


Research Article

Dynamically Resource Allocation in Beyond 5G (B5G) Network RAN Slicing Using Deep Deterministic Policy Gradient

Rizwan Munir ¹, Yifei Wei,¹ Chao Ma,² and Bizhu Yang²

¹Beijing University of Posts and Telecommunications, Beijing 100876, China

²China Academy of Information and Communications Technology, Beijing 100191, China

Correspondence should be addressed to Rizwan Munir; sardarrizwan.786@gmail.com

Received 13 October 2022; Revised 2 December 2022; Accepted 3 December 2022; Published 21 December 2022

Academic Editor: Xianfu Chen

Copyright © 2022 Rizwan Munir et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network slicing makes it possible for future applications with a variety of adaptability requirements and performance requirements by splitting the physical network into several logical networks. Radio access network (RAN) slicing's main goal is to assign physical resource blocks (RBs) to mMTC, eMBB, and uRLLC services while ensuring the Quality of service (QoS). Consequently, it is challenging to determine the optimal strategies for 5G radio access network (5G-RAN) slicing because of dynamically changes in slice needs and environmental data, and conventional approaches have difficulty addressing resource allocation issues. In this paper, we present an energy-efficient deep deterministic policy gradient resource allocation (EE-DDPG-RA) method for RAN slicing in 5G networks to choose the resource allocation policy that increases long-term throughput while satisfying the requirements of B5G systems for quality of service. This method's main goal is to remove unnecessary actions in order to lower the amount of available action space. The numerical outcomes demonstrate that the proposed approach outperforms boundaries by enhancing deep-rooted throughput and effectively managing resources.

1. Introduction

The fifth generation (5G) of the mobile network is being added in order to satisfy user expectations and the business requirements of network service providers in 2020 and beyond. By 2035, it will be valued at more than \$12.3 trillion, predicted in [1]. The 5G standard makes it possible to produce a state-of-the-art end-to-end network with completely mobile communication. The 5G system is well-matched with an extensive variety of existing use cases, each of which has a specific set of service needs. Multiple services are typically grouped into the three categories of mMTC, eMBB, and uRLLC [2]. The requirements for the eMBB are substantially dissimilar from those for the uRLLC and mMTC. Due to their specs, low data transfer volume, low power consumption, and delay resilience, mMTC applications stand out [3]. For run-time interaction, various uRLLC and mMTC platforms support higher throughput and reduced latency. eMBB applications stand out because of their higher data rates, bandwidth, and mobility support over a large service area. We demand substantial networks with step-based den-

sities, considerably higher bandwidths, network connectivity, full coverage mobility, hyper security, and secrecy due to the tremendous growth of users, potential uses, traffic volume, and business practices [4].

Modern network slicing (NS) makes it possible to switch from a static to a dynamic network infrastructure. Network slicing is the main advance of 5G technologies, which uses network virtualization, software-defined networks, and fog computing as enablers to provide a range of network capabilities based on user needs [5]. The ability to independently change each slice is how the network works, assigns the proper amount of network resources in line with business needs, and enhances the overall flexibility, robustness, dependability, and traffic models. A physical network might be divided into numerous logical networks using network slicing. The authors of [6] optimize the distribution of diverse resources and offer suitable assistance to numerous consumers of various services. Based on the needs of the slice, an end-to-end digital network can adaptively offer various services. Each network slice can offer resources, including transmission power, processing resources, resource

blocks (RBs), and bandwidth. Each slice runs autonomously from the others because of their separation; hence, issues with one slicing do not influence the functionality of the other slices [7].

Core network (CN) and radio access network (RAN) comprise a network slice. 5G core network slicing has garnered much interest compared to RAN slicing, which has so far attracted minimal interest from the research community. Allocation of resources is a major problem with RAN slicing. RAN slicing continues to be a key difficulty for users while maintaining the quality of service (QoS) needs as the radio access network environment changes in wireless link transmission conditions, user expectations, and user density. Resource selection in the RAN slice is more difficult than core network slicing when user movement and radio channel circumstances are considered [8]. The primary purposes of network slicing in 5G networks are resource scheduler allocation [9]. Resources were distributed statically in the previous research, with a set number of resources going to each slice. This would cause resource under- or overutilization, rendering the remaining resources useless and creating multiple difficulties for different mobile services in order to maintain QoS standards.

Resources will be underutilized without an adaptive resource utilization approach, which can cause issues for consumers using different services. In 5G network slicing, the MDP can be viewed as resource allocation to consider the importance of both SE (spectrum efficiency) and EE (energy efficiency) in the network. Allocation of resources is an NP-hard issue that is practically unsolvable when dealing with enormous volumes of data. A machine learning strategy can resolve NP-hard resource scheduling issues [10]. Deep reinforcement learning (DRL), a machine learning component, has recently grown in popularity and is useful for decision-making. DRL development is expanding in robotics, cyber security, and video games [11, 12].

This paper introduces an enhance, efficient deep deterministic policy gradient resource allocation (EE-DDPG-RA) framework based on RAN architecture to increase the radio resource allocation effectiveness of MVNOs. Radio resources are distributed across eMBB, uRLLC, and mMTC users using a Markov decision approach (MDP). For the purpose of allocating system dynamics RB and energy infrastructure to each client in a 5G network slice, a DRL-based resource allocation mechanism is being taken into consideration. When assigning resources to multiple users under this system, each customer's requirements in each slice are considered because the channel circumstances changed. This is the first article that, as far as the author is aware, addresses RAN resource allocation through a partnership of deep learning and reinforcement learning. To help RAN make accurate decisions, the significance of online choice components and projections can be dynamically adjusted.

- (i) The primary contributions of this study are, in brief, listed below
- (ii) A dual optimization goal of RB allocation and energy minimization is proposed for the resource

scheduling problem to reduce energy consumption and satisfy QoS criteria

- (iii) An MDP can describe the continuous control problem known as the dual optimal problem because of its broad solution area
- (iv) The DDPG resource scheduling (DDPG-RS) algorithm is proposed to obtain the optimal resource scheduling scheme founded on the advantages of DDPG in solving persistent control problems and the scalability problem
- (v) The DDPG approach enhances the entire system's performance by dynamically allocating the above-mentioned resources to each slice
- (vi) Finally, extensive simulations performed in Python confirm the usefulness of the suggested framework

The remainder of the essay is structured as follows: there is a study of the literature in Section 2. Section 3 discusses the framework and problem definition. In Section 4, we suggest the EE-DDPG-RA algorithm to solve the issue. The simulation results are presented with an explanation of the applicability and effectiveness of the suggested technique in Section 5 outcomes. In Section 6, the conclusion is extensively explained.

2. Literature Review

Numerous studies that looked at RAN slicing were published in [13] with deep slice. This deep learning strategy uses neural networks to tackle network access and load balancing concerns efficiently. Using the supplied KPIs, this study trains the network for inbound traffic monitoring and network slice projection for any user type. Load balancing and efficient resource consumption across the available network slices are made possible by intelligent resource allocation.

Both the business and academic communities consider slicing as the foundational innovation of the 5G network. Network slicing, according to the International Mobile Telecommunications Union (IMT) [14], is a crucial part of the 5G network. Several business sectors and organizations that establish standards, like the International Telecommunications Union, have been actively discussing machine learning methods for network slicing. For instance, the International Communication Union is creating groups based on machine learning to support future networks like 5G [15]. We noticed the use of network RAN slicing in [16]. The standard resources and radio hardware are parts of the wireless communication system known as the RAN slices; they are less elastic than the core network. In order to manage diverse requests from various mobile services, each slice of a RAN has a distinct air parameter.

