ReLAx: Deep Reinforcement Learning Based Resource Allocation for Next-G RANs

Ayman Younis, Chuanneng Sun*, Dario Pompili*
*Department of Electrical and Computer Engineering
Rutgers University–New Brunswick, NJ, USA
E-mails: {a.younis, chuanneng.sun, pompili}@rutgers.edu

2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS) | 979-8-3503-6399-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/MASS62177.2024.00064

***Abstract*—Next-Generation Radio Access Network (Next-G RAN) will leverage a novel architecture that accelerates the transition from inflexible networks to agile and adaptable networks. In this paper, we introduce a novel Deep Reinforcement Learning-based Resource Allocation (ReLAx) framework to deal with the joint optimization of UE association and power allocation in Next-G RAN systems. The ReLAx problem has been formulated to maximize the network Energy Efficiency (EE) under the constraints of Quality of Service (QoS), fronthaul link, functional split configuration, and transmit power budget. The optimization problem is cast as non-convex and NP-complete. A multi-task Deep Deterministic Policy Gradient (DDPG) method is proposed to solve the complexity, in which two actors are trained to generate UE association and power allocation, respectively. To speed up the training process and reduce computational resources, we introduce soft multi-task learning as a constraint during training so that one model would not drift too far away from the other one. Our real-time experiments on a fully containerized Next-G RAN testbed show the effect of functional splits on CPU utilization and system latency. In addition, simulation results show that the proposed resource allocation solution outperforms competing traditional algorithms, such as standard DDPG and Weighted Minimum Mean Square Error (WMMSE).**

***Index Terms*—Next-G RAN, resource allocation, deep reinforcement learning, OpenAirInterface, virtualization.**

I. Introduction

Motivation. In the near future, mobile data traffic is expected to continue to increase due to the increasing popularity of smart portable devices and the growing demand for emerging technologies, such as the Internet of Things (IoT), video streaming, and Augmented/Virtual Reality (AR/VR). According to a recent report from Cisco, by 2023, the total number of Internet users is expected to reach 5.3 billion, with an average 5G connection speed of 575 Mbps [1]. The increase in traffic patterns for Beyond 5G (B5G) services imposes significant challenges in meeting specific requirements such as Quality of Service (QoS), channel conditions, and service latency with existing mobile network architectures. Radio and computational resources can be seen as a real bottleneck in fulfilling the growing demands of B5G. On the other hand, adding more radio and computing resources at network sites could significantly increase the energy consumption of future mobile communication systems, particularly affecting the Operating Cost (OPEX) of network operators. Considering the limited communication radio resources and prohibitive signaling energy costs, it is essential to study novel practical RAN systems in which resource allocation algorithms can be applied effectively and efficiently.

Recently, Next Generation Radio Access Networks (Next-G RAN) has been presented as an emerging framework to enable the *virtualization* and *softwarization* technologies [2], [3]. The key feature of the Next-G RAN design is the flexible centralization of the core signal processing functions, performed by the digital baseband (PHY/MAC) processing in the Central Unit (CU) while retaining radio access and minimal communication functionalities at cell sites in the Distributed Units (DUs). Cooperation between the two main units, CU and DU, in an efficient way will open a path to enhance the overall network's significant metrics, including architecture planning, network operation, resource utilization, and back/mid/front-haul management. Consequently, various B5G wireless services, such as massive Machine-Type Communication (mMTC), enhanced Mobile Broadband (eMBB), and ultra-Reliable Low-Latency Communication (uRLLC), can be dynamically deployed and managed to satisfy the emerging demands of B5G applications.

Our Approach. In Reinforcement Learning (RL), the policy is trained while collecting data, making it a suitable choice to solve real-time decision-making problems, especially in dynamic resource allocation [4], [5]. As a result, the system model and prior data requirements in RL are less stringent. Furthermore, neural networks in DRL can enable the model to learn complex objective functions and handle large state and action spaces, such as multi-user systems [6] and robot controllers [7], [8]. A widely recognized approach in DRL is the Deep Deterministic Policy Gradient (DDPG) [9]. Using DDPG as a controller for optimizing variables is a promising direction, as DDPG is effective in generating continuous actions based on the state of the system. However, in this problem, two variables need to be optimized: the UE association, which is discrete, and the power allocation, which is continuous. Joint optimization of these two variables can result in a significant increase in the number of parameters required and a degradation of performance due to the presence of discrete and continuous action spaces. Therefore, we propose a twin-actor approach in which two actors are employed, each dedicated to one variable, and a centralized critic is utilized to jointly evaluate the performance of the actors.

In addition, we argue that the association and UE power allocation are closely related, implying some level of overlap

2155-6814/24/$31.00 ©2024 IEEE
DOI 10.1109/MASS62177.2024.00064

in the parameter spaces of the actors. Hence, in our proposed algorithm, we introduce a multi-task learning technique, DDPG, to significantly reduce the number of parameters and enhance the training efficiency. Compared to standard convex optimization methods, the DRL-based resource allocation algorithm can make real-time decisions based on the current state of the network. This type of intelligent decision-making is crucial for many B5G services, particularly those that require real-time, low-latency capabilities. In this paper, we present the system model for the Next-G system and formulate our ReLAx resource allocation algorithm, with the goal of maximizing overall Energy Efficiency (EE) in the Next-G RAN while satisfying constraints such as QoS requirements, transmission power budget, and limited fronthaul capacity.

Related Work. With the main target of integrating the full radio stack platforms and opening the way for virtual-cloud-based RAN ecosystems, the architecture of Next-G RAN will become intelligent and agile. However, how to properly manage various radio-computation resources in Next-G RANs has become a key challenge and research focus in the wireless communication field. For example, energy allocation has been studied in several works such as [10]. Fang *et al.* [11] have considered user fairness in a Multi-Carrier Non-Orthogonal Multiple Access (MC-NOMA) system. The joint problem of user assignment and power allocation has been included in [12], [13]. In addition, the authors in [14] have proposed a resource allocation solution that treats the problem as a bin-packing problem, with the aim of minimizing the number of active Virtual Machines (VMs) in the cloud center.

Meanwhile, DRL has emerged as a new research trend in B5G applications and has been demonstrated as a feasible tool to address dynamic resource allocation problems in cloud-based RAN systems [15]–[17]. In [15], the authors have proposed a Deep Q Network (DQN) method for power allocation in wireless networks. The model was initially trained in a simulator using the deep Q learning rule and then deployed in the real environment for fine-tuning. However, while the DQN performed well in discrete action spaces, its adoption in continuous power models may result in undesirable performance. In [16], the authors study a resource allocation method by designing a new DNN-based optimization approach solution consisting of a series of alternating direction methods of multiplier iterative schemes that assign the Channel State Information (CSI) values as the learned weights. Furthermore, the authors in [17] present a three-step deep reinforcement learning-based scheme that solves the joint sub-channel assignment and power allocation problem in an uplink multi-user NOMA system to maximize the network EE.

