IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 June 2023, accepted 7 July 2023, date of publication 19 July 2023, date of current version 27 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296851

RESEARCH ARTICLE

Deep Reinforcement Learning Based Resource Allocation for Network Slicing With Massive MIMO

DANDAN YAN, (Student Member, IEEE), BENJAMIN K. NG, (Senior Member, IEEE), WEI KE, (Member, IEEE), AND CHAN-TONG LAM, (Senior Member, IEEE)
Faculty of Applied Sciences, Macao Polytechnic University, Macau SAR, China

Corresponding author: Benjamin K. Ng (bng@mpu.edu.mo)

This work was supported by the Science and Technology Development Fund, Macau SAR, under Grant 0044/2022/A1.

ABSTRACT Network slicing is a critical technology for fifth-generation (5G) networks, owing to its merits in meeting the diversified requirements of users. Effective resource allocation for network slicing in Radio Access Networks (RAN) is still challenging owing to dynamic service requirements. Therein, automatic resource allocation based on environmental changes is of significant importance for network slicing. In this study, we used deep reinforcement learning (DRL) to allocate resources for network slicing in a RAN with the aid of massive multiple-input multiple-output (MIMO). The DRL agent interacts with the environment to execute autonomous resource allocation. We considered a two-level scheduling framework that aims to maximize the quality of experience (QoE) and spectrum efficiency (SE) of slices. The proposed algorithm can find a near-optimal solution. We used the standard DRL advantage actor-critic (A2C) algorithm to implement upper-level inter-slice bandwidth resource allocation that considers service traffic dynamics in a large timescale. Lower-level scheduling is a mixed-integer stochastic optimization problem with several constraints. We combined the proportional fair scheduling algorithm and the water filling algorithm to perform resource block (RB) and power allocation in a small timescale. The results show that the QoE and SE of all slices using the A2C algorithm achieved a significant performance improvement over the other algorithms. The efficiency of the proposed method was supported by the simulation results.

INDEX TERMS Network slicing, resource allocation, radio access networks (RAN), massive MIMO, advantage actor critic (A2C).

I. INTRODUCTION

There has been exponential growth in the amount of data transfer as an increasing number of devices are connected to wireless networks. The user experience is greatly affected by sharing a limited bandwidth over an extremely narrow spectrum. The existing fourth-generation (4G) network cannot meet the needs of a large amount of data communications within limited frequency bands. Mobile networks have mostly supported mobile devices from the early days of the third generation (3G) to the current 4G network. However, in the fifth generation (5G) era, mobile networks must provide services to devices with a variety of application types and quality of service (QoS) criteria. With the help of network slicing, many types can operate flexibly on the same physical infrastructure [1]. Various applications must meet QoS requirements, such as enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (uRLLC), and massive machine type communications (mMTC) [2]. Network slicing includes a range of network resource requirements, including those for computing and communication, as well as a range of performance characteristics, including latency and throughput. Because 5G cellular networks are expected to provide end users with faster data speeds and lower end-to-end latency, managing network resources in slices becomes

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

more difficult [3], [4], [5]. Owing to the scarcity of spectrum resources and the strict dynamic requirements of slice users, radio access networks (RAN) slicing presents more difficult technical challenges for real-time resource management. In this regard, automation is required to distribute resources to slices because it is difficult to manually regulate resource allocation from the changing state of slices using traditional mathematical model-based approaches. Artificial intelligence (AI) is useful for allocating resources for autonomous control because it can choose an appropriate control strategy based on the knowledge it has gained from analyzing previous data. AI technology, deep reinforcement learning (DRL), a model-free methodology, has been applied to allocate resources for network slicing with autonomous control. By allocating resources to slice the network using DRL algorithms, the performance was shown to be better than those of current state-of-the-art solutions in terms of QoS satisfaction and resource utilization [6]. Compared with existing systems, [7] showed that resource slicing based on DRL increases resource consumption, slice satisfaction, and throughput benefits. Reference [8] coupled slice Admission Control (AC) and network slicing problems using a multi-agent DRL. The suggested methodology can bring in up to 29.96% more revenue than methods based on heuristics methods. Additionally, their findings show that multi-agent DRL accelerates convergence and generates 8.62% more long-term infrastructure provider (InP) revenue than a single-agent DRL strategy. Reference [9] proposed a hierarchical deep learning framework that integrated punctured and orthogonal scheduling algorithms for a resource slice in a RAN. For the eMBB and URLLC slices, they improved the average service level agreement (SLA) satisfaction ratio (SSR), and for eMBB users, they increased the average aggregate throughput. By dynamically deploying suitable virtual base stations and distributing subchannels and power to each user of each slice, [10] proposed an efficient multi-agent DRL algorithm to jointly optimize three types of RAN slices according to throughput, average latency, and average interference plus noise power ratio (SINR). The suggested plan performed better than the alternatives did.

In view of the above results of DRL, this study designs an intelligent RAN slicing resource allocation strategy that can adapt to changing network conditions over time with a variety of time and resource granularities to meet the strict and specific slice criteria of the user.

II. RELATED WORK

The DRL agent obtains the status of the environment by automatically interacting with it. Based on the present condition, the agent chooses an action and performs it. The agent and the environment move on to the next state as a result of the system emitting a feedback value that rewards or punishes the action. Reinforcement learning has been used to allocate network slicing resources. References [11], [12], [13], [14], and [15] aimed to determine the best inter-slice bandwidth allocation that optimizes the utility of the system. System utility is related to the quality of experience (QoE) and spectrum efficiency (SE). They view fluctuating service requests as environmental states and resource allocation as environmental actions. Reference [11] leverages a deep Q network (DQN) to resolve resource management problems for network slicing scenarios. The convergence is not very good, and 50,000 updates are required to achieve stable performance. Reference [12] embedded long short-term memory (LSTM) into the actor-critic algorithm (A2C) to monitor user mobility and increase the utility of the system. The LSTM network is responsible for predicting the current state, which is combined with service requests from the past T states. A discrete normalized advantage function (DNAF) is added to traditional Deep Q-Learning (DQL) in [13] to speed convergence in a broad action space. They also incorporated a k-nearest-neighbor technique into the DQL to quickly find an action in the discrete space that is closest to the deterministic policy gradient descent (DPGD) outcome, since DPGD only works in a continuous action space. The proposed scheme converges quickly in a larger action space. It is possible to avoid calculating the Q value for each state-action combination by using the DPGD technique to break the Q value function into a state value function term and an advantage term. To approximate the action-value distribution, [14] used a deep distributional Q network (DDQN) driven by a generative adversarial network (GAN) and implemented the reward-clipping strategy to boost the stability of the GAN-training DDQN. To acquire the training problem with the built-in DDQN in GAN, they used Dueling GAN-DDQN to extract the state-value distribution and the action advantage function from the action-value distribution. Reference [15] proposed an intelligent resource management technique for network slicing and combined a graph attention network (GAT) with DQN.

In summary, although in the literature [11], [12], [13], [14], [15] various DRL methods had been proposed to implement inter-slice bandwidth allocation, they only considered the number of resource blocks allocated to each user without considering the specific resource block power. They leveraged round-robin scheduling methods within each slice which may be unfair for users with different channel conditions. Furthermore, the typical Shannon capacity in real-world applications does not account for users' delayed QoS requirements. As a result, it is not suitable to assess the effectiveness of delay-sensitive services [16]. For delay-sensitive network slicing services, Shannon formulation-based network capacity analysis may not be the best option. Therefore, to evaluate delay-sensitive applications that can provide statistical delay provisioning by ensuring a low probability of packet transmission delays, we use short packet transmission.

The technical contributions of this paper can be summarized as follows.

- For the user-delay requirement of the slice, we considered the short packet transmission, which provides

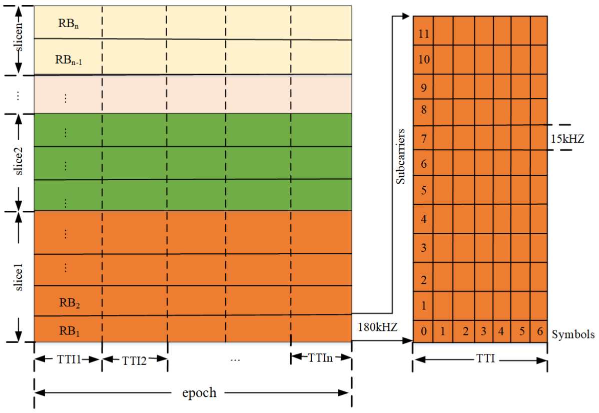


FIGURE 1. Illustration of the epoch, TTI, and PRB.

statistical delay provisioning approximated by finite block length theory and improves the QoE.