In this work, RAN slicing is considered since the RAN section of the network interacts closely with the competitive SPs, network operators, mobile customers, and the SDN scheduler responsible for all management plan decisions. Network slicing for the allocation of resources has been the

subject of many research studies. The efficiency of multitenant resource allocation during network slicing may be evaluated using game theory as a conceptual approach [26, 27]. Caballero et al. [28] developed a matching theoretic drive prioritization algorithm to assist the network in becoming unbiased concerning the networking source of the energy challenge. This enables the communication between infrastructure and service providers over an over-the-top (OTT) network. In [29], Sun et al. investigated a resource allocation technique called “share-constrained proportionality distribution” in a framework of diverse network games.

An explanation of the relationship between the phone devices (MUs) in fog RAN slicing, the global spectrum sharing supervisor, and the local cognitive radio controller was provided by Xiao and Krunz in [30]. None of the preceding articles has sufficiently defined the efficiency of resource scheduling. Xiao et al. in [31] considered complex network slicing for mobile edge computing systems in the context of energy recovery techniques that are improving and becoming more accessible. A Naive Bayes technique was recommended in order to obtain the ideal resource-slicing architecture between certain edge nodes. By trusting on the movements of the statistics network, this method reinforces the necessity for network density a priori statistical knowledge. With network or spectrum resource slicing as their primary constraints, these initiatives can offer basic mobile services [32].

On the contrary, network slicing becomes more agile and adaptable to a changing network environment due to intelligent learning. For extremely large service slices, the authors of [33] developed a primary concern admittance system that included two layers of approaches and the heuristics technique. However, the 5G network is dynamically ingrained [34]. The internet provider must optimize how resources are allocated among the layers in order to satisfy the shifting slice requirements because the consumption of resources and the volumes of network activity vary over time in the slice.

Despite significant efforts, the literature on the dynamic and effective regulation of RAN slicing still had several holes. We believe that not enough research has been done on dynamic scheduling algorithms for network slicing. Q-learning is used to improve resource allocation to a single vertex in a VNE in the dynamic resource strategy described in [17, 35]. Dispersion of resources differs from virtual network environment (VNE) distribution. Dynamic resource planning in 5G network segmentation is becoming more and more difficult as we deal with reliant virtualized network functions (VNFs) with prior orders and variable resource necessities, as well as separate slices with different QoS criteria. Building a flexible resource-scheduling technique for the various QoS requirements of various network slice services is essential to be able to maximize service productivity and resource consumption effectiveness [18, 19].

3. The System Model and Problem Formulation

System Description and Assumption

3.1. Business Model. The main characteristics of the system are listed below, coupled with an illustration of the most

basic wireless network configuration in Figure 1 showing a variety of supplies used by tenants:

- (i) Each tenant dynamically distributes different resources to many user equipment (UE) units following the service level agreement (SLA). Customers can access multiple resource blocks to find various service slices assigned to other UEs. The UE might be a network device powered by the Internet of Things (IoT). According to priority, each slice provides services to a set of users in real time
- (ii) Every tenant buys a portion of the network and asks the network operator for a physical resource block (PRB) on their portion’s behalf. The infrastructure provider then maintains the network
- (iii) The major component is the controller, which distributes networking PRB to the slicing and the relevant slices’ customers
- (iv) Because so many services are available, the control will constantly adjust the resource allocation approach for each slice to fit its needs
- (v) In addition, the controller learns from prior errors and assigns power and other resources to the UEs following the observed rate or queuing information of the specific slice. It is relevant in the following two situations:
 - (a) The controller can allocate resources according to any resource allocation strategy to schedule UEs in order to avoid deadlock in the case of a huge queue
 - (b) By sharing a channel with other users, users may cause interference problems that increase the likelihood of a service interruption

Consequently, the control must change the channel allocation strategy for the network slices to ensure the QoS slices.

3.2. System Model. We are considering a transmission situation in which a base station provides service to users across various randomly selected coverage zones. $U = 1, 2, 3, \dots$ characterises a user’s set. The base station, DU, and other parties exchange CSI whenever a data center is connected to one, as well as the user equipment (Figure 1). Available physical blocks that may be allocated to the eMBB, uRLLC, or mMTC exist within each of the s distributed systems that comprise up the physical network topology. $x, y,$ and z stand in for the slicing for eMBB, URLLC, and mMTC, respectively. While in eMBB, URLLC, or mMTC, there are $x, y,$ and z total network slices, correspondingly ($x + y + z = N$). To allocate the foundation network allocation to the network element’s eMBB, URLLC, and mMTC slices, we used three binary vectors, $_eI,$ $_uJ,$ and $_(m)L$. Table 1 describes the abbreviations used throughout the paper. Table 2 depicts the RL-based resource allocation algorithms, which

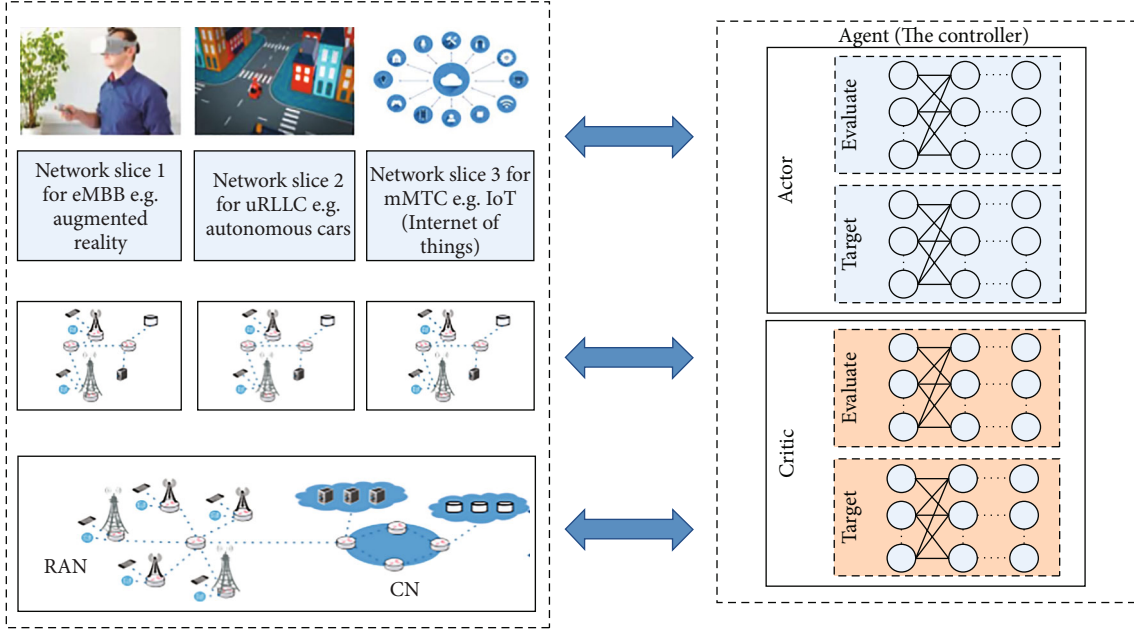


FIGURE 1: System scenario.

represent the main focus, optimization objectives, and their use case scenarios.

$$\alpha_e(x) = 1, \quad (1)$$

$$0 \text{ otherwise,}$$

$$\alpha_u(y) = 1, \quad (2)$$

$$0 \text{ otherwise,}$$

$$\alpha_m(z) = 1, \quad (3)$$

$$0 \text{ otherwise.}$$

In wireless communication, the prevalent fading channel model is taken into consideration. We look into the claim that variations may not impact the effectiveness of the transmission channel because the user-driven learning method employed in the DRL-based heterogeneous network tries to manipulate explicit channel coefficient information. Written specifically, the channel coefficient $\text{coeff}(bs, u)$ between both the user u and ground station (BS) is

$$\text{Coeff}_{bs,u} = \sqrt{\beta_{bs,u}} g_{bs,u}. \quad (4)$$

In this scenario, the substantially faded coefficient is (bs, u) , and the limited fading factor is $g(bs, u) \in (0, 1)$. In this arrangement, the following factors influence the high bandwidth rate Rate_u of UE k :

$$\text{Rate}_u = \log_2 \left(1 + \frac{\left| \sum_{bs=1}^{\text{BS}} \text{coeff}_{bs,u}^* w_{bs,u} \right|^2}{\sigma_k^2} \right), \quad (5)$$

TABLE 1: Abbreviation used in paper.