Although DNN-based methods have led to significant improvements in solving resource management in cloud-based wireless systems, these studies often overlook the challenges of the system and depend heavily on simplified assumptions in modeling the radio-computation resources of CUs and DUs. Although these studies address resource allocation problems from various individual perspectives, they do not take into account the dynamic nature of these problems in the Next-G RAN scenario, where functional splits are supported. In this paper, we present a DRL-based resource allocation solution for Next-G systems in realistic conditions, taking into consideration the required QoS and the limitations of fronthaul capacity and transmitted power. In addition, we validate our model through real-time experiments conducted on a fully containerized Next-G testbed.

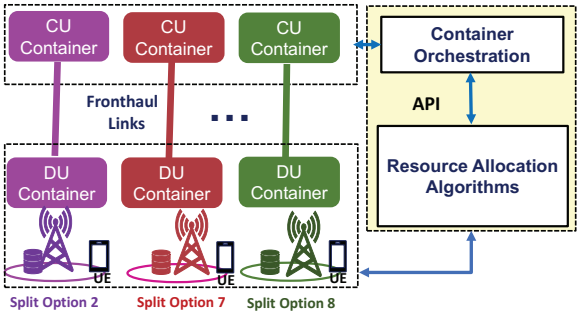


Fig. 1. The proposed Next-G architecture with resource allocation algorithms.

The main **contributions** of this work are listed below.

- We investigate resource allocation for Next-G RAN and formulate it as a Mixed-Integer Non-Linear Programming (MINLP) problem, considering constraints such as QoS, fronthaul capacity, functional splitting, and DU's power budget. The problem optimizes the UE association and the transmit power of the DU to maximize network EE, defined as the ratio of data rate to power consumption in different functional splitting scenarios.
- To address the complexity of the optimization problem formulated, we develop a deep learning-based framework named ReLAx, which improves upon the DDPG method. ReLAx can dynamically optimize the UE association and transmitted power in downlink Next-G systems.
- We set up a real-time Next-G testbed using the OpenAir-Interface (OAI) platform [18] and container virtualization, allowing wireless connections between the CU, DU and COTS UE. Experiments show that CU-DU CPU utilization depends on network parameters such as PRB and functional split options.
- Numerical simulations reveal that the proposed ReLAx framework can optimize network EE and outperform competing algorithms such as DDPG and Weighted Minimum Mean Square Error (WMMSE).

Paper Organization. In Sect. II, we present the system model. In Sect. III, the EE maximization problem is formulated, followed by the presentation of our proposed machine learning solution. The experimental results and numerical simulations are discussed in Sect. IV. Finally, the paper is concluded in Sect. V.

II. System Model

This section presents the network description, functional split model, wireless link model, and network power model.

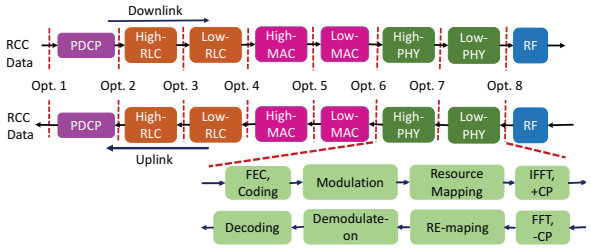


Fig. 2. Split options as specified by 3GPP [19].

A. Network Description

We consider a Next Generation Node B (gNB) to consist of multiple CUs connected to multiple DUs via a high-speed optical fiber fronthaul interface. The logical diagram of the Next-G downlink transmission, which comprises L DUs and U UEs, is shown in Fig. 1. The sets of DUs and UEs are denoted as $\mathcal{L} = \{1, 2, ..., L\}$ and $\mathcal{U} = \{1, 2, ..., U\}$, respectively. Orthogonal Frequency Division Multiple Access (OFDMA) techniques have been adopted to provide communication services in the downlink scenario. As part of the Next-G RAN, 3GPP has proposed eight different functional split options between the DU and the CU, as shown in Fig. 2, which are defined in 3GPP TR 38.801. Therefore, we assume that the functional split technique has been integrated into the gNB while formulating the essential modes of the Next-G system. In general, there are significant benefits to enabling flexible, functional split orchestration in the Next-G RAN. Some of the benefits include cost reduction, traffic load balancing, and minimizing latency and fronthaul costs. It is worth noting that in a real-world Next-G testbed implementation, the CU and DU can be deployed using virtualization techniques. For example, on the OAI platform, each CU can be realized by a container image and paired with one DU container image. To connect our model with a real experimental Next-G testbed, we adopt the DU-to-CU model.

B. Functional Split Model.

Placing all RAN functions in the CU pool can lead to maximizing energy savings; however, fully adopting a centralized RAN architecture is not always feasible. For example, physical layer processes such as FFT, parallel/serial, and cyclic prefixes (as shown in Fig. 2) have strict latency requirements and generate high traffic on the limited back/mid-haul interface located at the CU. As a result, these processes are usually implemented in the DU (e.g., the 7.2 functional split option in O-RAN [20]). High PHY layer and MAC/RLC processes that require high performance also have strict constraints, such as in LTE where the round-trip latency tolerance of the MAC layer using synchronous HARQ technique is limited to $3ms$ [21]. However, in the 5G MAC layer using the fully asynchronous HARQ technique, strict latency requirements are no longer an issue, and the round-trip time mainly depends on the type of service being provided. In light of the above, it can be concluded that various RAN processes can be supported by different 5G services, such as uRLLC, eMBB, and mMTC.

TABLE I
EXPERIMENTAL SPLIT OPTIONS AND CPU LOAD FOR NEXT-G RAN.