- We applied massive multiple-input multiple-output (MIMO) to improve the system capacity, which is studied assuming short packet communication.
- We formulated a joint QoE and SE optimization problem using a two-level scheduling strategy. In upper-level scheduling, we use the A2C algorithm to schedule inter-slice resources. In lower-level scheduling, resource block (RB) assignment and power allocation (RAPA) are jointly optimized for intra-slices by proportional fair (PF) scheduling and water filling (WF) algorithms. A corresponding resource block is allocated to each user, and power is allocated to each resource block.
- The proposed scheme exhibits faster convergence with lower computational complexity.

The rest of this paper is structured as follows. In section III, the proposed system model is introduced. It formulates the problem objectively and provides the proposed RAN resource-slicing control approach. The precise implementation plan and algorithm are outlined in section IV. We present a system-level simulation result and an analysis of the proposed scheme in section V. Finally, section VI concludes the paper.

In this study, matrices, vectors, and scalar quantities are denoted by bold-faced upper case letters, bold-faced lower case letters, and light-faced lower case letters, respectively. The notations used in the system model are summarized in Table 1.

III. SYSTEM MODEL

A. SIGNAL TRANSMISSION MODEL

We focus on a multi-user radio access network with a single base station (BS) that constitutes a wireless MIMO communication system. The transmitter is equipped with M antennas. The receiver is equipped with Q antennas. We set $M \geqslant Q$ in this study. We implemented inter-slice bandwidth allocation at each epoch and intra-slice resource block scheduling at each Transmission Time Interval (TTI). Each epoch is divided into several TTI ($\varDelta T$-TTI), each TTI is 0.5ms long and indexed by $t \in \mathcal{T} = \{1, 2, \ldots T\}$. Each TTI divides the bandwidth into a number of physical resource blocks (PRBs), designated by the symbol $j \in \mathcal{F} = \{1, 2, \ldots, F\}$. There exist a set of slices $n \in \mathcal{N} = \{1, 2, \ldots, N\}$ and a set of users (or user equipment) $i \in \mathcal{U} = \{1, 2, \ldots U\}$. A schematic of this process is shown in Figure 1. Each RB consisted of 12 subcarriers, each of which has seven Orthogonal Frequency Division Multiplexing (OFDM) symbols and a subcarrier spacing of 15 kHz. As a result, each RB has a 180 kHz bandwidth. The user equipment (UE) in this scenario moves within each epoch to generate different distributions, and the slice user shares the same movement model. The movement pattern refers to the user moving randomly around at a certain speed, the direction of movement obeys a uniform distribution within $[-180^\circ, 180^\circ]$. The set $\mathcal{U}_n$ is specifically referred to as the UE connected to the slice $n \in N$. The data rate at which the i-th user acquired the j-th physical resource block (PRB) at the t-th TTI, is determined as follows [17]:

$$r_{i,j,t} = B \cdot \log_2 \det \left(\mathbf{I}_Q + \frac{\gamma}{M} \mathbf{H}_{i,j,t} \mathbf{H}_{i,j,t}{}^{*} \right), \tag{1}$$

where $\det(\mathbf{A})$ denotes the determinant of matrix, $\mathbf{A}$. $\mathbf{I}_\mathrm{Q}$ denotes the identity matrix of dimension $Q \times Q$, and * denotes the transpose conjugate operator. Also $\gamma = \frac{\rho P_{i,j,t}}{\sigma^2}$, where the transmit power on the j-th PRB of the t-th TTI is denoted by $P_{i,j,t}$, ρ denotes the path loss, the shadowing effect between the UE and BS, and σ^2 is the power of additive white Gaussian noise (AWGN). $\mathbf{H}_{i,j,t} \in \mathbb{C}^{\mathrm{Q} \times \mathrm{M}}$ models the small-scale fading coefficients of the j-th PRB, based on the rayleigh fast fading channel [18]. The bandwidth of the PRB is B.

One unique aspect of the new uRLLC service, as opposed to conventional services, is the short packet transmission. The data rate of the short packet transmission cannot be accurately described by Shannon's capacity theory. Alternatively, finite block length theory can be used to approximate the possible data rate for short packet transmission [19]. The maximal rate achievable with MIMO transmission given the error probability ϵ and block length $\mathfrak{n}$ of the packet is closely approximated by

$$r_{i,j,t} = B \cdot \left\{ \mathrm{C}(\gamma) - \sqrt{\frac{V(\gamma)}{\mathfrak{n}}} \frac{Q^{-1}(\epsilon)}{In2} \right\}, \tag{2}$$

where $\mathrm{C}(\gamma) = \log_2 \det \left(\mathbf{I}_Q + \frac{\gamma}{M} \mathbf{H}_{i,j,t} \mathbf{H}_{i,j,t}{}^{*} \right)$ is the capacity, $\mathrm{V}(\gamma)$ denotes the channel dispersion [17], [20]

$$\mathrm{V}(\gamma) = Q - \sum_{q=1}^{Q} \left(1 + \frac{\gamma}{M} \cdot \lambda_{j,q} \right)^{-2}, \tag{3}$$

where $\lambda_{j,q}$ denotes the q-th unordered eigenvalue of $\mathbf{H}_{i,j,t} \mathbf{H}_{i,j,t}{}^{*}$. $Q^{-1}(\bullet)$ denotes the inverse of the Gaussian cumulative distribution function [21], [22] given by

$$Q^{-1}(\epsilon) = \sup\{x \in \mathbb{R}, \mathrm{Q}(x) \leq \epsilon, 0 < \epsilon < 1\}, \tag{4}$$

$$\mathrm{Q}(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{(-\frac{t^2}{2})} dt. \tag{5}$$

TABLE 1. Notation summary of system model.

Notation	Description
$\mathcal{U}$	The user set
$\mathcal{N}$	The slice set
T	The Transmission Time Interval set
f	The Physical resource block set
M	Number of transmission antennas
Q	Number of receiving antennas
$r_{i,j,t}$	The data rate of the i-th user acquired the j-th PRB at the t-th TTI
$r_{i,t}$	The i-th UE's instantaneous data rate at the t-th TTI
$a_{i,j,t}$	Resource block allocation binary indicator
B	The bandwidth of a PRB
ϵ	Transmission error probability
$\mathfrak{n}$	Transmission block length
$l_{i,t}$	The i-th user's queue length in the t-th TTI
L_i	Packet size
$A_{i,t}$	The instantaneous packet arrival (in bits) for user i during the t-th TTI
$\mathcal{U}_n^{actv}$	Activated users of slice n
Q_i	The data packet of the i-th user
$D_{i,m}$	The transmission delay of the m-th packet of the i-th user
$D_{n,\max}$	The threshold of the data packet transmission delay of slice n
$R_{n,\min}$	The threshold of transmission rate of slice n

Consequently, the instantaneous data rate for the i-th UE at the t-th TTI can be expressed as

$$r_{i,t} = \sum_{j=1}^{F} a_{i,j,t} r_{i,j,t}, \tag{6}$$

where the binary variable $a_{i,j,t} = 1$ indicates that the i-th UE obtains the j-th PRB in the t-th TTI; otherwise, $a_{i,j,t} = 0$. Additionally, only one UE per TTI may have a PRB allocated to it.

B. QUEUING MODEL OF UE

On the back-end server, it is assumed that each user has a data queue where incoming packets are held before being distributed following the first-come, first-served (FCFS) principle. The i-th user's queue length (expressed in packets) in the t-th TTI is shown and developed as follows [22], [23]:

$$l_{i,t+1} = \max\left\{l_{i,t} - r_{i,t}/L_i, 0\right\} + A_{i,t}. \tag{7}$$

L_i and $A_{i,t}$, denote overall packet size (in bits) and instantaneous packet arrival (in bits), respectively, for user i during the t-th TTI. Slice n's UEs that have a non-zero queue length $\mathcal{U}_n^{actv} = \left\{i \left| l_{i,t} > 0, i \in \mathcal{U}_n \right.\right\} \subseteq \mathcal{U}_n$ are referred to as being activated. We presume that the queue buffer has a limited capacity and that discarding packets due to buffer overflow is unavoidable. We introduce the queue length threshold in an effort to reduce the likelihood of a new arrival packet dropping.