Notation	Description
BS	Base station
RB	Resource block
eMBB	Enhanced mobile broad band
uRLLC	Ultrareliable low latency communication
mMTC	Massive machine type communication
QoS	Quality of service
RL	Reinforcement learning
DL	Deep learning
DRL	Deep reinforcement learning
AWGN	Additive white Gaussian noise
DPG	Deterministic policy gradient
DDPG	Deep deterministic policy gradient
UE	User equipment
RAN	Radio access network
CN	Core network
DU	Digital unit
CSI	The channel state information
TTI	Transmission time interval
CINR	Carrier-to-interference and noise ratio
TN	Transport network

where the terms $w_{bs,u}$ refers to noise power, and σ_u^2 refers to the downstream subcarrier parameter from the BS (BS) to customer u . We used the NOMA systems to overcome the disturbance from the surrounding subchannels. We suppose that the base station uses a range of frequency bands in order to reduce intercellular interference.

TABLE 2: RL-based resource allocation.

Ref.	Algorithm	Focus	Optimization objective	Use case/vertical app	Training	Development
[17]	DQN	RAN	Improve resource consumption and slice isolation	Continuous bit rate and lowest bit rate	Centralized	Simulation
[18]	Q-learning		Maximization of resource use while assembling the elements of successful communication	Haptic	Centralized	Simulation
[19]	Q-learning, SARSA, and Monte Carlo	RAN	Assurance of efficient resource use while fulfilling the demands for low latency	Internet of Things	Centralized	Simulation
[20, 21]	DDQN and duelling DQN	RAN	Maximize long-term profits while offering the services that different multitenant customers require	Manufacturing, automotive, and utilities	Centralized	Emulation (TensorFlow)
[22]	DQN	RAN	Maximize the utilisation of radio resources while preserving QoS	eMBB, mMTC, and URLLC	Centralized	Simulation
[23]	DQN	E2E (RAN, TN, CN, edge)	SFC traffic variations should be accommodated when VNF placement is optimised	eMBB	Centralized	Emulation (OpenAI gym)
[24, 25]	LSTM	RAN	It is necessary to maximize spectrum efficiency and the SLA satisfaction ratio	VoLTE, eMBB, and URLLC	Centralized	Simulation
[8]	A3C	RAN	Making the most of resources while preserving slice separation	Undeclared	Distributed	Emulation (TensorFlow)

3.3. Slice User Scheduling Model. Users are connected to the proper routing slice in this system based on the numerous requests they make once the network has determined what kind of product or service the user needs. Each slice is designed to serve a specific user type and has specific virtual resource requirements, such as those for internet and power. This article develops a distinct strategy for distributing resources across each slice. To keep the traffic queue full, each user continuously buffers incoming packets. To decrease the amount of time, evidence needs to be delayed in the pipeline until being transmitted requires careful queue scheduling to maximize system capacity. The queues must be properly scheduled to increase system capacity and decrease buffer queuing wait times. Consider for a moment that network slices are capable of supporting a variety of services. Each slice contains $u_s, s = \{1, 2, 3, \dots\}$, etc. users. The user arrives at time slot t as $A(t) = \sum_{s=1}^S A_s(t)$. In comparison, A is the number of people who can physically fit in the space at once. The total amount of people coming across all slices is the same as the number of users entering at time slot t :

$$A(t) = \sum_{s=1}^S A_s(t). \quad (6)$$

$A_s(t)$ shows the number of people who initiated slice s while time slot t . Figure 2 displays the slice s request queue as $Q_s^F(t), Q_s^F(t) < \infty$. First users access their particular network slices in reply to the demands of various services at a time t . The slice administrator then distributes all slice users among various resource modules in compliance along with the opportunity scheduling approach [36]. The NOMA sys-

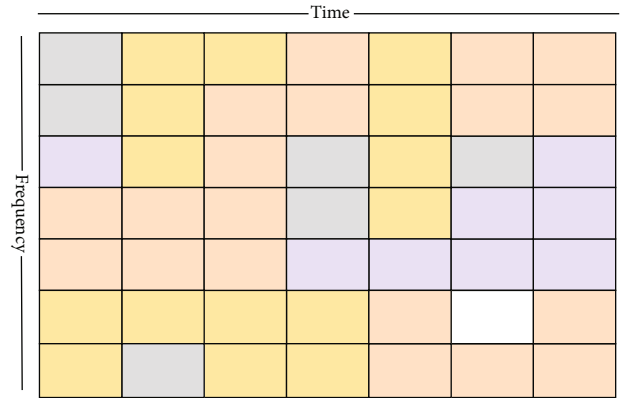


FIGURE 2: Allocation of resource block using slices.

tem's receivers employ successive interfering elimination (SIC) to multiple users of different power levels onto the similar subchannel. The procedure specifies that multiplexed users with higher channel gains can decode and remove noise from multiplex customers with smaller channel quality and accuracy [37]. Users often receive lower power allocations when there are strong channel gains, whereas users typically receive larger power allocations when there are low channel gains [22]. A scheduling system has been developed to guarantee that clients connected to a single subchannel have different channel gains. For accessing various resource blocks (RBs) n of section s , it is described as $Q_{s,n}^S(t), Q_{s,n}^S(t) < \infty$ which is defined as the likelihood that the consumer of slice s will be processed on RB $n, n \in N$, where N shows the set of RBs, and $|N| = N$. The following

is a calculation for the total likelihood that slice i user will receive attention during time slot t :

$$\sum_{n \in N_s} P_{sn} = 1. \quad (7)$$

In this equation, N_s denotes a group of RBs with slice s assigned to each of them. In order to slice s , denoted by N_s , they divide a group of accessible bandwidths among one another. Equation (2) indicates that the system accommodates all targeted users. The user holds duration for sliced s on base network n while time slot t is represented as follows:

$$Q^s(t) = Q_{s,n}^s(t). \quad (8)$$

Slice s on RB n 's user queue length for prime time t is specified as $Q^s(t) = Q_{s,n}^s(t)$. An expression for queue data storage at the time is provided after a statement for the queue caching time:

$$Q(t) = \sum_{n \in N} \sum_{s \in N} Q_{s,n}^s(t). \quad (9)$$

Using equation (3) as an example, average queue caching Q is as follows:

$$\bar{Q} = \lim_{t \rightarrow \infty} \sup \frac{1}{T} \sum_{t \in T} E\{Q(t)\}. \quad (10)$$

3.4. Resource Management Model. The slice operator takes each user's channel circumstances into account while deciding which RB to allocate them to. The system's networks in this study are entirely independent, uniformly dispersed Rayleigh fading channels, while the study's channel noise is multiplicative white Gaussian noise. Resource blocks are a type of network resource used by the RAN.

