increase more than double without scalability issues for the agent.

In practical 6G networks, which are decentralized, O-RU clustering and O-DU selection are critical for efficient CF mMIMO systems [12]. Moreover, the distance between the user and O-RAN entities significantly impacts the QoS. Consequently, in real scenarios, users are not served by all O-RAN entities; rather, natural clusters of entities capable of better serving specific users are formed. This clustering ensures the agent's adaptability to real-world implementations.

On the other hand, while having more than three preferences (or reward elements) can be considered a problem with a large number of objectives, our simulations show that the MORL agent was successful with four preferences. Moreover, previous studies [30] demonstrated the algorithm's effectiveness with up to 6 preferences.

The size of the optimal policy also influences performance [30]. To analyze this, we reviewed the agent's performance in a binary tree environment using the Coverage Ratio (CR)

and Adaptation Error (AE) metrics, as defined in [30]. CR evaluates the agent´s ability to recover optimal solutions in the convex coverage set (i.e., policies offering the best trade-offs between objectives for varying preferences), while AE measures the agent's real-time policy adaptation to specified preferences. CR and AE are formulated as:

$$CR = \frac{\text{Area covered by learned policies}}{\text{Area of the true Pareto frontier}} \tag{7}$$

and

$$AE(\lambda) = V^*(\lambda) - V^\pi(\lambda) \tag{8}$$

where $V^*(\lambda)$ represents the optimal value for preference λ, and $V^\pi(\lambda)$ is the value of policy π learned by the algorithm for the same λ.

The Envelope MORL agent in this work outperforms other algorithms, such as the Multi-Objective Fitted Q-Iteration [34], the Conditional Neural Network with Optimistic Linear Support [35], and the Scalarized Q-update with Q-learning [36] in binary trees with depths of 5,6, and 7 and 6 preferences both in terms of CR and AE [30]. Moreover, the agent's success in trees with depths of 8 and four preferences. While environments with smaller solution sets yield higher CR and AE values, the Envelope MOQ-learning algorithm remains the most stable and effective approach.

In terms of sample efficiency and policy adaptation, the Envelope MOQ-learning algorithm demonstrates promising performance [30], making it a more efficient solution.

Finally, the use of a MORL agent to accommodate multiple service types is motivated by the fact that a single neural network and one training phase can produce a policy adaptable to multiple preferences after training. This approach eliminates the need for training separate agents for each service type [11], thereby reducing computational complexity and enhancing energy efficiency. For large-scale systems requiring multiple agents, adding one MORL agent can replace numerous single RL agents, further improving scalability and efficiency.

VI. Conclusion and Future Steps

In this research work, we proposed a dynamic network slicing scheme integrated into xAPPs in Near-RT RIC of O-RAN. This intelligent scheme leverages the extensive connectivity and smooth mobility enabled by CF mMIMO. Our primary objective was to increase the computational and energy efficiency through several steps. First, we proposed a MORL-based approach to deliver multiple service types using a single training phase for one agent. Second, we employed an agent that offers scalability, sample efficiency, and stability in real-world scenarios. Furthermore, our proposed scheme provides a slice of O-DUs, O-RUs, their connecting links, and resource blocks for every user which allows the rest of O-RAN entities to stay in the offline mood and improves the EE in the RAN. Extensive simulations and numerical analyses demonstrated that this approach successfully meets the required QoS for various service types. The results were validated by analyzing multiple KPIs and observing the behavior of the algorithm in our simulated environment.