Split s	Split type	DU $\leftrightarrow$ CU
z_{j1}	No split, all at CU	$\leftrightarrow f_1, f_2, f_3$
z_{j2}	F1 split	$f1 \leftrightarrow f2, f3$
z_{j3}	IF 4.5 split	$f_1, f_2 \leftrightarrow f_3$
z_{j4}	No split, all at DU	$f_1, f_2, f_3 \leftrightarrow$

Split function	CPU load
f_1	63%
f_2	21%
f_3	15%

Therefore, we examine four practical CU/DU configurations based on the functional split selections outlined in Tab. I. In our model, we denote the set of Next-G RAN functions and the set of functional split options as $\mathcal{F}$ and $\mathcal{Z}$, respectively. A functional split $z_{js}, \forall j \in \mathcal{L}, s \in \{1, 2, 3, 4\}$, is performed at gNB j if all RAN functions above and including f_s are executed in the CU, while all RAN functions below f_s are executed at the DU. Hence, in the functional split $z_{js} \in \mathcal{Z}$, the CPU utilization at the CU (ω_s) is equal to the sum of the processing load of all RAN functions above and including f_s. That is, $\omega_s = \sum_{j \geq s} \varrho_j$, where ϱ_j represents the CPU requirement at the functional split s. Based on the experimental results in Sect. IV-A, we have generated Tab. I to depict the CPU processing load for downlink traffic. Furthermore, the CU to DU functional split indicator, z_{js}, is defined such that $z_{js} = 1$ indicates that gNB j operates in functional split s, while $z_{js} = 0$ implies otherwise.

C. Wireless Link Model

We assume that each UE can establish a wireless connection with the DU through uplink and downlink cellular links. In addition, UE is considered to be static, and cell channels are assumed to be constant during each decision-making and resource allocation algorithm procedure. In this paper, we adopt the downlink OFDMA system as the scheme for the proposed Next-G RAN model. Consequently, the operational frequency band B is divided into N equal sub-bands, each with a size of $W = B/N$ [Hz]. To maintain the orthogonality property in our model, we assume that each UE is assigned to one sub-band for downlink transmission. Thus, each DU can serve a maximum of N UEs simultaneously. Furthermore, we take into account both large-scale and small-scale fading. We assume that large-scale fading is consistent across all sub-bands, while small-scale fading is frequency-sensitive and flat. Let $g^n_{j,u}$ represent the channel gain from DU j to UE u on sub-band n. It is calculated as follows,

$$g^n_{j,u} = \varpi_{j,u} |h^n_{j,u}|^2, \forall j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}, \quad (1)$$

where $\varpi_{j,u}$ denotes the large-scale fading, including path loss and shadowing, and $h^n_{j,u}$ represents the small-scale Rayleigh fading. To model small-scale fading, we utilize Jake's model [22] and model it as a first-order complex Gaussian-Markov process. The update rule is as follows,

$$h^n_{j,u} = \rho h^n_{j,u} + \sqrt{1 - \rho^2} e^n_{j,u}, \forall j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}, \quad (2)$$

where $\rho = J_0(2\pi f_d T)$ represents the correlation between consecutive fading blocks, with J_0 being the zero-th order Bessel function of the first kind and f_d being the maximum Doppler frequency. T is the time interval between successive

channel gain estimations. $e^n_{j,u}$ represents the channel innov tion process and is modeled as a circularly symmetric comple Gaussian distribution. A high value of ρ indicates a substanti change in the channel since the last estimation, which can b due to either a long time interval between estimations or a hig maximum Doppler frequency. Let $\mathcal{N} = \{1, ..., N\}$ be the s of available sub-bands at each DU. We denote the sub-chann association between UE u and sub-channel n of DU j as,

$$x^n_{ju} = \begin{cases} 1, & \text{UE } u \text{ associated with DU } j \text{ on sub-channel } n \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

Let p^n_{ju} denote the transmission power from DU j o sub-band n to UE u. The Signal-to-Interference-plus-Nois Ratio (SINR) from DU j on sub-band n to UE u is define by,

$$\gamma^n_{ju} = \frac{p^n_{ju} h^n_{ju}}{\sum_{k \in \mathcal{K} \setminus \{j\}} \sum_{r \in U} x^n_{kr} p^n_{kr} h^n_{kr} + \sigma^2}, \quad (4)$$

where σ^2 is the variance of Additive White Gaussian Noise (AWGN). Then, the maximum achievable data rate of UE u using the sub-channel n in DU j can be calculated as,

$$R^n_{ju}(\mathcal{X}, \mathcal{P}) = W \log_2 (1 + \gamma_{ju}), \forall j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}, \quad (5)$$

where $\gamma_{ju} = \sum_{n \in N} \gamma^n_{ju}$ is the total SINR, $\mathcal{X} = \{x^n_{ju} | j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}\}$ and $\mathcal{P} = \{p^n_{ju} | j \in \mathcal{L}, n \in \mathcal{N}, u \in \mathcal{U}\}$ are to represent the UE assignment and power allocation, respectively. Hence, the sum-rate of the network R^T can be written as,

$$R^T(\mathcal{X}, \mathcal{P}) = \sum_{j \in \mathcal{L}} \sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{U}} R^n_{ju}(\mathcal{X}, \mathcal{P}) \quad (6)$$

In Next-G RAN, the processing of baseband signal between the CU and DU is transmitted through a fronthaul interface, standardized as *F1 interface* in 3GPP [2]. This kind of fronthaul transmission requires a high-speed data rate—10× higher than the original baseband signal data rate [23]. For this reason, the fronthaul link is considered the bottleneck of cloud-based RANs. To that end, 3GPP proposed a novel functional splitting technique to flexibly manage and control data rate transmission between the CUs and the DUs in Next-G RAN. Specifically, the functional split can significantly reduce the transmission cost by shifting part of the baseband signal processing operations from the CU to DUs [24]. In this paper, we define the fronthaul capacity constraint by,

$$\sum_{u \in \mathcal{U}} \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{Z}} x^n_{ju} z_{js} R^n_{ju}(\mathcal{X}, \mathcal{P}) \leq C_j, \quad (7)$$

where C_j represents the fronthaul capacity of DU j. Let C^{max}_j be considered as the fronthaul capacity of DU j. Hence, C_j can be expressed as $C_j = C^{max}_j / \epsilon$, where ϵ is the ratio of the bandwidth that is demanded from the baseband transmission between CUs and DUs. Hence, the value of C_j is mainly based on fronthaul transmission technologies (e.g., optical fiber technology).

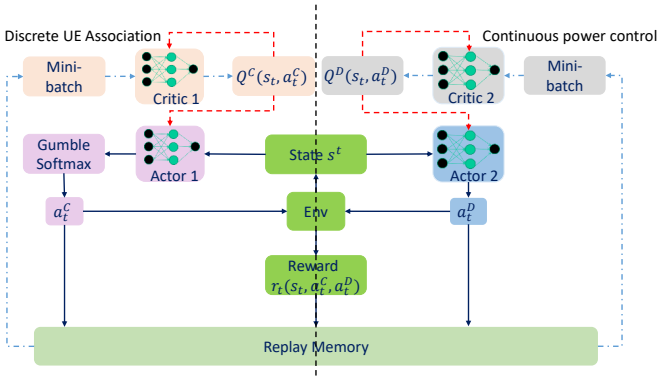


Fig. 3. The framework and workflow of ReLAx. Solid lines indicate data flow; red dashed lines and blue dash-dotted lines represent forward and backward gradient propagation.