A resources block (RB) is split into a frequency domain and a time domain in Figure 2. The frequency is divided into subcarrier units. TTI units are used to measure time. The proportional fair scheduler gives a UE an RB for every TTI. When standardized by the overall data transfer rate of all UEs, the scheduler assigns far more RBs to the UE with the greatest data rate. Therefore, fair distribution may be carried out, and RBs may be allocated to UEs even when the information rate is very low, as it is for UEs close to the cell edge. The subcarriers of bandwidth B are divided into numerous $U(s, c)$ consumers of slice l that are clustered on the multicarrier c in the equation $C = 1, 2 \dots C$.

The whole purpose of substantial subcarrier c is characterised by the symbol Pow_c , where $P_c = \sum_{i=1}^{u_{s,c}} \text{pow}_i$, mpow . This study examines the downlink transmission and ranks each user according to their channel gains, as specified by the notation, in which all the I is the encoding of the number of consumers inside the subcarrier $|u_{1,c}|^2 < \dots < |u_{i-1,c}|^2 < |u_{i,c}|^2 < \dots < |u_{s,c}|^2$. Users can be recognized based on power level and channel gain. The subcarrier c 's overlying signal on the transmission connector through the NOMA transmission is represented by the symbol Sig_c , which is as follows:

$$\text{Sig}_m = \sum_{i=1}^{u_{s,c}} \sqrt{\text{pow}_{i,c}} \text{Trans}_{i,c}. \quad (11)$$

The terms $\text{Trans}_{i,m}$ and $\text{pow}_{i,c}$ in this context, refer, respectively, to the data transmission of user I on multicarrier c as well as the energy supplied to the customer I on subcarrier c . In RAN slicing, resource block isolation gives each slice access to the greatest number of immediately available RBs. In order to avoid using more RBs than were permitted, each slice also allocates RBs to its UEs. An equation for the information from client I on multicarrier c that the receiver picked up is as follows:

$$y_{i,c} = h_{i,c} \text{Sig}_c + \omega_{i,c}. \quad (12)$$

In (6), he detailed Rayleigh fading network parameter between several BSs to the i -th user on transmit antenna c and $h_{i,c}$ that are 0 complex AWGN random variables with variances of σ_c^2 and $\omega_{i,c}$, respectively.

Shannon's capability equation and the SIC technique at the transmitters can be used to determine the maximum attainable data frequency of the i -th consumer on subcarrier c as follows:

$$\text{Shannon}_{i,m} = B_c \log_2 \left(1 + \frac{\text{Pow}_{i,c} \Gamma_{i,c}}{1 + \sum_{j=i+1}^{u_{s,c}} \text{pow}_{j,c} \Gamma_{j,c}} \right). \quad (13)$$

According to (7), each consumer can be viewed as having a rate that will be considerably impacted by how much power is given to new users. Here, B_c is the subcarrier's bandwidth B_c , and $\Gamma_{(i,c)}$ is the i -th user's CINR, which is characterised as follows:

$$\Gamma_{i,c} = \frac{|u_{i,c}|^2}{\sigma_c^2}. \quad (14)$$

The preceding is an equation for the entire slice rate; meanwhile, we supposed that slice S has access to C subcarriers:

$$\begin{aligned} \text{Rate}_s^C &= \sum_{c=1}^C \sum_{u=1}^{u_{s,c}} \text{Rate}_{u,c}, \\ \text{Rate}_s^C &= \sum_{c=1}^C \sum_{u=1}^{u_{s,c}} B_c \log_2 \left(1 + \frac{\text{Pow}_{i,c} \Gamma_{i,c}}{1 + \sum_{j=i+1}^{u_{s,c}} \text{pow}_{j,c} \Gamma_{j,c}} \right). \end{aligned} \quad (15)$$

The determined number of available bandwidths can be allotted to satisfy the slicing performance and latency necessities. The NOMA calculation includes the fading channels c global path loss as follows:

$$\begin{aligned} \text{Pow}_{\text{total}}^c &= 1 - \text{PowRate} \{ \forall r_{i,c} > \text{Rate}_s^{\min} \}, i = \{ 1, 2, \dots, u_{s,s} \}, l \\ &= \{ 1, 2, \dots, S \}. \end{aligned} \quad (16)$$

3.5. Problem Formulation. The problem requires a decrease in

the overall quantity of experience delay $\alpha_u(y)$ for many interconnections and URLLC users $\alpha_u(y)$ for mMTC in order to achieve an effective allocation of resources for MVNOs and obtain a greater sum of data rate $\alpha_e(x)$ for eMBB users. So, in a coupled issue, we express the maximizing and minimization problems of an MVNO *MiM* as follows:

$$\text{Maximization}_{\text{Performance}} = \left\{ \sum_{i=U_i} \alpha_i^x, \alpha_j^y, \alpha_i^z - \sum_{j \in U_j} \alpha_j^y \text{Delay}_j + \sum_{l \in U_l} \alpha_l^z \text{Massive}_l \right\}, \quad (17)$$

$$\text{Subject to } 0 \leq \text{Performance}_i \leq \text{Performance}_{\max}, \quad (18a)$$

$$\sum_{i=U_i} \text{Performance}_i \leq 1, \quad (18b)$$

$$0 \leq \text{Performance}_i \leq \text{Performance}_{\max}, \quad (18c)$$

$$\text{Throughput}_i \geq \text{Throughput}_i^{\min}, \quad (18d)$$

$$\text{Delay}_j \leq \text{Delay}_j^{\max}, \quad (18e)$$

$$\text{Massive}_l = \text{Massive}_l^{\max}. \quad (18f)$$

While determining these values, the user's requirements and the protection of resources' upper limits must be taken into account. The constraint (18a) ensures that the frequency fraction allocated falls between the range of 0 and $\text{Performance}_{\max}$, the maximum value. There is a guarantee that the channel capacity allotted to customers will not exceed the bandwidth B_i offered from the InP by constraint (18b). Constraint (18d) ensures that an eMBB user's data rate must exceed a predetermined minimum standard. To ensure that the maximum number of devices connected by an mMTC user is more than a predetermined threshold, constraint (18e) states that the URLLC subscriber's data packet delay should not exceed a specific limit (18f). Our network slicing also aims to increase system throughput while guaranteeing that the quality of service requirements of various network slices are satisfied. To achieve the aim, we must consider three crucial aspects of the system throughput:

Throughput of eMBB network slices (Thru_x)

Throughput of URLLC network slices (Thru_y)

Throughput of mMTC network slices (Thru_z)

3.5.1. Throughput/Efficiency of eMBB Network Slice. The symbol Thru_x, u denotes the throughput for such eMBB network slice request that a UE makes to the mobile operator:

$$\text{Thru}_x, u = \sum_{x=1}^X \alpha_e(x) f_{x,u}, R_s, \quad (19)$$

where the pair $f_{x,u}$ designates the resource bandwidth provided to a UE u inside the x -th eMBB slice (x, u). The frequency restriction of UE should be bigger than Thru_x, u , and it should be highlighted.

$$\text{Thru}_x, u \geq \text{Thru}_{u,\min}. \quad (20a)$$

The corresponding sum throughput for the eMBB particular portion is as follows:

$$\text{Thru}_x = \sum_{s=1}^S \text{Thru}_{x,u}. \quad (20b)$$

3.5.2. Throughput of uRLLC Network Slice. The UE k URLLC slice's throughput is

$$\text{Thru}_y, u = \sum_{y=1}^Y \alpha_u(y) f_{y,u}, R_s. \quad (21a)$$

The symbol $f_{j,u}$ designates the resource bandwidth given to the UE k inside the j -th URLLC slice. According to our research, one packet of data should theoretically be sent within a single URLLC frame. The frame time should not be exceeded by the maximal network delay D as [14].

$$\frac{F_{y,k}}{\text{Thru}_y, u} \leq D_{y,u,\max}, \quad (21b)$$

where $D(y, k, \max)$ is the maximum segment latency of user equipment k in the y -th URLLC particular portion, and $F(y, k)$ is the packet size to user equipment k in the y -th URLLC data object. The URLLC network slice's connected sum throughput is $\text{Thru}_y = \sum_{s=1}^S \text{Thru}_{y,u}$. The packet length for user equipment k in the y -th URLLC set of resources is $F_{y,k}$, and $D_{y,k,\max}$ maximal is the maximum segment delay of UE k in the y -th URLLC set of resources. The related sum throughput for the URLLC slice is $\text{Thru}_y = \sum_{s=1}^S \text{Thru}_{y,u}$.