This work can be improved by incorporating more service types and different metrics for the reward array. The network topology could also be expanded or optimized by increasing the number of O-RUs and O-DUs. Combining the Gym-based environment with other network-related simulators, would enable testing the trained agent under different conditions, such as analyzing area traffic density, which could not be addressed in our user-centered episodic simulations. Moreover, the integration of the proposed intelligent approach in the xApp framework of Near-RT RIC in O-RAN has not been discussed in this work. As a future direction, implementing the proposed approach on experimental platforms [37], [38] could offer valuable insights and practical validation. Further improvements can be achieved by adopting a CF mMIMO system that integrates both sensing and communication. In an Integrated Sensing and Communication (ISAC) CF mMIMO system, the signals transmitted for data communication are also used for environmental sensing. This approach leverages distributed sensing capabilities, enabling more efficient use of spectrum and hardware resources, while significantly enhancing overall system efficiency.

Acknowledgement

This work has received funding from the European Union under the ADROIT6G project (Grant agreement ID: 101095363).

References

[1] C.-X. Wang *et al.*, "On the road to 6g: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[2] A. Abouaomar, A. Taik, A. Filali, and S. Cherkaoui, "Federated deep reinforcement learning for open ran slicing in 6g networks," *IEEE Communications Magazine*, vol. 61, no. 2, pp. 126–132, 2022.

[3] J. Wang, J. Liu, J. Li, and N. Kato, "Artificial intelligence-assisted network slicing: Network assurance and service provisioning in 6g," *IEEE Vehicular Technology Magazine*, vol. 18, no. 1, pp. 49–58, 2023.

[4] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane, "Communication and computation o-ran resource slicing for urllc services using deep reinforcement learning," *IEEE Communications Standards Magazine*, vol. 7, no. 1, pp. 66–73, 2023.

[5] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint network slicing and mobile edge computing in 5g networks," in *in Proc. IEEE ICC*, 2019.

[6] A. T. Ajibare and O. E. Falowo, "Resource allocation and admission control strategy for 5g networks using slices and users priorities," in *in Proc. IEEE AFRICON*, 2019.

[7] X. Yang, Y. Wang, I. C. Wong, Y. Liu, and L. Cuthbert, "Genetic algorithm in resource allocation of ran slicing with qos isolation and fairness," in *2020 IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2020, pp. 1–6.

[8] A. A. Khan, M. Abolhasan, W. Ni, J. Lipman, and A. Jamalipour, "An end-to-end (e2e) network slicing framework for 5g vehicular ad-hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7103–7112, 2021.

[9] J. A. Hurtado Sánchez, K. Casilimas, and O. M. Caicedo Rendon, "Deep reinforcement learning for resource management on network slicing: A survey," *Sensors*, vol. 22, no. 8, p. 3031, 2022.

[10] A. M. Nagib, H. Abou-zeid, and H. S. Hassanein, "Accelerating reinforcement learning via predictive policy transfer in 6g ran slicing," *IEEE Transactions on Network and Service Management*, 2023.

[11] N. Ghafouri, J. S. Vardakas, K. Ramantas, and C. Verikoukis, "A multi-level deep rl-based network slicing and resource management for o-ran-based 6g cell-free networks," *IEEE Transactions on Vehicular Technology*, pp. 1–12, 2024.

[12] ——, "Rl-based high-level radio unit clustering and distributed unit assignment in user-centric cell-free mmimo for oran-based 6g," in *ICC 2024 - IEEE International Conference on Communications*, 2024, pp. 2065–2070.

[13] T. Huang, W. Yang, J. Wu, J. Ma, X. Zhang, and D. Zhang, "A survey on green 6g network: Architecture and technologies," *IEEE access*, vol. 7, pp. 175 758–175 768, 2019.

[14] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane, "Communication and computation o-ran resource slicing for urllc services using deep reinforcement learning," *IEEE Communications Standards Magazine*, vol. 7, no. 1, pp. 66–73, 2023.

[15] R. Dangi and P. Lalwani, "Optimizing network slicing in 6g networks through a hybrid deep learning strategy," *The Journal of Supercomputing*, pp. 1–21, 2024.

[16] M. A. Hossain, A. Xiang, A. Kiani, T. Saboorian, J. Kaippallimalil, and N. Ansari, "Ai-assisted e2e network slicing for integrated sensing and communication in 6g networks," *IEEE Internet of Things Journal*, 2023.