C. ANALYSIS OF QUALITY OF EXPERIENCE (QoE)

The QoE of users served by slice n is represented by the transmission success rate of the data packets [24] and can be expressed as:

$$QoE_n = \frac{\sum_{i\in\mathcal{U}_n} \sum_{m\in Q_i} \nu_{i,m}}{\sum_{i\in\mathcal{U}_n} |Q_i|}, \tag{8}$$

where Q_i denotes the data packet for the i-th user. The total number of data packets sent to user i by the BS is denoted by $|Q_i|$. If the packets are successfully transferred, the binary variable $\nu_{i,m}$ indicates whether the service's rate and delay limits are satisfied.

$$\nu_{i,m} = \begin{cases} 1, r_{i,t} \geqslant R_{n,\min}, D_{i,m} \leqslant D_{n,\max} \\ 0, \text{otherwise} \end{cases}, \tag{9}$$

where $D_{i,m}$ denotes the transmission delay of the m-th packet of the i-th user. $D_{n,\max}$ and $R_{n,\min}$ are the thresholds for the data packet transmission delay and transmission rate, respectively, according to the n-th slice communication requirements.

IV. PROBLEM FORMULATION AND ALGORITHM DECISION

In this study, we calculated the network slice's overall utility, which is connected with the service's QoE performance and the network slice SE. The utility function of slice n in the k-th epoch is then given by:

$$U_{n,k} = \zeta \cdot U_{n,k}^{QoE}(w,d) + \mu \cdot U_{n,k}^{SE}(w,d). \tag{10}$$

The use of DRL for radio resource slicing bandwidth allocation is covered in this section. The ζ and μ are the weights of QoE and SE, respectively. The group of slices has varying demands $d = \left\{d_1, d_{2,}, \ldots, d_N\right\}$ and shares the total bandwidth W. To optimize the long-term reward expectations $\mathbb{E}\left\{R(w,d)\right\}$, the bandwidth sharing solution is $w = \{w_1, w_2, \ldots, w_N\}$, where the notation $\mathbb{E}(\cdot)$ denotes the expectation of the random argument in $(\cdot)$. The objective maximization can be formtated as:

$$P : arg_{\mathrm{W}} \max \left\{ U = \left[\sum_{k=0}^{\infty} \sum_{n\in\mathcal{N}} U_{n,k} \right] \right\} \tag{11}$$

$$\text{s.t. } C1 : w_1 + w_2 + \ldots + w_N = W \tag{11a}$$

$$C2: \sum_{i\in\mathcal{U}_n^{actv}}\sum_{j=1}^{F} a_{i,j,t}\cdot B \leqslant w_n, \forall n\in\mathcal{N} \tag{11b}$$

$$C3: a_{i,j,t}\in(0,1), \forall_i\in\mathcal{U}_n^{actv}, j\in\mathcal{F}, n\in\mathcal{N} \tag{11c}$$

$$C4: \sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{U}_n^{actv}} a_{i,j,t}\leqslant 1, \forall_j\in\mathcal{F} \tag{11d}$$

$$C5: \sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{U}_n^{actv}}\left(\sum_{j\in\mathcal{F}} a_{i,j,t}p_{i,j,t}\right)\leqslant P_{\max}. \tag{11e}$$

where constraint (11a) represents the total allocated bandwidth, which is equal to the total system bandwidth. (11b) states that the sum of users' bandwidths sharing the slice is less than or equal to the allocated bandwidth of that slice. (11c) represents the binary variable of the resource block allocation. (11d) represents the indicator of a resource block allocation of only one user in a TTI. (11e) represents the restricted power allocation. The optimization of the utility function in (11) is resolved by two-level scheduling. In the upper-level scheduling, we use the A2C algorithm to determine slice bandwidth within constraint (11a), which takes the (11a) as action and chooses from action space in (18). The lower-level scheduling determines the user bandwidth and power within the constraints (11b) to (11e). We calculated user rates based on user bandwidth and power, which is related to QoE and SE. Then we calculated the reward by SE and QoE. The reward was feedback to the A2C agent to update its parameters and thus influence the choice of action for the next epoch. Our approach to solve (11) using the A2C and mapping the RAN scenario to the context of Markov decision process (MDP) is in similar spirit as [12].

A. UPPER-LEVEL SCHEDULING BY A2C

1) OVERVIEW OF A2C

In reinforcement learning (RL) [11], an agent generates actions and manages the elements of the environment. S is a collection of states, and A is a collection of actions. The agent assesses the state $s\in S$ of the surrounding environment at each time step t. Regardless of whether the action $a\in A$ taken by the agent is advantageous or disadvantageous to the state, it is output to the state following its policy $\pi\left(a_t|s_t\right)$ and is rewarded r_t. The agent gains knowledge on how to boost prospective benefits as

$$R_t=\sum_{t=0}^{T}\Upsilon^t r_t, \tag{12}$$

where the discount factor Υ $(0\leqslant\Upsilon\leqslant 1)$ is a constant that deducts future rewards, and T represents how many time steps are still left in the training session or episode. (13) evaluates the future rewards that will be received. A drawback is that the reward fluctuates stochastically according to the state's initial value. Consequently, the predicted reward value following a state action is

$$Q\left(s_t,a_t\right)=E\left[R_t|s_t,a_t\right]=E\left[r_t+\Upsilon\max_{a\in A}Q\left(s_{t+1},a\right)\right]. \tag{13}$$

A2C method [12] employed a combination of policy and value-based techniques to reduce the variance of the reinforcement algorithm and the training agent more effectively and quickly. In the agent role, the actor executes the present action distribution while being judged by the critic. The advantage function is defined as follows:

$$A\left(s_t,a_t\right)=\underbrace{r_{t+1}+\Upsilon V\left(s_{t+1};\theta_c\right)}_{Q(s_t,a_t)}-V\left(s_t;\theta_c\right). \tag{14}$$

A2C maximizes the following policy objective:

$$L\left(\theta_a\right)=-\mathbb{E}_t\left[\log\pi_\theta\left(a_t\,|s_t\;;\theta_a\right)A\left(s_t,a_t\right)+\varphi H\left(\pi\left(s_t;\theta_a\right)\right)\right]. \tag{15}$$

The parameterization of a stochastic policy is given by π_θ. In an algorithm that alternates between sampling and optimization, $A\left(s_t,a_t\right)$ is an estimator of the advantage function at time-step t, and $\mathbb{E}_t\left[\bullet\right]$ is the expectation reflecting the empirical average across a small batch of data. $H\left(\pi\left(s_t:\theta_a\right)\right)$ is the entropy term used to favor exploration during the training process. The action entropy's weight is represented by the component φ.

The gradient of the actor is obtained by taking the gradient of objective $L\left(\theta_a\right)$. The actor θ_a and the critic θ_c parameters are updated as follows:

$$\theta_a=\theta_a+\nabla L\left(\theta_a\right), \tag{16}$$

$$\theta_c=\theta_c+A\left(s_t,a_t\right)\frac{\partial V\left(s_t;\theta_c\right)}{\partial\theta_c}. \tag{17}$$

2) PROPOSED A2C DEEP REINFORCEMENT LEARNING ALGORITHM

The proposed A2C network slicing resource allocation architecture is shown in Figure 2. The environment refers to the 5G network communication, in which multiple slices are produced based on the same physical infrastructure. We divided the scheduling procedure into two levels, with the intelligent agent responsible for adjusting the inter-slice bandwidth in accordance with the dynamics of service traffic over a long period. The agent first obtains state s_t from the wireless network environment and then takes s_t as input of the actor network. The actor network produces a slice bandwidth allocation action based on this policy. This action is fed back into the environment. Based on the agent's behavior, the lower-level controlling policy immediately executes radio resource scheduling in accordance with the dynamics of the physical layer over a brief period. After scheduling, the environment stepped into the next state, s_{t+1}, and we received a reward that was given to the agent. The critic network took the state s_t and the next state s_{t+1} as input to produce $V\left(s_t\right)$ and $V\left(s_{t+1}\right)$. The temporal difference (TD) error was calculated using the reward R, $V\left(s_t\right)$, and $V\left(s_{t+1}\right)$. The critic network's parameter θ_c is updated via TD error. The TD error and policy are used to update the actor-network parameter θ_a.

State: In our formulation, the state is defined as, $\boldsymbol{s}_t = \{d_1, d_{2,}, \dots, d_N\}$, where each observation vector represents the number of packets in each slice during the k-th epoch.