3.5.3. Throughput of mMTC Network Slice. The mMTC slice of user equipment k shares characteristics with the uRLLC and eMBB slices in terms of their throughput by

$$\text{Thru}_z, u = \sum_{z=1}^Z \alpha_m(z) f_{z,u}, R_s. \quad (22a)$$

The UE inside the l -th mMTC slice has a resource bandwidth (f_l, k) assigned to it. Such as the eMBB and URLLC slices, the mMTC slicing is not subject to a rate/latency requirement. The formula for the suitable sum flow of an mMTC slicing is $\text{Thru}_z = \sum_{s=1}^S \text{Thru}_{z,u}$.

In decision, the following equation can be used to determine the total network bandwidth $T(t)$ entirely at the time t :

$$\text{Thru}_{\text{Total}}^t = \text{Thru}_x^t + \text{Thru}_y^t + \text{Thru}_z^t. \quad (22b)$$

The problem of improving system throughput across T time frames is expressed as follows:

$$\text{Prob} : \max \left\{ \text{Thru}_x^t + \text{Thru}_y^t + \text{Thru}_z^t \right\} \sum_{t=1}^T T_{\text{total}}^{(t)}, \quad (23)$$

where the components of binary data are, appropriately, $\text{Thru}_x^t + \text{Thru}_y^t + \text{Thru}_z^t$. Remember that the extra slices are saved as a backup.

Due to the requirements of the dynamical slices and the presence of data gained in the long-term optimum objective, the novel optimization issue is highly difficult. As a result, it is challenging to solve it directly using the traditional optimization procedure. The problem can be formulated using an MDP and the necessary reinforcement learning solution approaches.

4. Basics of Deep Reinforcement Learning

Reinforcement learning (RL) is a field of artificial intelligence and intelligent systems that deals with the issue of a learning agent that is placed in a setting to accomplish a task. The RL agent must learn by trial and failure how to behave in order to acquire the highest reward, in contrast to reinforcement methods, where the learner's structure receives instances of good and bad performance [23]. In order to do this work, the agent must perceive the environment's state at some level and act accordingly to create a new state. The agent's action results in a reward, which encourages it to repeat the same behaviour in the future.

Modelling the environment's state transitions depending on the agent's behaviours is also required to formulate the challenge eventually. As a result, an MDP is created that has the functionalities of S , A , R , and T , where S denotes a set of environmental states, A denotes a set of potential actions within a state, T denotes the function that switches between states based on the actions, and R denotes the reward for the specific pair of S and A .

4.1. DRL-Based Resource Allocation Model. We outline the MDP's formulation in this section. We establish the subspace, the activity floor plan, and the value function for rewards in formulating the MDP issue.

4.1.1. State Space. Each agent keeps track of the status of the environment at each temporal step t . The sort of unique visitors and their obtained features is observed for each virtual network mobile operator (MVNO). The sorts of users are required since they establish the SLA's requirements (SLA). To allocate bandwidth effectively, estimating the strengthening between each related user on the communications platform is required. Each MVNO periodically collects the channel gains. In actuality, every MVNO sends out model validation to all of its customers. Each user then calculates the channel state data and transmits it back to their MVNO through the controller.

The observed condition of MVNO m_i at period t is designated as $\text{State}_i(t)$.

$$\text{State}_i(t) = \{\text{channel}_{\text{Gain}_i}(t), U_i(t)\}. \quad (24)$$

The list of user categories for the MVNO is characterised by U_i and U_t , where $\text{channel}_{\text{Gain}_i}(t)$ reflects the signal strength among MVNO and its users during the time slot t . The three numbers U_e, U_u , and U_m are used to specify the

many user groups and represented the priority of each category. Users of URLLC are typically given higher priority ratings since they have stricter delay requirements.

4.1.2. Action Space. Each MVNO receives the required bandwidth fraction B_i during each time slot from RIC. Users of an MVNO are given B_i fractions. The following is the operation zone for every MVNO during time slot t :

$$\text{Act}_i(t) = [0, \text{Performance}_{\text{max}}]. \quad (25)$$

Each action, $a_i \text{Act}_i(t)$, is represented as a row vector, $\text{Performance}_{i,j,l}(t)$.

4.1.3. Reward Function. An MVNO desires an activity $a_i \text{Act}_i(t)$ at time step t , in exchange for which it is given a reward $a_i \in \text{Reward}_i(t)$. Since the goal is to reduce the delay, the incentive should be defined as a function of the latency for uRLLC users regarding the data flow for eMBB users and the maximum number of devices connected.

We specify a reward associated with each end user's contentment, with

$$\text{Reward}_i(t) = a_i^x, a_i^y, a_i^z. \quad (26)$$

These words can be used to convey the total reward:

$$\text{Reward}_i(t) = \sum_{i=1}^N \text{Reward}_{(i)} \quad \text{if } a_i \text{ is valid, } -0.1 \text{ otherwise.} \quad (27)$$

If the average of the components is far less than 1 and the fractions assigned result in latency and data speeds that match the SLA values, the action $a_i \in a_i$ is deemed valid. A significant reward is provided if the action is invalid in order to deter the agent from making a similar decision in later phases.

4.2. Proposed EE-Deep Reinforcement Learning-Based Resource Allocation Algorithm. The valuation and policy-based subcategories of prototype reinforcement learning systems can be used to categorise policy modification. Value-based solutions give the agent the ability to acquire the best policy by helping them comprehend the value function. The action space is always there in this piece. The value-based approach of the linear system and the naive discretization of the action space lead to the dimensional curse and the loss of crucial information about the structure of the action domain. The policy-based approaches make use of parameterized policies to successfully train probabilistic policies for addressing high-dimensional data action and state and action space challenges.

The following is a representation of the unpredictable policy function π_θ at time step t : when a policy is parameterized, action in θ at state s follows the posterior distribution with parameter.

$$\pi(a|\text{state}, \theta) = P(\text{Act}_t = a | \text{State}_t = \text{state}, \theta_t = \theta). \quad (28)$$

$\text{Obj}(\pi) = \mathbf{E}_{s \sim p^r, a \sim \pi_\theta} [\sum_t r(\text{state}_t, a_t)]$. According to the

objective function's specification, the subsidised official visitor probability for a policy serves as a representation of the expected return, denoted by p^π . The gradient descent technique [38] uses the steep descent to get the optimum parameter π . This is a representation of the parameter update:

$$\theta_{t+1} = \theta_t + \alpha \nabla \theta_t \text{Obj}(\pi_{\theta_t}). \quad (29)$$

The sequential policy gradient (SPG) must carry out complex calculations when the reaction is a high-dimensional vertex in order to sample the action again for stochastic policy. Instead of frequently sampling actions, the deterministic policy gradient (DPG) [39] immediately generates a deterministic behaviour policy. The DPG optimization problem gradient is described as follows:

$$\nabla \theta^{\text{obj}}(\mu_\theta) = \left[E_{\text{state} \sim p^\mu} \left[\nabla_{\theta} \mu_\theta(s) \nabla_a Q^\mu(\text{state}|a) \Big|_{a=\mu_\theta(\text{state})} \right] \right]. \quad (30)$$

DPG-based techniques result in deterministic strategies as opposed to studying the environment. Outside of official policy, exploitation, and exploration can coexist. Enough action exploration is guaranteed via a stochastic behaviour policy. The goal strategy is deterministic and effectively makes use of efficient deterministic policies. As a result, the actor-critic (AC) technique, which is detailed in the next section, is used in the learning framework of the DPG technique.