D. Computational Power Model

The power consumption for the Next-G RAN downlink is modeled to be two main parts, the power consumption of the CU and the power consumption of the DU.

CU-power consumption. In general, CUs can usually be implemented virtually by virtual machines or containers. In this way, the capacities of the CU containers can be dynamically modified to deal with variable traffic loads and channel states. Thus, the power consumption of the CUs depends on computing the workload size while processing the baseband signals from DUs [25]. Hence, we can model the CU power computation to handle the baseband traffic from DU j as,

$$P^{\mathrm{CU}}_j = P^{\mathrm{C}}_j + \alpha_j \sum_{u \in \mathcal{U}} \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{Z}} x^n_{ju} z_{js} \omega_s, \forall j \in \mathcal{L}, \quad (8)$$

where P^{C}_j is the static power of CU j corresponding container of DU j. α_j is the container power consumption factor determined by the architecture, traffic size, functional splitting mode, and hardware equipment of the CU pool.

DU-power consumption. Similarly, we can assume that DU power consumption consists of two main parts: static and dynamic power consumption. Static power consumption is needed to run the DU container, while dynamic power consumption is usually proportional to DU transmitted power, traffic workload, and network configurations. Hence, the power consumption of DU j can be modeled as,

$$P^{\mathrm{DU}}_j = P^{\mathrm{D}}_j + \beta_j \sum_{u \in \mathcal{U}} \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{Z}} x^n_{ju} z_{js} (\omega_1 - \omega_s) p^n_{ju}, \quad (9)$$

where P^{D}_j models the static power consumption of DU j, and β_j is the power factor of DU j characterizing the link between the dynamic power consumption and the traffic load. The value of β_j is determined based on the architecture of the DU, traffic load, and the type of functional split mode. Hence, the power factor parameters β_j and α_j are detailed in Sect. IV-A. Based on the above considerations, the total network power consumption model of the Next-G RAN can be expressed as $P(\mathcal{X}, \mathcal{Z}, \mathcal{P}) = P^{\mathrm{CU}}_j + P^{\mathrm{DU}}_j$.

III. Energy Efficiency Maximisation

In this section, we formulate the EE maximization problem, followed by the proposed solution.

A. Problem Formulation and Relaxation

To effectively utilize radio resources such as the radio spectrum and transmit power, and optimize the computation capacity, including fronthaul capacity and CU-DU computation capacity while ensuring the QoS requirements of UEs, we define the network EE of the total system as a more comprehensive objective for downlink Next-G systems. Thus, the network EE of Next-G RAN is defined as,

$$\zeta_{\mathrm{EE}}(\mathcal{X},\mathcal{Z},\mathcal{P}) = R^T(\mathcal{X},\mathcal{P})/P(\mathcal{X},\mathcal{Z},\mathcal{P}) \tag{10}$$

The adaptive EE function in (10) quantitatively describes the impact of the network's achievable data rate and total power consumption on system performance. The main resource allocation problem can be formulated as,

$$\underset{\mathcal{X},\mathcal{Z},\mathcal{P}}{\mathrm{Max}} \ \ \zeta_{\mathrm{EE}}(\mathcal{X},\mathcal{Z},\mathcal{P}) \tag{11a}$$

$$\text{s.t.} \ \ \sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{n\in\mathcal{N}}\sum\nolimits_{s\in\mathcal{Z}} x^n_{ju} z_{js} R^n_{ju} \le C_j, \forall j \in \mathcal{L}, \tag{11b}$$

$$\sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{n\in\mathcal{N}} p^n_{ju} \le P^{\max}_j, \forall j \in \mathcal{L}, \tag{11c}$$

$$\sum\nolimits_{j\in\mathcal{L}}\sum\nolimits_{n\in\mathcal{N}} x^n_{ju} R^n_{ju} \ge R^{\min}_u, \forall j \in \mathcal{L}, \tag{11d}$$

$$\sum\nolimits_{u\in\mathcal{U}} x^n_{jun} \le 1, \forall j \in \mathcal{L}, n \in \mathcal{N}, \tag{11e}$$

$$\sum\nolimits_{s\in\mathcal{Z}} z_{js} = 1, \forall j \in \mathcal{L}, \tag{11f}$$

$$x^n_{ju} = \{0,1\}, \forall j \in \mathcal{L}, u \in \mathcal{U}, n \in \mathcal{N}, \tag{11g}$$

The constraints in (11) can be described as follows; the fronthaul capacity is modeled as the maximum tolerated data rate that can be transmitted on the fronthaul link, as previously reported in literature such as in [26], [27]. Therefore, constraint (11b) limits the fronthaul capacity of DU j to the maximum fronthaul capacity of the system, C_j; constraint (11c) sets the transmission power budget for each DU; constraint (11d) ensures that each UE's data rate requirement exceeds its minimum data rate, $R^{\min}_u$; constraint (11e) restricts each UE to a single sub-band per allocation; constraint (11f) requires each gNB j to use a single functional split option per iteration; and constraint (11g) enforces binary resource allocation in Next-G RAN.

The objective function in problem (11) is expressed in fractional form and is non-convex. Furthermore, the presence of binary variables $\mathcal{X}$ and $\mathcal{Z}$ makes the optimization problem in (11) a MINLP problem, which is known to be NP-hard and challenging to solve [28]. Similar to [29], the primary problem in (11) is reformulated as,

$$\underset{\mathcal{X},\mathcal{Z},\mathcal{P}}{\mathrm{Max}} \ \ R^T(\mathcal{X},\mathcal{P}) - \psi P(\mathcal{X},\mathcal{Z},\mathcal{P}) \tag{12a}$$

$$\text{s.t.} \ \ (11\text{b}) - (11\text{f}), \tag{12b}$$

where the symbol ψ represents the weight assigned to network power consumption.

B. RL Problem Formulation

The major challenge in solving the resource allocation problem in (11) is that the integer variable x^n_{ju} makes the optimization problem a MIP problem that is, in general, non-convex and NP complete [30]. In addition, in real wireless network environments, the QoS, fronthaul link, and transmit power requirements update dynamically. Therefore, it is generally infeasible to adopt traditional optimization solutions (e.g., standard convex solutions) to handle resource management complexities. Hence, a deep reinforcement method is proposed to deal with these challenges. Specifically, we will first provide a general background of traditional ML approaches to solve optimization problems, and then present our proposed solution to solve the problem in (11).