Action: Action is the quantity of bandwidth that the BS allots to each slice. The action space is defined as follows:

$$\boldsymbol{a} = \left[\left(w_{0,0}, w_{0,1}, w_{0,2}\right), \left(w_{1,0}, w_{1,1}, w_{1,2}\right), \dots \left(w_{p,0}, w_{p,1}, w_{p,2}\right)\right], \tag{18}$$

where$\left(w_{p,0}, w_{p,1}, w_{p,2}\right)$ denotes the bandwidth for three slices of the p-th action.

Reward: The reward, described in (19) [12], is given by:

$$r1 = \begin{cases} (Q_u - 0.7) \cdot 10, \text{ if } Q_v \geqslant 0.98\text{and } Q_e \geqslant 0.95 \\ -5, \text{ if } Q_v < 0.98\text{or } Q_e < 0.95 \end{cases}. \tag{19}$$

where Q_u, Q_v, and Q_e represent, respectively, the QoE of uRLLC, VoLTE, and eMBB. Once the specified service level agreement (SLA) requirements have been satisfied as (19), the agent is encouraged to modify its reward as $r2 = 4 + (SE - 200) * 0.1$ if $SE > 200$; otherwise $r2 = 4$.

The pseudocode of the upper level is shown in algorithm 1.

Algorithm 1 A2C-Based Upper-Level Scheduling Procedure

```
Input: $\{d_1, d_2, \dots, d_N\}$;
Input: QoE, SE, Utility;
Initial queue buffer, latency buffer, replay $D$, and user location.
Users move.
Users activate.
Obtain state $s = \{d_1, d_2, \dots d_N\}$ from environment.
for $i$ in range($\Psi$) do
    Choose action by state $s$.
    Map action to slice RB $\left(w_{p,1}, w_{p,2}, w_{p,3}\right)$.
    for $t$ in range($T$) do
        RBs allocation by algorithm 2, obtain user bandwidth, and power allocation by algorithm 3, obtain RBs power.
        if $t == T - 1$ then
            break.
        else
            Clear the queue of packets and delay cache.
        end if
    end for
    Calculate the reward utility, observe the next state $s_{t+1}$.
    Calculate temporal difference (TD) error by (14). Update parameter of actor network $\theta_a$ by (16) and critic network $\theta_c$ by (17) per learn steps.
    $s_t = s_{t+1}$.
    Parameter Reset.
    Users move.
    Users activate.
end for
return QoE, SE, Utility;
```

B. LOWER-LEVEL SCHEDULING

Following the selection of the slice bandwidth by the upper-level control policy during the k-th period, each TTI implements a radio resource allocation plan. Allocating PRB and transmitting power to the active UE in accordance with the lower-level strategy presents the next issue. We allocate the PRB to the user using the PF scheduling algorithm, which makes full use of the time-frequency characteristics of the channel to schedule the users with better channel conditions as much as possible and to schedule every user as much as possible. The number of RB allotted to each UE was determined by the PF scheduling method using the user priority set by the instantaneous available data rate over the average data rate. User priority is computed as $p = \frac{r_i(t)}{R_i(t)}$, and the average data rate [25] is defined as

$$\overline{R_{i,t}} = \left(1 - \frac{1}{t_c}\right)\overline{R_{i,t-1}} + \frac{1}{t_c} r_{i,t}. \tag{20}$$

It is important to note that the $\overline{R_{i,t}}$ value is updated for every TTI by using a weighted moving average technique and accounting for the real data quota. And $r_{i,t}$ is the data rate that the i-th user can reach during the prior subframe. According to the PF scheduling algorithm, a user cannot always be scheduled to communicate in a multi-user cell. This is because if the user is communicating all the time, it can cause $\overline{R_{i,t}}$ to continue increasing. As shown by the priority formula of the user PF algorithm, the priority also decreases.

A higher priority was given to users near the base station. Owing to bad channel conditions, UEs farther from the base station have a lower priority and cannot be scheduled for a long period. The average data rate is reduced. The priority of users increases again. Then t_c denotes the update time window parameters, which affect the average rate and contain information about how long the channel conditions have been in the past. When t_c is larger, the user rate is averaged over the rate values in more slots, and the long-term fairness is better. In contrast, when t_c is smaller, the user average rate only refers to the rate value at the latest slot, which leads to an abrupt change in the user channel and affects the scheduling result of the scheduling algorithm more easily.

The PF scheduling algorithm determines the number of RBs required for each user. We can then map the RB index to each user. We then need to allocate power to each PRB. To distribute power and obtain the best throughput, we commonly used the water-filling algorithm, a well-known information theory procedure. The goal of water filling is to benefit from better channel conditions, as these enable the faster and more efficient transmission of more power and data. When the quality of the channel decreases, less power and a lower rate are transmitted over the channel. The channel does not require more power and a faster data rate is achieved if the instantaneous channel signal-to-noise ratio(SNR) is

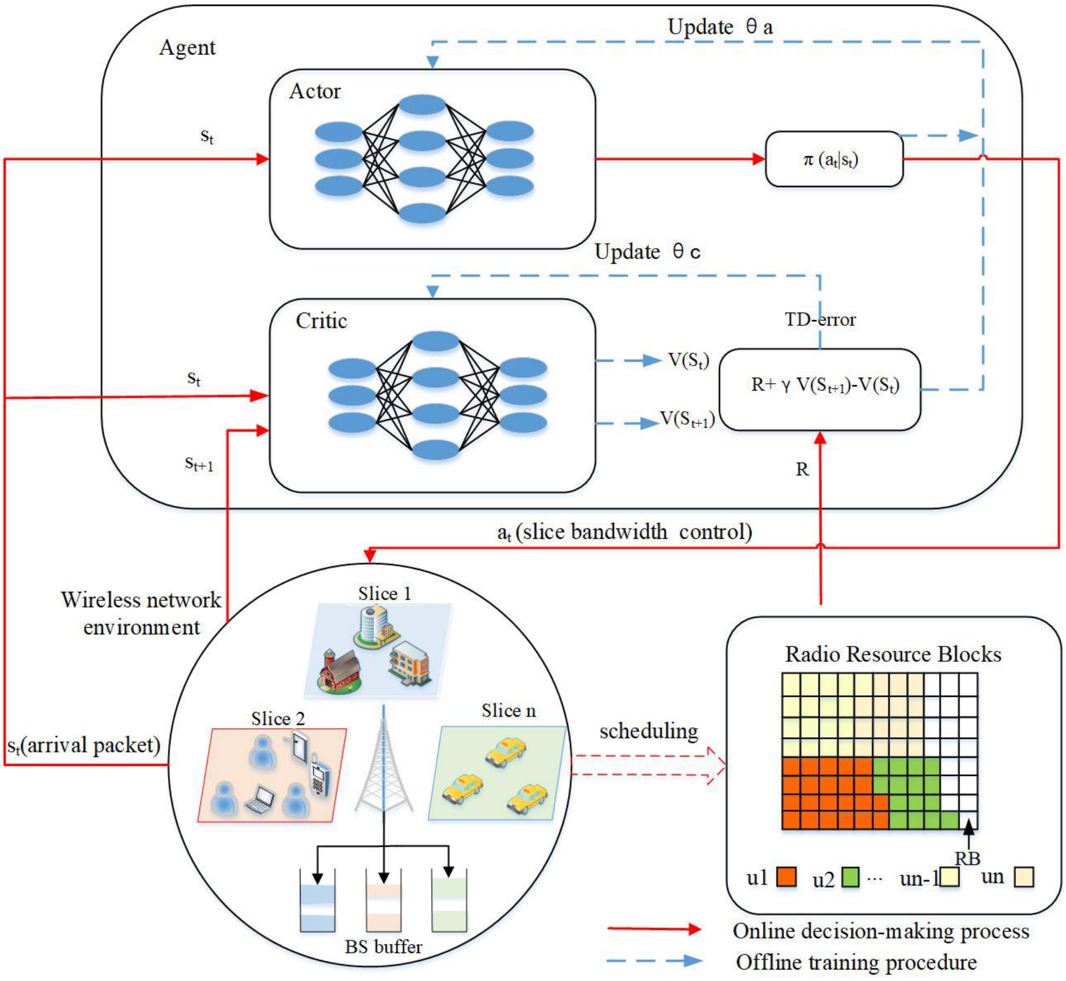


FIGURE 2. Illustration of the proposed A2C network slicing resource allocation architecture.