4.2.1. Actor-Critic Method. The actor-critic method combines the advantages of value-based techniques and policy-based procedures. To put it another way, the actor generates behaviour from a state that a policy function provides. The critic develops an action-value function and usages the TD-error to assess the action's effectiveness (loss function). The actor then uses the DPG technique to upgrade the policy variable with the critic's output. The critical updates the action function f using gradient descent [40]. Additionally, the function approximations parameterized by θ^Q and θ^μ , the activity default value and the regulation variable are taken into account. The following changes are made to the linear combination parameter:

$$\begin{aligned} \delta_t &= r_t + \gamma Q(s_{\text{tate}_{t+1}, \mu}(\text{state}_{t+1} | \theta^\mu)) \Big|_{\theta^Q} - Q(\text{state}_t, a_t | \theta^Q), \\ \theta^Q(t+1) &= \theta^Q(t) + \alpha_c \delta_t \nabla_{\theta^Q} Q(\text{state}_t, a_t | \theta^Q). \end{aligned} \quad (31)$$

The actor uses the DPG mechanism to update policy parameters θ^μ :

$$\theta^\mu(t+1) = \theta^\mu(t) + \alpha_a \nabla_{\theta^\mu} \mu(\text{state}_t | \theta^\mu) \nabla_a Q(\text{state}_t, a_t | \theta^Q) \Big|_{a=\mu(\text{state}_t)}. \quad (32)$$

4.2.2. Deep Deterministic Policy Gradient-Based Resource. An algorithm for allocating resources using deep reinforcement learning is introduced in this section. With the aid of

a special dual deep stochastic policy gradient technique, the resource provisioning issue is addressed. The actor-critic technique is unbalanced when deep neural networks are utilized with function approximations. The experience replay training method of the deep Q network algorithm [41] can destroy the correlation between succeeding data [42]. Based on the rewards of the DQN algorithm and actor-critic method, the deep deterministic policy gradient (DDPG) algorithm successfully operates over the continuous state domain. The DDPG architecture is presented in detail in Figure 2 [43, 44]. The solid red line and the blue lines with full dots represent the training processes for actor or critic networks, respectively.

(1) *Experience Replay.* The agent communicates with the environment to collect data tuples $(\text{state}_t, a_t, r_t, \text{state}_{t+1})$ and keep them in replaying buffer D . The critique and actor randomly select a minibatch of subsampling from D to modify the dynamic programming variable and the regulation function parameter.

(2) *Target Network.* Deep neural networks used to execute Q-learning directly have been shown to be unstable. The network update usually differs from the original because two protocols target networks, and the predicted network shares a set of parameters. Duplicates of the actor network $\mu'(\text{State} | \theta^{\mu'})$ and critic network $Q'(\text{state}, a | \theta^{Q'})$ are generated in order to ascertain the target value. DDPG employs $\theta' \leftarrow \tau \theta + 1(1 - \tau)\theta'$ soft target updates for the target networks' weights. The learning stability could be enhanced with $\tau \ll 1$.

5. Experiments and Results

Simulation studies using TensorFlow and Python were conducted to evaluate the dominance of the proposed DDPG-based training set for the allocation of resources in RAN slicing. A resource item and two resources, mainly power capacity and bandwidth, are assigned to each physical point. Five network slices, or a total of 25 VNFs, are randomly distributed within the network during each episode's deployment. The needed resources for each VNF from the inside of a slice are evenly split between one and twenty resource units during each system cycle. The simulations we could obtain are shown in the following figures. Using the following resources, we selected different slices: 150 megahertz is the total bandwidth, and 175J of energy resources is available.

Furthermore, in response to demands from the end user, we altered the resources required for each job. In order to fulfil the required quantity, it distributes resources as equally as is practical, raising, or lowering them to the level that most closely satisfies the needs of each slice. When a slice demands more resources, the agents will try their best to accommodate the request or assign as many of the resources as is practical. However, after the resources have been allocated, the agents will not reduce the resources, even if resource utilization is low. In this method, the best agent

emphases on reducing the SLA breach. The random agent assigns a random amount of the demanding resources to each request.

In addition, the DDPG environment looks like this. The motivation discount in our simulations is fixed at 0.9, the learning rate for a performer is set to 0.001, and the learning rate for a critic is set to 0.001. This research will discuss the simulation's findings in the sections that follow. The efficient deep causal gradient descent algorithm needs two continuously trained efficient deep deterministic policy gradient-based algorithms. At each encounter, T time frames are used to repeatedly train the second efficient deep deterministic policy gradient-based algorithm for user space adaptability and the first efficient deep deterministic policy gradient-based method for slice-level adaptive. Slice-level performers and critic networks are taught once client-side actors and critic networks have been trained. The second DDPG method takes as input the result of the first DDPG algorithm's actor program.

The differences in reward systems based on the number of episodes are shown in Figure 3. We can see that the proposed DDPG-resource allotment approach converges after about 200 sessions. We map the reward according to the amount of training sessions during the training phase. We see that as the number of training sessions climbs, the overall incentive of DRL-NS increases quickly. As a result, as shown in Figure 4, the compensation rises with each episode and reaches a point of stability after around 200 episodes.

Figure 4 also demonstrates that the tenant's general utility is rising, which is the main objective of our suggested plan, as indicated in the problem formulation section. RB penetration and MVNO used to trade off against each other. The MVNO rents more RBs from the systems integrator, increasing MVNO consumption to provide more transmission resources to network operations and generate cash.

The bandwidth resource distribution with the number of episodes is shown in Figure 5. When tried to be compared to end user queries in slice 2, slice 1's requests from bandwidth-hungry end customers require more bandwidth. In contrast, slice 2's end users' queries are distinct from those in slice 1's end users' requests. Therefore, high bandwidth resources were allocated to slice 1 using the proposed DDPG dynamic resource allocation algorithm, while other resources were allocated to slice 2. According to Figure 5, the suggested plan allocated slice 1 after around 20–30% other resources and 70–80% bandwidth resources. The trends for other resource distributions to slices are comparable. Variations in resource capacity affect how much energy the MVNO operator or controller gets. The size of the resource capacity determines how much money the MVNO controller will make from resource allocation.