Problems that can be modeled as a Markov Decision Process (MDP) can be solved using RL algorithms. An MDP consists of a set of states $\mathcal{S}$, which characterizes the properties of the system, and a set of actions $\mathcal{A}$. In addition, the RL agent is deployed with a policy $\pi : \mathcal{S} \to \mathcal{A}$ parameterized by θ, which provides decisions given the state. After executing an action, the environment will change according to a state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. The agent will receive a reward from the environment as a function of state and action $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The goal of RL algorithms is to maximize the total expected return $R = \sum_{t=0}^{T} \gamma^t r_t$, where T is the maximum steps, and γ is a discount factor.

The state, action, and reward in our proposed solution are described as follows.

State. We encode all relevant information about the system to assist actors in making decisions and define the state as a tuple that includes the current allocation of sub-bands, power, and channel gain. $s_t = \{\mathcal{X}_{t-1}, \mathcal{Z}_{t-1}, \mathcal{P}_{t-1}, H_{t-1}\}$, where $\mathcal{X}_{t-1}$, $\mathcal{Z}_{t-1}$, $\mathcal{P}_{t-1}$, and H_{t-1} are the values of $\mathcal{X}$, $\mathcal{Z}$, $\mathcal{P}$, and the channel gain from previous iterations. The actors are supposed to make informed decisions based on this information.

Action. The action is a set of variables to optimize, with the binary action $a^D_t \in \{0,1\}$ representing the selection of subbands for variable $\mathcal{X}$. The continuous action $a^C_t \in \mathbb{R}$ represents the variable $\mathcal{P}$. To comply with the maximum power constraint outlined in (11c), we normalize the power levels at each DU to ensure they do not exceed P^{max}_j.

Reward. The reward is the metric by which the action is evaluated based on the state, with higher rewards indicating better performance by the agent. Therefore, we define the reward as the EE that we aim to maximize $r_t = \zeta_{\mathrm{EE}}(\mathcal{X},\mathcal{Z},\mathcal{P}) = R^T(\mathcal{X}_t,\mathcal{P}_t)/P(\mathcal{X}_t,\mathcal{Z}_t\mathcal{P}_t)$.

C. ReLAx Design in Next-G RAN System

In the optimization problem (11), we face two types of challenges: (i) we are optimizing a discrete and a continuous variable simultaneously, and the classic RL algorithm could lead to a slow convergence rate and bad performance; (ii) as the number of DUs and UEs increases (that is, the dimension of $\mathcal{X}$ and $\mathcal{P}$ increases), the required amount of parameters increases significantly (a.k.a. the curse of dimensionality [31]).

To solve the challenges, we propose a dual DDPG framework, ReLAx. The proposed algorithm consists of a pair of actors and a centralized critic where one actor handles the sub-band allocation problem, and the other deals with the power allocation problem. However, the two variables are not completely independent of each other, and there could be some level of overlap in the models' parameter spaces. Thus, in ReLAx, we adopt the multi-task training concept so that the actors can share parameters to reduce the number of parameters needed to be trained. To illustrate, in classic multi-task learning [32], tasks share a common model but with different output layers. In this way, the model can learn the correlation between variables and improve its performance. Fig. 3 shows the computational diagram of the proposed framework where the red dashed lines indicate the direction of backpropagation and the blue dashed line represents the feedforward process in the agent update. We can see that the state for the two actors is the same, and they are supposed to make decisions on different aspects given the state.

The update rules for the actors follow the DDPG update rule. However, the variable $\mathcal{X}$ is discrete, and DDPG can handle only continuous output. To overcome this challenge, we apply the Gumble Softmax trick [33]. This trick can transform the continuous output from DDPG to discrete outputs in a differentiable way. Let θ^C and θ^D denote continuous and discrete actors, respectively. The gradient for the actors can be written as,

$$\nabla_\theta J(\theta^C) = \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_\theta \mu_{\theta^C}(s_t)\nabla_a Q^\mu(s_t, a_t^C)|_{a_t=\mu_{\theta^C}(s_t)}\right] \tag{13}$$

$$\nabla_\theta J(\theta^D) = \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_\theta \mu_{\theta^D}(s_t)\nabla_a Q^\mu(s, g(a_t^D))|_{a_t=\mu_{\theta^D}(s_t)}\right], \tag{14}$$

where a_t^C and a_t^D stand for the continuous and discrete actions, i.e., UE association and power allocation, and $g(\cdot)$ represents the Gumbel Softmax. To optimize the critic, we need information from both actors at the same time. The intuition behind this is that, if we regard these two actors as separate agents, this problem becomes a Multi-Agent Reinforcement Learning (MARL). If the critic can only observe one agent at a time, the whole environment will become dynamic and difficult to optimize, and a centralized critic can solve this problem [34]. The loss function for the critic can be written as,

$$\begin{aligned}\mathcal{L}(\theta^Q) &= \mathbb{E}_{s_t, a_t^C, a_t^D, r_t, s_{t+1}}\left[\left(Q^*(s_t, a_t^C, a_t^D|\theta^Q) - y\right)^2\right] \\ y &= r(s_t, a_t^C, a_t^D) + \gamma \max_{a_{t+1}^C, a_{t+1}^D} \bar{Q}^*(s_{t+1}, a_{t+1}^C, a_{t+1}^D).\end{aligned} \tag{15}$$

IV. Performance Evaluation

We first discuss the experimental settings and results of the Next-G testbed, then evaluate ReLAx's performance in resource allocation through numerical simulations.

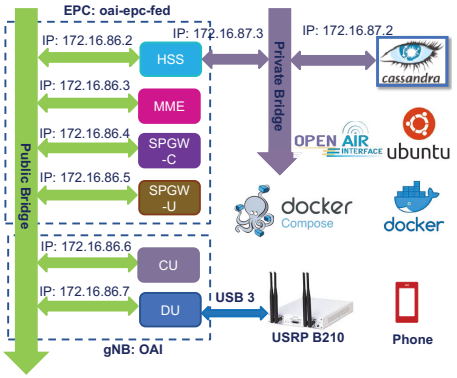


Fig. 4. Logical illustration of the fully containerized-based Next-G testbed.

A. Testbed Experiment

We present our testbed in this paper to support our resource allocation model in Sect. II, particularly by modeling the computational power consumption in the CU and DU.