below the cutoff value [20]. The objective of this part is to optimize the sum rate of the entire system under the limiting total power restriction. Denote p_j as the assigned power to the j-th RB.(1) can be rewritten as [26]:

$$r_{i,j,t} = B \cdot \sum_{n=1}^{\eta} \log_2 \left(1 + \frac{\rho p_{i,j,t} \lambda_{j,n}}{M\sigma^2}\right), \tag{21}$$

where $\eta = \min(M, Q)$ is the rank of the channel matrix $\mathbf{H}_{i,j,t}$ and $\lambda_{j,n}$ $(n = 1, 2, \ldots, \eta)$ are the positive eigenvalues of $\mathbf{H}_{i,j,t}\mathbf{H}_{i,j,t}{}^*$. The objective function for a long packet can be represented as

$$\max \sum_{j=1}^{F} \sum_{n=1}^{\eta} B \cdot \log_2 \left(1 + \frac{\rho p_j \lambda_{j,n}}{M\sigma^2}\right)$$
$$St. \sum_{j=1}^{F} p_j \leqslant P_{\max}, \tag{22}$$

where, $P_{\max}$ is the total power. This is a non-convex optimization issue that has been extensively researched in the literature, and it is then transformed into a convex-constrained optimization problem [27], [28]. We adopt the water-filling [29] power allocation and the power for RB p_j as

$$p_j = \left[\frac{1}{\mathrm{LIn2}} - \frac{\eta\sigma^2}{\frac{\rho}{M}\sum_{n=1}^{\eta}\lambda_{j,n}}\right]^+, \tag{23}$$

$p_j = \left[u - \frac{\eta\sigma^2}{\frac{\rho}{M}\sum_{n=1}^{\eta}\lambda_{j,n}}\right]^+$, where u denotes $\frac{1}{\mathrm{LIn2}}$ and $[x]^+ = \max\{0, x\}$, u is the water level that satisfies

$$\sum_{j=1}^{F} p_j = \sum_{j=1}^{F} \left(u - \frac{\eta\sigma^2}{\frac{\rho}{M}\sum_{n=1}^{\eta}\lambda_{j,n}}\right)^+ = P_{\max}, \tag{24}$$

so, $u = \left(P_{\max} + \sum_{j=1}^{F} \frac{\eta\sigma^2}{\frac{\rho}{M}\sum_{n=1}^{\eta}\lambda_{j,n}}\right)/F$.

The objective function for a short packet can be represented by (26). We verified that the first-order approximation turns the objective function into a convex function by using the Taylor expansion approach to approximate the problem

(26) [30]. We obtained the first-order Taylor expansion as

$$\sqrt{1-\frac{\sum_{q=1}^{Q}\frac{1}{\left(1+\frac{\rho p_j\lambda_{j,q}}{M\sigma^2}\right)^2}}{Q}} = 1-\frac{1}{2}\sum_{q=1}^{Q}\frac{1}{Q\left(1+\frac{\rho p_j\lambda_{j,q}}{M\sigma^2}\right)^2} + o\left(\frac{1}{2}\sum_{q=1}^{Q}\frac{1}{Q\left(1+\frac{\rho p_j\lambda_{j,q}}{M\sigma^2}\right)^2}\right). \tag{25}$$

In particular, the approximation $\sqrt{1-\frac{\sum_{q=1}^{Q}\frac{1}{\left(1+\frac{\rho p_j\lambda_{j,q}}{M\sigma^2}\right)^2}}{Q}} \approx 1-\frac{1}{2}\sum_{q=1}^{Q}\frac{1}{Q\left(1+\frac{\rho p_j\lambda_{j,q}}{M\sigma^2}\right)^2}$ is tight in the case as the value of $\frac{1}{2}\sum_{q=1}^{Q}\frac{1}{Q\left(1+\frac{\rho p_j\lambda_{j,q}}{M\sigma^2}\right)^2}$ is very small. Then, problem (26), as shown at the bottom of the next page, can be approximated as (27), as shown at the bottom of the next page, we have the following key lemma:

Lemma 1. Problem (27) is convex. Proof as follows:

$$\frac{\partial r_i}{\partial p_j}=\sum_{q=1}^{Q}\left\{\frac{\mathfrak{M}}{\left(1+p_j\mathfrak{M}\right)\cdot In2}-\frac{Q^{-1}\left(\varepsilon\right)}{\sqrt{Q\mathfrak{n}}In2}\cdot\frac{\mathfrak{M}}{\left(1+p_j\mathfrak{M}\right)^3}\right\}, \tag{28}$$

where $\mathfrak{M}=\rho\lambda_{j,q}/M\sigma^2$.

$$\begin{aligned}\frac{\partial^2 r_i}{\partial^2 p_j}&=\sum_{q=1}^{Q}\left\{-\frac{\mathfrak{M}^2}{\left(1+p_j\mathfrak{M}\right)^2\cdot In2}+\frac{3Q^{-1}\left(\varepsilon\right)}{\sqrt{Q\mathfrak{n}}In2}\cdot\frac{\mathfrak{M}^3}{\left(1+p_j\mathfrak{M}\right)^4}\right\}\\&=\sum_{q=1}^{Q}\left\{\frac{\mathfrak{M}^2}{\left(1+p_j\mathfrak{M}\right)^2\cdot In2}\right.\\&\quad\left.\cdot\left(\frac{3Q^{-1}\left(\varepsilon\right)}{\sqrt{Q\mathfrak{n}}}\cdot\frac{1}{\left(1+p_j\mathfrak{M}\right)^2}-1\right)\right\}\\&\leqslant\sum_{q=1}^{Q}\left\{\frac{\mathfrak{M}^2}{\left(1+p_j\mathfrak{M}\right)^2\cdot In2}\cdot\left(\frac{1}{\left(1+p_j\mathfrak{M}\right)^2}-1\right)\right\}\leqslant 0.\end{aligned} \tag{29}$$

Hence, $-\frac{\partial^2 r_i}{\partial^2 p_j}\geqslant 0$. In addition, we have $-\frac{\partial^2 r_i}{\partial^2 p_i p_j}=0$, $\forall i\neq j$, thus, it can be written as a diagonal matrix for the hessian matrix of it. Since this matrix is positive semidefinite, the problem (27) is convex. The pseudocode for the lower level is shown in Algorithms 2 and 3. In Algorithm 3 $|Channels|$ denotes the numbers of Channels.

C. COMPUTATIONAL COMPLEXITY ANALYSIS

The number of neurons π_a^l in the l-th layer of the actor neural network and the epoch steps T_U of upper-level policy training can be used to represent the computational complexity of the learning method for the upper-level control policy (i.e., Algorithm 1) as $\mathcal{O}\left(T_U\left(\sum_{l=0}^{L_{actor}}\pi_a^l\pi_a^{l+1}+\sum_{l=0}^{L_{critic}}\pi_c^l\pi_c^{l+1}\right)\right)$. $L_{actor}(L_{critic})$ (22) denotes the number of hidden layers in the actor (critic)

Algorithm 2 Proportional Fairness Scheduler Procedure

```
Input: Slice RB $\{w_1, w_2, \ldots, w_N\}$;
Output: Allocate user RB $\{RB_1, RB_2, , \ldots, RB_U\}$;
$index = 0$.
for $n$ in range($N$) do
    Calculate available RBs for slice $n$, namely $w_n$.
    Calculate require RBs slice user $\{RB_{n,1}, RB_{n,2}, \ldots, RB_{n,u}\}$, $u \leqslant \mathcal{U}_n^{actv}$.
    Calculate instantaneous available data rate of the $i$-th user of the slice $n$, $r_{n,i,t-1}$ and average data rate $\overline{R_{n,i,t-1}}$.
    Calculate the priority of the user, and sort users by priority.
    index 1=index
    for $i$ in range($u$) do
        Calculate the required resource blocks according to the user's priority order.
        if $w_n \geqslant RB_{n,i}$ then
            $aRB_{n,i} = RB_{n,i}$ ($aRB_{n,i}$ denote the assign RBs for user $i$ of slice $n$).
            $w_n - = aRB_{n,i}$
        else
            if $w_n \geqslant 0$ then
                $aRB_{n,i} = w_n$
                $w_n = 0$
            else$aRB_{n,i} = 0$
            end if
        end if
        $RB_{n,i} = [index1, index1 + aRB_{n,i}]$.
        $index1 = index1 + aRB_{n,i}$
    $index + = w_n$
    end for
end for
Obtain user RB $\{RB_1, RB_2, , \ldots, RB_U\}$.
```

Algorithm 3 Water-Filling-Based Power Allcoation

```
Input: User RB $\{RB_1, RB_2, , \ldots, RB_U\}$, total power $p_{\max}$;
Output: RBs power $\{P_{RB1}, P_{RB2} \ldots, P_{RBF}\}$;
Sort resource blocks according to their channel coefficients.
Remove channels; $Rems = 0$.
Calculate the water line.
Power allocation to resource blocks, calculate RBs power.
while $P_{sum} > P_{\max}$) and ($Rems < |Channels|$ do
    $Rems + = 1$.
    Calculate the water line.
    Power allocation to resource blocks.
end while
Divide the remaining power equally among the remaining channels.
Obtain RBs power $\{P_{RB1}, P_{RB2}, \ldots, , P_{RBF}\}$.
```

network, while π_c^l is the number of neurons in the l-th layer of the critic neural network. Algorithm 2's lower-level control policy's learning procedure's complexity may be calculated using the formula $\mathcal{O}(T_L \mathcal{U}_{active})$, where T_L denotes TTI steps. The number of active users is represented by $\mathcal{U}_{active}$.