[17] F. Song, H. Xing, X. Wang, S. Luo, P. Dai, and K. Li, "Offloading dependent tasks in multi-access edge computing: A multi-objective reinforcement learning approach," *Future Generation Computer Systems*, vol. 128, pp. 333–348, 2022.

[18] A. Feriani *et al.*, "Multiobjective load balancing for multiband downlink cellular networks: A meta-reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2614–2629, 2022.

[19] S. Sharma and W. Yoon, "Multiobjective reinforcement learning based energy consumption in c-ran enabled massive mimo," *Radioengineering*, vol. 31, no. 1, pp. 155–163, 2022.

[20] "O-RAN ALLIANCE," https://www.o-ran.org/, Accessed: 1.01.2023.

[21] S. M. A. Zaidi, M. Manalastas, H. Farooq, and A. Imran, "Mobility management in emerging ultra-dense cellular networks: A survey, outlook, and future research directions," *IEEE Access*, vol. 8, pp. 183 505–183 533, 2020.

[22] Y. Xiao, P. Mähönen, and L. Simić, "Mobility performance analysis of scalable cell-free massive mimo," in *in Proc. IEEE ICC*, 2022.

[23] J. Jiang, F. Yan, Y. Ye, W. Sutthiphan, J. Zhang, and B. Ai, "Traffic demand-oriented cell-free massive mimo network," *IEEE Wireless Communications Letters*, 2023.

[24] A. Mesodiakaki *et al.*, *The 6G Architecture Landscape: European Perspective*. Belgium: European Commission, 2023.

[25] P. Chang and G. Miao, "Energy and spectral efficiency of cellular networks with discontinuous transmission," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2991–3002, 2017.

[26] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, "5g urllc: Design challenges and system concepts," in *in Proc. 15th ISWCS*, 2018.

[27] M. Al-Ali and E. Yaacoub, "Resource allocation scheme for embb and urllc coexistence in 6g networks," *Wireless Networks*, pp. 1–20, 2023.

[28] Q. Chen, J. Wu, J. Wang, and H. Jiang, "Coexistence of urllc and embb services in mimo-noma systems," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 839–851, 2022.

[29] X. An, J. Wu, W. Tong, P. Zhu, and Y. Chen, "6g network architecture vision," in *in Proc. EuCNC/6G Summit*, 2021.

[30] R. Yang, X. Sun, and K. Narasimhan, "A generalized algorithm for multi-objective reinforcement learning and policy adaptation," *Advances in neural information processing systems*, vol. 32, 2019.

[31] L. T. Watson and R. T. Haftka, "Modern homotopy methods in optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 74, no. 3, pp. 289–305, 1989.

[32] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[33] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[34] A. Castelletti, F. Pianosi, and M. Restelli, "Tree-based fitted q-iteration for multi-objective markov decision problems," in *The 2012 international joint conference on neural networks (IJCNN)*. IEEE, 2012, pp. 1–8.

[35] H. Mossalam, Y. M. Assael, D. M. Roijers, and S. Whiteson, "Multi-objective deep reinforcement learning," *arXiv preprint arXiv:1610.02707*, 2016.

[36] A. Abels, D. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher, "Dynamic weights in multi-objective deep reinforcement learning," in *International conference on machine learning*. PMLR, 2019, pp. 11–20.

[37] J. L. Herrera, S. Montebugnoli, P. Bellavista, and L. Foschini, "Enabling reusable and comparable xapps in the machine learning-driven open ran," in *2024 IEEE 25th International Conference on High Performance Switching and Routing (HPSR)*, 2024, pp. 37–42.

[38] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Colo-ran: Developing machine learning-based xapps for open ran closed-loop control on programmable experimental platforms," *IEEE Transactions on Mobile Computing*, vol. 22, no. 10, pp. 5787–5800, 2023.