Next-G RAN Testbed Architecture. We have utilized an open-source project, OAI [18] to construct our experimental prototype. OAI has been extensively tested and validated to be fully compatible with the 5G protocol stack for the gNB and UE, enabling the end-to-end deployment of a 5G network [18]. As illustrated in Fig. 4, we have implemented a RAN consisting of two containers, CU and DU, deployed using *Docker*, an open-source platform that automates the deployment, scaling and management of applications and services in containers. Furthermore, we have used oai-epc-fed, an implementation of the 3GPP specifications for Evolved Packet Core (EPC) networks, to implement the core network. The oai-epc-fed comprises the following network elements: Mobility Management Entity (MME), Home Subscription Server (HSS), and Packet Gateway and Service Gateway (SPGW-C-U). All components of *oai-epc-fed* have been deployed in containers. Besides, we have utilized Software Defined Radio (SDR) boards, specifically the Ettus USRP B210, which covers a frequency range of 70 MHz to 6 GHz and supports 2×2 MIMO with a maximum sample rate of 62 MS/s. All containers are executed using *Docker Compose* [35], which is a tool for defining and running multi-container Docker applications, and a YAML file has been created to specify the configuration of these containers. All containers are hosted in a workstation tower with an Intel Xeon E5-1650 processor, which features 12 cores running at a clock speed of 3.5 GHz, and 32 GB of RAM. As for the UE, we utilized a Samsung Galaxy S9 smartphone running on the Android 10 operating system. For network configuration, our Next-G RAN prototype is run with three functional split options: Option F1 (PDCP/RLC, as specified in Option 2 of the 3GPP TR 38.801 standard), Option IF4.5 (Lower PHY/Higher PHY, also known as Option 7.x in the 3GPP TR 38.801 standard) and Option LTE eNB.

Latency and Throughput Cost. To measure latency on the Next-G testbed, we record the Round-Trip Time (RTT) values while sending downlink traffic between the EPC container and the UE. Fig. 5(a) describes the relationship between RTT and

TABLE II
MEASURED THROUGHPUT FOR DOWNLINK.

No. PRB	F1 Split Mbit/s	IF4.5 Split Mbit/s	eNB Mbit/s
25	17.7	17.6	17.3
50	35.7	35.5	35.1
100	73.8	73.6	72.2

packet size when CU/DU is running on functional split Option F1. In each iteration, we send 500 Internet Control Message Protocol (ICMP) echo request packets from the SPGW-U to the UE. It can be seen that the RTT linearly increases as the packet size increases. To determine the maximum downlink throughput on the Next-G testbed, we use the User Datagram Protocol (UDP) as the transport protocol between the SPGW-U container and the UE. Specifically, we use *iPerf3* tool to generate the downlink UDP traffic between the UE and the testbed for a fixed duration of 120 seconds. In Tab. II, we report the maximum throughput of the three network configurations for different PRB values. We observe that the throughput value increases linearly with the number of PRBs for the three functional split options. However, Option F1 slightly outperforms compared to Option eNB and Option IF4.5. That is because Option F1 has less latency and bandwidth cost compared to the others. In general, the F1 functional split typically includes only the layer of the Packet Data Convergence Protocol (PDCP) and Radio Link Control (RLC) layers, which are both located closer to the user data compared to the other functional split options. This proximity to the user data can result in lower latency by reducing the number of protocol processing steps required before the data is transmitted to the end user.

Impact Functional Splits on CPU Utilization. To understand the CU-DU CPU power consumption in relation to UEs' traffic requests in the Next-G system, as discussed in Sect. II-D, we aim to study the correlation between functional split options and CPU usage at the CU and DU. In this experiment, we measure the percentage of CPU utilization using the *docker stats* command in Ubuntu, which provides real-time data on the performance of the containers running. We conducted the experiment by repeatedly sending downlink UDP traffic from the SPGW-U to the UE with varying PRB values in two functional split configurations, Option F1 and Option IF4.5, and recorded the CPU utilization percentage during the process. The percentage of CPU utilization has been measured for the split options F1 and IF4.5 as shown in Fig. 5(b) and Fig. 5(c), respectively. One of the key elements of Fig. 5(b) is that the CPU utilization of DU is reduced by 25.5% when we move from PRB 100 to PRB 50. However, lower CPU reduction, which is 2.7%, when moving from PRB 100 to PRB 50 in CU. The reason CPU is consumed higher in DU than in CU, in Option F1, is that the higher PHY operations such as RLC/MAC, L1/high, tx precode, rx combine, and L1/low operations reside in DU for split Option F1, while CU has only PDCP and RCC operations. However, in Fig. 5(c), the trend of CPU consumption is different from Option F1. It can be observed that the highest CPU consumption occurred in CU, while CU CPU consumption is reduced by 14.9% when we move from PRB 100 to PRB 50. In general, power usage in the Next-G system can be significantly minimized if we adapt the computational CU/DU resources, such as CPU cycles per second [14].

TABLE III
VALUE OF ρ FOR DIFFERENT INTERVAL TIME T.

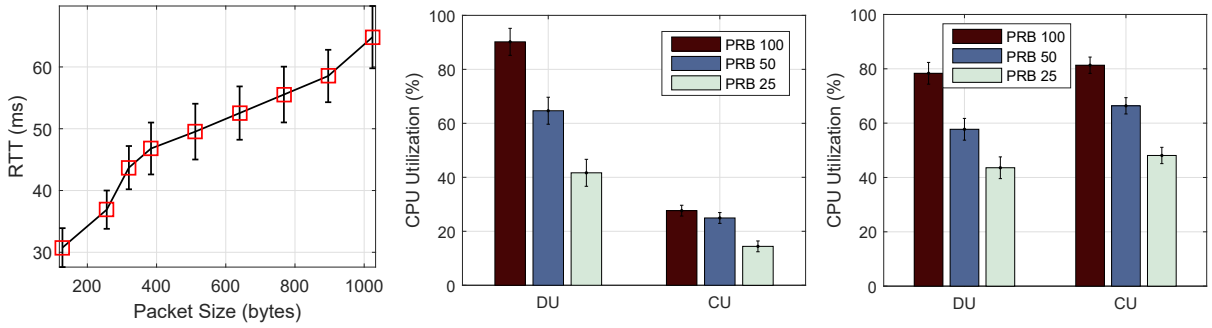
T (ms)	1	10	100	1000	10000
ρ	1.00	0.90	0.22	0.07	0.02

Remark: *Based on experimental results in Figs. 5(b), and 5(c), we can conclude that the power factor parameters α_j and β_j, in* (8) *and* (9) *respectively, primarily depend on the Next-G RAN configurations, such as the number of PRBs and functional split options. The value of α_j increases as we move from Option 1 to Option 8. However, the value of β_j decreases as we move from Option 1 to Option 8. Specifically, the scenario where $\alpha_j > \beta_j$ occurs when the CPU consumption in the CU is higher than in the DU, as shown in Fig. 5(c). On the other hand, the scenario where $\alpha_j \leq \beta_j$ is estimated.*