Figures 6 and 7 depict the system throughput and allocation of energy resources to the delay needs. Slices 3 and 4's end user requests are slightly distributed normally. About half of the requests necessitate a significant amount of energy, whereas the other half demand system throughput. As a result, slices 3 and 4 received roughly equal amounts of energy resources and delayed requirements, respectively, using the proposed DDPG dynamic resource allocation

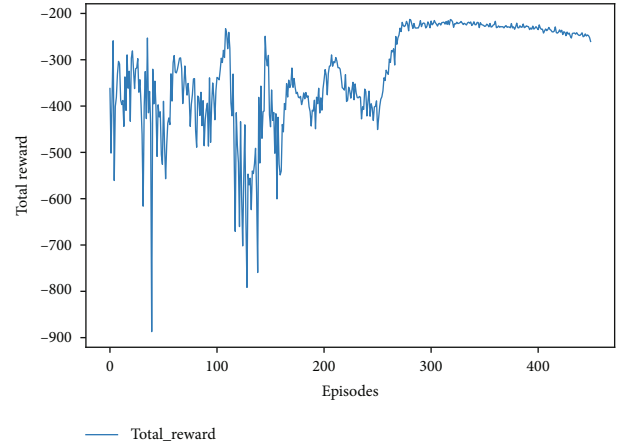


FIGURE 3: Total reward.

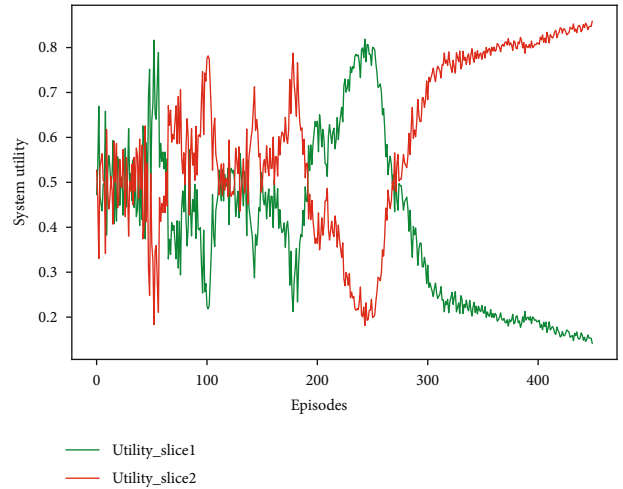


FIGURE 4: System utility.

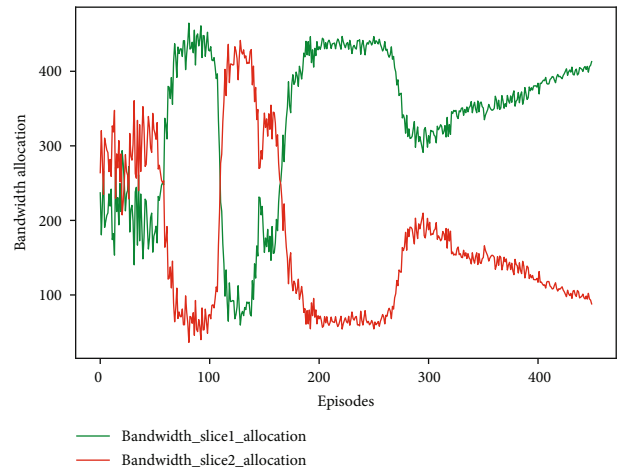


FIGURE 5: Bandwidth allocation.

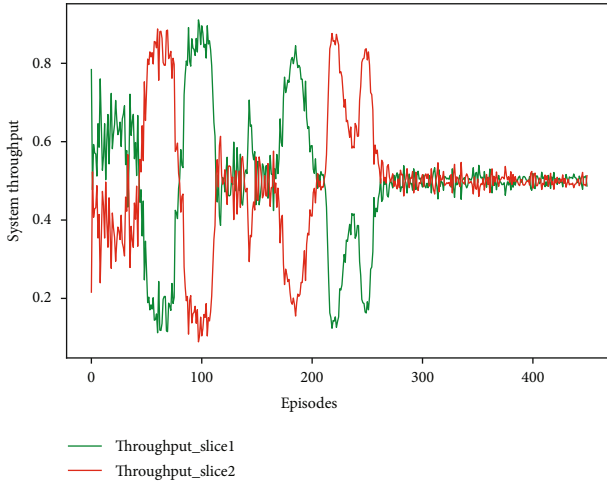


FIGURE 6: System throughput.

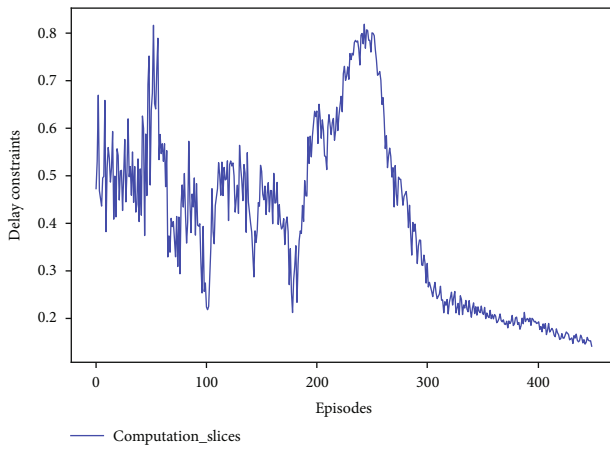


FIGURE 7: Energy consumption.

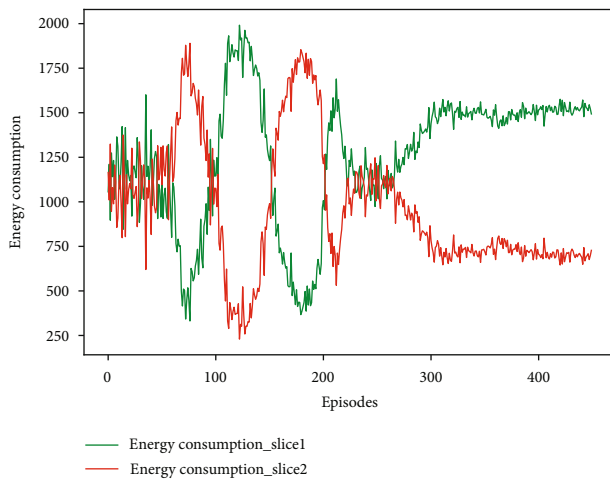


FIGURE 8: Delay constraints.

method. Figures 6 and 7 show that slices 3 and 4 get about 50% of the energy and 50% of the delay, respectively.

The DDPG algorithm is used for these wholly dynamic resource assignments that should be emphasized. Figure 7 is an exception, where resource assignment varies by 400 episodes, possibly due to different end user request patterns. The effectiveness of preserving resource actual values of an MVNO controller's usefulness is depicted in Figure 7. Every algorithm gradually becomes more useful as additional resources are made available. As a result, InP's provision of adequate resources is what makes MVNO controllers more useful. The proposed DRL-based algorithm offers the most utility. Figure 8 shows how to distribute resources to improve system performance and reduce the need for delays. The end user requests slice that correspond to the customary resource allocation. The first half of the questions are intensive, but the second half of the questions need the system to be able to handle a lot of work.

6. Conclusion

This article looked at the number of MVNOs that use resources when RAN slicing is in effect. We concentrated on how to apply machine learning to create reliable slicing patterns in various wireless communication environments. After that, we suggested a DDPG to create a deep reinforcement-learning system for distributing power and bandwidth simultaneously. Slices of the eMBB, URLLC, and mMTC types were taken into account in our scenarios. We organized the problem as a specific virtual network mobile operator's MDP in order to allocate radio resources to different user types (eMBB, mMTC, and URLLC). In our proposal, we combined the benefits of policy-based and value-based reinforcement learning techniques into an actor-critic approach. Since the fractional bandwidth values are constant, we simultaneously train a Q-function and a policy using a deep deterministic gradient. This gradient has an ongoing effect. To enhance how many MVNOs manage radio resource allocation cooperatively, we developed a (EE-DDPG-RA) DRL-based technique on a RAN architecture. Under numerous simulated situations with non-i.i.d. and uneven distribution of the end users, the effectiveness of the suggested (EE-DDPG-RA) DRL technique is demonstrated. Experience has shown that, in comparison to models created independently by each MVNO, the model trained to employ (EE-DDPG-RA) DRL is more resistant to environmental changes.