The computational complexity of Algorithms 1 and 2 grows linearly with the number of epoch steps, TTI steps, and active users in the neural network and quadratically with the number of neurons in the neural network layer.

V. SIMULATION ANALYSIS

A. SIMULATION ENVIRONMENT SETTINGS

In this study, we consider a typical downlink cellular network system with one BS, three different service types (VoLTE, eMBB, and uRLLC), and related slices in a simulation region with a radius of 100 m, where there are 120 UEs. Within the cell service region, there exists a random distribution of three different user types. The actor network and cirtic network learning rates were set to 0.002 and 0.01, respectively. Additionally, the entropy regularization used to promote exploration was set to 0.001. Users move at the beginning of each epoch and remain constant within each epoch, slices of the same type share the same movement pattern and will bounce symmetrically back into the cell when the user moves out of the cellular boundary. The specific parameter settings are shown in Table 2 [12].

B. BASELINE ALGORITHM

1) HARD-SLICING

Hard-slicing is a technique in which all the RBs are divided by the number of slices. Each service slice is always allocated $\frac{1}{N}$ of the entire bandwidth because there are N types of services. $w_{n,t} = \frac{1}{N} \cdot W$, w_n is the whole bandwidth of the n-th slice in t-th TTI.

2) DQN [11]

When the experience is set as the state, action, and reward simultaneously, the agent receives continuous experience input over time. Time-series correlations between the data are shown when learning is performed in the order that experiences arrive. As a result, the DQN uses experience replay to erase the correlation. Learning is randomly taken from the memory after the events have been briefly stored there to eliminate any correlation. The target value is y_t is given by

$$y_t = r_{t+1} + \Upsilon \max_{a \in A} Q(s_{t+1}, a; \theta), \tag{30}$$

where θ denotes the neural network's parameters. The agent learns the value of the action by updating θ to approach y_t. For effective learning, the DQN also has a fixed goal Q-network and an experience replay. DQN refers to the training data as a target value and visualizes it as an action value. The error was the TD error. Consequently, the loss function that must be reduced is

$$L(\theta) = E\left[(y_t - Q(s_t, a_t : \theta))^2\right]. \tag{31}$$

The goal update interval is referred to as the update interval θ^*. A gradient-based strategy is frequently used to obtain θ

$$\theta \leftarrow \theta - \alpha \nabla L(\theta). \tag{32}$$

3) DUELING DQN

Using the dueling DQN method, (s, a), the value function V(s) and advantage function A(s, a) are separated from the Q-values [31]. The reward from state s is represented by the value function V(s). Formula A(s, a) can be used to determine the relative superiority of one action over the other actions of the advantage function. By combining the value V and advantage A for each activity, we were able to derive Q-values:

$$Q(s, a, \theta, \alpha, \beta) = V(s, \theta, \beta) + A(s, a, \theta, \alpha). \tag{33}$$

We set the highest value in the advantage function to zero, and all other values to negative numbers, forcing the maximum Q-value to be equal to V. As a result, we will have

$$\max \sum_{j=1}^{F} B \left\{ \sum_{q=1}^{Q} \log_2\left(1 + \frac{\rho p_j \lambda_{j,q}}{M\sigma^2}\right) - \sqrt{\frac{1}{\mathfrak{n}}\left(Q - \sum_{q=1}^{Q}\left(1 + \frac{\rho p_j \lambda_{j,q}}{M\sigma^2}\right)^{-2}\right)} \frac{Q^{-1}(\epsilon)}{In2} \right\}$$

$$\sum_{j=1}^{F} p_j \leqslant P_{\max}. \tag{26}$$

$$\max \sum_{j=1}^{F} B \left\{ \sum_{q=1}^{Q} \log_2\left(1 + \frac{\rho p_j \lambda_{j,q}}{M\sigma^2}\right) - \sqrt{\frac{Q}{\mathfrak{n}}} \frac{Q^{-1}(\epsilon)}{In2} \left(1 - \frac{1}{2Q} \sum_{q=1}^{Q}\left(1 + \frac{\rho p_j \lambda_{j,q}}{M\sigma^2}\right)^{-2}\right) \right\}$$

$$\sum_{j=1}^{F} p_j \leqslant P_{\max}. \tag{27}$$

TABLE 2. **Default parameter setting for network slicing.**

Parameter	Value		
Total system bandwidth	10MHZ		
The length of each epoch	1000ms(2000TTI)		
Total power	46 dBm		
Maximum Queue Length	5		
Transmission error probability	0.001		
MIMO antenna	16 × 4 (transmitting antennas, receiver antennas)		
Scheduling	Proportional fair per TTI		
Shadow fading	log-normal		
Path loss	$145.4 + 37.5 \log_{10} d$		
Service type	VoLTE	eMBB	uRLLC
Speed	1m/s	4m/s	8m/s
Inter-Arrival Time per packet	Uniform: Min: 0, Max: 160ms	Truncated Pareto: Exponential Para:1.2, Mean:6ms, Max:12.5ms	Exponential: Means 180ms
Packet size	400 Byte	800 Byte	200 Byte
Maximum tolerant packet delay	10ms	10ms	1ms
Rate	51kbps	500Mbps	100Mbps

a precise value for V that can be used to add all the advantages and arrive at a solution:

$$Q(s,a,\theta,\alpha,\beta) = V(s,\theta,\beta) + \left(A(s,a,\theta,\alpha) - \max_{a' \in |\mathcal{A}|} A\left(s,a';\theta,\alpha\right)\right). \tag{34}$$

An alternate module substitutes an average for the max operator:

$$Q(s,a,\theta,\alpha,\beta) = V(s,\theta,\beta) + \left(A(s,a,\theta,\alpha) - \frac{1}{|\mathcal{A}|} A\left(s,a';\theta,\alpha\right)\right). \tag{35}$$

The target Q value is computed using the following formula, and the proposed model must approximate the target Q value:

$$y_t = r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a; \theta, \alpha, \beta). \tag{36}$$

Our model is trained to approximate y_t, and using the gradient descent method, all parameters are gradually updated to reduce the mean square error $\|Q(s_{t+1}, a; \theta, \alpha, \beta) - y_t\|^2$ as follows:

$$\theta \leftarrow \theta - \phi\nabla\theta, \alpha \leftarrow \alpha - \phi\nabla\alpha, \beta \leftarrow \beta - \phi\nabla\beta, \tag{37}$$

where ϕ denotes the learning rate.

C. EXPERIMENT RESULT

In this section, we contrast the Dueling DQN, DQN, and hard slicing algorithm simulation results with those of the proposed A2C algorithms.

The first aim was to test the spectrum efficiency of the different methods. The variations in the spectrum efficiency of the system with respect to the iteration index are shown in Figure 3. It can be observed that the hard algorithm has a fixed spectrum efficiency of approximately 250. The spectrum efficiency of the DQN and Dueling DQN do not converge during the entire iteration, as they fluctuate in the range of 50-500. In contrast, Dueling DQN has less fluctuation than DQN. The A2C has a stable spectrum efficiency of approximately 400 and a fast convergence rate, which converges after approximately 2000 iterations. It can be observed that A2C has the best results regarding the effectiveness of the spectrum.