B. Numerical Simulations

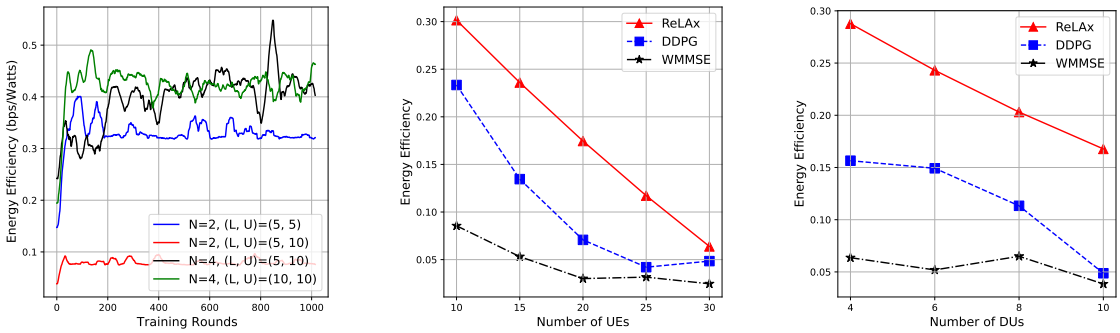
The simulation results are mentioned here to evaluate the performance of our proposed algorithm. The simulations are conducted using Python and the Pytorch toolkit. ReLAx has an actor with three in-between Fully-Connected (FC) layers of size 64, 128, and 128, as well as two separate FC output layers to handle both discrete and continuous actions. Additionally, ReLAx includes two critics that have a similar architecture, processing the state through an FC layer of size 64 before concatenating it with the action. The concatenated will be processed by an FC layer of size 128. The nonlinear function used in the network is the Rectified Linear Unit (ReLU), and the learning rates used for the actor and critic are 0.01 and 0.05, respectively. Our proposed solution for the joint UE association and transmit power allocation problem is compared against two popular approaches; DDPG and WMMSE. The DDPG method is a classic reinforcement learning approach with an actor-critic structure. In contrast, the WMMSE method, which has been adapted to optimize the power control problem, is an optimization technique [10].

Convergence of the ReLAx Algorithm. To show that the proposed framework is capable of achieving good performance in the resource allocation scenario, in Fig. 6 (a), the reward (that is, the EE defined in (10)) is displayed against training rounds. We observe three comparisons; (i) for the first two curves, we observe that, with the same number of sub-bands and DUs, the one with 5 UEs performs better than the one with 10 UEs. The result indicates that, under the same conditions, the more UEs there are, the worse the performance of the model; (ii) for the second and third scenarios, we can see that, with the same number of DUs and UEs, the model with 4 subbands outperforms that with 2 subbands. The result shows that more sub-bands can improve the model's performance; (iii) for the last two scenarios, we can observe that, with the same amount of UEs and sub-bands, the two models achieve similar performance, which is because the number of DUs



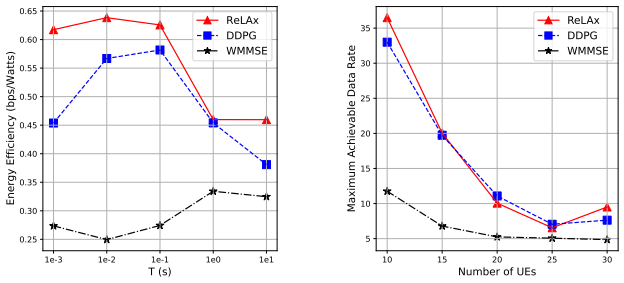
(a) (b) (c)

Fig. 5. Fully containerized Next-G testbed experimental results for different configurations; (a) RTT measurement for different packet sizes; (b) CPU utilization of functional split Option F1 for downlink traffic; (c) CPU utilization of functional split Option IF4.5 for downlink traffic.



(a) (b) (c)

Fig. 6. The network EE versus (a) Training rounds; (b) The number of UEs; and (c) The number of DUs.



(a) (b)

Fig. 7. (a) The network EE versus the time duration between two successive channel estimations; and (b) The maximum achievable data rate compared to the number of UEs.

is sufficient in both scenarios and, therefore, the two models converge to a similar level.

Impact of the Number of UEs. Fig. 6(b) shows the ReLAx approach compared to other different methods versus the number of UEs. We can observe that when the amount of UEs increases, the performance of all three models degrades because the dimension of action spaces has increased. Furthermore, WMMSE and DDPG have similar bad performances. For WMMSE, the reason for this poor performance is that it only optimizes the set of the power variable $\mathcal{P}$ and, thus, when the number of UEs increases, it cannot assign links to good sub-bands. As for DDPG, the action space becomes too large for it to learn a good representation, and therefore, its performance is much worse than ReLAx. 5 DUs and 5 subbands are used with $T = 0.01$ s.

Impact of the Number of DUs Fig. 6(c) shows ReLAx's performance when varying the number of DUs compared to the other two methods. We can observe that the energy efficiency is not significantly impacted by the number of DUs except for a small drop when the number of DUs. The drop could be caused by the static power consumption of the DU and CU because the improvement in the data rate cannot compensate for the increase in power consumption. 3 subbands and 20 UEs are used with $T = 0.01$ s.

Time processing of the ReLAx algorithm. T is the time interval between two channel estimations, and the greater it is, the more the channels change. From Fig. 7(a), we observe that while increasing T, the performance of all three models deteriorates, which is expected because, from an RL point of view, the environment changes more significantly between two steps with a higher T, so the RL/WMMSE agent cannot depend on the knowledge it learned in the previous steps. The value of the channel change factor ρ is shown in Tab. III. We can see that when T changes from 0.001s to 0.01s, the channel is relatively steady, and when T reaches 0.1s, the value of ρ becomes 0.22, which indicates that the channel becomes very difficult to estimate. Due to this phenomenon, ReLAx drops rapidly after $T = 1 \times 10^{-2}$ s because the decision it makes

is to maximize the reward in the environment from the last step. Conversely, the other two models' performances are flat compared to ReLAx's, and this is because they do not learn much from interacting with the environment, and thus their actions are somehow random. Due to this randomness, their performance tends to be flat no matter how the environment changes. 5 DUs, 20 UEs, and 3 subbands are used.