We pointed up certain key concerns in order to fully execute the application of DDPL in a larger meaning. In the future, network slicing may benefit significantly from the use of DRL, in our opinion. However, you should carefully consider network slicing because it involves a number of elements before implementing DDPG: for network slicing to succeed, a flexible and dynamic slice management strategy is required, (a) limiting the acceptance of fresh slice requests. How to use DDPG also provides a fascinating difficulty because the state and action spaces must adjust to the modifications made to the "slice" space if new slice requests arise. A quick policy-learning method is needed because of user

activity and the time-varying nature of wireless channels in (b) policy learning cost. However, the cost of policy training today is still less than the required learning rate. As a result, there are still many intriguing questions that have not been addressed.

Data Availability

There is no data included with the manuscript for publication.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The study is funded by the National Natural Science Foundation of China (61871058).

References

- [1] F. Grijpink, A. Ménard, H. Sigurdsson, and N. Vucevic, "The road to 5G: the inevitable growth of infrastructure cost," vol. 8, McKinsey& Company, 2018.
- [2] A. Sengupta, A. Rico Alvarino, A. Catovic, and L. Casaccia, "Cellular terrestrial broadcast-physical layer evolution from 3GPP release 9 to release 16," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 459–470, 2020.
- [3] S. R. Pokhrel, J. Ding, J. Park, O.-S. Park, and J. Choi, "Towards enabling critical mMTC: a review of URLLC within mMTC," *IEEE Access*, vol. 8, pp. 131796–131813, 2020.
- [4] S. Zhang, Y. Wang, and W. Zhou, "Towards secure 5G networks: a survey," *Computer Networks*, vol. 162, article 106871, 2019.
- [5] S. Shi, W. Yang, J. Zhang, and Z. Chang, "Review of key technologies of 5G wireless communication system," *MATEC Web of Conferences*, vol. 22, article 01005, 2015.
- [6] Z. Wang, F. Yifei Wei, Y. Richard, and Z. Han, "Utility optimization for resource allocation in multi-access edge network slicing: a twin-actor deep deterministic policy gradient approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 5842–5856, 2022.
- [7] Q. Chen, X. Wang, and Y. Lv, "An overview of 5G network slicing architecture," in *AIP Conference Proceedings*, Busan, South Korea, 2018.
- [8] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, "Intelligent resource scheduling for 5G radio access network slicing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7691–7703, 2019.
- [9] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: a survey on principles, enabling technologies, and solutions," *IEEE Communication Surveys and Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.
- [10] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: a brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.
- [11] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with deep reinforcement learning," pp. 1–9, 2013, <http://arxiv.org/abs/1312.5602>.
- [12] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [13] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "Deep-Slice: a deep learning approach towards an efficient and reliable network slicing in 5G networks," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0762–0767, New York, NY, USA, 2019.
- [14] ITU-T Telecommunication Standardization Sector of ITU, "Framework of network slicing with AI-assisted analysis in IMT-2020 networks," 2020, SERIES Y: Annual Report - Future Networks, https://www.itu.int/rec/dologin_pub.asp?lang=s&id=T-REC-Y.3156-202009-I!!PDF-E&type=items.
- [15] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: enabling enterprises' own software-defined cellular networks," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 146–153, 2016.
- [16] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, "Game theory in wireless and communication networks: theory, models, and applications," Cambridge University Press, 2012.
- [17] R. Mijumbi, J. L. Gorricho, J. Serrat, M. Claeys, F. De Turck, and S. Latré, "Design and evaluation of learning algorithms for dynamic resource management in virtual networks," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–9, Krakow, Poland, 2014.
- [18] L. Le, T. N. Nguyen, K. Suo, and J. He, "Efficient embedding VNFs in 5G network slicing: a deep reinforcement learning approach," vol. 1, no. 1, 2022, <https://arxiv.org/abs/2207.11822>.
- [19] M. Masoudi, O. T. Demir, J. Zander, and C. Cavdar, "Energy-optimal end-to-end network slicing in cloud-based architecture," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 574–592, 2022.
- [20] A. Aijaz, "Hap-SliceR: a radio resource slicing framework for 5G networks with haptic communications," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2285–2296, 2018.
- [21] A. Nassar and Y. Yilmaz, "Reinforcement learning for adaptive resource allocation in Fog RAN for IoT with heterogeneous latency requirements," *IEEE Access*, vol. 7, pp. 128014–128025, 2019.
- [22] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimizing 5G infrastructure markets: the business of network slicing," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.
- [23] W. B. Powell, *Approximate dynamic programming: solving the curses of dimensionality*, John Wiley & Sons, 2nd edition, 2011.
- [24] K. Xiong, S. S. R. Adolphe, G. O. Boateng, G. Liu, and G. Sun, "Dynamic resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Access*, vol. 7, pp. 115440–115453, 2019.
- [25] S. Troia, R. Alvizu, and G. Maier, "Reinforcement learning for service function chain reconfiguration in NFV-SDN metro-core optical networks," *IEEE Access*, vol. 7, pp. 167944–167957, 2019.
- [26] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: fundamentals and applications," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 52–59, 2015.

- [27] E. Datsika, A. Antonopoulos, D. Yuan, and C. Verikoukis, "Matching theory for over-the-top service provision in 5G networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5452–5464, 2018.
- [28] P. Caballero, A. Banchs, G. De Veciana, and X. Costa-Perez, "Network slicing games: enabling customization in multi-tenant networks," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.
- [29] Y. Sun, M. Peng, S. Mao, and S. Yan, "Hierarchical radio resource allocation for network slicing in fog radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3866–3881, 2019.
- [30] Y. Xiao and M. Krunz, "Dynamic network slicing for scalable fog computing systems with energy harvesting," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2640–2654, 2018.
- [31] Y. Xiao, M. Hirzallah, and M. Krunz, "Distributed resource allocation for network slicing over licensed and unlicensed bands," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2260–2274, 2018.
- [32] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.
- [33] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.
- [34] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management & prioritization in 5G mobile systems," in *European wireless 2016; 22th european wireless conference*, pp. 1–6, Oulu, Finland, 2016.
- [35] P. Rost, C. Mannweiler, D. S. Michalopoulos et al., "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, 2017.
- [36] 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 Communications Letters*, vol. 24, no. 9, pp. 2005–2009, 2020.
- [37] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, "Dynamic SDN-based radio access network slicing with deep reinforcement learning for URLLC and eMBB services," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2174–2187, 2022.
- [38] T. Chen, A. Mokhtari, X. Wang, A. Ribeiro, and G. B. Giannakis, "Stochastic averaging for constrained optimization with application to online resource allocation," *IEEE Transactions on Signal Processing*, vol. 65, no. 12, pp. 3078–3093, 2017.
- [39] E. F. Morales and J. H. Zaragoza, "An introduction to reinforcement learning," in *Decision Theory Models for Applications in Artificial Intelligence*, pp. 63–80, IGI Global, 2012.
- [40] M. Paczkowski, "Low-friction composite creping blades improve tissue properties," *Pulp & Paper*, vol. 70, no. 9, pp. 125–129, 1996.
- [41] S. Black, "A growing trend: 3D printing of aerospace tooling," *Composites World*, vol. 1, no. 7, pp. 22–31, 2015.
- [42] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, vol. 12, 1999.
- [43] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [44] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2016, <https://arxiv.org/abs/1509.02971>.