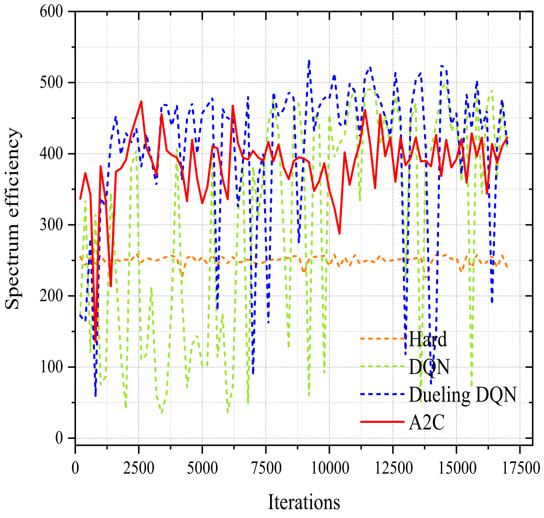


FIGURE 3. **Comparison of spectrum efficiency for different methods.**

A performance comparison of QoE for each service is presented in Figure 4-6. It can be seen that all three slices met our expectations by learning from the viewpoint of QoE using the A2C algorithm. In Figure 4, the VoLTE slice, its QoE is stable at 1 by A2C, because its requirement is easy to meet both in rate and packet delay. The QoE by Dueling DQN, DQN, and hard slicing algorithms have some fluctuation, not converging in the entire training process. The QoE by Dueling DQN, DQN, and hard slicing algorithm in the range of 0.994 to 1 has a similar convergence performance.

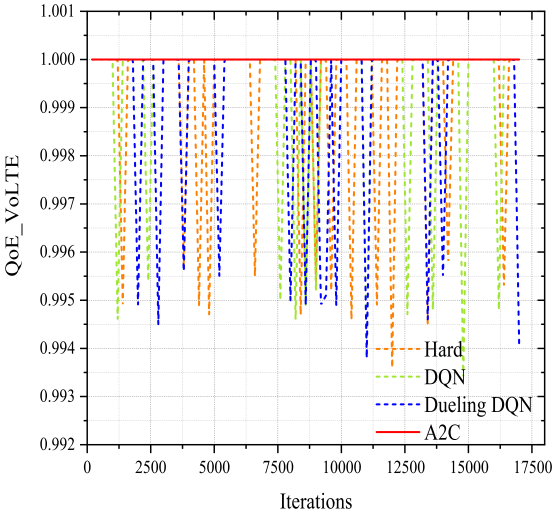


FIGURE 4. QoE of VoLTE slice.

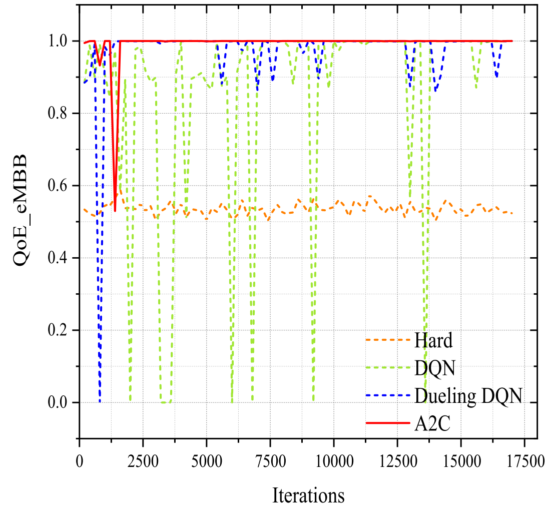


FIGURE 5. QoE of eMBB slice.

The QoE of eMBB is shown in Figure 5, where the QoE of eMBB by A2C slicing converges at a fast speed and maintains a steady state with a higher value of approximately 1. The Dueling DQN and DQN do not converge during the entire iteration, but the Dueling DQN has much less fluctuation. The QoE of the hard algorithm was stable at a lower value of approximately 0.5.

The QoE of the uRLLC is shown in Figure 6. The uRLLC QoE performance of A2C was stable at approximately 1. The Dueling DQN, DQN algorithms, and hard do not have convergence in the whole iteration process. It can be seen that despite the uRLLC slice having strict latency requirements, we can achieve better QoE performance using short packet transmission. The QoE by Dueling DQN, DQN, and hard slicing algorithm in the range of 0.996 to 1 has a similar convergence performance.

As specified in (11), we can combine three RL algorithms, one for each slice. QoE and SE each have importance weights of $\zeta = [1, 1, 1]$ and $\mu = 0.01$, respectively. The utility function can demonstrate how well RAN slicing control works. In other words, the QoE and SE performance of the network slices improves with increasing utility function values. Figure 7 shows the utility values during the training. The utility function values ranged widely between 2 and 8. The utility value by the Dueling DQN and DQN algorithms does not converge during all iterations. Hard slicing has the worst utility value around 5. The A2C has a stable utility of approximately 7, with a fast convergence rate of approximately 2000 iterations. This shows that the utility value of A2C is superior to Dueling DQN, DQN, and hard slicing algorithms. From the above results, it is clear that utility and spectrum efficiency have similar trends, which stem from (11).

The packet drop rate is shown in Figure 8. Obviously, the packet drop rate by A2C is the lowest, approaching 0, followed by the hard algorithm. The DQN and Dueling DQN have a high packet loss rate compared to A2C.

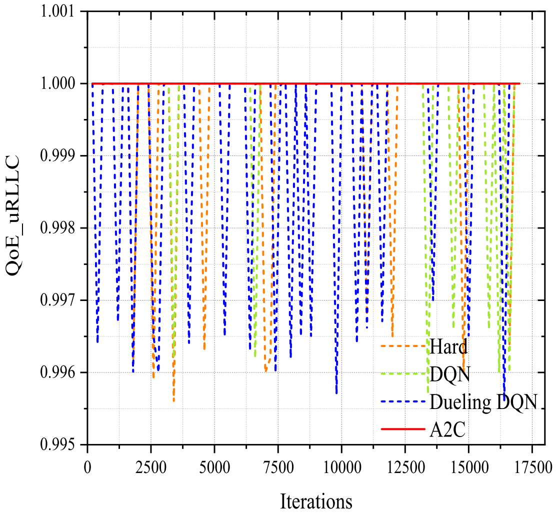


FIGURE 6. QoE of uRLLC slice.

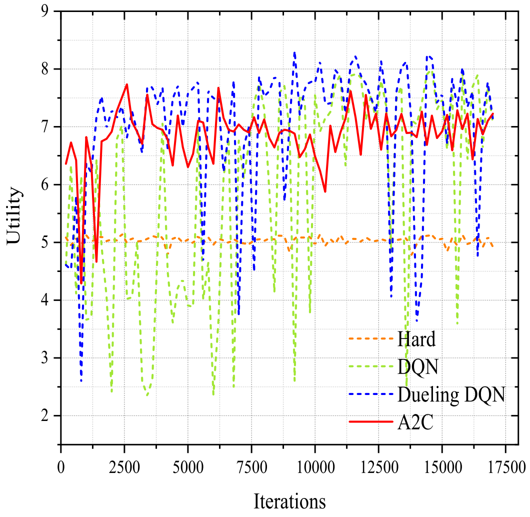


FIGURE 7. Utility function of all network slicing.

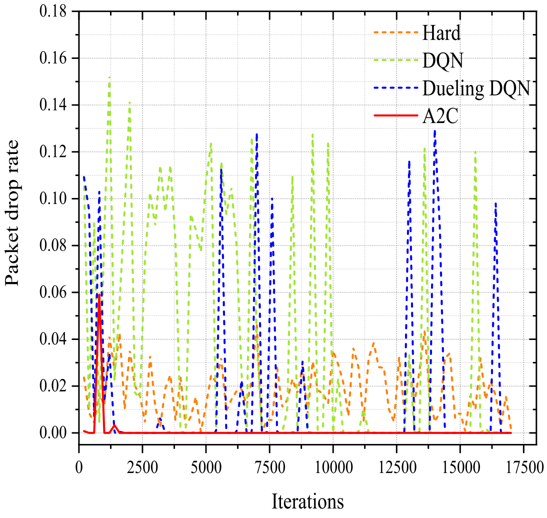


FIGURE 8. **Packet drop rate of all network slicing.**

Overall, the slicing resource allocation by the A2C algorithm has an advantage in terms of convergence speed, QoE, and packet drop rate over the other three algorithms. The problem with Dueling DQN and DQN is that they do not converge. The hard slicing algorithm has a lower uRLLC QoE and utility than the other three algorithms. Similar trends are observed for all four algorithms mentioned above when assessing their efficiency and utility.