Impact of Number of UEs on Maximum Achievable Data Rate. In Fig. 7(b), we show the results of the maximum achievable data rate against the number of UEs. As the number of UEs increases, ReLAx, DDPG, and WMMSE show a performance drop, which is expected because the network is getting more and more crowded. In addition, ReLAx and DDPG have achieved similar maximum achievable data rates but have a significant performance difference in terms of energy efficiency (see Fig. 6(b)). This shows that ReLAx can do a better job in power allocation than DDPG according to (10). 5 DUs and 3 subbands are used.

V. Conclusion

We studied the problem of maximizing network Energy Efficiency (EE) in Next Generation Radio Access Network (Next-G RAN) while considering practical constraints like Quality of Service (QoS) requirements, transmit power, and fronthaul capacity. Based on real-world data collected from a programmable, real-time Next-G testbed, we established conclusions to better understand the network power consumption model in the Next-G system. The proposed optimization problem is classified as a Mixed-Integer Non-Linear Programming (MINLP) problem, which is, in general, non-convex and NP-complete. Therefore, we proposed a Deep Reinforcement Learning (DRL)-based algorithm—the modified dual DDPG method, named ReLAx. Simulation results coupled with real-time experiments on a fully containerized Next-G testbed show that the proposed approach solution outperforms competing algorithms, such as DDPG and WMMSE.

Acknowledgment. This work was supported by the US National Science Foundation under Grant No. ECCS-2030101.

References

[1] Cisco, "Cisco Annual Internet Report (2018–2023)," *White paper*, 2020.
[2] ETSI, "NG-RAN; Architecture description," *3GPP TS 38.401 Ver. 15.2.0 Rel. 15*, 2018.
[3] L. Chettri and R. Bera, "A comprehensive survey on internet of things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, 2019.
[4] X. You, C.-X. Wang, J. Huang, X. Gao, Z. Zhang, M. Wang, Y. Huang, C. Zhang, Y. Jiang, J. Wang, *et al.*, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China Information Sci.*, vol. 64, pp. 1–74, 2021.
[5] C. Sun, Y. Zhou, G. Jung, T. X. Tran, and D. Pompili, "CaRL: Cascade reinforcement learning with state space splitting for O-RAN based traffic steering," *arXiv:2312.01970*, 2023.
[6] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3133–3174, 2019.
[7] C. Sun, S. Huang, and D. Pompili, "HMAAC: Hierarchical multi-agent actor-critic for aerial search with explicit coordination modeling," in *Proc. IEEE ICRA*, pp. 7728–7734, 2023.
[8] V. Sadhu, C. Sun, A. Karimian, R. Tron, and D. Pompili, "AerialDeepSearch: Distributed multi-agent deep reinforcement learning for search missions," in *Proc. IEEE MASS*, pp. 165–173, 2020.
[9] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv:1509.02971*, 2015.
[10] A. Younis, T. Tran, and D. Pompili, "Energy-efficient resource allocation in C-RANs with capacity-limited fronthaul," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 473– 487, 2021.
[11] F. Fang, Z. Ding, W. Liang, and H. Zhang, "Optimal energy efficient power allocation with user fairness for uplink MC-NOMA systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1133–1136, 2019.
[12] K. Wang, W. Zhou, and S. Mao, "On joint BBU/RRH resource allocation in heterogeneous cloud-RANs," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 749–759, 2017.
[13] X. Huang, W. Fan, Q. Chen, and J. Zhang, "Energy-efficient resource allocation in fog computing networks with the candidate mechanism," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8502–8512, 2020.
[14] A. Younis, T. X. Tran, and D. Pompili, "Bandwidth and energy-aware resource allocation for cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6487–6500, 2018.
[15] F. Meng, P. Chen, and L. Wu, "Power allocation in multi-user cellular networks with deep Q learning approach," in *Proc. IEEE ICC*, pp. 1–6, 2019.
[16] X. Liao, J. Shi, Z. Li, L. Zhang, and B. Xia, "A model-driven deep reinforcement learning heuristic algorithm for resource allocation in ultra-dense cellular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 983–997, 2019.
[17] X. Wang, Y. Zhang, R. Shen, Y. Xu, and F.-C. Zheng, "DRL-based energy-efficient resource allocation frameworks for uplink NOMA systems," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7279–7294, 2020.
[18] "OpenAirInterface." http://www.openairinterface.org/, 2024.
[19] 3GPP TR 38.801, "Study of new radio access technology: Radio access architecture and interfaces," *Release 14*, 2017.
[20] A. U. T. Yajima, T. Uchino, and S. Okuyama, "Overview of O-RAN Fronthaul Specifications," *NTT Docomo Tech. J.*, vol. 21, no. 1, 2019.
[21] W. Erik, "4G/5G RAN architecture: How to a split can make the difference," *Ericsson Technol. Rev*, vol. 93, no. 6, pp. 1–15, 2016.
[22] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li, "Spectrum and power allocation for vehicular communications with delayed CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 458–461, 2017.
[23] J.-i. Kani, J. Terada, K.-I. Suzuki, and A. Otaka, "Solutions for future mobile fronthaul and access-network convergence," *IEEE J. Lightwave Technol.*, vol. 35, no. 3, pp. 527–534, 2016.
[24] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 146–172, 2019.
[25] M. Khan, R. S. Alhumaima, and H. S. Al-Raweshidy, "Reducing energy consumption by dynamic resource allocation in C-RAN," in *Proc. IEEE EuCNC*, pp. 169–174, 2015.
[26] Y. Zhou, J. Li, Y. Shi, and V. W. Wong, "Flexible functional split design for downlink C-RAN with capacity-constrained fronthaul," *IEEE Trans. Veh. Technol*, vol. 68, no. 6, pp. 6050–6063, 2019.
[27] V. N. Ha, L. B. Le, *et al.*, "Coordinated multipoint transmission design for cloud-RANs with limited fronthaul capacity constraints," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7432–7447, 2015.
[28] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys Operat. Research Manage. Sci.*, vol. 17, no. 2, pp. 97–106, 2012.
[29] Z. Zhou, M. Dong, K. Ota, J. Wu, and T. Sato, "Energy efficiency and spectral efficiency tradeoff in device-to-device (D2D) communications," *IEEE Wireless Commun. Lett.*, vol. 3, no. 5, pp. 485–488, 2014.
[30] E. D. Andersen and K. D. Andersen, "Presolving in linear programming," *Springer Math. Programming*, vol. 71, no. 2, pp. 221–245, 1995.
[31] R. E. Bellman, *Adaptive control processes: A guided tour*. Princeton U., 2015.
[32] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv:1706.05098*, 2017.
[33] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv:1611.01144*, 2016.
[34] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv:1706.02275*, 2017.
[35] "Docker Compose." https://docs.docker.com/compose/, 2024.