VI. CONCLUSION

In this paper, we presented a DRL joint massive MIMO resource allocation method in RAN slicing that aims to optimize the long-term QoE of network slices while also maximizing the SE. The main components of the proposed solution are a lower-level controller and an upper-level controller. To improve QoE and SE performance, the upper-level controller applies the A2C algorithm to allocate the bandwidth between slices in each epoch according to the user mobility at a rough granularity. The lower-level PF and WF controller adapter schedules each network slice's active UEs' PRB and power allocation at a fine granularity, which is combined with MIMO to increase the rate. Compared with the Dueling DQN, DQN, and hard schemes, the simulation results demonstrate that A2C provides higher QoE and SE performance and stable convergence control performance. The results prove that uRLLC slices with high latency requirements have better results with short packet transmission. In addition, it has a very low packet loss rate. Future work will attempt to further improve the convergence speed using a more effective algorithm.

REFERENCES

[1] A. Banchs, G. de Veciana, V. Sciancalepore, and X. Costa-Perez, "Resource allocation for network slicing in mobile networks," *IEEE Access*, vol. 8, pp. 214696–214706, 2020.

[2] F. Fossati, S. Moretti, P. Perny, and S. Secci, "Multi-resource allocation for network slicing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1311–1324, Jun. 2020.

[3] P. Caballero, A. Banchs, G. de Veciana, X. Costa-Pérez, and A. Azcorra, "Network slicing for guaranteed rate services: Admission control and resource allocation games," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6419–6432, Oct. 2018.

[4] O. U. Akgul, I. Malanchini, and A. Capone, "Anticipatory resource allocation and trading in a sliced network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.

[5] Y. Xiao, M. Hirzallah, and M. Krunz, "Distributed resource allocation for network slicing over licensed and unlicensed bands," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2260–2274, Oct. 2018.

[6] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2454–2465, Sep. 2019.

[7] G. Sun, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and J. Wei, "Autonomous resource slicing for virtualized vehicular networks with D2D communications based on deep reinforcement learning," *IEEE Syst. J.*, vol. 14, no. 4, pp. 4694–4705, Dec. 2020.

[8] M. Sulaiman, A. Moayyedi, M. Ahmadi, M. A. Salahuddin, R. Boutaba, and A. Saleh, "Coordinated slicing and admission control using multi-agent deep reinforcement learning," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1110–1124, Jun. 2023.

[9] M. Setayesh, S. Bahrami, and V. W. S. Wong, "Resource slicing for eMBB and URLLC services in radio access network using hierarchical deep learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 8950–8966, Nov. 2022.

[10] G. Zhou, L. Zhao, G. Zheng, Z. Xie, S. Song, and K.-C. Chen, "Joint multi-objective optimization for radio access network slicing using multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, early access, Apr. 20, 2023, doi: 10.1109/TVT.2023.3268671.

[11] R. Li, Z. Zhao, Q. Sun, I. Chih-Lin, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018.

[12] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, "The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sep. 2020.

[13] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, "Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1337–1341, Aug. 2019.

[14] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.

[15] Y. Shao, R. Li, Z. Zhao, and H. Zhang, "Graph attention network-based DRL for network slicing management in dense cellular networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Nanjing, China, Mar. 2021, pp. 1–6.

[16] Q. Ye, W. Zhuang, L. Li, and P. Vigneron, "Traffic-load-adaptive medium access control for fully connected mobile ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9358–9371, Nov. 2016.

[17] J. Zheng, Q. Zhang, and J. Qin, "Average achievable rate and average BLER analyses for MIMO short-packet communication systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12238–12242, Nov. 2021.

[18] M. Y. Abdelsadek, Y. Gadallah, and M. H. Ahmed, "Resource allocation of URLLC and eMBB mixed traffic in 5G networks: A deep learning approach," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[19] H. Yang, K. Zheng, K. Zhang, J. Mei, and Y. Qian, "Ultra-reliable and low-latency communications for connected vehicles: Challenges and solutions," *IEEE Netw.*, vol. 34, no. 3, pp. 92–100, May 2020.

[20] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[21] V. Y. F. Tan and M. Tomamichel, "The third-order term in the normal approximation for the AWGN channel," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2430–2438, May 2015.

[22] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq, and H. Abou-Zeid, "Intelligent radio access network slicing for service provisioning in 6G: A hierarchical deep reinforcement learning approach," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6063–6078, Sep. 2021.

[23] Y. Xu, Z. Zhao, P. Cheng, Z. Chen, M. Ding, B. Vucetic, and Y. Li, "Constrained reinforcement learning for resource allocation in network slicing," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1554–1558, May 2021.

[24] G. Chen, X. Mu, F. Shen, and Q. Zeng, "Network slicing resource allocation based on LSTM-D3QN with dual connectivity in heterogeneous cellular networks," *Appl. Sci.*, vol. 12, no. 18, p. 9315, Sep. 2022.

[25] G. Piro, L. A. Grieco, G. Boggia, R. Fortuna, and P. Camarda, "Two-level downlink scheduling for real-time multimedia services in LTE networks," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1052–1065, Oct. 2011.

[26] I. M. Duran, "Optimal power allocation in MIMO wire-tap channels," M.S. thesis, Departament de Teoria del Senyal i Comunicacions, Projecte/Treball Final de Carrera, UPC, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, 2011.

[27] W. Yu and J. M. Cioffi, "On constant power water-filling," in *Proc. ICC. IEEE Int. Conf. Commun. Conf. Rec.*, Helsinki, Finland, Jun. 2001, pp. 1665–1669.

[28] J. Kim, H.-W. Lee, and S. Chong, "Virtual cell beamforming in cooperative networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1126–1138, Jun. 2014.

[29] P. He, L. Zhao, S. Zhou, and Z. Niu, "Water-filling: A geometric approach and its application to solve generalized radio resource allocation problems," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3637–3647, Jul. 2013.

[30] Z. Xiao, G. Sun, Y. Hu, C. Shen, and A. Schmeink, "Channel capacity in the finite blocklength regime for massive MIMO with selected multi-streams (invited paper)," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2022, pp. 216–221.

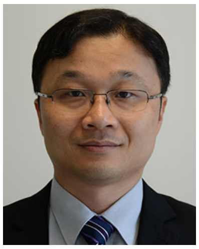
[31] T.-W. Ban, "An autonomous transmission scheme using dueling DQN for D2D communication networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16348–16352, Dec. 2020.

DANDAN YAN (Student Member, IEEE) received the M.S. degree in circuits and systems from the Chengdu University of Technology, China, in 2017. She is currently pursuing the Ph.D. degree in computer applied technology with Macao Polytechnic University. Her current research interests include resource allocation, machine learning, and deep reinforcement learning in communications.

BENJAMIN K. NG (Senior Member, IEEE) received the B.A.Sc., M.A.Sc., and Ph.D. degrees in engineering science and electrical engineering from the University of Toronto, in 1996, 1998, and 2002, respectively.

From 2005 to 2009, he was with Radiospire Networks Inc., Boston, MA, USA, where he held the position of Senior Communications Engineer focusing on the UWB and millimeter wave technologies. He joined Macao Polytechnic University, Macau, China, in 2010, where he is currently an Associate Professor with the Faculty of Applied Sciences. His research interests include wireless communications and signal processing, with an emphasis on MIMO, NOMA, and machine learning technologies.

WEI KE (Member, IEEE) received the Ph.D. degree from the School of Computer Science and Engineering, Beihang University. He is currently a Professor with the Computer Applied Technology Program, Macao Polytechnic University. His current research projects involve the design and implementation of open platforms for applications of computer vision and pattern recognition, including programming tools, environments, and frameworks. His research interests include programming languages, image processing, computer vision, and tool support for component-based engineering and systems.



CHAN-TONG LAM (Senior Member, IEEE) received the B.Sc. (Eng.) and M.Sc. (Eng.) degrees from Queen's University, Kingston, ON, Canada, in 1998 and 2000, respectively, and the Ph.D. degree from Carleton University, Ottawa, ON, Canada, in 2007. He is currently an Associate Professor with the Faculty of Applied Sciences, Macao Polytechnic University, Macau SAR, China. From 2004 to 2007, he participated in the European Wireless World Initiative New Radio (WINNER) Project. His research interests include mobile wireless communications, machine learning in communications, and computer vision in smart cities.